> This is documentation for WiX Toolset **v3**, which is no longer actively maintained.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: Create a Shortcut on the Start Menu

When installing applications it is a common requirement to place a shortcut on the user's Start Menu to provide a launching point for the program. This how to walks through how to create a shortcut on the start menu. It assumes you have a WiX source file based on the concepts described in How To: Add a file to your installer.

Step 1: Define the directory structure

Start Menu shortcuts are installed in a different directory than regular application files, so modifications to the installer's directory structure are required. The following WiX fragment should be placed inside a Directory element with the TARGETDIR ID and adds directory structure information for the Start Menu:

```
<Directory Id="ProgramMenuFolder">
    <Directory Id="ApplicationProgramsFolder" Name="My Application Name"/>
</Directory>
```

The ProgramMenuFolder Id is a standard Windows Installer property that points to the Start Menu folder on the target machine. The second Directory element creates a subfolder on the Start Menu called My Application Name, and gives it an id for use later in the WiX project.

Step 2: Add the shortcut to your installer package

A shortcut is added to the installer using three elements: a Component element to specify an atomic unit of installation, a Shortcut element to specify the shortcut that should be installed, and a RemoveFolder element to ensure proper cleanup when your application is uninstalled.

The following sample uses the directory structure defined in Step 1 to create the Start Menu shortcut.

```
<DirectoryRef Id="ApplicationProgramsFolder">
    <Component Id="ApplicationShortcut" Guid="PUT-GUID-HERE">
        <Shortcut Id="ApplicationStartMenuShortcut"
                  Name="My Application Name"
                  Description="My Application Description"
                  Target="[#myapplication.exe]"
                  WorkingDirectory="APPLICATIONROOTDIRECTORY"/>
        <RemoveFolder Id="CleanUpShortCut" Directory="ApplicationProgramsFolder" On="uninstall"/>
        <RegistryValue Root="HKCU" Key="Software\MyCompany\MyApplicationName" Name="installed"
Type="integer" Value="1" KeyPath="yes"/>
    </Component>
</DirectoryRef>
```

The DirectoryRef element is used to refer to the directory structure created in step 1. By referencing the ApplicationProgramsFolder directory the shortcut will be installed into the user's Start Menu inside the My Application Name folder.

Underneath the DirectoryRef is a single Component to group the elements used to install the Shortcut. The first element is Shortcut and it creates the actual shortcut in the Start Menu. The Id attribute is a unique id for the shortcut. The Name attribute is the text that will be displayed in the Start Menu. The description is an optional attribute for an additional application description. The Target attribute points to the executable to launch on disk. Notice how it references the full path using the `[#FileId]` syntax where `myapplication.exe` was previously defined. The WorkingDirectory attribute sets the working directory for the shortcut.

To set an optional icon for the shortcut you need to first include the icon in your installer using the Icon element, then reference it using the Icon attribute on the Shortcut element.

In addition to creating the shortcut the component contains two other important pieces. The first is a RemoveFolder element, which ensures the ApplicationProgramsFolder is correctly removed from the Start Menu when the user uninstalls the application. The second creates a registry entry on install that indicates the application is installed. This is required as a Shortcut cannot serve as the KeyPath for a component when installing non-advertised shortcuts for the current users. For more information on creating registry entries see How To: Write a registry entry during installation.

Step 3: Tell Windows Installer to install the shortcut

After defining the directory structure and listing the shortcuts to package into the installer, the last step is to tell Windows Installer to actually install the shortcut. The Feature element is used to do this. The following snippet adds a reference to the shortcut component, and should be inserted inside a parent Feature element.

```
<ComponentRef Id="ApplicationShortcut" />
```

The ComponentRef element is used to reference the component created in Step 2 via the Id attribute.

The Complete Sample

The following is a complete sample that uses the above concepts. This example can be inserted into a WiX project and compiled, or compiled and linked from the command line, to generate an installer.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Wix xmlns="http://schemas.microsoft.com/wix/2006/wi">
    <Product Id="*" UpgradeCode="PUT-GUID-HERE" Version="1.0.0.0" Language="1033" Name="My Application
Name" Manufacturer="My Manufacturer Name">
        <Package InstallerVersion="300" Compressed="yes"/>
        <Media Id="1" Cabinet="myapplication.cab" EmbedCab="yes" />

        <Directory Id="TARGETDIR" Name="SourceDir">
            <Directory Id="ProgramFilesFolder">
                <Directory Id="APPLICATIONROOTDIRECTORY" Name="My Application Name"/>
            </Directory>
            <!-- Step 1: Define the directory structure -->
            <Directory Id="ProgramMenuFolder">
                <Directory Id="ApplicationProgramsFolder" Name="My Application Name"/>
            </Directory>
        </Directory>

        <DirectoryRef Id="APPLICATIONROOTDIRECTORY">
            <Component Id="myapplication.exe" Guid="PUT-GUID-HERE">
                <File Id="myapplication.exe" Source="MySourceFiles\MyApplication.exe" KeyPath="yes"
Checksum="yes"/>
            </Component>
            <Component Id="documentation.html" Guid="PUT-GUID-HERE">
                <File Id="documentation.html" Source="MySourceFiles\documentation.html" KeyPath="yes"/>
            </Component>
```

```xml
            </DirectoryRef>

            <!-- Step 2: Add the shortcut to your installer package -->
            <DirectoryRef Id="ApplicationProgramsFolder">
                <Component Id="ApplicationShortcut" Guid="PUT-GUID-HERE">
                    <Shortcut Id="ApplicationStartMenuShortcut"
                              Name="My Application Name"
                              Description="My Application Description"
                              Target="[#myapplication.exe]"
                              WorkingDirectory="APPLICATIONROOTDIRECTORY"/>
                    <RemoveFolder Id="ApplicationProgramsFolder" On="uninstall"/>
                    <RegistryValue Root="HKCU" Key="Software\MyCompany\MyApplicationName" Name="installed"
Type="integer" Value="1" KeyPath="yes"/>
                </Component>
            </DirectoryRef>

            <Feature Id="MainApplication" Title="Main Application" Level="1">
                <ComponentRef Id="myapplication.exe" />
                <ComponentRef Id="documentation.html" />
                <!-- Step 3: Tell WiX to install the shortcut -->
                <ComponentRef Id="ApplicationShortcut" />
            </Feature>
        </Product>
</Wix>
```

✏️ Edit this page