FIREGIANT (/)

Products (/products/)   Pricing (/services/)   Documentation (/docs/)   Blog (/blog/)

Contact Us (/contact-us/)

🏠 Home (/) / WiX Toolset v3 Tutorial (/docs/wix/v3/tutorial/)
/ Getting Started (/docs/wix/v3/tutorial/getting-started/) / Where to Install?

Where to Install?

Standalone applications will be installed into their own
folders under `Program Files` ---we've already seen how
to accomplish this. But plugins, add-ons or similar
additional products, not meant for standalone use but to
accompany other programs already installed on the
system (either your own or some third party) has to learn
when and where to install themselves first. Asking the

user to provide this information would be both inelegant and possibly dangerous in many cases, thus we need means of querying the registry, to consult .ini files already on the system or to look for actual folders and files to determine what to do.

When we look for any of those items, the result will be stored in a property (a string variable). So, we start by specifying the property (note that the `Id` we use is the same name we've already used in our first sample, denoting the target folder we install to). Inside the `Property` tag, we launch a registry search. The attributes speak for themselves:

```
<Property Id="INSTALLDIR">
    <RegistrySearch Id='AcmeFoobarRegistry' Type
='raw' Root='HKLM' Key='Software\Acme\Foobar 1.0
' Name='InstallDir' />
</Property>
```

If the registry search was successful (that is, the registry entry specified does exist), its value will be assigned to our `INSTALLDIR` property, ready to be used for our purposes. To check this out, add this line after the `Media` tag in our previous sample and save it to SampleRegistry.wxs (or, just download it (/system/files /samples/SampleRegistry.zip) again). Compile it but before you start the installation, go into the registry and create the `HKEY_LOCAL_MACHINE\SOFTWARE\Acme\Foobar 1.0` key. Create a new string value named `InstallDir` and set it

About FireGiant

Now you can install with confidence. FireGiant arms you with accurate support from the creators of WiX.

Because there are no questions we can't answer or problems we can't solve, you can unleash the full functionality of WiX without a doubt.

to an empty folder you've just created anywhere on your system. Run the installer with logging enabled.

If you did everything right, our three sample files will appear in this new folder. Also note the shortcuts (in the Start Menu and on the Desktop) to point to this new location now.

Similar information can come from other sources as well. Let's assume a `\Windows\SampleRegistry.ini` file (you can only read these files if they are located in the system folder):

```
[Sample]
InstallDir=C:\InstallHere
```

Replace the previous section with this new one:

```
<Property Id="INSTALLDIR">
    <IniFileSearch Id='AcmeFoobarIniFile' Type='directory' Name='SampleRegistry.ini' Section='Sample' Key='InstallDir' />
</Property>
```

There might be cases when simply knowing the folder is not enough: you have to look into the folder and make sure a given file exists there. `Depth=n` can be used to instruct the installer to look *n* levels deeper than the specified `Path`. Zero or a missing `Depth` attribute means only to look in the specified folder, not below it. We use

square brackets in `Path` to tell the installer to use the value of the `INSTALLDIR` property---bracketed names will be looked up and if found, replaced with their actual value. If not found, the string will remain unchanged.

```
<Property Id="FILEEXISTS">
    <DirectorySearch Id="CheckFileDir" Path="[IN
STALLDIR]" Depth="0">
        <FileSearch Id="CheckFile" Name="Lookfo
r.txt" />
    </DirectorySearch>
</Property>
```

If the file has been found, its full path will be assigned to the `FILEEXISTS` property, otherwise it will be left unassigned. You can check this if you build this sample (you need both the previous `RegistrySearch` and this fragment) and then run it with logging enabled. Observe that if you first put the file `Lookfor.txt` into the folder specified in the registry, the log will contain a reference to `FILEEXISTS`, with the full path of the file as its value.

Although we don't yet have a user interface, it's already worth noting that properties meant to receive their value from any interaction with the user, passing that value to the installation logic (eg. destination folders or features selected by the user) need to be *public* properties. To ensure their public status, their name has to be in all uppercase letters.

It is also worth noting that Windows Installer has better ways of upgrading your product than to look for specific registry entries of the previous installation. You should use these features whenever you can---but you need some patience until we reach them during our discussion.

‹ Useful Extras (/docs/wix/v3/tutorial/getting-started/useful-extras/)

up (/docs/wix/v3/tutorial/getting-started/)

Conditional Installation › (/docs/wix/v3/tutorial/getting-started/conditional-installation/)

CONTACT US (/contact-us/) | BLOG (/blog/)

PRODUCTS (/products/)

› HeatWave (/docs/heatwave/)

› HeatWave Build Tools (/products/heatwave-build-tools/)

› FireGiant WiX v3 (/products/fg-wix/)

SERVICES (/services/)

› Support Programs (/services/support-programs/)

› Code Review (/services/code-review/)

> Phone Consultation (/services/phone-consultation/)

> Custom Development (/services/custom-development/)

DOCUMENTATION (/docs/)

> HeatWave (/docs/heatwave/)

> WiX v4 Tutorial (/docs/wix/tutorial/)

> WiX v3 Tutorial (/docs/wix/v3/tutorial/)

> FireGiant WiX v3 (/docs/fg-wix/)

© 2023 FIREGIANT (/)

> About Us (/about-firegiant/)

> Why WiX (/why-wix/)

> Leadership Team (/about-firegiant/leadership-team/)

> Privacy Policy (/about-firegiant/privacy-policy/)