

This is documentation for WiX Toolset **v3**, which is no longer actively maintained.

For up-to-date documentation, see the **[latest version](#)** (v4.0).

Version: v3

How To: Check for .NET Framework Versions

When installing applications written using managed code, it is often useful to verify that the user's machine has the necessary version of the .NET Framework prior to installation. The WiX support for detecting .NET Framework versions is included in a WiX extension, WixNetFxExtension. This how to describes using the WixNetFxExtension to verify .NET Framework versions at install time. For information on how to install the .NET Framework during your installation see [How To: Install the .NET Framework Using Burn](#).

Step 1: Add WixNetFxExtension to your project

You must add the WixNetFxExtension to your project prior to use. If you are using WiX on the command line, you need to add the following to your candle and light command lines:

```
-ext WixNetFxExtension
```

If you are using WiX in Visual Studio, you can add the extension using the Add Reference dialog:

1. Open your WiX project in Visual Studio.

2. Right click on your project in Solution Explorer and select **Add Reference...**
3. Select the **WixNetFxExtension.dll** assembly from the list and click Add.
4. Close the Add Reference dialog.

Step 2: Add WixNetFxExtension's namespace to your project

Once the extension is added to your project, you need to add its namespace to your project so you can access the appropriate WiX elements. To do this, modify the top-level **Wix** element in your project by adding the following attribute:

```
xmlns:netfx="http://schemas.microsoft.com/wix/NetFxExtension"
```

A complete Wix element with the standard namespace and WixNetFxExtension's namespace added looks like this:

```
<Wix xmlns="http://schemas.microsoft.com/wix/2006/wi"  
    xmlns:netfx="http://schemas.microsoft.com/wix/NetFxExtension">
```

Step 3: Reference the required properties in your project

WixNetFxExtension defines **properties for all current versions of the .NET Framework**, including service pack levels. To make these properties available to your installer, you need to reference them using the **PropertyRef** element. For each property you want to use, add the corresponding PropertyRef to your project. For example, if you are interested in detecting .NET Framework 2.0 add the following:

```
<PropertyRef Id="NETFRAMEWORK20"/>
```

Step 4: Use the pre-defined properties in a condition

Once the property is referenced, you can use it in any WiX condition statement. For example, the following condition blocks installation if .NET Framework 2.0 is not installed.

```
<Condition Message="This application requires .NET Framework 2.0. Please install the .NET Framework then run this installer again.">  
  <![CDATA[Installed OR NETFRAMEWORK20]]>  
</Condition>
```

Installed is a Windows Installer property that ensures the check is only done when the user is installing the application, rather than on a repair or remove. The NETFRAMEWORK20 part of the condition will pass if .NET Framework 2.0 is installed. If it is not set, the installer will display the error message then abort the installation process.

To check against the service pack level of the framework, use the *_SP_LEVEL properties. The following condition blocks installation if .NET Framework 3.0 SP1 is not present on the machine.

```
<Condition Message="This application requires .NET Framework 3.0 SP1. Please install the .NET Framework then run this installer again.">  
  <![CDATA[Installed OR (NETFRAMEWORK30_SP_LEVEL and NOT NETFRAMEWORK30_SP_LEVEL = "#0")]]>  
</Condition>
```

As with the previous example, Installed prevents the check from running when the user is doing a repair or remove. The NETFRAMEWORK30_SP_LEVEL property is set to "#1" if Service Pack 1 is present. Since there is no way to do a numerical

comparison against a value with a # in front of it, the condition first checks to see if the NETFRAMEWORK30_SP_LEVEL is set and then confirms that it is set to a number. This will correctly indicate whether any service pack for .NET 3.0 is installed.

 [Edit this page](#)