# About

The WiX Toolset lets developers create installers for Windows Installer, the Windows installation engine.

- The core of WiX is a set of build tools that build Windows Installer packages using the same build concepts as the rest of your product: source code is compiled and then linked to create executables; in this case .exe setup bundles, .msi installation packages, .msm merge modules, and .msp patches. The WiX command-line build tools work with any automated build system. Also, MSBuild is supported from the command line, Visual Studio, and common CI/CD build systems like GitHub Actions.
- WiX includes several extensions that offer functionality beyond that of Windows Installer. For example, WiX can install IIS web sites, create SQL Server databases, and register exceptions in the Windows Firewall, among others.
- With Burn, the WiX bootstrapper, you can create setup bundles that install prerequisites like the .NET Framework and other runtimes along with your own product. Burn lets you download packages or combine them into a single downloadable .exe.
- The WiX SDK includes managed and native libraries that make it easier to write code that works with Windows Installer, including custom actions in both C# and C++.

You can also follow via Twitter at @wixtoolset

## System requirements

### Running packages built with WiX

In general, packages you build with WiX will work on Windows 7 or later. Code that you introduce -- for example, custom actions or bootstrapper applications -- can require later versions of Windows.

### Building packages with WiX

To use WiX as a .NET tool or as an MSBuild SDK via `dotnet build`, you must have a .NET 6 SDK installed. See its system requirements and download here.

To use WiX as an MSBuild SDK via `msbuild`, you must have .NET Framework 4.7.2 or later installed. WiX runs on ARM64 systems as ARM64

## License

The WiX toolset is released under the Microsoft Reciprocal License (MS-RL). A reciprocal license is used to ensure that others who build on the effort of the WiX community give back to the WiX community. Specifically the license requires that fixes and improvements to the WiX toolset must be published using the same license.

Sometimes the reciprocal license is incorrectly interpreted to also apply to bundles, packages, and custom actions built using the WiX toolset. The Outercurve Foundation has previously provided this statement below to clarify which now the .NET Foundation reaffirms:

> The WiX toolset (WiX) is licensed under the Microsoft Reciprocal License (MS-RL). The MS-RL governs the distribution of the software licensed under it, as well as derivative works, and incorporates the definition of a derivative work provided in U.S. copyright law. OuterCurve Foundation (and the .NET Foundation) does not view the installer packages generated by WiX as falling within the definition of a derivative work, merely because they are produced using WiX. Thus, the installer packages generated by WiX will normally fall outside the scope of the MS-RL, and any of your source code, binaries, libraries, routines or other software components that are incorporated in installer packages generated by WiX can be governed by other licensing

> terms.

The full text of the MS-RL license is reproduced below. It can also be found in the LICENSE.TXT file included with the source code.

## Microsoft Reciprocal License (MS-RL)

This license governs use of the accompanying software. If you use the software, you accept this license. If you do not accept the license, do not use the software.

1. Definitions

   The terms "reproduce," "reproduction," "derivative works," and "distribution" have the same meaning here as under U.S. copyright law.

   A "contribution" is the original software, or any additions or changes to the software.

   A "contributor" is any person that distributes its contribution under this license.

   "Licensed patents" are a contributor's patent claims that read directly on its contribution.

2. Grant of Rights

   (A) Copyright Grant- Subject to the terms of this license, including the license conditions and limitations in section 3, each contributor grants you a non-exclusive, worldwide, royalty-free copyright license to reproduce its contribution, prepare derivative works of its contribution, and distribute its contribution or any derivative works that you create.

   (B) Patent Grant- Subject to the terms of this license, including the license conditions and limitations in section 3, each contributor grants you a non-exclusive, worldwide, royalty-free license under its licensed patents to make, have made, use, sell, offer for sale, import, and/or otherwise dispose of its contribution in the software or derivative works of the contribution in the

software.

3. Conditions and Limitations

(A) Reciprocal Grants- For any file you distribute that contains code from the software (in source code or binary format), you must provide recipients the source code to that file along with a copy of this license, which license will govern that file. You may license other files that are entirely your own work and do not contain code from the software under any terms you choose.

(B) No Trademark License- This license does not grant you rights to use any contributors' name, logo, or trademarks.

(C) If you bring a patent claim against any contributor over patents that you claim are infringed by the software, your patent license from such contributor to the software ends automatically.

(D) If you distribute any portion of the software, you must retain all copyright, patent, trademark, and attribution notices that are present in the software.

(E) If you distribute any portion of the software in source code form, you may do so only under this license by including a complete copy of this license with your distribution. If you distribute any portion of the software in compiled or object code form, you may only do so under a license that complies with this license.

(F) The software is licensed "as-is." You bear the risk of using it. The contributors give no express warranties, guarantees or conditions. You may have additional consumer rights under your local laws which this license cannot change. To the extent permitted under your local laws, the contributors exclude the implied warranties of merchantability, fitness for a particular purpose and non-infringement.

# Governance

This project is led and managed by a benevolent dictator. The benevolent dictator is responsible for the general strategic direction in addition to the day-to-day maintenance of the project. The community guides the decisions of the benevolent dictator through active engagement and contribution.

This project has also adopted the code of conduct defined by the Contributor Covenant to clarify expected behavior in our community. For more information see the .NET Foundation Code of Conduct.

## Roles and Responsibilities

### Benevolent dictator (project leads)

In the WiX Toolset the role of Benevolent Dictator (project leads) is shared between Rob Mensching and Bob Arnson. The project leads are expected to understand the community as a whole and strive to satisfy as many conflicting needs as possible, while ensuring that the project survives in the long term.

In many ways, the role of the benevolent dictator is less about dictatorship and more about diplomacy. The key is to ensure that, as the project expands, the right people are given influence over it and the community rallies behind the vision of the project leads.

Additionally, .NET Foundation staff considers the project leads as the primary point of contact or first point of contact for the project for purposes of business operations including domain registrations, and technical services (e.g. code-signing).

### Committers

Committers are contributors who have made sustained valuable contributions to the project and are now relied upon to both write code directly to the repository. In many cases they are programmers but it is also possible that they contribute in a different role. Typically, a committer will focus on a specific aspect of the project, and will bring a level of expertise and understanding that earns them the respect of the community and the project lead. The role of committer is not an official one; it is simply a position that influential members of the community will find themselves in as the project lead looks to them for guidance and support.

Committers have no authority over the overall direction of the project. However, they do have the ear of the project leads. It is a committer's job to ensure that the lead is aware of the community's needs and collective objectives, and to help develop or elicit appropriate contributions to the project. Often, committers are given informal control over their specific areas of responsibility, and are assigned rights to directly modify certain areas of the source code. That is, although committers do not have explicit decision-making authority, they will often find that their actions are synonymous with the decisions made by the leads.

**How to become a Committer:** Be a regular Contributor then be appointed by the Benevolent Dictator.

**Contributors**

Contributors are community members who submit pull requests for the project. These pull requests may be a one-time occurrence or occur over time. The expectation is that that contributors will submit pull requests that are small at first. Ideally larger or more complex contributions are sent once the contributor has built confidence in the quality of their pull requests with the community.

Before a contributor's first pull request is put into the repository they must sign a Contribution License Agreement. The pull request can be submitted and discussed but it cannot be committed to the repository without the appropriate paperwork in place.

**How to become a Contributor:** Participate in the project as a developer.

**Users**

Users are community members who have a need for the project. They are the most important members of the community: without them, the project would have no purpose. Anyone can be a user; there are no specific requirements.

Users should be encouraged to participate in the life of the project and the community as much as possible. User contributions enable the project team to ensure that they are satisfying the needs of those users. Common user activities include (but are not limited to):

- Advocating the use of the project

- Informing developers of the project strength and weaknesses from a new user's perspective
- Providing moral support (a 'thank you' goes a long way)
- Writing documentation
- Filing bug reports and feature requests
- Participating on the mailing lists and discussion forum

Users who continue to engage with the project and its community will often find themselves becoming more and more involved. Such users may then go on to become Contributors, as described above.

**How to become a User:** Use the WiX toolset and participate on the mailing lists and discussion forum.

# Contribution License Agreement

The WiX Toolset copyright is held by the .NET Foundation. Before we can accept any contributions you must have a Contribution License Agreement on file. Fortunately the process is very easy.

When you submit your first pull request, there will be a prompt to accept the CLA in the comments. After reading the CLA you can accept it by responding with a comment saying `I have read the CLA Document and I hereby sign the CLA`. This will only necessary for your first pull request.

So go out there! Participate in design discussions. Get your code written. Test it. And test it some more. Post the pull request for code review. Then respond to the CLA prompt and your contribution will be ready to be merged into the WiX Toolset.

Happy coding!

Edit this page