components, you need to use specific WiX attributes and conditions. The core principle involves building an installer that targets the 64-bit platform and using WiX variables like `VersionNT64` to control conditional logic.

1. Configure the WiX Project for 64-bit

First, ensure your WiX setup project in Visual Studio is set to build for a 64-bit platform.

- In **Visual Studio 2022**, right-click your WiX project in the **Solution Explorer** and select **Properties**.
- Go to the **Build** tab (or the equivalent in the WiX v3/v4 properties).
- Set the **Platform** to `x64`. If `x64` is not available, you may need to open the **Configuration Manager** (from the Build menu or by right-clicking the Solution in Solution Explorer) and add the `x64` platform for both the solution and the WiX project.
- Alternatively, in the `.wixproj` file itself, you can add `<InstallerPlatform>x64</InstallerPlatform>` to a property group.

2. Use `ProgramFiles64Folder` in `.wxs`

When targeting 64-bit systems, use the correct Directory identifiers to ensure files are installed in the `C:\Program Files\` directory (instead of `C:\Program Files (x86)\`):

xml

```xml
<Directory Id="TARGETDIR" Name="SourceDir">
    <!-- Use ProgramFiles64Folder for 64-bit installations -->
    <Directory Id="ProgramFiles64Folder">
        <Directory Id="INSTALLFOLDER" Name="My 64-bit App" />
    </Directory>
</Directory>
```

3. Implement Launch Conditions

Add the following inside your `<Product>` element in your `.wxs` file:

xml

```xml
<Condition Message="This application is only supported on a 64-bit operating system
    VersionNT64
</Condition>
```

This condition checks if `VersionNT64` is set (meaning it's a 64-bit OS). If it's not set, the installer displays the custom error message and stops the installation.

4. Conditionally Install Specific Files (Advanced)

If you are using a single installer (often a WiX Burn bundle) to conditionally install certain components or files based on the target machine's architecture, you can use component conditions.

For example, to install a specific 64-bit DLL only on 64-bit systems:

xml

```xml
<Component Id="My64BitComponent" Guid="*">
    <File Id="My64BitDLL" Source="path\to\your\64bit.dll" KeyPath="yes" />
    <!-- This condition ensures the component is only installed on 64-bit OS -->
    <Condition>VersionNT64</Condition>
</Component>
```

You would then include this component in a feature. This approach is generally used within a WiX **Bundle** (`.wixproj` type), which can manage both 32-bit and 64-bit MSI packages from a single bootstrapper executable.

Launch condition to ensure that 64-bit installer is used on ...

Aug 11, 2011 — Related * WiX: Prevent 32-bit installer from running on 64-bit Windows. * Using Wix to create 32bit and 64bit installers from one ...

Stack Overflow

Mar 16, 2010 — Related * Using Wix to create 32bit and 64bit installers from one .wxs file. * Launch condition to ensure that 64-bit installer is...

Stack Overflow

Is there a way to make a combined 32-bit and 64-bit installer in Wix?

Mar 20, 2014 — Related * How to implement single installer for 32 & 64 platforms. * WiX: Prevent 32-bit installer from running on 64-bit Windows.

Stack Overflow

Show all

Dive deeper in AI Mode

Videos

4:52

How To Install WiX Toolset 3.14.1 on Windows 11 x64

YouTube · Ken

Dec 19, 2024

Install Wix Toolset 3.0 or later to create EXE MSI files using JPackage.

5:10

How to Create an MSI installer with WiX Toolset

YouTube · Advanced Installer & PacKit

Feb 17, 2025

Learn to create an MSI **installer** with **Wix** Toolset, including adding project output, modifying product details, and adding shortcuts.

19:13

How to create installation package with WiX Toolset

YouTube · EN64

Feb 6, 2025