

# Limnor Studio – User’s Guide

## *Installer*

Last modified: May 15, 2015

### Contents

1	Setup Project.....	3
2	General Information .....	6
2.1	Banner Image File .....	6
2.2	Company Folder Name .....	7
2.2.1	%Program Files%.....	7
2.2.2	{CompanyFolderName}.....	8
2.2.3	{Setup Project Name}.....	8
2.3	Dialog Image File .....	8
2.4	Setup Icon File .....	9
2.5	Version and Last-Compiled-Version.....	10
3	Files to be distributed .....	11
3.1	Distribute files from projects .....	11
3.2	Distribute extra files.....	11
3.3	Add files from a folder .....	12
3.4	Create sub folders .....	13
3.5	Distribute Files to System Folder .....	15
3.6	Distribute Files to Common Data Folder.....	15
4	Shortcuts .....	16
4.1	Icons .....	16
4.2	Add shortcut .....	17
4.2.1	Select icon .....	17
4.2.2	Specify location .....	17
4.2.3	Shortcut name.....	17
4.2.4	Target .....	17
5	Cultures.....	18

5.1	Add culture.....	19
5.2	Application Name.....	19
5.3	Welcome Message.....	20
5.4	Setup License File.....	21
5.5	MSI File Information .....	24
5.5.1	File Summary Information .....	24
5.5.2	Code page for MSI file information.....	25
5.5.3	Display Version in File Information.....	26
6	Feedback .....	27

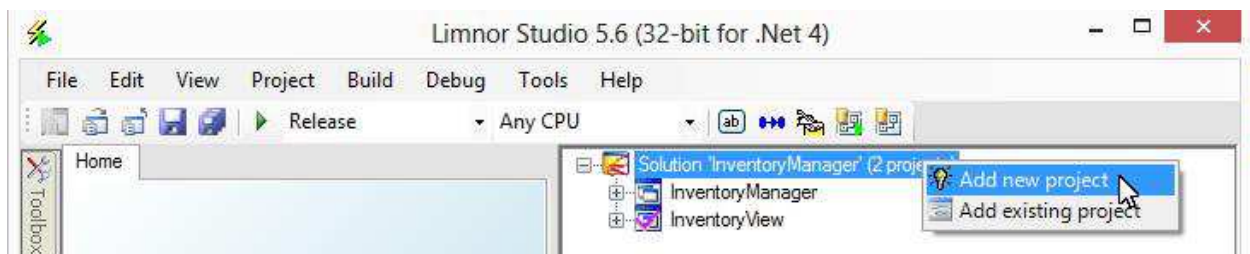
## 1 Setup Project

An installer makes it easy for your clients to setup your software product in their computers. An installer packages all the files consisting your software product into one single file, a \*.msi file, to be distributed to your clients. This msi file is the installer for your software product. It may also perform installation time actions such as creating shortcuts and program groups.

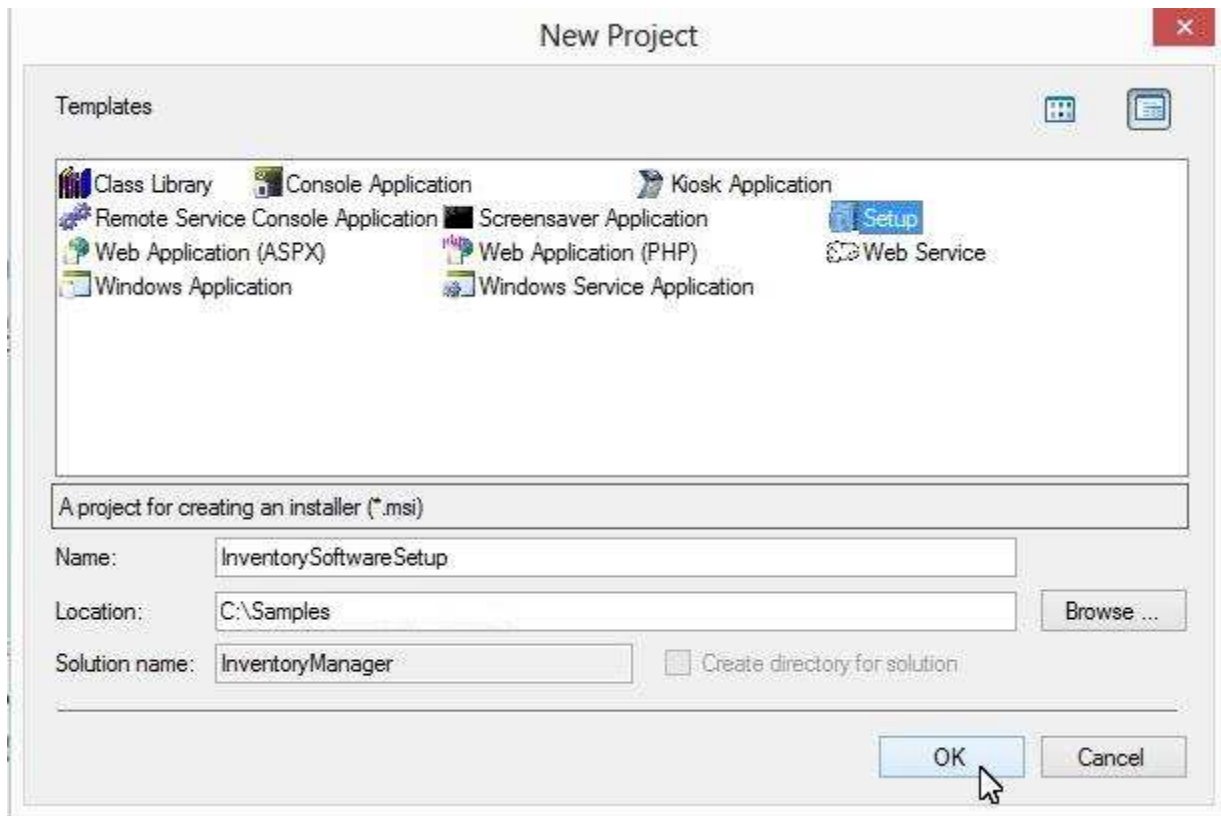
An installer is created by a Setup project.

Suppose we developed a product consists of two projects in one solution: Inventory View and Inventory Manager. We want to create an installer to distribute the software product. We may do it by adding a new Setup project to the solution.

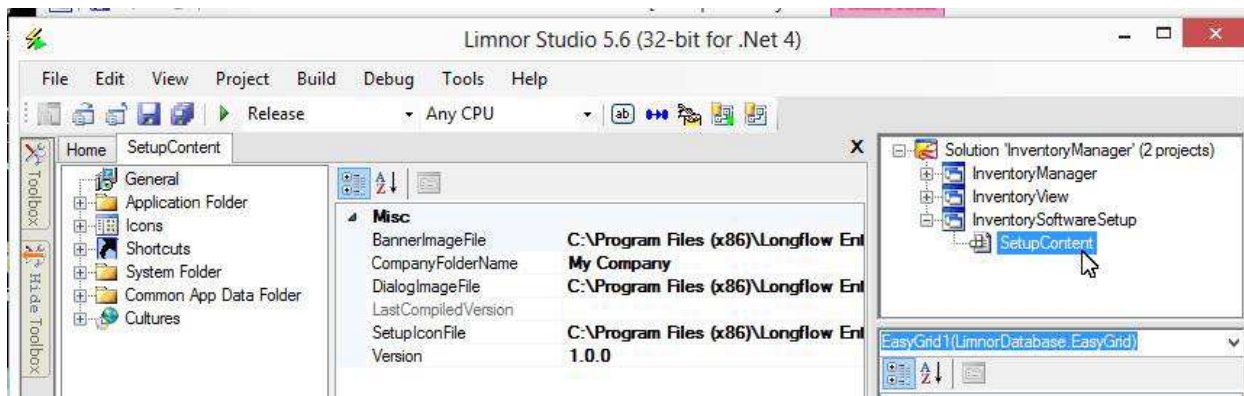
Right-click the solution, choose “Add new project”:



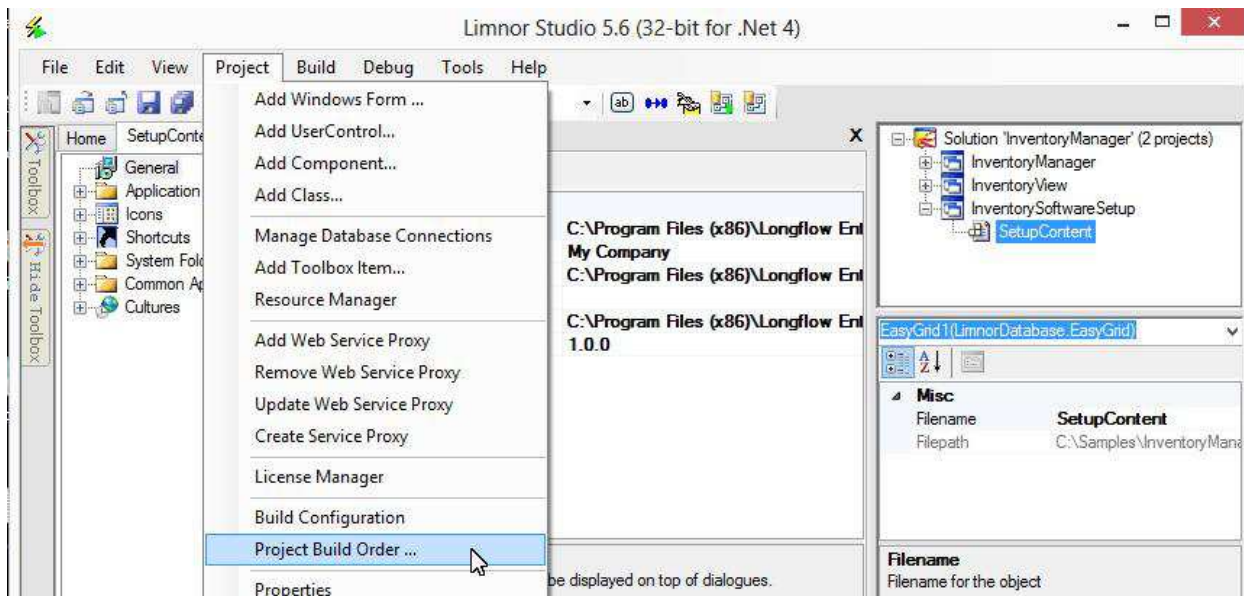
Select Setup as the project type, give a name for the project, and choose the project location:



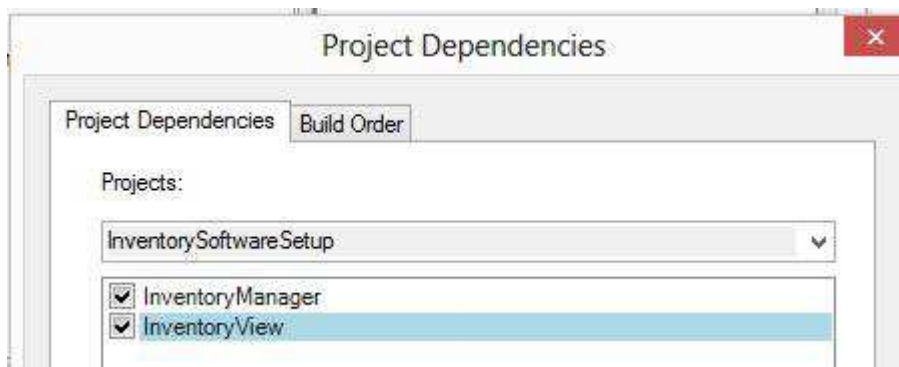
The new setup project is created and added to the solution. It has a SetupContent.limnor file for collecting setup information.



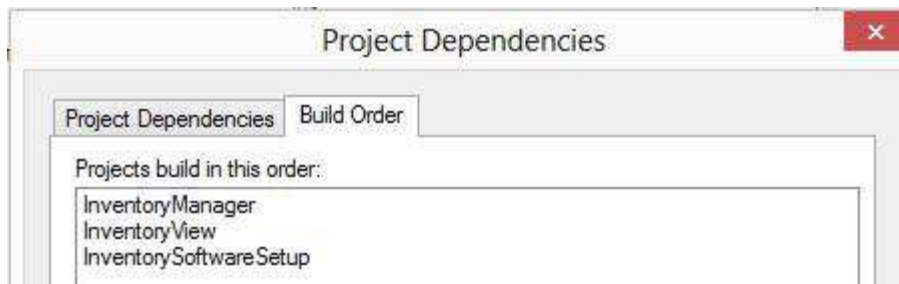
Since the setup project needs to collect and package the compiled files from other projects, it should be compiled after all other projects are built. We may set "Project Build Order ..." to make sure of it:



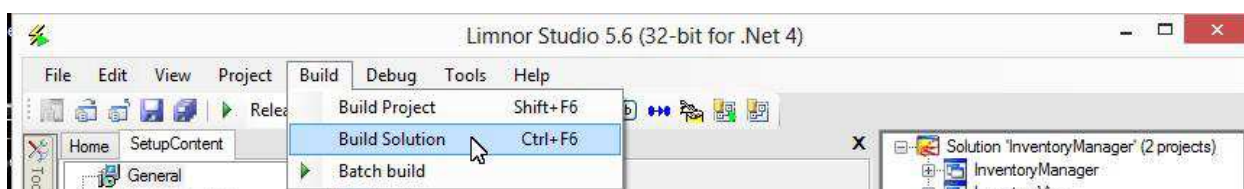
Make the setup project depends on all other projects:

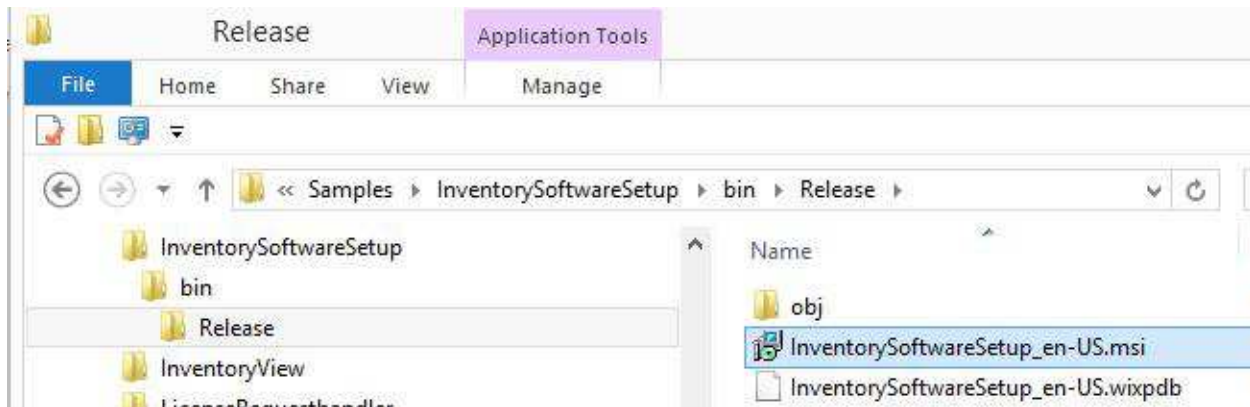


Now the setup project will be the last one to be built:



On building the solution, an msi file is created:



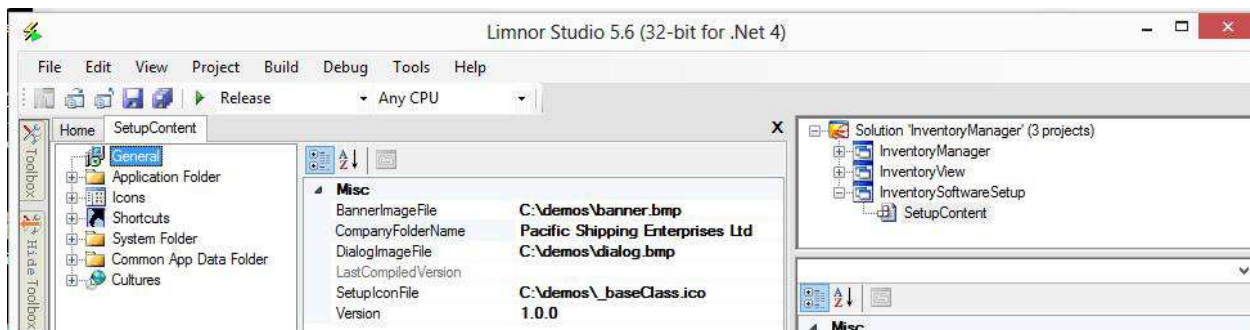


This msi file is the installer with all the files packaged inside. It can be distributed to your customers.

Below we will go through the information collection file SetupContent.limnor.

## 2 General Information

Select the General node to enter general setup information:



### 2.1 Banner Image File

This is an image to be displayed on top of dialogue boxes. For example, if **BannerImageFile** is using such an image file:



Setup dialogue boxes will look like something as following.



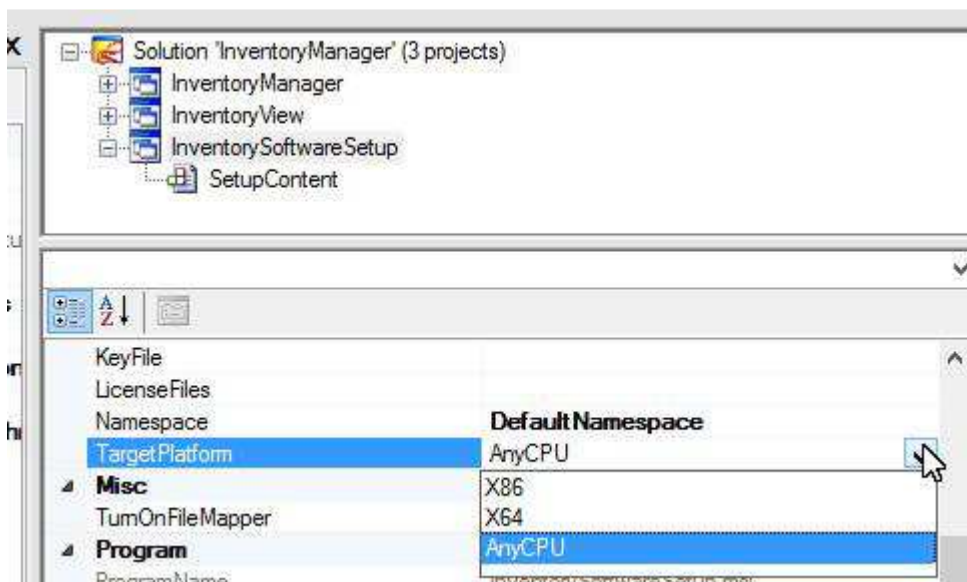


## 2.2 Company Folder Name

By default, your software will be installed in a folder formed by **%Program Files%\{CompanyFolderName}\{Setup Project Name}**, each part is explained below.

### 2.2.1 %Program Files%

%Program Files% is determined by Windows installation and setup target platform. You may change the target platform of a setup project:



Suppose in a customer computer the Windows is installed on C: drive; if **TargetPlatform** is X64 then %Program Files% is C:\Program Files; if **TargetPlatform** is not X64 then %Program Files% is C:\Program Files (x86).

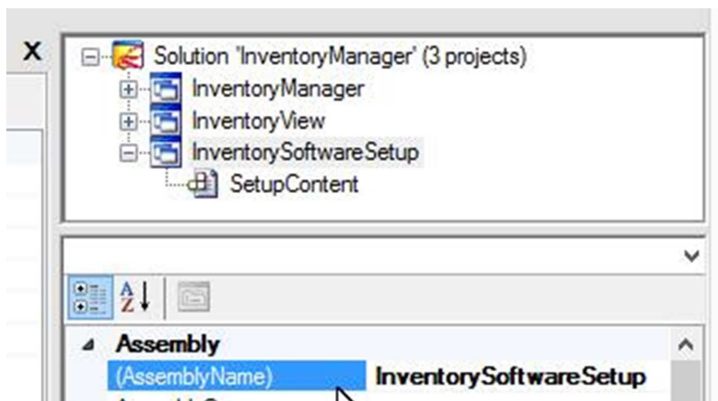
### 2.2.2 {CompanyFolderName}

Usually all your software products are installed in sub-folders of one “Company folder” under %Program Files% folder. CompanyFolderName allows you to specify this “Company folder”. In the above example, it is using “Pacific Shipping Enterprises Ltd”.

Note that since it is a folder name, you should not use characters not allowed for folder names.

### 2.2.3 {Setup Project Name}

Each setup project uses its own installation folder. The folder name is the project assembly name.



Using the above example, the default installation folder will be similar to that shown in following screenshot.



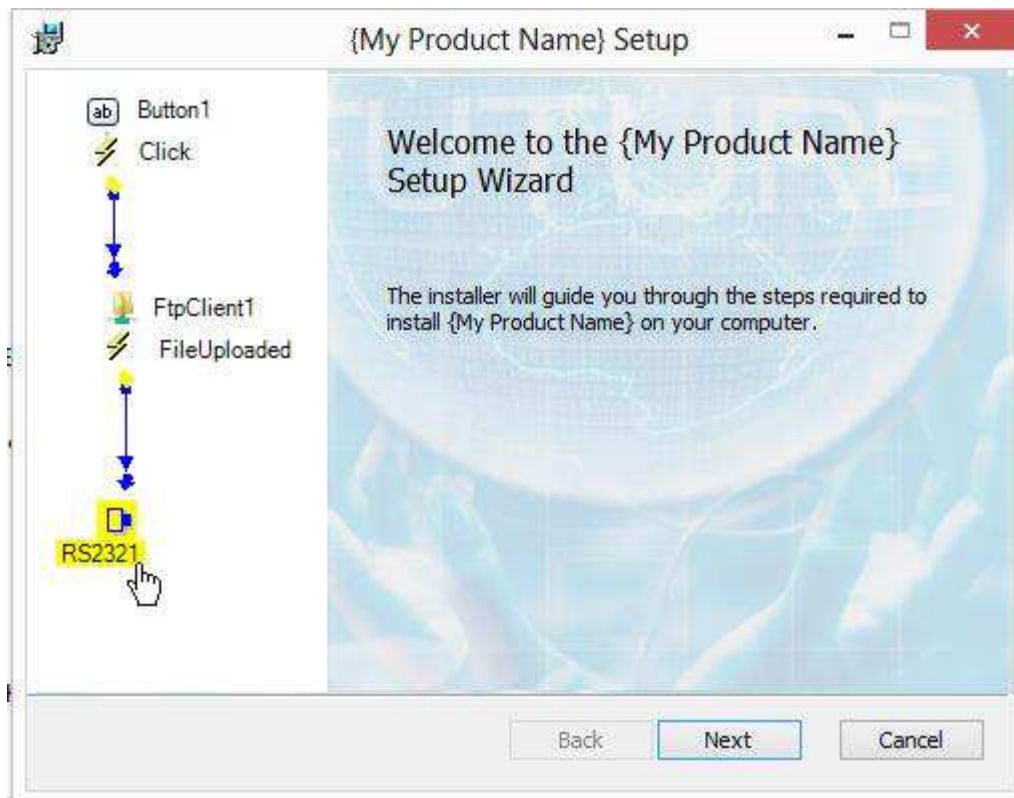
## 2.3 Dialog Image File

DialogImageFile is an image file to be displayed as background of dialogue boxes. Suppose following image is used:



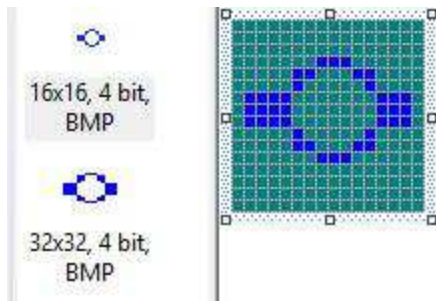


Setup dialogue boxes will be displayed similar to the following screenshot:

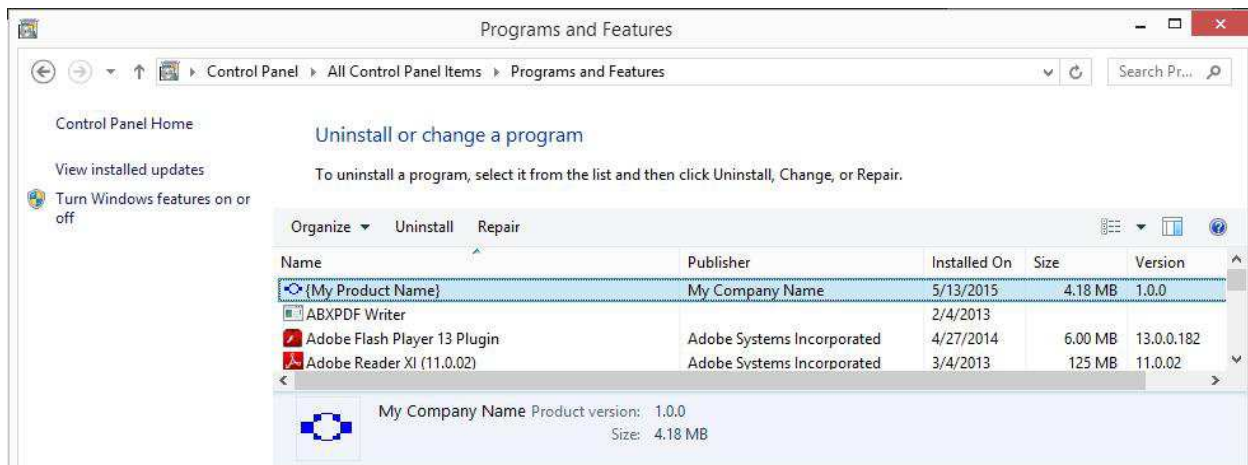


## 2.4 Setup Icon File

SetupIconFile is an icon file to be used by Windows “Programs and Features” in the Control Panel in listing this installation. You may use an icon containing 16x16 and 32x32 images. For this example icon file C:\demos\\_baseClass.ico is used:



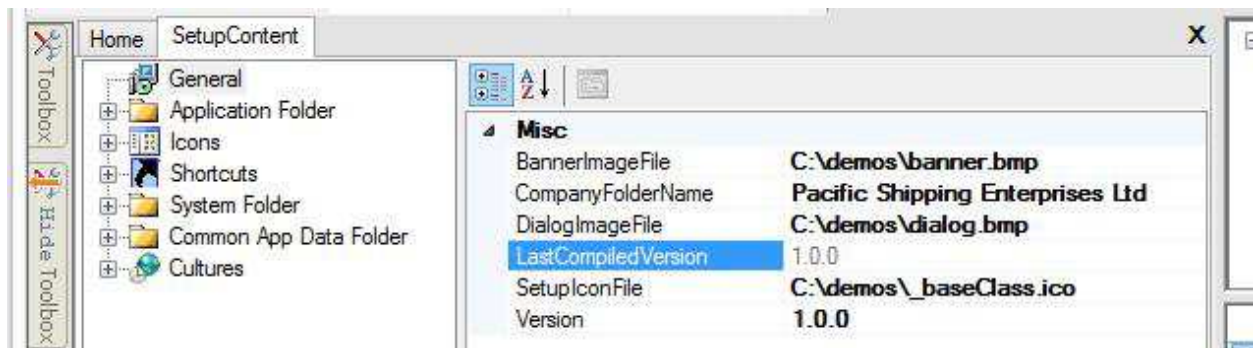
The installation appears in “Programs and Features” using the above icon:



## 2.5 Version and Last-Compiled-Version

The installer version must be in a format of 3 numbers separated by dots, for example, 1.2.3.

On compiling the setup project to an MSI file, the Version is copied to LastCompiledVersion:

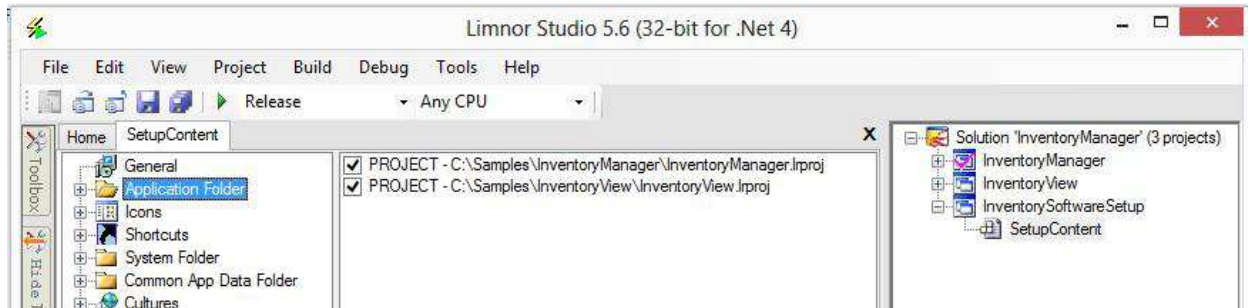


Your customers run the MSI to install your software. You modify your software and compile the setup project again to generate a new MSI file. Before making a new MSI file, the Version must be changed to be different than the version of previously generated installer so that the new installer can upgrade previously installed product.

## 3 Files to be distributed

### 3.1 Distribute files from projects

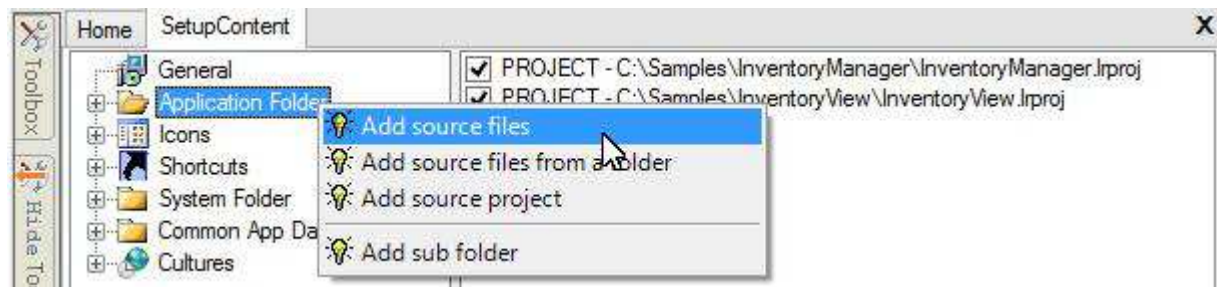
Select node “Application Folder (TARGETDIR)” for specifying the files to be distributed:



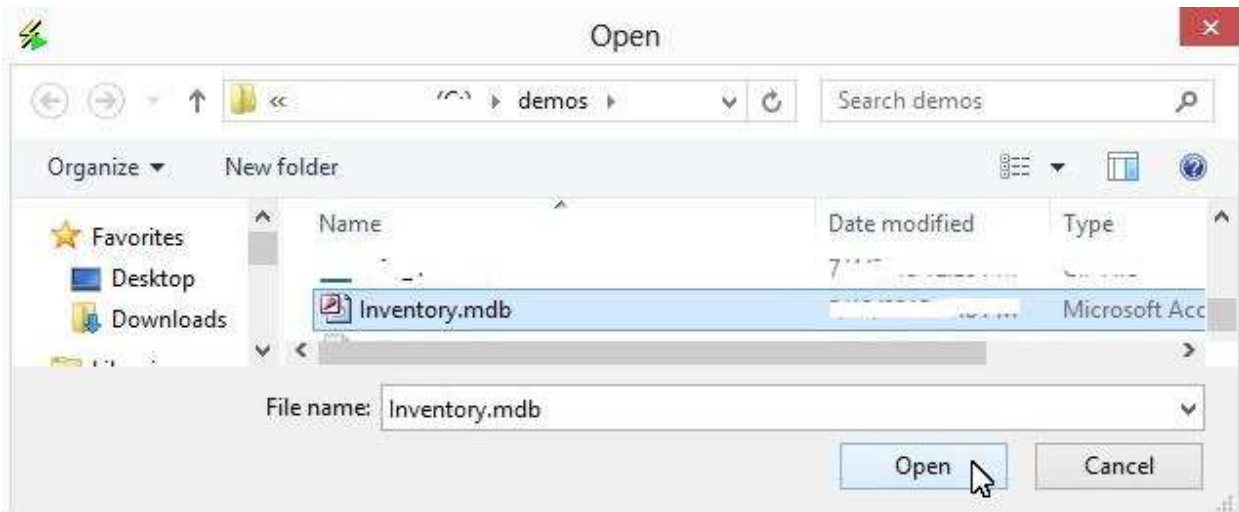
By default the two other projects in the solution, InventoryView and InventoryManager, are included in a checked list, indicating that compiled files from these projects will be included in the distribution. If you want exclude a project from the distribution then uncheck the project in the list.

### 3.2 Distribute extra files

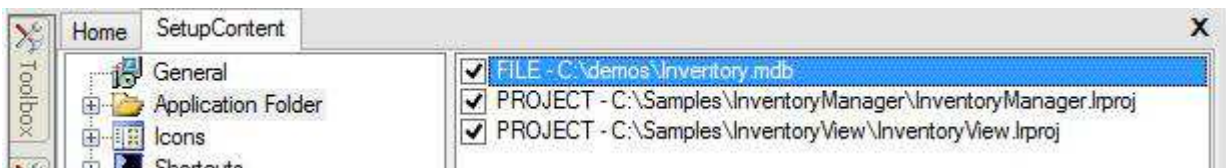
Apart from files from projects, other files can also be installed. For example, suppose we want to distribute the database file. Right-click the node “Application Folder (TARGETDIR)”, choose “Add source files”:



Select the database file to be distributed:



The database file appears in the list:

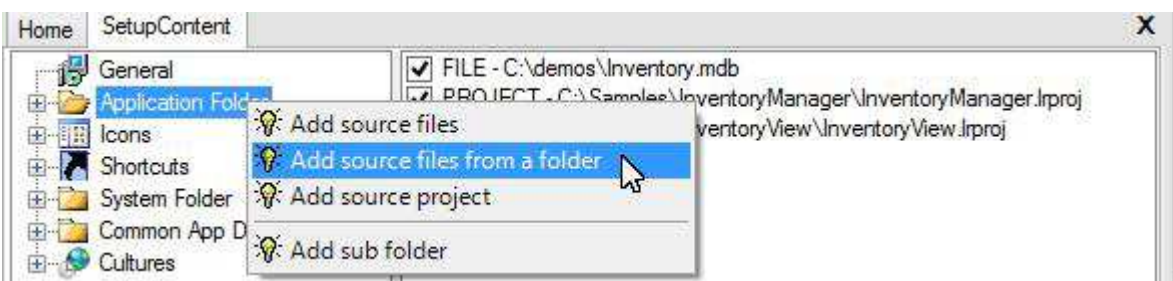


We may add more files in this way.

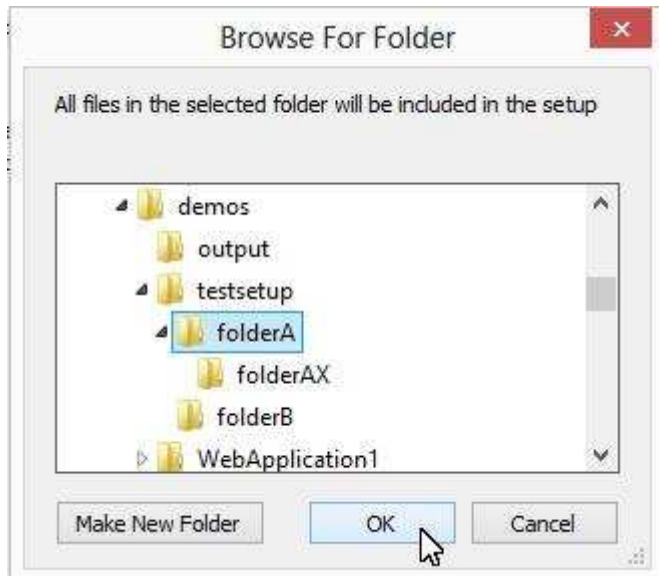
If we change our mind and want to remove a file from the installer then uncheck the file.

### 3.3 Add files from a folder

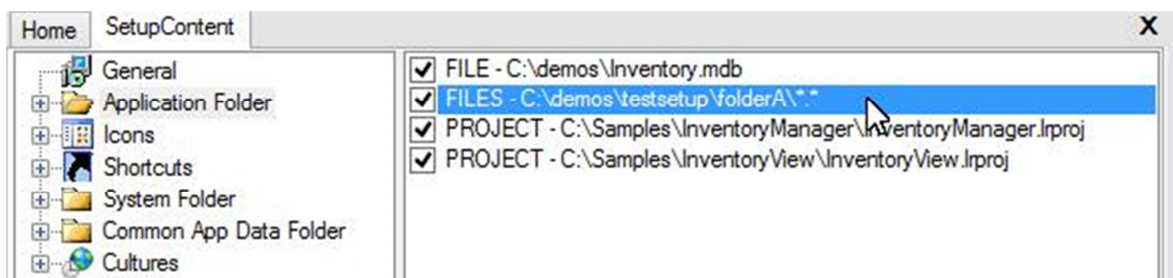
Instead of including individual files, we may also specify that all files from a folder should be included. Right-click the node “Application Folder (TARGETDIR)”, choose “Add source files from a folder”:



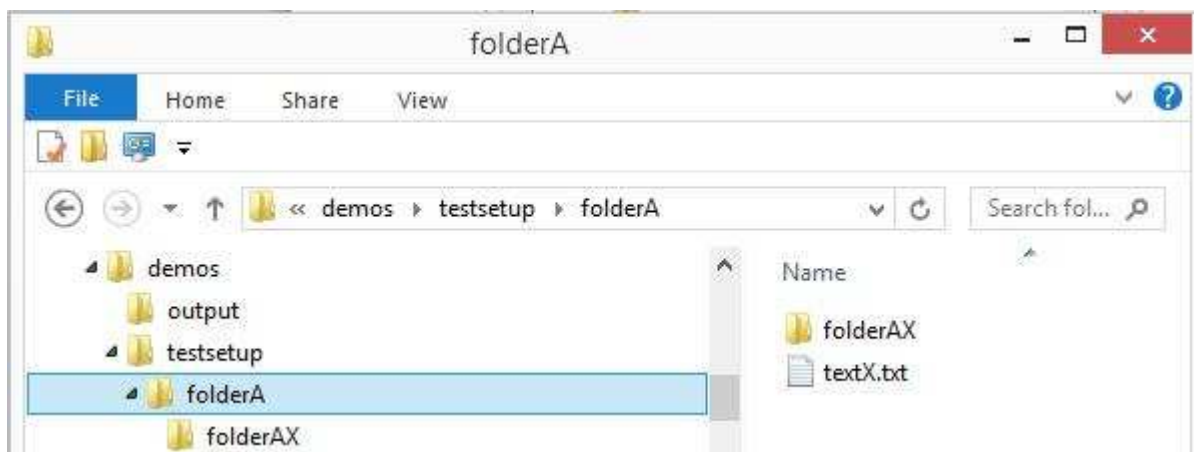
Select a desired folder:



The folder is included as a source for files:



Note that sub folders will also be installed. In this sample, folderA contains a file textX.txt and a sub folder folderAX, they will be installed under the installation folder.



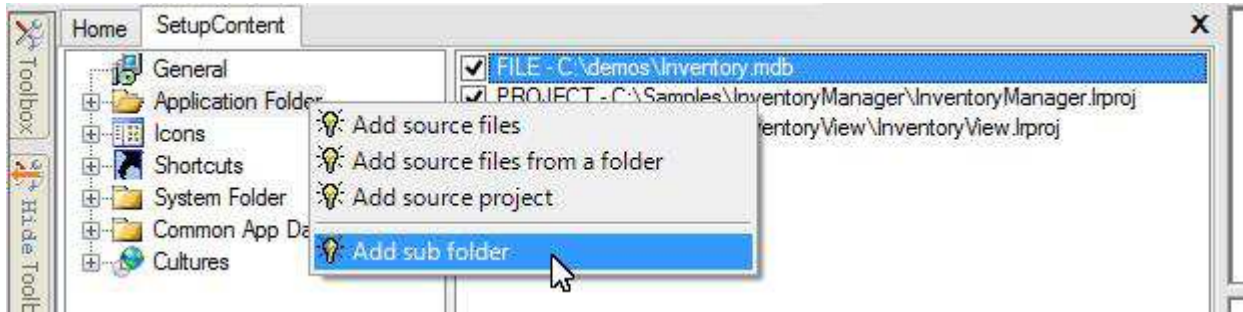
### 3.4 Create sub folders

The node "Application Folder (TARGETDIR)" represents the installation folder in the target computer. All files listed will go to the installation folder.

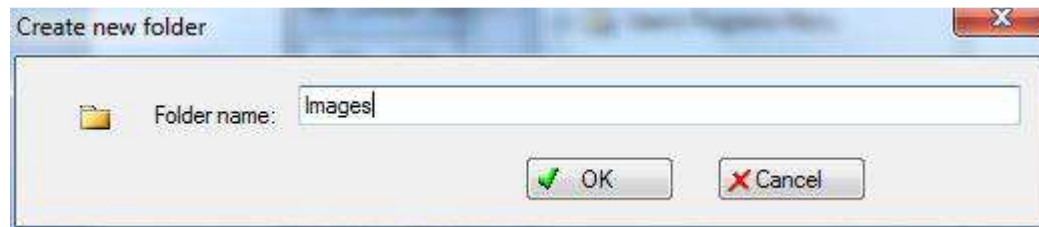


We may also create sub folders under the installation folder.

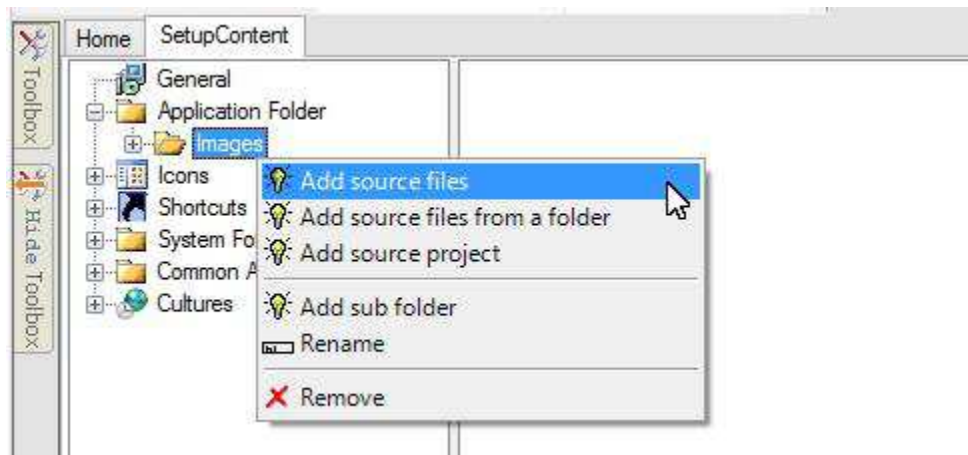
Right-click the node “Application Folder (TARGETDIR)”, choose “Add sub folder”:



Give a folder name:



The sub folder appears under the node “Application Folder (TARGETDIR)”. Right-click the sub folder, a context menu appears:



Add source files – Select one or more files to be installed in this sub folder.

Add source files from a folder – Select a folder to include all files from the folder to be installed in this sub folder.

Add source project – Select a project to install distribution files of the project to this sub folder.

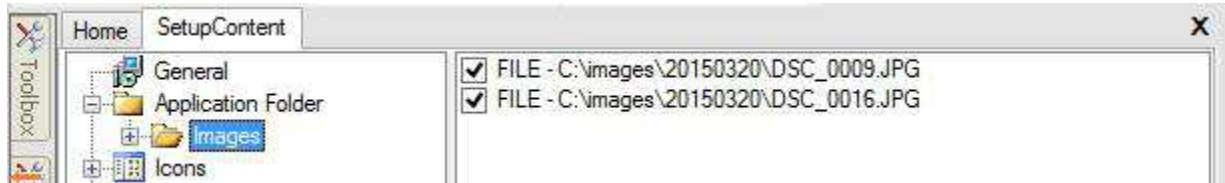
Add sub folder – Create a sub folder of this sub folder.



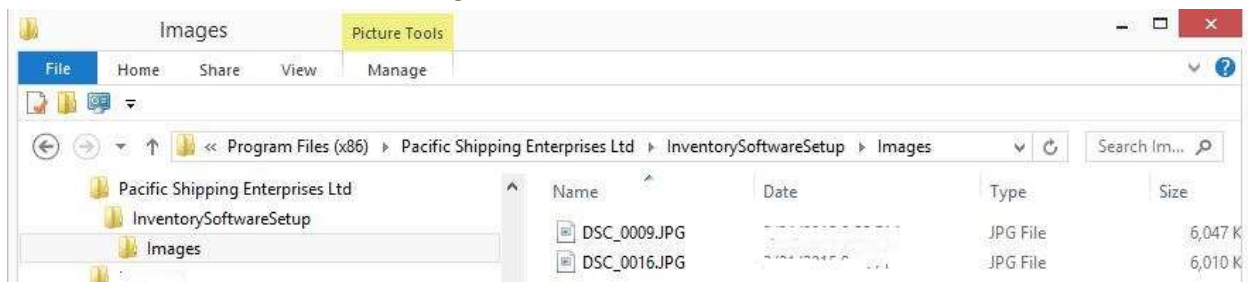
Rename – Change the sub folder name

Remove – Remove the sub folder

Files added to the folder will appear in the list. We may uncheck files to remove them.

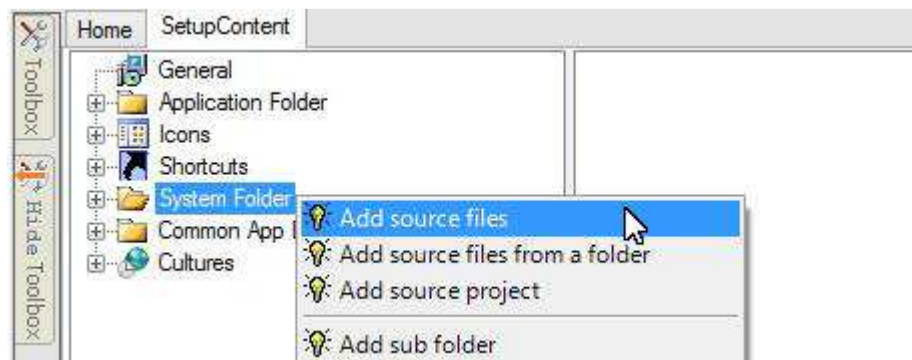


After compilation, we get the installer InventorySoftwareSetup.msi. Double-click it to run it. It installs the files. We can see a sub folder Images is created under the installation folder.



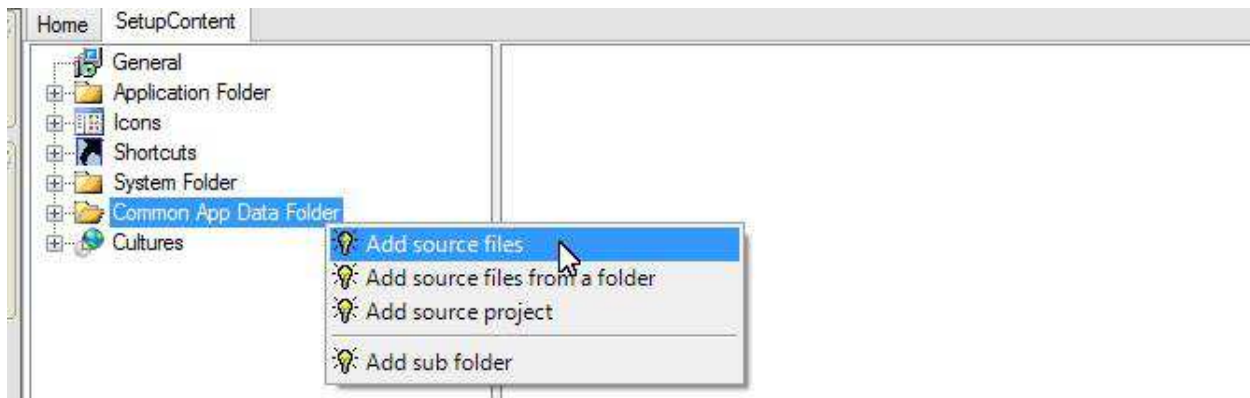
### 3.5 Distribute Files to System Folder

If some program files, usually DLL files, are used in more than one of your software product then you may want to distribute such files to Windows System folder.



### 3.6 Distribute Files to Common Data Folder

If your software uses some data files from Common Application Data folder then such data files should be installed into the Common Application Data folder.



## 4 Shortcuts

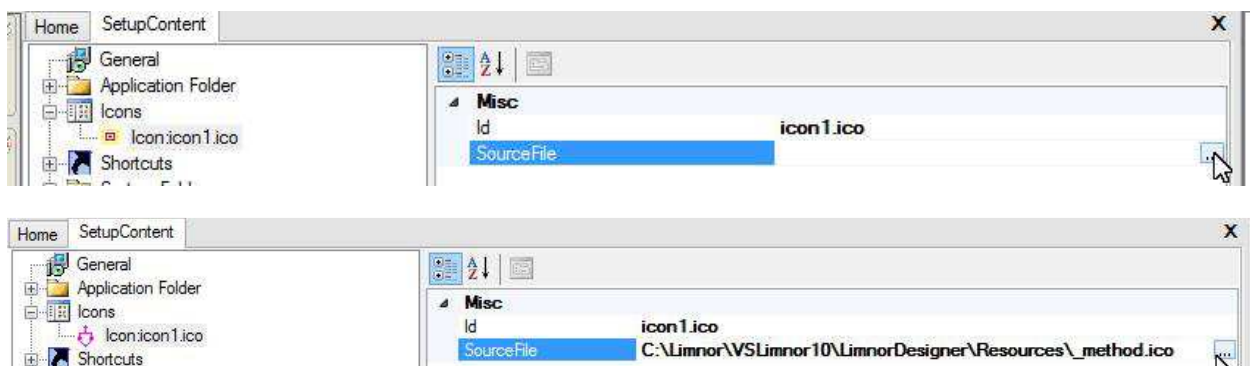
You create shortcuts for launching your software.

### 4.1 Icons

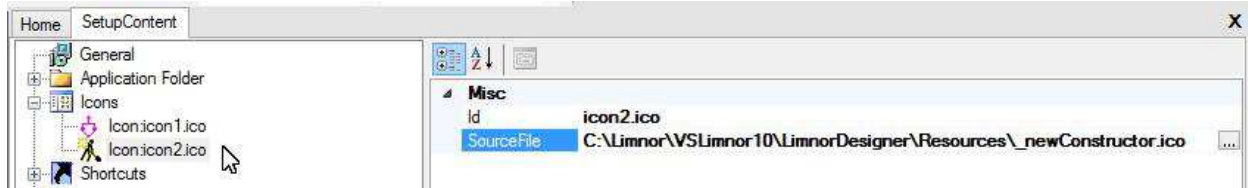
Each shortcut needs an icon. Shortcuts may use same icons. Before creating shortcuts, we need to include icons in the setup for the shortcuts to use.



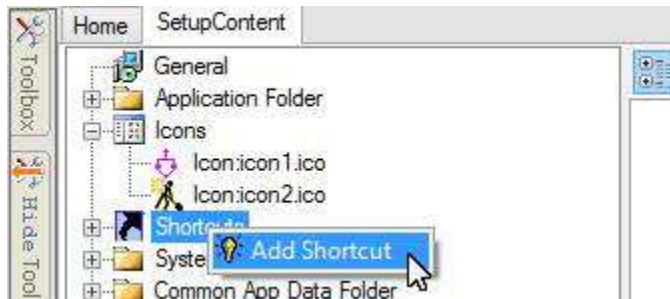
Select an icon file:



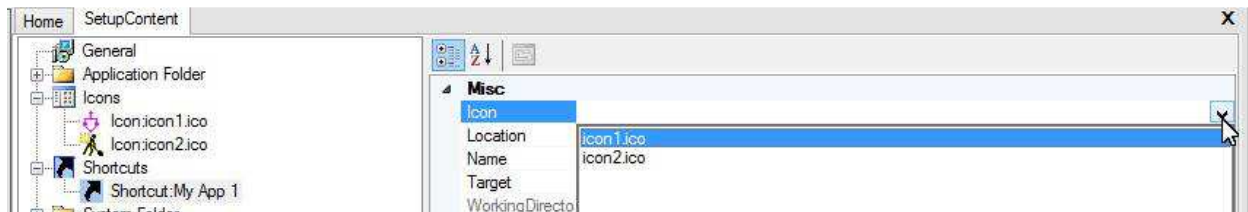
Add another icon:



## 4.2 Add shortcut



### 4.2.1 Select icon



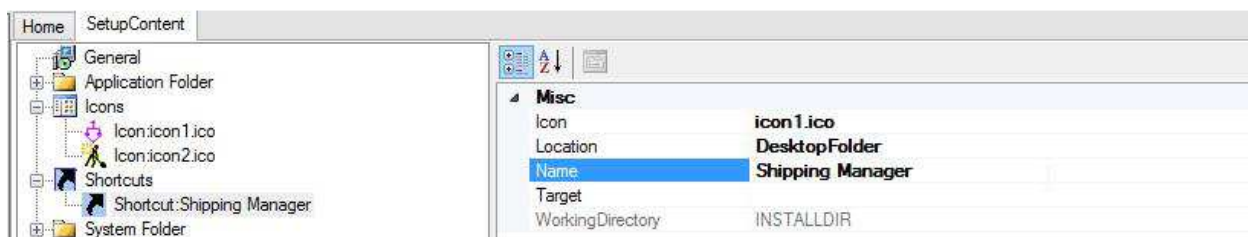
### 4.2.2 Specify location

A shortcut can be placed at desktop or at program menu.



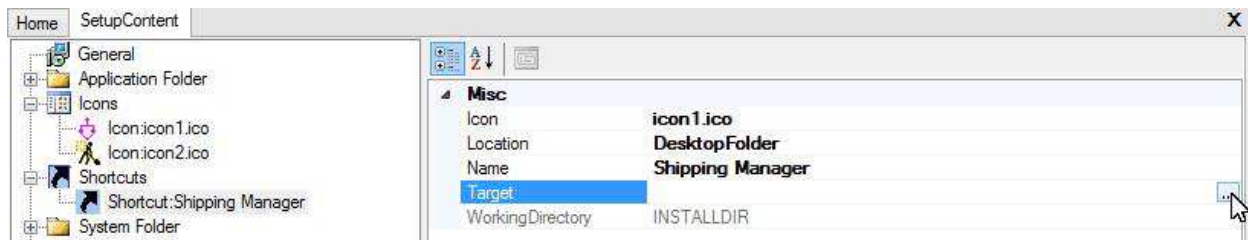
### 4.2.3 Shortcut name

A shortcut is displayed by its icon and name.

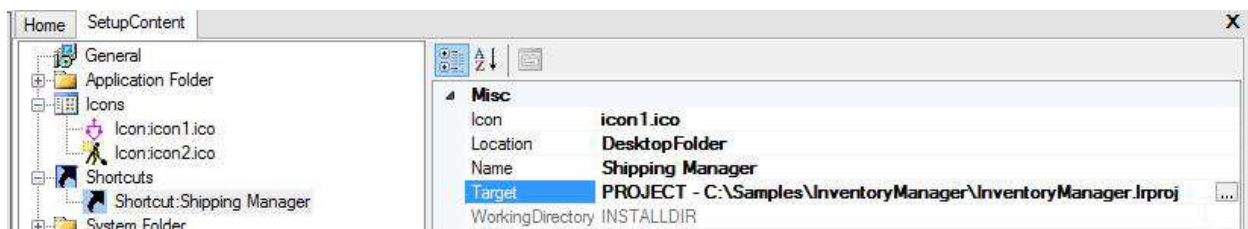
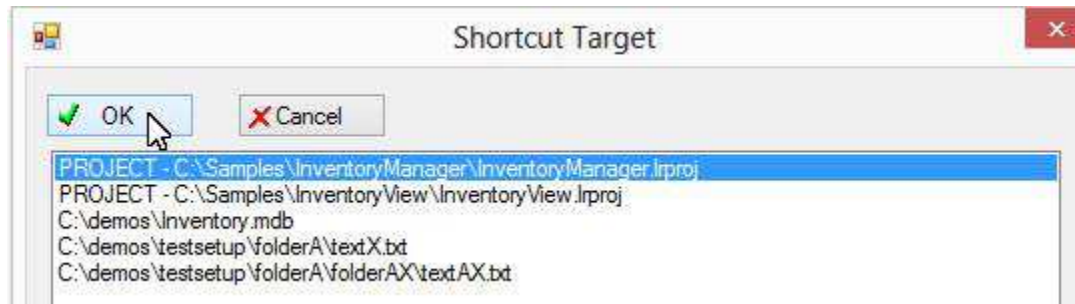


### 4.2.4 Target

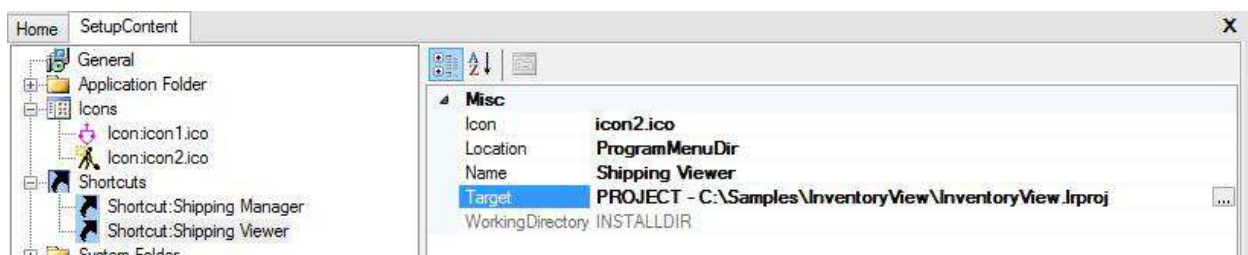
Target is a program to be launched by a shortcut.



Files included in the setup are listed for a selection. If a project file is selected then the EXE file compiled from the project will be used as the shortcut target:



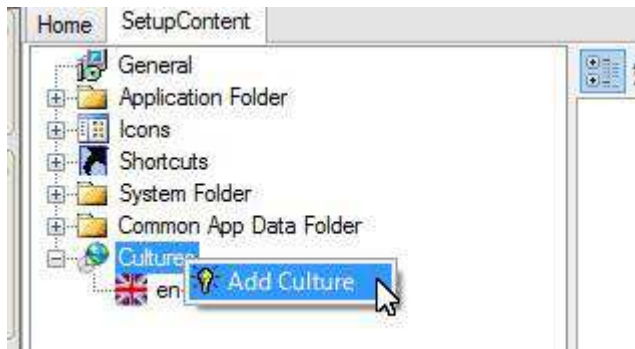
Add another shortcut:



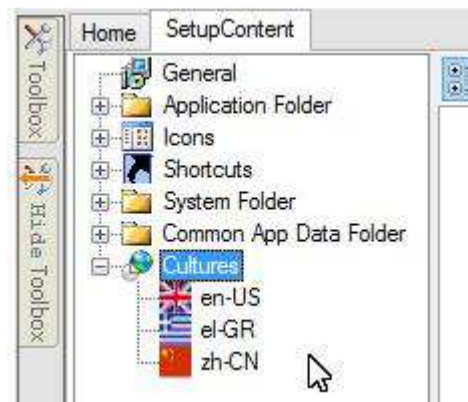
## 5 Cultures

You may specify one or more cultures for a setup project. For each culture, one corresponding MSI file is generated. A setup project must have at least one culture.

## 5.1 Add culture



Each culture is listed under Cultures.



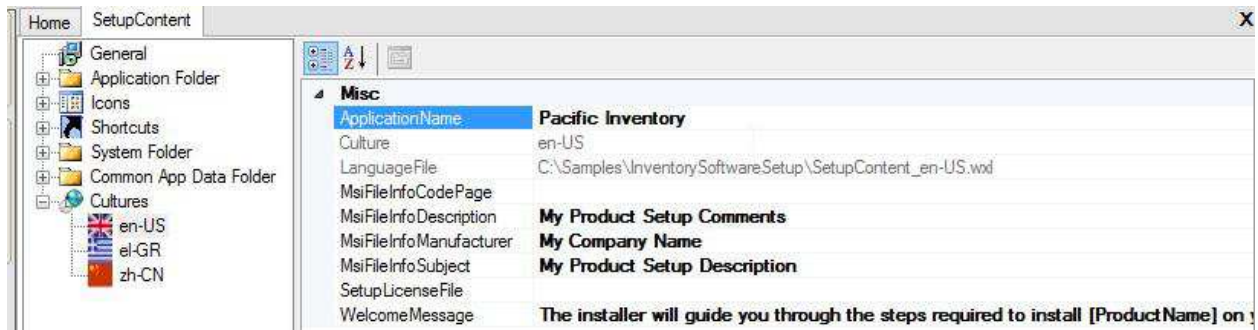
Note that not all cultures are supported by Microsoft's Windows Installer. You need to make a build to see if there are errors.

Select each culture to specify localized information.

## 5.2 Application Name

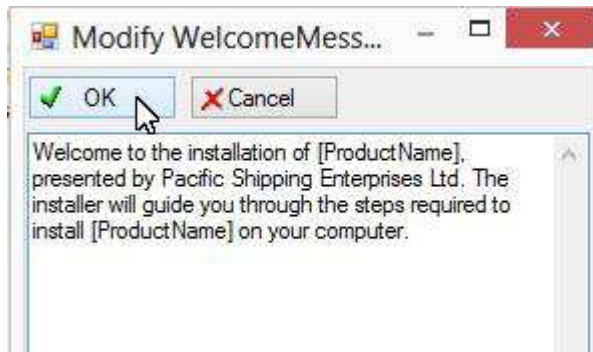
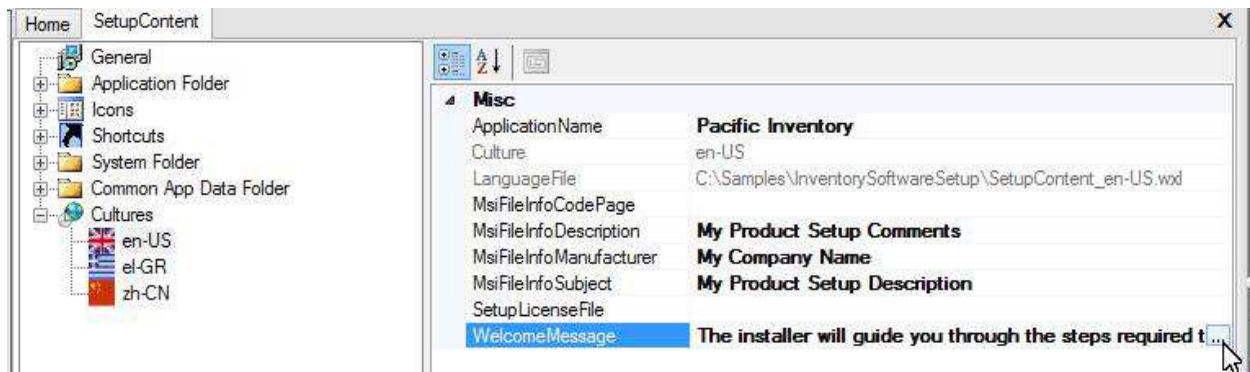
Application name will be displayed in setup user interface.





### 5.3 Welcome Message

Welcome message is displayed at the beginning of the setup.



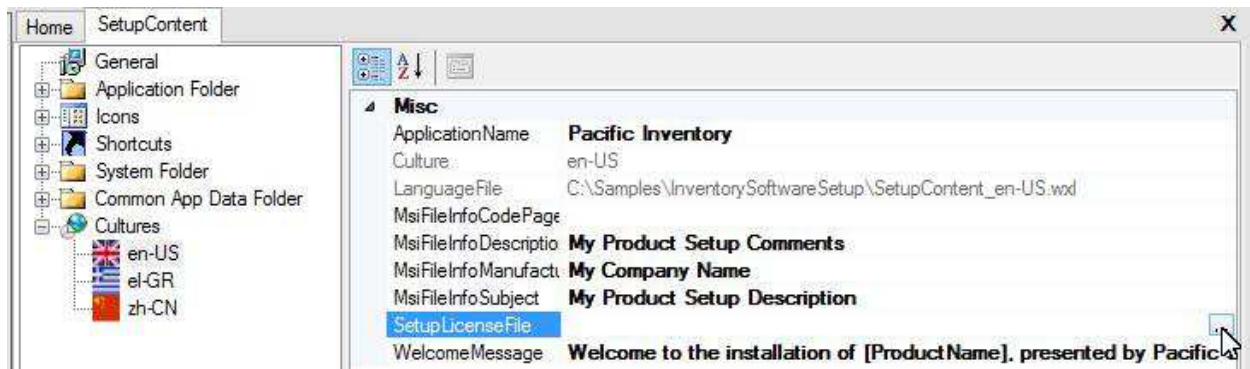
The Application Name and the Welcome Message will be displayed in setup dialogues:

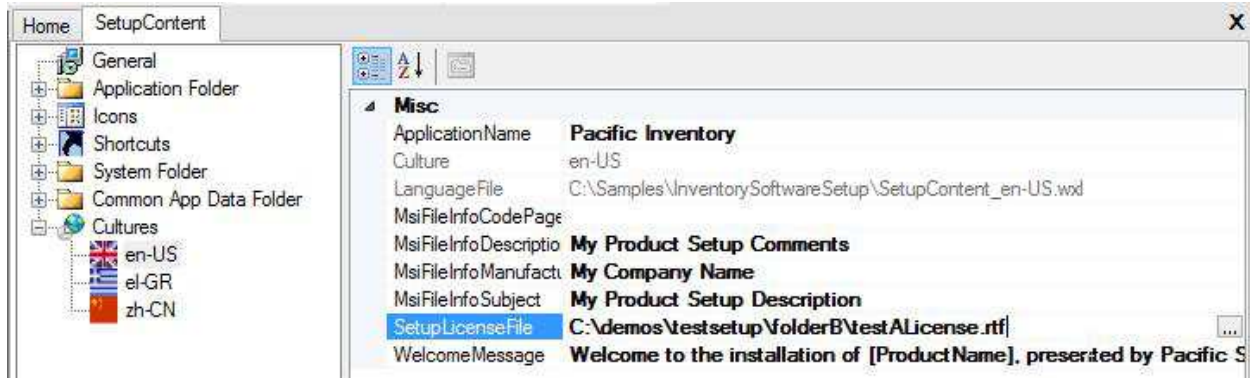




## 5.4 Setup License File

Setup License File is a license file to be displayed at the beginning of setup. The file should be in Rich Text Format (RTF).

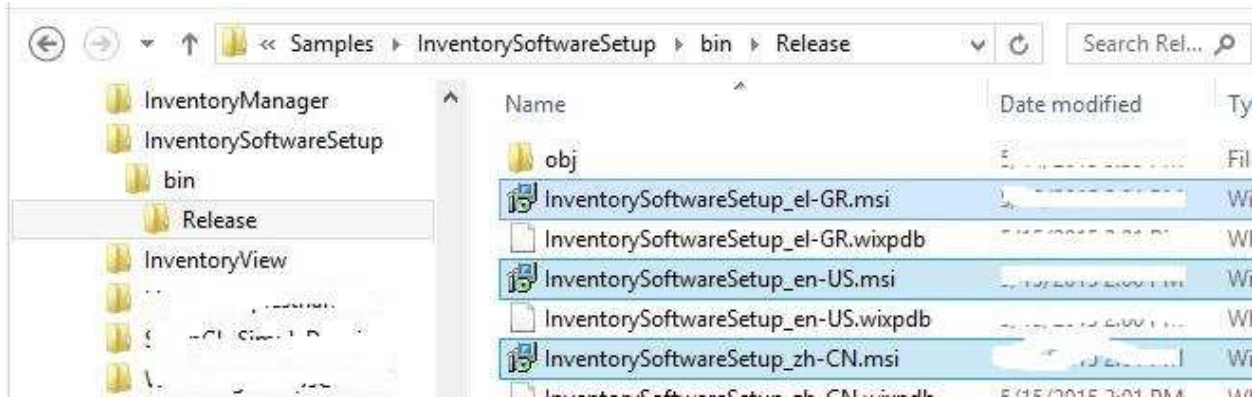




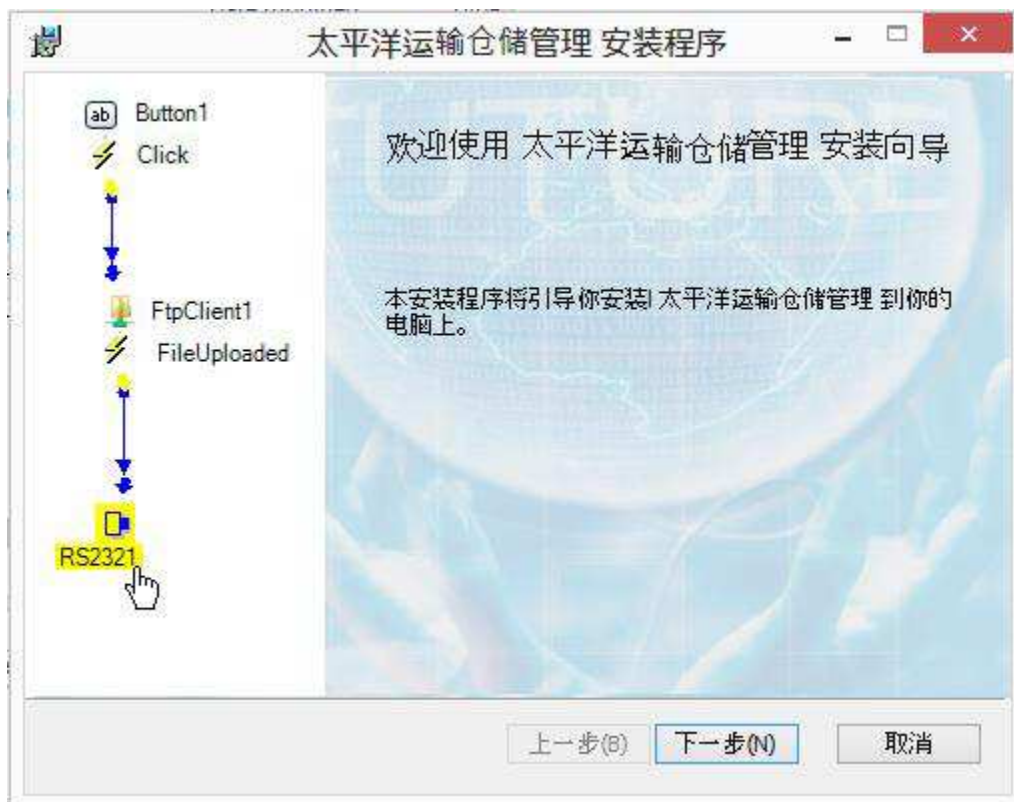
At the beginning of setup, the license file will be displayed:



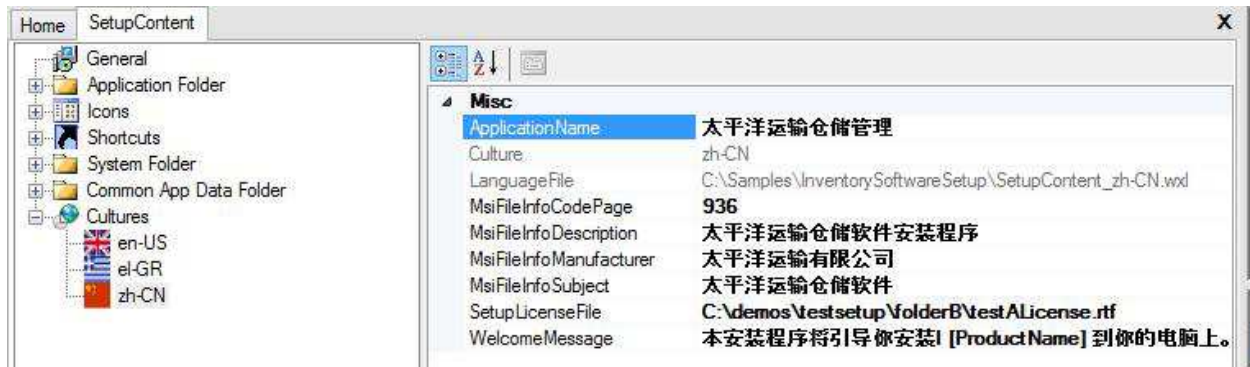
Note that for each culture, a separate MSI file is generated:



Each MSI file uses corresponding language in its user interface.



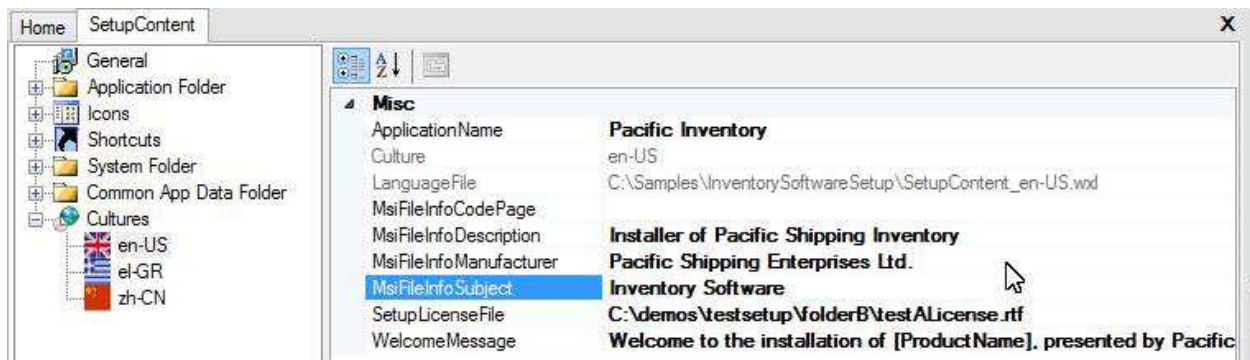
The above user interface is generated using following information:



## 5.5 MSI File Information

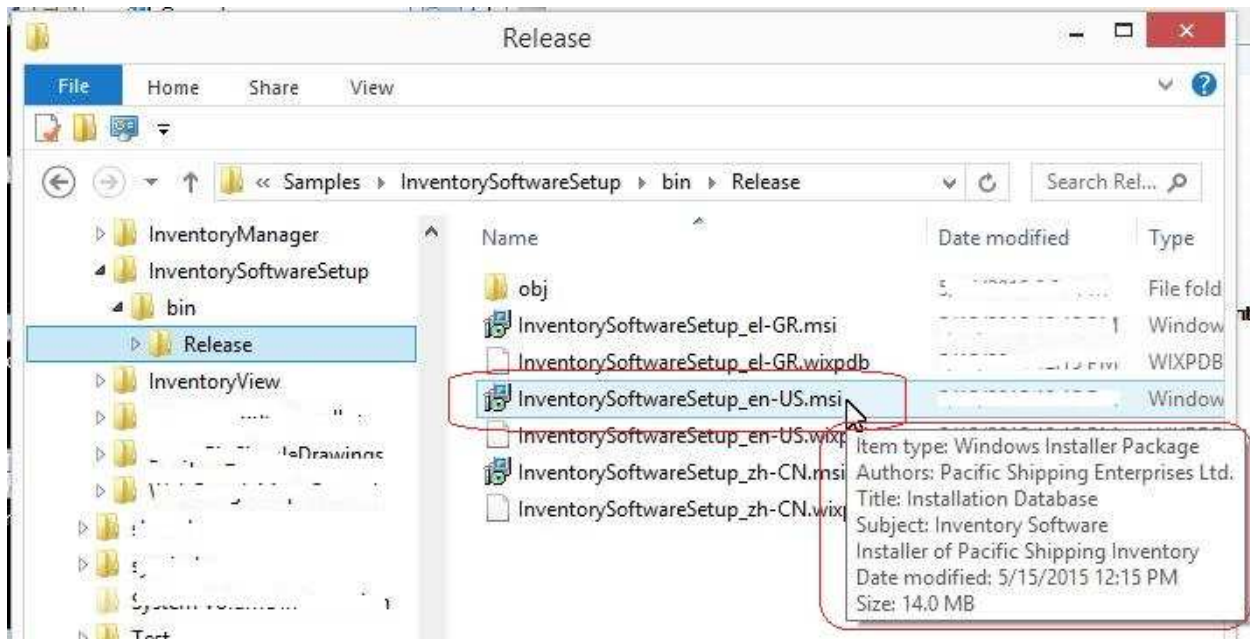
### 5.5.1 File Summary Information

When a MSI file is generated, you may specify file information through **MsiFileInfoManufacturer**, **MsiFileInfoSubject**, and **MsiFileInfoDescription**.



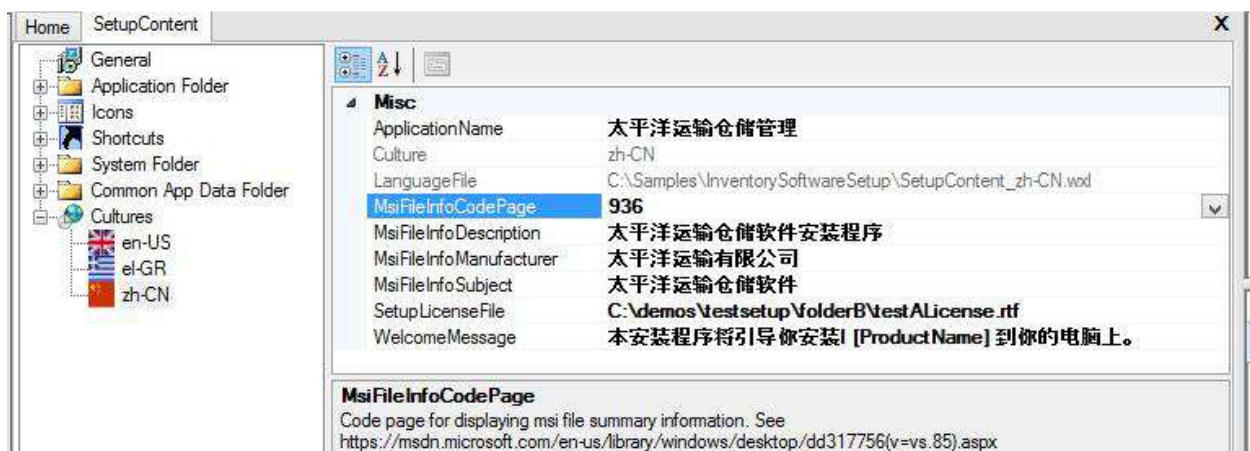
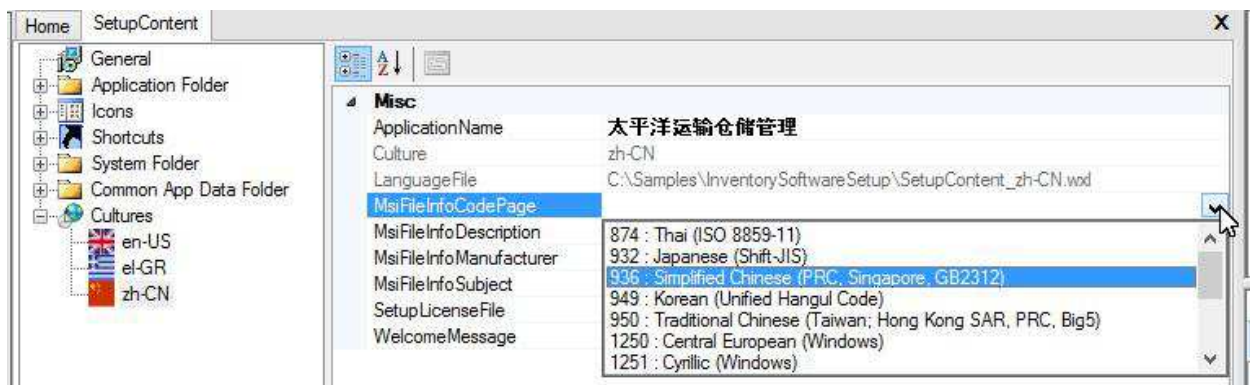
**MsiFileInfoManufacturer** is for Author of MSI file, **MsiFileInfoSubject** is for Subject of MSI file, and **MsiFileInfoDescription** is for Description of MSI file.

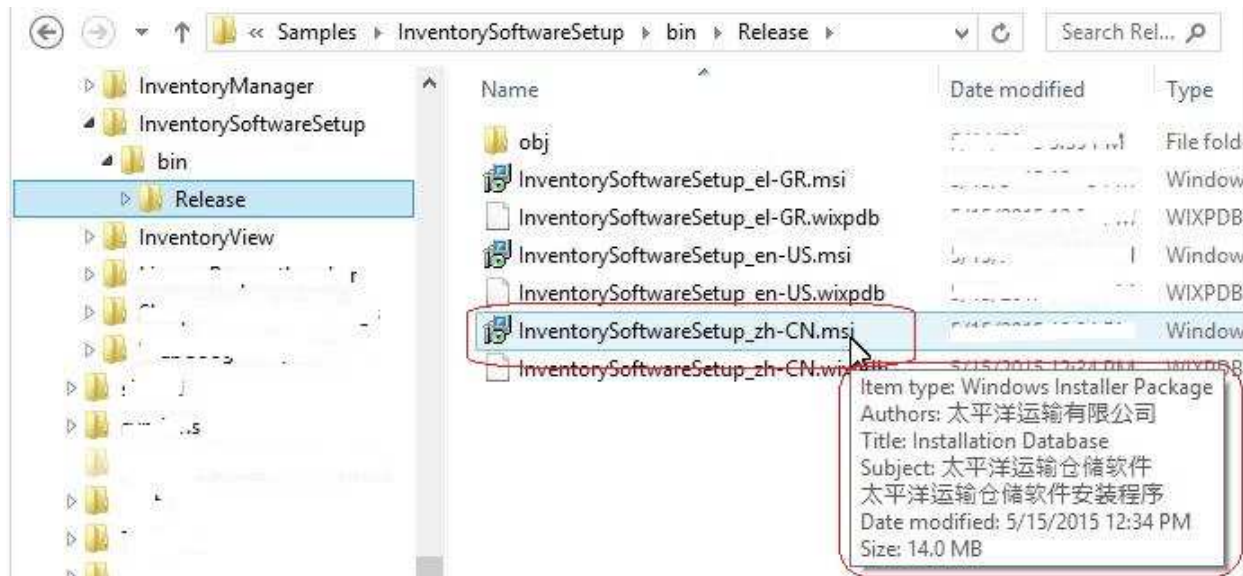




### 5.5.2 Code page for MSI file information

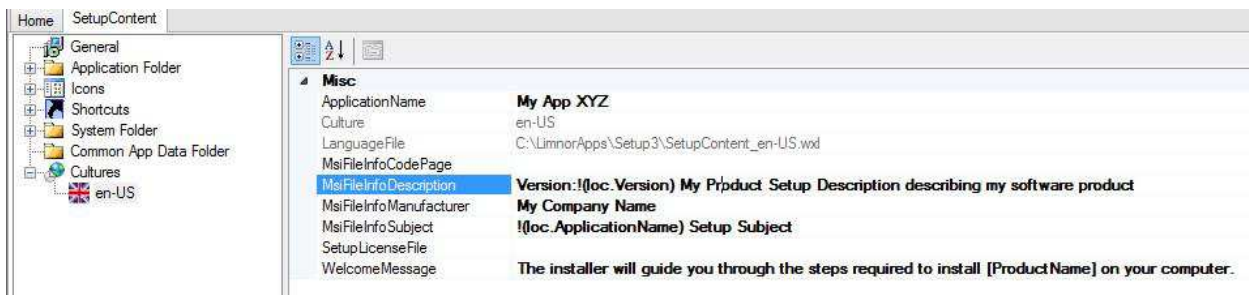
A code page should be specified to match the language used for writing MSI file information.



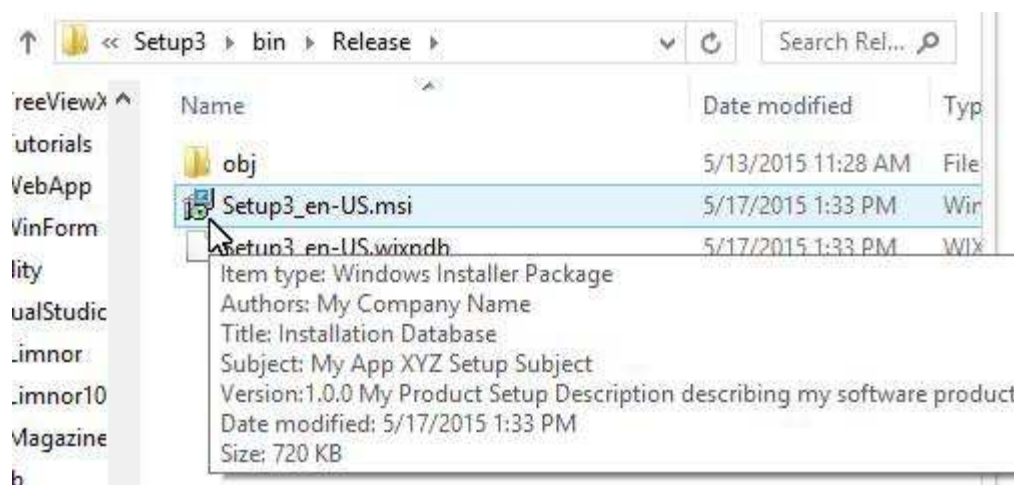


### 5.5.3 Display Version in File Information

MSI file information does not include a field for displaying version information. You may use a variable `!(loc.Version)` to represent the installation version in **MsiFileInfoManufacturer**, **MsiFileInfoSubject**, and **MsiFileInfoDescription**. Following example shows such usage:



The above example generates following MSI file information:





In the above example, a variable !(loc.ApplicationName) is used in MsiFileInfoSubject. This variable refers to the setting of ApplicationName. Note that this same variable is represented by [ProductName] in WelcomeMessage.

## 6 Feedback

Please send your feedback to [support@limnor.com](mailto:support@limnor.com).