

Create an installer for website with WIX, part 2

5 Replies

In [part 1](#) we saw how we can use a MS Build script to harvest all the files that needs to be installed and to create our msi installer.

When you run the installer we created you saw it had a very minimalistic UI, better none UI at all. In part 2 we're going to add a user interface for the installer. We'll make use of the build in UI elements of WiX but also add a new dialog so the end user can enter the database connection values and credentials.

The WiX library has already a lot of build in UI dialogs and flows that you can use out of the box. If you download the [source code from codeplex](#) you can take a look at the standard UI. (unzip and then go to `\src\ext\UIExtension\wixlib\`)

The dialogs we wanted to show our end customers where:

- Welcome screen
- Let the user select the install directory (default `c:\inetpub\`)
- Let the user fill out database connection values (database server IP or name, database name, database user and password)
- Confirmation to install
- Show some progress of the installation
- Show a install succeeded screen

The WixUI_InstallDir default UI comes closest to what we want to implement. So we start from that default UI theme.

Create our own UI

Add a new file in our setup project and name it DemoUI.wxs. and copy the default WixUI_InstallDir file content into this file.

You can see the flow the UI will follow. First a welcome screen, then a license screen, then the installation directory screen, a verify ready dialog and a maintenance dialog. Except for the license screen this is

exactly what we needed.

Our first job will be to remove the license screen from our UI flow. You can delete the lines 60 and 61 but we'll have to take care of the references to the license screen.

On line 58 you'll see that the welcome dialog next button will send you to the license screen, alter that so it will send you to the installation directory screen.

Same for line 63 where the back button will send you to the license screen. Update the Value tag so it references the welcome dialog instead.

Reference our own UI

We still have to tell our installer that he has to use our own UI instead of the default one. In our DemoUI.wxs change the Id of the UI element (line 33) from "WixUI_InstallDir" to "MyUI".

In our Product.wxs we'll have to reference our own UI by adding a "UIRef" element in the "Product" element with Id "MyUI". (line 11)

And we have to tell the UI the default folder he has to use in the Install directory dialog by adding the "WIXUI_INSTALLDIR" property that we reference to our "INETPUB" property.

Update our MS build file

Now that we have a new source file (DemoUI.wxs) we have to update our MS Build file to include our user interface.

Open up the setup.build file and update the Itemgroups containing the WiX source files and WiX compiled files.

In our WIX target we'll have to tell the light.exe tool that we are using parts of the WixUIExtension library.

And last but not least I added a new target to clean up all previous created and build files to avoid conflicts.

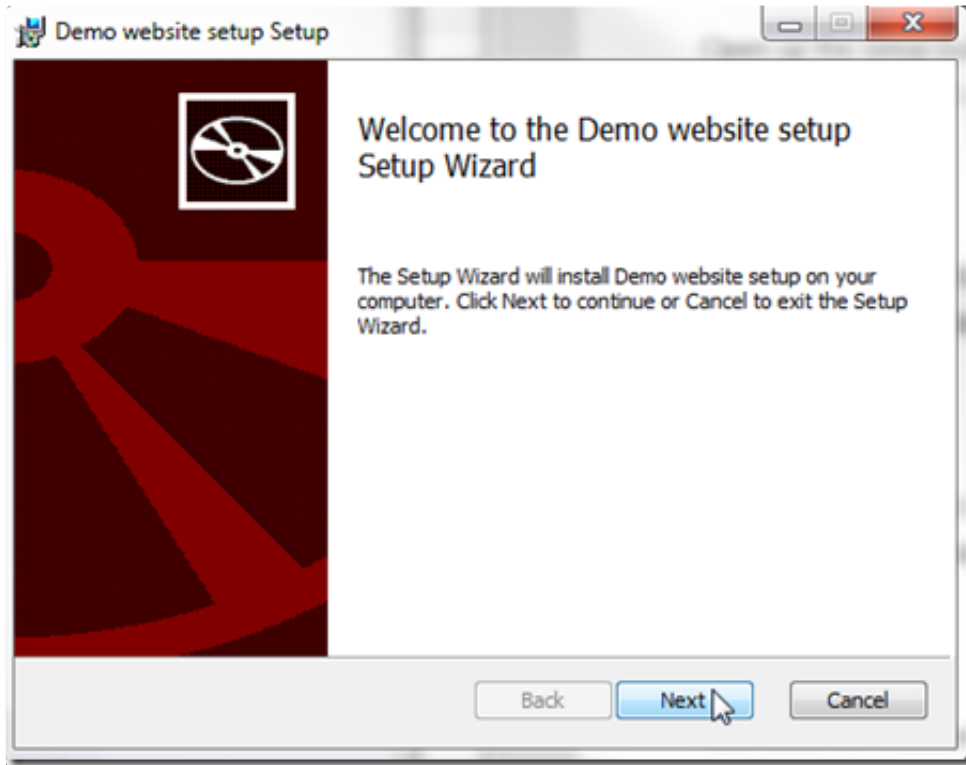
Now build the installer by opening a Developer Command Prompt for Visual Studio 2012, change the

directory to the setup project and enter the next command:

```
msbuild /t: Build;PublishWebsite;Harvest;WIX;DeleteTmpFiles setup.build
```

and hit enter.

In the \bin\release folder of our setup project you'll find our freshly build DemoWebsite_Setup.msi. Double-click the installer and you'll see the welcome screen as intended.



If we click next the destination folder is prefilled with the c:\inetpub folder but can be changed by the end user. Clicking next will show the “ready for installation” and the “installation succeeded” screens.

During installation we'll see a progress bar showing the status of the installer. Running the installer will, just like in the last post, install the necessary files in the C:\inetpub\DemoWebsite folder (if no other location is selected by you).

If you run the installer again you'll get the default repair/remove interface.

Add the SQL connection screen

Now that we have a default UI for our installer we can add the dialog to let the end user enter the database connection values.

Add properties

The database connection values that will be entered by the end user will need to be reused in a later process of installation of the database. Therefore we have to store them in properties.

In our Product.wxs file we add four properties (DB_USER, DB_PASSWORD, DB_SERVER, DB_DATABASE) and give them some default values (line 7 to 10).

Add a new dialog

Now we can add the new dialog, add a new Installer file in our setup project and name it UIDialogs.wxs. I added the controls we need.

We'll see a label and a text box for our 4 properties and a next and cancel button that we'll link in our UI file.

Insert the dialog in the UI flow

Open up our DemoUI.wxs file again and add our new dialog between the "Install directory" and "verify ready" dialogs. Add 2 references, one for the next button and one for the back button.

The next button we reference to the "VerifyReadyDlg" and the back button to the "InstallDirDlg".

We've got to alter the flow to insert our dialog by changing line 5 (reference the next button of "InstallDirDlg" to our one) and change line 12 (reference the back button of "VerifyReadyDlg" to our one).

Update our MS build file

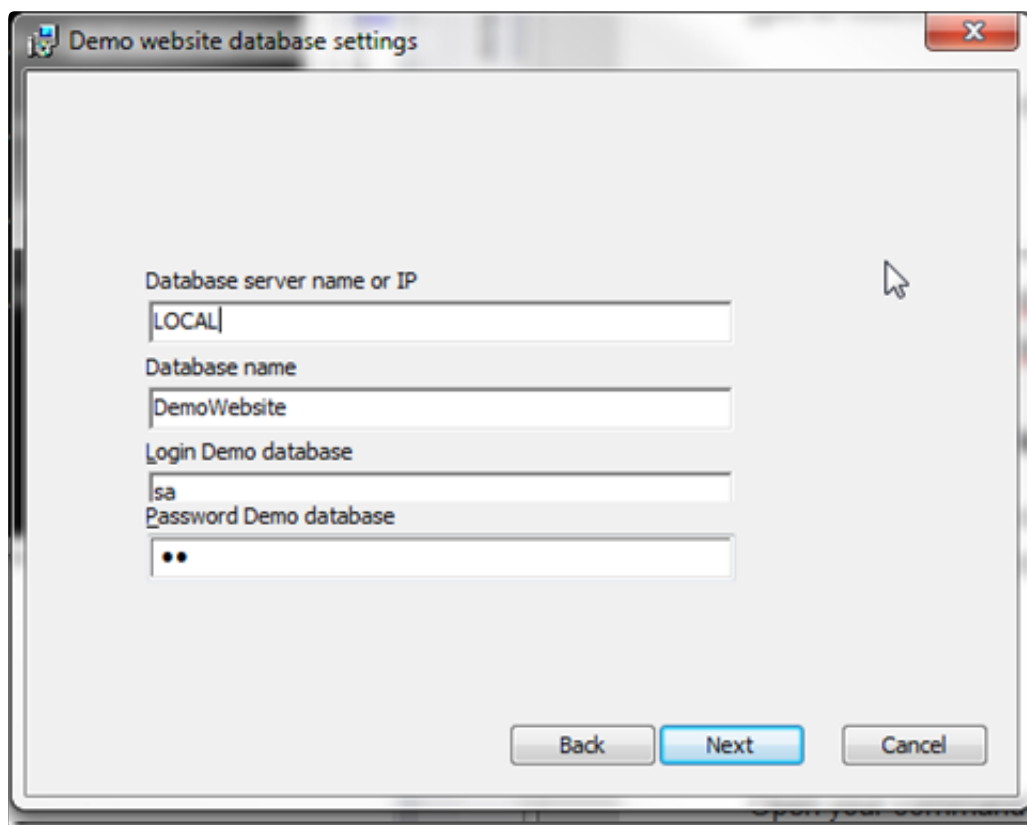
We have to tell our script there is a new source file to handle, the dialog file. Open up the MS build script again and alter the ItemGroup settings by adding the new file.

Open your command prompt again and run the next statement:

```
msbuild /t: Build;PublishWebsite;Harvest;WIX;DeleteTmpFiles setup.build
```

and hit enter.

When our installer is build, run it and you'll see the new dialog appear after the installation directory dialog.



Check the installer

You may see the properties now in the UI you want off course to be sure they are set properly. You can check that by letting the install process create a log file.

Browse to the `/bin/release/` folder in the setup project and open a command prompt. On the prompt type the next command:

```
msiexec /I Demowebiste_Setup.msi /!v demo.log
```

You'll see that during the installation process a "demo.log" file will appear in the `/bin/release/` folder. In that

log file you will see the parameters are set to the entered values (second line in the screenshot)

```
Property(C): WIXUI_INSTALLDIR = INETPUB
Property(C): DB_SERVER = LOCAL
Property(C): DB_DATABASE = DemoWebsite
Property(C): DB_USER = sa
Property(C): DB_PASSWORD = *****
Property(C): ALLUSERS = 1
Property(C): ARPNAME = 1
Property(C): INETPUB = C:\Inetpub\
Property(C): TARGETDIR = C:\
Property(C): SourceDir = C:\project\Github\WIXDemo\WIXInstallerDemo\DemoWebsite.Setup\bin\Release\
Property(C): dir9BAF78203B38579D00D575979F7B8750 = C:\Inetpub\DemoWebsite\Content\themes\
Property(C): Manufacturer = Me
Property(C): ProductCode = {D16E42B5-3C6F-4EE5-B2F4-727BF8B74A92}
Property(C): ProductLanguage = 1033
Property(C): ProductName = Demo website setup
```

Next parts

This concludes the custom UI we wanted to create. In the next chapters we'll use the entered information to create a database.

Of course there is (a lot) of room to enhance the UI with custom bitmaps, text, ... More information you can find in the [WiX documentation page](#).

The items in yellow are handled, the next post will be probably on how to add a website in IIS.

- Install the .NET 4.5 framework if that isn't installed already
- Install the MVC 4 framework if that isn't installed already.
- Create a folder and copy all needed files to run the application (done)
- Create a new database on an existing SQL server and prefill the database with the correct tables and values. (the connection details and database name should be entered by the end user running the installer)
- Create a new website in IIS 7.5 (create website and application pool running under .NET 4.5)
- Alter the config file so the correct connection settings are used(entered by the end user)

Complete source code

[Can be found on Github.](#)

Other posts in this series

- [Create an installer for website with WIX, part 1](#) (install files, create MS Build script)
- [Create an installer for website with WIX, part 2](#) (Custom UI)
- [Create an installer for website with WIX, part 3](#) (Install website in IIS)
- [Create an installer for website with WIX, part 4](#) (Create database and run scripts)