DelphiDabbler SIBuilder

Tutorial

Overview

This tutorial walks you through development of a simple installation project using *SIBuilder*. The project we will develop will use the bonus in-process COM server registration program distributed with *SIBuilder 3*.

The project will do the following:

- Provide an uninstall program.
- Ask the user to either confirm the default installation path or provide a different one.
- Display a license that the user must accept to continue with the installation.
- Install the main application file and a read me file in the root of the installation folder.
- Install a COM server DLL that is required by the application into the `DelphiDabbler` sub-directory of the `Common Files` folder.
- Register the application with Windows and register the DLL as a shared DLL and as a COM server
- Run a batch file to dynamically create a file that will be displayed the first time the program is run and will be stored in the main install folder. The batch file will then be deleted.
- Display a "readme" file in Windows *Notepad* at the end of the installation.

> ***Note:** Some of the facilities described above are unlikely to be included in a real world project, but have been included here in order to demonstrate some of the major facilities of the program.*

Before we can begin to create the install project we need to collect the required files. The next section provides information on where to find these files.

Preparation

We need five files in order to create the installation project:

- `RegInProcSvr.exe` – to be installed in the install path and registered as an application with Windows.
- `ReadMe.txt` – to be installed in the install path and to be displayed after the installation completes.
- `RegCOMSvrs.dll` – to be installed in the `DelphiDabbler` sub folder of the `Common Files` folder and registered as a COM server and shared DLL.
- `RegInProcSvr.lic` – contains the license to be displayed by the installer (not permanently installed).

- `CreateFile.bat` – to be run by the installer to create the file that is displayed the first time Windows starts up after the installation.

Of these files `RegInProcSvr.exe` and `RegCOMSvrs.dll` are provided with your *SIBuilder* installation - see the `Extras` sub folder of your *SIBuilder* install path for `RegInProcSvr.exe` and the `DelphiDabbler` sub folder of the `Common Files` folder for `RegCOMSvrs.dll`.

The other three files are not provided with the *SIBuilder* distribution, but a zip file containing them can be downloaded from DelphiDabbler.com. Unzip the files into some convenient location as we will be accessing them later.

Now we have the files we need, we can get started. Run *SIBuilder 3* and select the *File* | *New Blank* menu option. This will start a new project and display the Project Details page ready to start entering details of the project.

Project Details

Enter the main details of the project on the Project Details page as follows.

Project Identification section

Enter `RegInProcSvr` in the *Project ID* edit box

> This name should uniquely identify the project – it must contain only alphabetic and numeric characters along with the underline character.

Enter `COM Server Registrar by DelphiDabbler` in the *Project name* edit box.

> This is the description of the program that is displayed in the *Add/Remove Programs* applet accessed from the Windows Control Panel.

Installation Path section

Enter `[Program Files]\DelphiDabbler\RegInProcSvr` in the *Installation path* edit box.

> This will cause the program to be installed in the `DelphiDabbler\RegInProcSvr` sub-folder of the target system's `Program Files` folder. As an alternative, if the folder already exists on your system you can click the button at the right of the edit box and choose the required folder from the dialog box that is displayed. Special paths, such as `Program Files`, are recognised by *SIBuilder* and are converted to a macro such as `[Program Files]`.

Check the *Editing of installation path permitted* and *Ask user for path on installation* check boxes.

> This causes the install program to pause and prompt the user for an installation path: the path entered in the *Installation Path* edit box will be used if the user accepts the default.

Install Program section

Enter `RIPSInst.exe` in the *Install program name* edit box.

> This is the name that will be given to the install program – do not include a path in the name since *SIBuilder* creates the program in the same folder that the project file is stored in.

Check the *Create uninstall program* check box.

> This causes *SIBuilder* to create an uninstall program as well as the install program. This program will always be named `UnInstall.exe`.

Previous Installations section

Ensure that the *Uninstall previous installations in same folder* check box is checked.

> This enables the installer to uninstall any previous program in the same folder before performing the installation. This option is required to enable us to register shared DLLs properly, which we will do later.

We could also specify other programs to be uninstalled before proceeding with the installation by listing them in the *Uninstall project(s)* edit box, but we do not need to do this in this example.

> ***Note:** this option is only available from SIBuilder release 3.1.*

Save the project

Save the project now by selecting the *File | Save* menu option and choosing a location to save the file. The install program will also be stored in this location. Save the file as `RegInProcSvr.sip`.

Now we have provided all the basic details of the project, the next step is to inform *SIBuilder* we want the installer to display a license.

License Information

We will now specify the license to be displayed by the installer. Move to the *License Info* page by clicking the *License Info* button at the left of the main window.

Ensure a license is displayed

First we check the *Display license* check box to tell *SIBuilder* we want to include a license in the project. The remainder of the controls on the page will now be enabled.

License File

We now need to specify a file containing the license we want to display. We use the file `RegInProcSvr.lic` we downloaded earlier.

Enter the license file name (including path to where you unzipped it) in the *Source license file* edit box. The file name can be typed into the edit box, but it may be easier to click the button that appears at the right of the edit box and to select the file from the dialog box that is displayed. Clicking OK in the dialog box displays the file name in the edit box.

Dialog Box Properties

Next we configure the license dialog box as follows.

Ensure that *Accept & Decline* is selected in the *Which buttons* drop down list.

> This makes the dialog box display both *Accept* and *Decline* buttons. When the dialog box is displayed by the installer, the user must click the *Accept* button to proceed with the installation (clicking *Decline* aborts the installation).

Select *Large* in the *Dialog size* drop down list.

> The installer can display the license dialog in three sizes – small, medium and large. We need to display a large dialog box to avoid unnecessary word wrapping.

Test the Dialog box

The dialog box can now be tested to ensure that it displays as expected. Click the *Display License* button to display the dialog. Click either the *Accept* or *Decline* buttons to close the dialog box.

> ***Note:*** *The displayed dialog box will include a help button, while the version displayed by the installer will not.*

We have now completed configuration of the license dialog box. The next step is to determine which operating systems the program can be installed on.

Target Operating System

It is now time to specify the valid operating systems for the program installation. Move to the *Target Operating Systems* page by clicking the *Target OS* button at the left of the main window.

Let us assume the program will run on any operating system except Windows 95. Therefore we clear the *Windows 95* and *Windows 95 OSR2* check boxes. When this is done, the panel below the check boxes will display *Windows NT Platform, Windows 98, Windows 98 SE, Windows ME*.

That's all there is to configuring the operating systems – the installer will now refuse to run on Windows 95. It's at last time to choose the files to be installed.

Files

We are now ready to so specify the files we need. In this installation we need the following files:

- `RegInProcSvr.exe` – to be installed in the install path and registered as an application with windows.
- `RegCOMSvrs.dll` – to be installed in the `DelphiDabbler` sub folder of the `Common Files` folder and registered as a COM server and shared DLL.
- `ReadMe.txt` – to be installed in the install path and to be displayed after the installation completes.
- `CreateFile.bat` – temporary batch file to be used to create a file to be displayed the next time Windows is run.

Move to the *Files* page by clicking the *Files* button to the left of the page.

File are installed within groups – there will be one empty *Main* group already on the page. We will use this group to install `RegInProcSvr.exe` and `ReadMe.txt`. We will also require further groups to install `RegCOMSvrs.dll` and `CreateFile.bat` since these files are installed to a different folder to the other files, and each group is installed to a specified folder.

The following steps show how to specify the files to be installed.

Add RegInProcSvr.exe to the project

We need to add `RegInProcSvr.exe` to the project and configure the file so that it only overwrites earlier versions of itself and so the program is registered as an application with Windows. We do this as follows:

1. Select the *Main* group, click the *Add Files* button and, using the file dialog that is displayed, find `RegInProcSvr.exe` in the `Extras` sub-folder of your *SIBuilder* installation. Click *OK* to add the file to the group.
2. Select the newly added file and click the *Edit File* button to display the *Edit File Install Properties* dialog box.
3. Select *Only overwrite file with lower file version* in the *File overwriting options* drop down list.
4. Check the *Register application path with Windows* check box.
5. Click OK to close the dialog box.

Add the ReadMe file to the project

We need to add the readme file to the project since we want to display it in *NotePad* after the installation has completed. We also want to keep the file for later use.

1. Using the *Add Files* button again, locate and add the `ReadMe.txt` file to the project.
2. Select the file and click *Edit File* to display the *Edit File Install Properties* dialog box once again.
3. Select the *Only overwrite file with older time stamp* option in the *File overwriting options* drop down list so that, should the file already exist, only earlier copies will be overwritten.

Add RegCOMSvrs.dll to the project

This file needs installing into a different folder to the above, so we need to create a new group for it. We also want to register the DLL as a COM server and as a shared DLL.

1. To create a new group for the DLL click the *Add Group* button and in the resulting *Edit Group Properties* dialog box make the following entries:
    1. In the *Group name* edit box enter a unique group name, e.g. `Shared`.
    2. In the *Installation path* combo box, drop down the list and select `[Common Files]` which will be copied into the edit box. Place the cursor at the end and add `\DelphiDabbler` so that the entry reads `[Common Files]\DelphiDabbler`.
    3. Ensure the *Remove folder if empty* and *Delete only files that were installed* radio buttons are selected.
    4. Close the dialog box using the *OK* button – the new group will appear in the tree.
2. Click the *Add Files* button and select the `RegCOMSvrs.dll` file from `DelphiDabbler` sub folder of the `Common Files` folder.
3. Select the newly added file, click *Edit File* to display the *Edit File Install Properties* dialog box for the DLL.
4. Select *Only overwrite file with lower file version* in the *File overwriting options* drop down list.

5. Check the *Register as shared DLL with Windows* and *Register COM in-process server with Windows* check boxes.
6. Close the dialog box by clicking *OK*.

Add CreateFile.bat to the project

This batch file is not required once it has been run by the installer. We create a new temporary group for the file.

1. Create the new group by clicking the *Add Group* button to display the *Edit Group Properties* dialog box and make the following entries:
    1. Give the group a new name in the *Group name* edit box, e.g. `Temp`.
    2. Enter `[Temp]` in the *Installation path* combo box. This specifies the system's temporary folder as the place in which to install the batch file.
    3. Select the *Do not remove folder* radio button in the *Folders* group box – we don't want to delete the user's temporary folder!
    4. Select the *Temporary files deleted by installer* radio button in the *Files* group box – to make the installer delete the file before closing.
    5. Click *OK* to close the dialog.
2. Click the *Add Files* button and select the `CreateFile.bat` file. The default properties for this file are acceptable so there is no need to edit these.

We have now added all the required files to the project. Next we need to specify any programs to be run by the installer.

Programs

Move to the *Programs* page by clicking the *Programs* button at the left of the window.

We want to run two programs:

- We run *Notepad* on the target system to display `ReadMe.txt`. *NotePad* will be found in the system's `Windows` folder.
- We run `CreateFile.bat` from the temporary folder to create a file in the installation folder called `InstallFiles.txt` (Later in the tutorial we configure the registry to get this file displayed when Windows next starts).

Both these programs will be run by the installer – we don't run any programs from the uninstaller.

Display the ReadMe.txt file

We set up a program entry to display `ReadMe.txt` in Windows *Notepad* as follows:

1. Click the *Add Program* to display the *Edit Program Details* dialog box.
2. We specify the program to be run in the *Program* combo box. Drop down the combo's list and select `[Windows]`. Now place the cursor after `[Windows]` in the combo's edit box and enter `\Notepad.exe`.
3. We only want to execute the program from the installer so select *Installation only* in the *Execute on* drop down list.

4. We pass the read me file as a parameter to *NotePad*. The file is installed in the main install path. Enter `[Install Path]\ReadMe.txt` into the Parameter edit box and click the *Add* button.
5. That's it – click *OK*.

Run the file creator batch file

We now enable the `CreateFile.bat` batch file to be run, passing the install and shared files folder.

1. Click the *Add Program* to display the *Edit Program Details* dialog box once more.
2. The batch file we want to run is part of the install project so we can select it from the *Program* combo's drop down list – it is listed as `[Temp]\CreateFile.bat`.
3. Again select *Installation only* in the *Execute on* drop down list.
4. We need two parameters:
    1. The first parameter is the install folder: enter `[Install Path]` in the *Parameter* edit box and click *Add*.
    2. The second parameter is the shared folder: enter `[Common Files]\DelphiDabbler` in the *Parameter* edit box and click *Add*.
5. Click *OK* to accept the program.

We have now created the programs we need to run. It's now time to configure the registry changes we need to make.

Registry

We need to configure the registry to display `InstallFiles.txt` in the installation folder the next time Windows starts.

Move to the *Registry* page by clicking the *Registry* button at the left of the window.

We need to create the registry key

```
HKEY_LOCAL_MACHINE\
  Software\
    Microsoft\
      Windows\
        CurrentVersion\
          RunOnce
```

and set a value that provides details of the program to be run.

To create the key proceed as follows:

1. Rather than creating each sub key individually we can use a predefined template to enter most of the required key: so click the *Templates* button to display *Choose Registry Template* dialog box. Drop down the *Choose template* drop list, select the *Current Version (HKLM)* entry and click *OK*. Entries for the key are now displayed in the registry tree.
2. We need to create a *RunOnce* sub key of the key we've just entered, so select the *CurrentVersion* sub key in the tree and click the *Add Key* button to display the *Edit Registry Key* dialog box. Enter `RunOnce` in the *Name of key* edit box and select *Never delete key* in the *Key deletion options on uninstall* drop down list – we don't want our uninstaller deleting this standard registry entry. Click *OK*.

We now need to create a string value indicating that we want to "run" `InstallFiles.txt`. Add the value as follows:

1. Select the *RunOnce* key in the registry tree and click the *Add Data Item* button to display the *Edit Registry Value* dialog box.
2. Give the value a name, e.g. `ViewInstallFiles`, by entering it in the *Name* edit box.
3. We are creating a string type, so select the *String* radio button in the *Data type* group box.
4. We want the installer to expand the path macros we will be using in the file path so check the *Expand Path Macros* check box.
5. The registry value is the name of the file to be displayed. (We rely on Windows using the default text editor to display the file). We enter the value in stages in the *Value* edit box:
    1. Enter a double quote.
    2. Use the *Insert path macro* button and select *[Install Path]* from the drop down menu.
    3. Complete the parameter file name by entering `\InstallFiles.txt` followed by a closing double quote.
    4. Alternatively simply type `"[Install Path]\InstallFiles.txt"` in the *Value* edit box.
6. Click *OK* to accept the new value. It will now appear under the *RunOnce* node in the tree.

> ***Note** that path macros will only be expanded if they form part of a recognisable file name: therefore passing `[Install Path]\InstallFiles.txt` as a parameter to* Notepad *would not work because the path macro would not be expanded.*

Now we have completed the registry entries we are ready to build the project.

Build the Installer

We're now in a position to build the project. Click the *Build Installer* button on the left of main window to display the *Build Installer* page.

First, select the kind of compression to be used. Do this by choosing one of the options from the compression drop down list. We will use *ZLib*.

Click the *Build* button to create the installer. After the build completes a message will be displayed telling whether the it succeeded. The install file is created in the folder where you saved the project file. It will have the name `RIPSInst.exe` (or whatever you called it on the Project Details page).

You may wish to create a report of showing the files that have been installed: click the *Report* button to create it and the *View Report* button to display it. The report will be stored in a file in the same folder in which the project was saved.

> ***Note:** The report will be created in either plain text or HTML. You specify which report type you require by selecting the* Tools | Options *menu option and selecting the required value from the* Report Kinds *drop down list.*

We're finished! Run the install program you just created to check that everything works. Restart Windows to make sure that `InstallFiles.txt` gets displayed in *Notepad*. If there are any errors reload the project file and check everything has been entered correctly.

> *If you are running the installer on Windows NT, 2000 or XP please make sure you have administrator privileges or the installation will fail.*
>
> *On Windows Vista the OS will automatically prompt for an administrator password if running as a standard user. If running as an administrator an elevation prompt will appear.*

To uninstall the program, go to the *Add/Remove Programs* Control Panel applet and uninstall the program from there.