



EBook Gratis

APRENDIZAJE

wix

Free unaffiliated eBook created from
Stack Overflow contributors.

#wix

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con wix.....	2
Observaciones.....	2
¿Qué es WiX?.....	2
¿Cómo funciona WiX?.....	2
Requisitos del sistema WiX.....	2
Versiones.....	3
Examples.....	3
Instalación o configuración.....	3
Advertencias.....	3
Detalles.....	4
Configuración simple.....	4
Creditos.....	8

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [wix](#)

It is an unofficial and free wix ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official wix.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con wix

Observaciones

¿Qué es WiX?

El [conjunto de herramientas de WiX](#) permite a los desarrolladores crear instaladores para Windows Installer, el motor de instalación de Windows. Es de [código abierto](#) y parte de la [Fundación .NET](#).

El núcleo de WiX es un conjunto de herramientas de compilación que crean paquetes de Windows Installer usando los mismos conceptos de compilación que el resto de su producto: el código fuente se compila y luego se vincula para crear ejecutables; en este caso, los paquetes de instalación .exe, los paquetes de instalación .msi, los módulos de fusión .msm y los parches .msp. Las herramientas de construcción de línea de comandos de WiX funcionan con cualquier sistema de construcción automatizado. Además, MSBuild es compatible desde la línea de comandos, Visual Studio y Team Build.

WiX incluye varias extensiones que ofrecen funcionalidad más allá de la de Windows Installer. Por ejemplo, WiX puede instalar sitios web de IIS, crear bases de datos de SQL Server y registrar excepciones en el Firewall de Windows, entre otros.

Con Burn, el bootstrapper de WiX, puede crear paquetes de configuración que instalen requisitos previos como .NET Framework y otros tiempos de ejecución junto con su propio producto. Burn le permite descargar paquetes o combinarlos en un solo archivo .exe descargable.

El WiX SDK incluye bibliotecas administradas y nativas que facilitan la escritura de código que funciona con Windows Installer, incluidas acciones personalizadas en C # y C ++.

¿Cómo funciona WiX?

El código fuente de WiX está escrito en formato XML con una extensión de archivo .wxs. Las herramientas de WiX siguen el modelo tradicional de compilación y enlace utilizado para crear ejecutables a partir del código fuente.

En el momento de la compilación, los archivos fuente de WiX se validan con respecto al esquema central de WiX y luego los procesa un preprocesador, compilador y enlazador para crear el resultado final. Hay un conjunto de herramientas WiX que se pueden usar para producir diferentes tipos de salida.

Requisitos del sistema WiX

WiX soporta tanto .NET 3.5 y 4.0 y posteriores. El soporte MSBuild de WiX requiere .NET 3.5, que no se instala de forma predeterminada en Windows 8 y Windows Server 2012 y posteriores.

En la próxima versión de WiX (v3.11), se requerirá .NET 4.0; la construcción con .NET 3.5 ya no será compatible.

Versiones

Versión	Fechas de lanzamiento
3.11	2017-05-05
3.10.3	2016-07-04
3.9 R2	2015-01-21
3.8	2013-11-28
3.7	2012-12-24
3.6	2012-09-03
3.5	2011-01-19
3.0	2009-06-19
2.0	2007-10-05

Examples

Instalación o configuración

Descargue e instale el conjunto de herramientas de WiX desde wixtoolset.org .

El instalador de WiX Toolset proporciona también la integración con [Visual Studio](#) , después de la instalación, debería poder crear proyectos específicos de WiX.

Advertencias

Se necesitan derechos de administrador.

Algunas versiones de WiX son compatibles solo con una versión particular de Visual Studio:

- V3.11 y luego no incluye las extensiones para Visual Studio, debes descargar las extensiones para tus versiones de Visual Studio
- V3.10 e inferior no funciona con Visual Studio 2017, pero incluye las plantillas de proyecto

para Visual Studio

Detalles

El instalador está construido con WiX y presenta una ventana inusual:



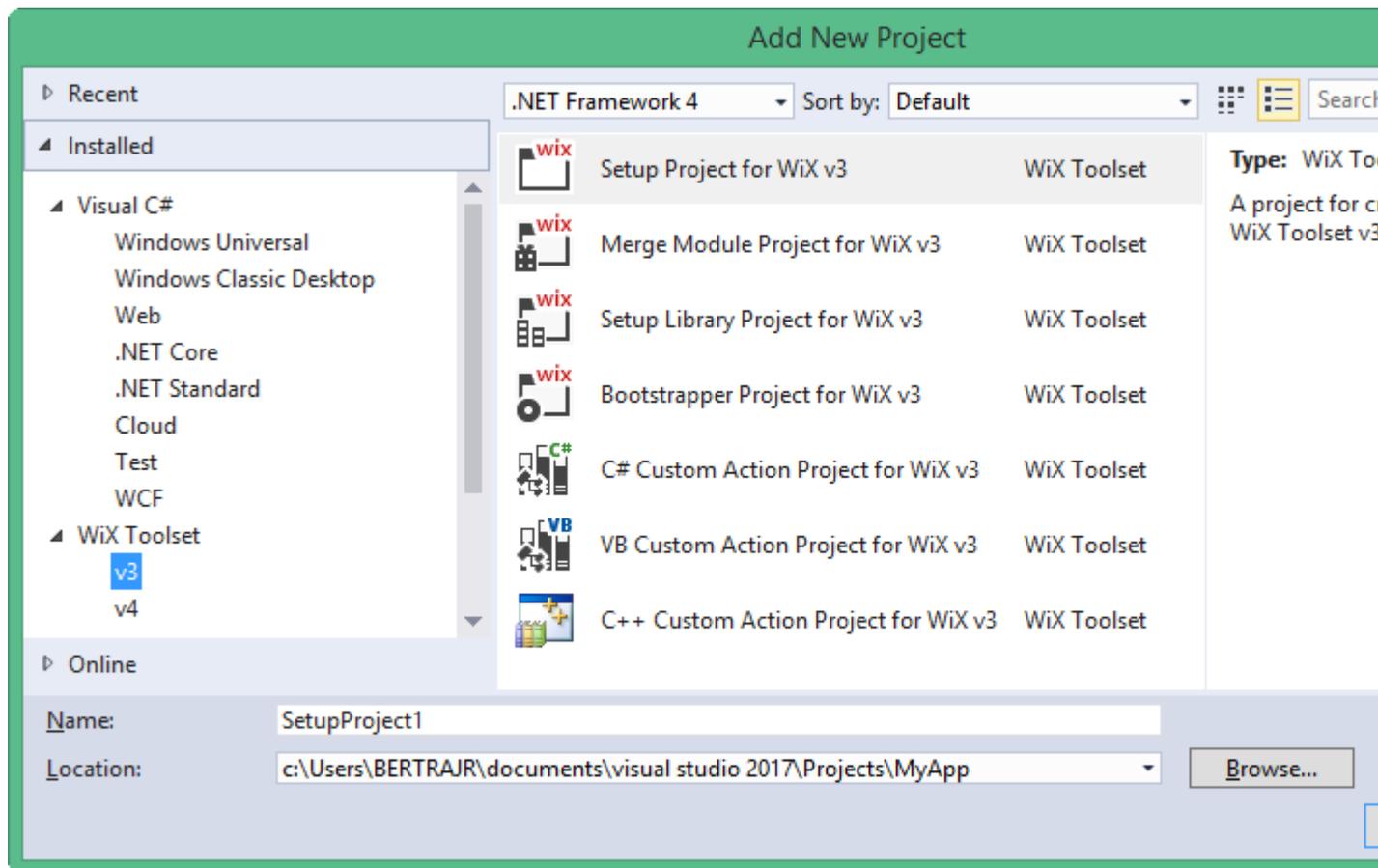
Está compuesto por 6 partes:

- WiX Toolset: muestra la versión instalada por el instalador e [inicia el sitio web de Wix Toolset](#) con un clic
- Licencia: mostrar la licencia
- Instalar: iniciar la instalación
- Hasta la fecha: verifique si hay una nueva versión disponible
- Noticias: lanza las [novedades de WiX](#).
- Salir: cerrar el instalador

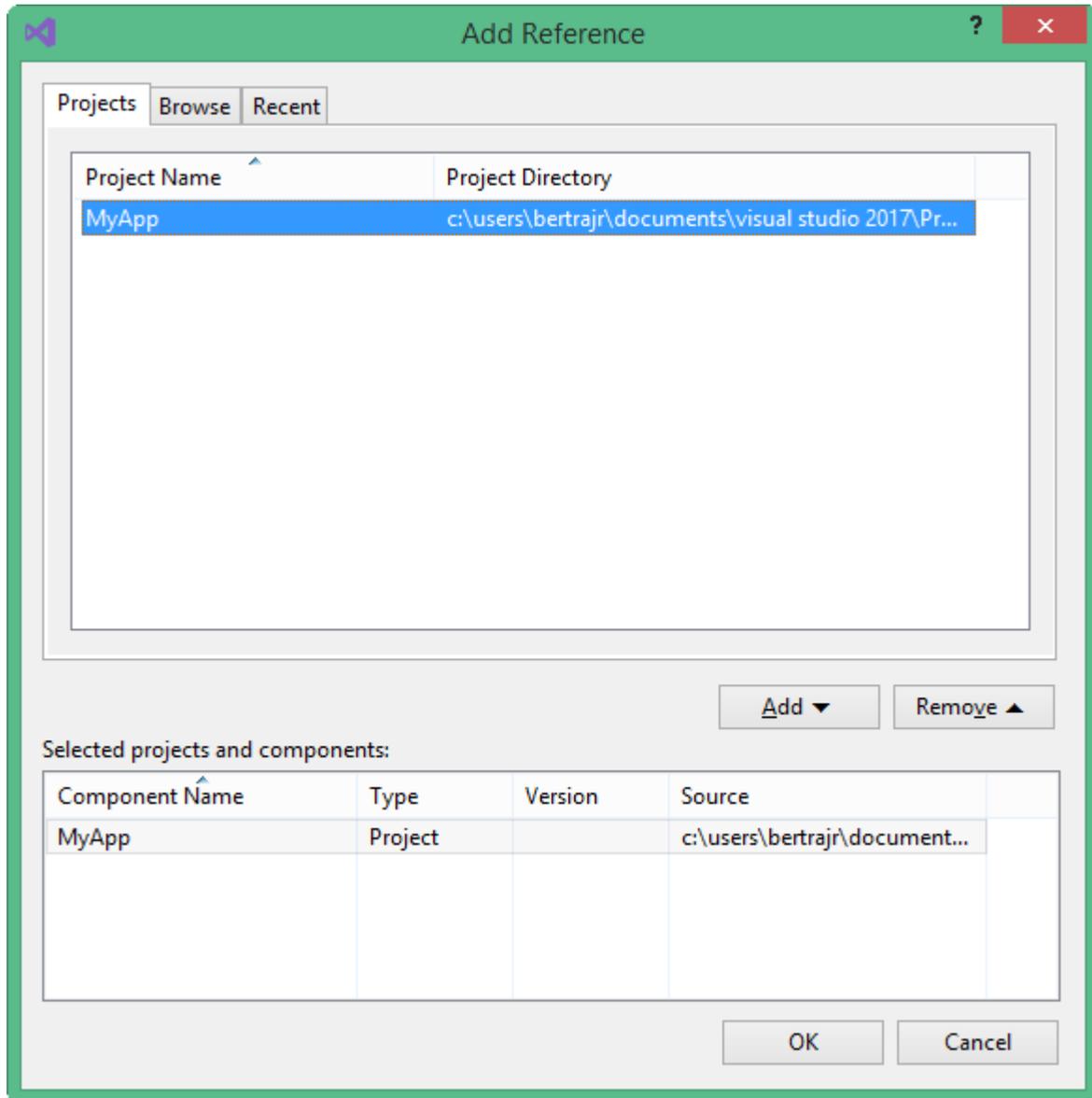
Configuración simple

En este ejemplo, se supone que ya existe una solución con una aplicación llamada `MyApp` .

- Agrega un nuevo proyecto a la solución:



- En el proyecto de instalación, agregue una nueva referencia a `MyApp` desde la pestaña Proyectos:



- En el archivo `Product.wxs` , `Product.wxs` el atributo `Manufacturer` del nodo `Product` con `HelloWorld` :

```
<Product Id="*" Name="MyApp.Setup" Language="1033" Version="1.0.0.0" Manufacturer="HelloWorld" UpgradeCode="52f2c69b-5901-4d18-bb96-8c1c86cd1a3e">
```

En el nodo `Fragment` que contiene los nodos del `Directory` , envuelva el último con un nuevo `Directory` :

```
<Directory Id="ManufacturerFolder" Name="!(bind.property.Manufacturer)">
  <Directory Id="INSTALLFOLDER" Name="MyApp.Setup" />
</Directory>
```

En el nodo `ComponentGroup` , elimine el comentario de los nodos comentados y elimine `TODO` luego agregue un nodo de `File` en el `Component` :

```
<File Source="$ (var.MyApplication.TargetPath) " />
```

El atributo `Fuente` especifica dónde encontrar el archivo para empaquetar durante la compilación.

En lugar de valores de código rígido para estos atributos en nuestro código fuente, usamos las variables del preprocesador de WiX que se pasan al compilador de WiX.

- Construye el proyecto WiX.

¡Eso es! Ahora tiene un instalador en funcionamiento que instala y desinstala la aplicación.

Product.wxs **completo de** Product.wxs :

```
<?xml version="1.0" encoding="UTF-8"?>
<Wix xmlns="http://schemas.microsoft.com/wix/2006/wi">
  <Product Id="*" Name="MyApp.Setup" Language="1033" Version="1.0.0.0"
  Manufacturer="HelloWorld" UpgradeCode="52f2c69b-5901-4d18-bb96-8c1c86cd1a3e">
    <Package InstallerVersion="200" Compressed="yes" InstallScope="perMachine" />

    <MajorUpgrade DowngradeErrorMessage="A newer version of [ProductName] is already
  installed." />
    <MediaTemplate />

    <Feature Id="ProductFeature" Title="MyApp.Setup" Level="1">
      <ComponentGroupRef Id="ProductComponents" />
    </Feature>
  </Product>

  <Fragment>
    <Directory Id="TARGETDIR" Name="SourceDir">
      <Directory Id="ProgramFilesFolder">
        <Directory Id="ManufacturerFolder" Name="!(bind.property.Manufacturer)">
          <Directory Id="INSTALLFOLDER" Name="MyApp.Setup" />
        </Directory>
      </Directory>
    </Directory>
  </Fragment>

  <Fragment>
    <ComponentGroup Id="ProductComponents" Directory="INSTALLFOLDER">
      <Component Id="ProductComponent">
        <File Source="$ (var.MyApp.TargetPath)" />
      </Component>
    </ComponentGroup>
  </Fragment>
</Wix>
```

Lea Empezando con wix en línea: <https://riptutorial.com/es/wix/topic/4487/empezando-con-wix>

Creditos

S. No	Capítulos	Contributors
1	Empezando con wix	Community , cristallo , Jérémie Bertrand