Microsoft Ignite

Nov 19–22, 2024

Register now ⟩

✕

Learn | Discover ⌄   Product documentation ⌄   Development languages ⌄   Topics ⌄   🔍   Sign in

.NET   Languages ⌄   Features ⌄   Workloads ⌄   APIs ⌄   Troubleshooting   Resources ⌄   Download .NET

Learn / .NET / .NET Framework /

⊕   ✎   ⋮

Regasm.exe (Assembly Registration Tool)

Article • 03/22/2022 • **12 contributors**   ⟲ Feedback

In this article

Syntax

Parameters

Remarks

Examples

See also

The Assembly Registration tool reads the metadata within an assembly and adds the necessary entries to the registry, which allows COM clients to create .NET Framework classes transparently. Once a class is registered, any COM client can use it as though the class were a COM class. The class is registered only once, when the assembly is installed. Instances of classes within the assembly cannot be created from COM until they are actually registered.

To run the tool, use Visual Studio Developer Command Prompt or Visual Studio Developer PowerShell.

Syntax

```Console
regasm assemblyFile [options]
```

Parameters

⊡ Expand table

Parameter	Description
assemblyFile	The assembly to be registered with COM.

⊡ Expand table

Download PDF

Option	Description
`/codebase`	Creates a Codebase entry in the registry. The Codebase entry specifies the file path for an assembly that's not installed in the global assembly cache. Don't specify this option if you will subsequently install the assembly that you're registering into the global assembly cache. It is strongly recommended the *assemblyFile* argument that you specify with the `/codebase` option be a strong-named assembly.
`/registered`	Specifies that this tool will only refer to type libraries that have already been registered.
`/asmpath:directory`	Specifies a directory containing assembly references. Must be used with the `/regfile` option.
`/nologo`	Suppresses the Microsoft startup banner display.
`/regfile` [: *regFile*]	Generates the specified .reg file for the assembly, which contains the needed registry entries. Specifying this option does not change the registry. You cannot use this option with the `/u` or `/tlb` options.
`/silent` or `/s`	Suppresses the display of success messages.
`/tlb` [: *typeLibFile*]	Generates a type library from the specified assembly containing definitions of the accessible types defined within the assembly.
`/unregister` or `/u`	Unregisters the creatable classes found in *assemblyFile*. Omitting this option causes Regasm.exe to register the creatable classes in the assembly.
`/verbose`	Specifies verbose mode; displays a list of any referenced assemblies for which a type library needs to be generated, when specified with the `/tlb` option.
`/?` or `/help`	Displays command syntax and options for the tool.

> ⓘ **Note**
>
> The Regasm.exe command-line options are case insensitive. You only need to provide enough of the option to uniquely identify it. For example, `/n` is equivalent to `/nologo` and `/t: outfile.tlb` is equivalent to `/tlb: outfile.tlb`.

Remarks

You can use the `/regfile` option to generate a .reg file that contains the registry entries instead of making the changes directly to the registry. You can update the registry on a computer by importing the .reg file with the Registry Editor tool (Regedit.exe). The .reg file does not contain any registry updates that can be made by user-defined register functions. The `/regfile` option only emits registry entries for managed classes. This option does not emit `TypeLibID` or `InterfaceID` entries.

When you specify the `/tlb` option, Regasm.exe generates and registers a type library describing the types found in the assembly. Regasm.exe places the generated type libraries in the current working directory or the directory specified for the output file. Generating a type library for an assembly that references other assemblies may cause several type libraries to be generated at once. You can use the type library to provide type information to development tools like Visual Studio. Don't use the `/tlb` option if the assembly you are registering was produced by the Type Library Importer (Tlbimp.exe). You cannot export a type library from an assembly that was imported from a type library. Using the `/tlb` option has the same effect as using the Type Library Exporter (Tlbexp.exe) and Regasm.exe, with the exception that Tlbexp.exe does not register the type library it produces. If you use the `/tlb` option to register a type library, you can use the `/tlb` option with the `/unregister` option to unregister the type library. Using the two options together will unregister the type library and interface entries, which can clean the registry considerably.

When you register an assembly for use by COM, Regasm.exe adds entries to the registry on the local computer. More specifically, it creates version-dependent registry keys that allow multiple versions of the same assembly to run side by side on a computer. The first time an assembly is registered, one top-level key is created for the assembly, and a unique subkey is created for the specific version. Each time you register a new version of the assembly, Regasm.exe creates a subkey for the new version.

For example, consider a scenario where you register the managed component, myComp.dll, version 1.0.0.0 for use by COM. Later, you register myComp.dll, version 2.0.0.0. You determine that all COM client applications on the computer are using myComp.dll version 2.0.0.0 and you decide to unregister myComponent.dll version 1.0.0.0. This registry scheme allows you to unregister myComp.dll version 1.0.0.0 because only the version 1.0.0.0 subkey is removed.

After registering an assembly using Regasm.exe, you can install it in the global assembly cache so that it can be activated from any COM client. If the assembly is only going to be activated by a single application, you can place it in that application's directory. Using the `/codebase` option is an alternative to using the global assembly cache; however, the location of the assembly during registration is recorded globally and activation will fail if the assembly is moved. If the assembly isn't found through probing, the `/codebase` option will load the assembly in a load-from context that has additional considerations documented in Assembly.LoadFrom.

Examples

The following command registers all public classes contained in `myTest.dll`.

Console	Copy

```
regasm myTest.dll
```

The following command generates the file `myTest.reg`, which contains all the necessary registry entries. This command does not update the registry.

Console	Copy

```
regasm myTest.dll /regfile:myTest.reg
```

The following command registers all public classes contained in `myTest.dll`, and generates and registers the type library `myTest.tlb`, which contains definitions of all the public types defined in `myTest.dll`.

Console	Copy

```
regasm myTest.dll /tlb:myTest.tlb
```

See also

- Tools
- Tlbexp.exe (Type Library Exporter)
- Tlbimp.exe (Type Library Importer)
- Registering Assemblies with COM
- Developer command-line shells

○ **Collaborate with us on GitHub**

The source for this content can be found on GitHub, where you can

.NET **.NET feedback**

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

also create and review issues and pull requests. For more information, see our contributor guide.

Open a documentation issue

Provide product feedback

English (United States)   Your Privacy Choices   ☼ Theme ⌄