🏠 WiX tools and concepts WiX extensions and custom actions

WiX extensions and custom actions

WiX extends the functionality of Windows Installer by providing *WiX Extensions* that typically include compiler extensions to provide custom elements and attributes in your WiX authoring and custom actions to implement the extended functionality.

The following WiX extensions are provided by the WiX team:

Extension	Documentation
WixToolset.Bal.wixext	Bal schema
WixToolset.ComPlus.wixext	Complus schema
WixToolset.Dependency.wixext	Dependency schema
WixToolset.DifxApp.wixext	Difxapp schema
WixToolset.DirectX.wixext	Directx schema
WixToolset.Firewall.wixext	Firewall schema
WixToolset.Http.wixext	Http schema

Extension	Documentation
WixToolset.Iis.wixext	Iis schema
WixToolset.Msmq.wixext	Msmq schema
WixToolset.Netfx.wixext	Netfx schema
WixToolset.PowerShell.wixext	Powershell schema
WixToolset.Sql.wixext	Sql schema
WixToolset.UI.wixext	UI schema
WixToolset.Util.wixext	Util schema
WixToolset.VisualStudio.wixext	Vs schema

Using extensions

To use a WiX extension during the build of your project, you need to load the extension and reference its namespace in your WiX authoring.

Loading extensions using the Wix.exe command-line tool

To use a WiX extension when using the Wix.exe command-line .NET tool, you need to:

1. Add the extension to the WiX extension cache.
2. Use the extension when building.

For example:

```
wix extension add -g WixToolset.Util.wixext
wix extension add -g WixToolset.Bal.wixext
wix build Bundle.wxs Bundle.en-us.wxl -ext WixToolset.Util.wixext -ext WixToolset.Bal.wixext
```

Loading extensions in a .wixproj MSBuild project

WiX extensions are available as NuGet packages on NuGet.org. You can reference them as `PackageReference` items in your .wixproj MSBuild projects:

```
<ItemGroup>
  <PackageReference Include="WixToolset.Bal.wixext" />
  <PackageReference Include="WixToolset.Util.wixext" />
</ItemGroup>
```

You might need to specify a version for WiX extension packages (such as when they're prerelease packages):

```
<ItemGroup>
  <PackageReference Include="WixToolset.Bal.wixext" Version="5.0.0-preview.1" />
  <PackageReference Include="WixToolset.Util.wixext" Version="5.0.0-preview.1" />
</ItemGroup>
```

You need to restore package references. When using the `dotnet build` command, restore is automatic. When using .NET Framework MSBuild, you can use the `-Restore` switch on your `msbuild` command line to run package restore before building.

Declare the extension namespace

When you need to use a custom element from a WiX extension in your WiX authoring, you need to use the namespace of that extension. For example, the WixToolset.VisualStudio.wixext extension uses the `http://wixtoolset.org/schemas/v4/wxs/vs` namespace for its elements so to use the WixToolset.VisualStudio.wixext extension element `FindVisualStudio`, you might have authoring that looks like this:

```xml
<Wix
    xmlns="http://wixtoolset.org/schemas/v4/wxs"
    xmlns:vs="http://wixtoolset.org/schemas/v4/wxs/vs">

  <Package>
    <vs:FindVisualStudio />

    <PropertyRef Id="VS2022_ROOT_FOLDER" />
  </Package>
</Wix>
```

Edit this page