## ReadMe: WPF Application Installer

This document provides additional information and tips for the installer you've created for your WPF application.

### Included Files

When you build your setup project, you will get the following important files in the output directory (usually Debug or Release):

* **setup.exe**: This is a bootstrapper that can check for and install any necessary prerequisites (like the .NET runtime) before running the main installer. It is generally the file you should distribute to your users.
* **.msi file**: This is the Windows Installer package. It contains all of your application's files and the logic to install them. The setup.exe will run this file.

### Adding an Application Icon

To give your installed application and its shortcuts a custom icon:

1. Add your .ico file to your WPF application project.
2. In your WPF project's properties, under the "Application" tab, set the icon.
3. When you create a shortcut in the setup project, it should automatically pick up the icon from the application's primary output.

[09-21-2025, MA, Added note: It is not as simple as described above. The icon file must be added to the Application Folder of the installer, and the icon must be set for the shortcuts to this icon file. This is done in the shortcut’s Properties window.]

### Specifying Prerequisites

Your WPF application will require a specific version of the .NET runtime to be installed on the user's machine. You can configure your installer to check for this.

1. Right-click on your setup project in the Solution Explorer and select **Properties**.
2. In the property pages, click the **Prerequisites...** button.
3. In the Prerequisites dialog, you can select the required .NET Desktop Runtime.
4. Choose whether to download the prerequisites from the component vendor's website or from a location you specify.

### Custom Actions

For more advanced installation scenarios, you can use "Custom Actions." These allow you to run an executable, DLL, or script during the installation process.

1. Right-click on your setup project and select **View > Custom Actions**.
2. You can then add custom actions to the Install, Commit, Rollback, or Uninstall phases.

This is useful for tasks like:

* Creating a database during installation.
* Modifying configuration files.
* Running a licensing or registration program.

By following the step-by-step guide and keeping these tips in mind, you can create a professional and reliable installer for your WPF application.