



# Open Source at Microsoft

The Windows Installer XML (WiX) toolset: A pioneering open source project at Microsoft helps developers build high-quality installation packages

Back in 1999, before XML, in practice, was the next big thing, Microsoft developer Rob Mensching decided to take on a side project using the technology. He took something he knew quite well—Windows Installer—and combined it with something he wanted to learn—XML. What he ended up with is the Windows Installer XML (WiX), a toolset that builds Windows installation packages from XML source code.

An active project licensed under the Common Public License on the Sourceforge.net repository since 2004, WiX also happens to be the first Microsoft tool released as open source. It is also widely used by development teams within Microsoft.

### **WiX: A match for the open source approach**

There are a number of reasons why it made sense to use an open source approach with WiX. As Mensching explains:

First off, WiX was never part of my job description at Microsoft. I work on the project in my free time.

Second, WiX is a very developer-oriented project and thus providing source code access increases the pool of available developers. As a development tool for developers, it was a natural fit for open source.

Tools for developers such as a compiler/linker that creates an installation package are suitable for open source development because the community is naturally going to be made up of developers. Mensching contrasts this with an “Open Office” project, for example. This wouldn’t be as strong a match for open source because a very large part of the user base

would not be able to contribute to the code because they are not developers.

Nonetheless, open source development for WiX does benefit users. Customers today expect a complete setup solution bundled with any product they purchase. And because setup is the very first user experience for any product, having a reliable setup is critical.

Mensching initially shared the WiX tool internally at Microsoft with other developers who were interested in using it to build their installation packages. He gave them the complete tool, source code and all. Soon, these developers were looking at the code, pointing out specifically where the bugs were, and eventually adding complete features to the code. By harnessing that collective feedback, the open source model within Microsoft was clearly benefiting Mensching and his small group of five developers also working on WiX as a “side project.”

Comments Mensching:

Many parts of the open source development process appeal to me.

Back in 1999 and 2000, I wanted to improve understanding within Microsoft of what the open source community was about, and I wanted to do this by providing an example.

### **Release early, release often**

Another reason WiX has been successful as an open source project is the “release early, release often” philosophy Mensching and other committed volunteer developers have taken.

Version 2.0 of the WiX toolset became available in April 2004, and it became stable in 2006.

Mensingh and his team still put out weekly releases, and version 3.0 is currently in the works. A group of developers meets on Thursday nights to write code, and they release a new build each Friday. Because the WiX project has no dedicated QA engineers, the team releases the code and makes adjustments each week based on feedback from members of the community. Members of the external community have provided what Mensching calls “great, continuous” feedback, and have even contributed code back to the project.

WiX has proven to be so popular with Microsoft development teams that many Microsoft software products, including Microsoft® SQL Server™ 2005 and the 2007 Microsoft® Office system, are packaged using WiX.

### **WiX integrates into the build process**

The WiX toolset provides a command line environment that developers can integrate into their build processes to build MSI and MSM setup packages. MSI and MSM are installer packages and merge modules, respectively.

Written in C#, the toolset requires the Microsoft® .NET Framework to run. WiX integrates with the usual process of creating applications. Instead of offering a GUI-based assembly of information, it uses a text file based on XML format to describe all the elements of the installation process. WiX can easily be made part of an automated application build process.

Additionally, developers can use WiX to integrate the setup development process with the overall application development process, instead of developing the installation after the fact. This helps prevent separate development teams from having to collect all the pieces of information from the original developers or

trying to extract them from the source code when the main application has already been released.

VisualSVN—a company that provides a simple way to manage changes to source code inside Visual Studio using Subversion—has taken advantage of WiX for these very reasons. Explains Danil Shopyrin of the VisualSVN team:

The WiX toolset provides a peaceful way to make an installation. With WiX, we take control of everything in our installer. Moreover, it's easy to integrate WiX in our development and build processes. The most exciting advantage is that installation is built from plain XML-based scripts that are very friendly for us. For example, it's easy to explore changes within installer using standard diff viewer. And, last but not at least, we receive the pure standard install package.

### **A measure of success for WiX**

WiX is now used by Microsoft product groups, community developers and businesses incorporating WiX technologies into their products, and leading commercial open-source products alike. In its three years on Sourceforge.net, WiX has been downloaded 480,000 times, and it currently averages approximately 16,000 downloads a month.

The setup of the 2007 Microsoft Office system was developed entirely with WiX. In Microsoft® Office Enterprise 2007 alone, the installation touches over 4,600 files and over 53,000 registry keys. See the table below to see the impact WiX has had on other Microsoft products.

WiX Impact on Microsoft Products	
Microsoft product	WiX touches
Microsoft Office Enterprise 2007	Over 4,600 files and over 53,000 registry keys
Microsoft® Office Visio 2007	Over 1,000 files and over 10,500 registry keys
Microsoft SQL Server 2005 engine	Almost 400 files and almost 30,000 registry keys
Microsoft SQL Server 2005 tools	Over 2,000 files and almost 28,000 registry keys
Windows Live Messenger®	Almost 100 files and almost 1,000 registry keys
Windows Live Writer®	Almost 50 files and almost a 150 registry keys
Microsoft® Silverlight™ 1.0	3 files and almost 90 registry keys

Says Mensching, “Source code is just an inanimate object. Community is what really makes it all work. I pushed for the WiX toolset to be released as an open source project because I wanted to build a community around solving software installation problems.”

Additionally, numerous other Microsoft teams including Windows and Visual Studio use the tool.

Outside of Microsoft, many developers in varied industries, including six major banks, are using the tool, as does MySQL, the popular open source database software. And several companies have successfully built businesses on top of WiX. For example, Indigo Rose’s Setup Factory for Windows Installer relies on the WiX toolset.

But perhaps the greatest measure of the success for WiX is the thriving community around it. Mensching says that about 20 e-mail messages are posted daily asking “how-to” questions, and several members in the external WiX community have been answering them regularly. One developer even created and shared a 70-page WiX tutorial.

## Copyright

Information in this document, including URL and other Internet Web site references, is subject to change without notice and is provided for informational purposes only. The entire risk of the use or results from the use of this document remains with the user, and Microsoft Corporation makes no warranties, either express or implied. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in examples herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

© 2007 Microsoft Corporation. All Rights Reserved.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Microsoft, Microsoft SQL Server 2005, Microsoft .NET Framework, 2007 Microsoft Office system, Microsoft Office Enterprise 2007, Microsoft Office Visio 2007, Microsoft SQL Server 2005, Microsoft Silverlight, Windows Live Messenger, Windows Live Writer are either registered trademarks or trademarks of

Microsoft Corporation in the United States and/or other countries.

All other trademarks are property of their respective owners.