

# **Enveloper for Software and Data Protection**

## **Using Pen Drive Protection**

---

### **User Guide**

**[For Standard Edition]**

## Contents

	Page No
1. General	3
2. How to Download	4
3. Pre-Requisites	5
4. Application User interface	6
5. Target System	7
6. Input Files	8
7. Security Information	10
8. Add Files	11
8a Adding Files	12
8b Adding Folders	14
8c Renaming files and folders	15
8d Adding Drives	16
8e Setting File Properties	17
8f Remove a file or folder	20
8g Save file list	21
8h Load file list	22
9. Output File	23
10. Error Messages while creating envelope	24
11. Error Messages while running enveloped exe	25

# 1. General

Software Envelope allows binding multiple files in to a single executable. The new enveloped executable is a binary file and it is very difficult to reverse engineer the file.

Normally when you use a software protection tool, you use a DLL to verify authorisation key from a hardware [it can be a dongle, USB flash drive, or any other device].

The hacker or cracker normally create an identical DLL to by-pass the functions or produces a duplicate hardware.

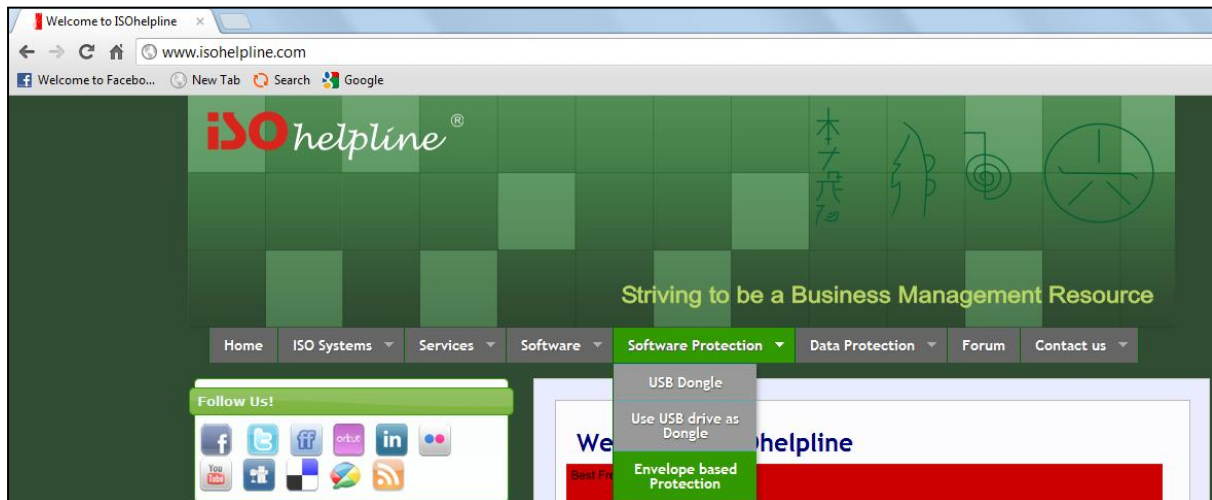
When you envelope the DLL and all other files in to a single executable, by-pass technique do not work as you have a single executable.

Envelopes can be used for one or more of following:

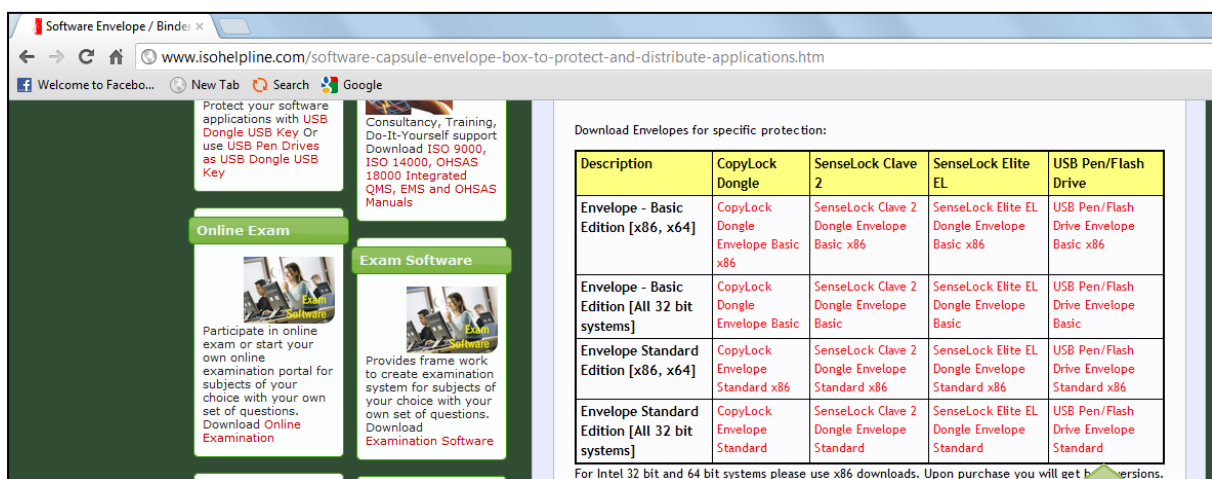
- Bind main program and DLL's together
- Bind main program and its data file [for example a PDF with PDF Reader]
- Run legacy software on advanced OS [for example a VB6 application on Vista and Win 7] by creating a single file which will not need installation
- Prevent reverse engineering of software
- Protect software with dongles / USB key or any other security tool.

Normally it is very easy to reverse engineer a .net program or any other program as disassemblers are easily available, normally free from web. Envelope will safeguard your programs from disassemblers.

## 2. How to Download



Go to [www.isohelpline.com](http://www.isohelpline.com), click “Software Protection” >> “Envelope based Protection”



Download the envelope suitable for your system.

1. **X86, x64**  
Select this option if **YOUR** Machine has Intel based CPU or compatible.
2. **All 32 bit systems**  
Select this option if **YOUR** Machine has non-Intel compatible CPU (for example: AMD Processor)

### Versions

**Basic:** Allows creating single protected EXE from an executable and [optional] a data file.

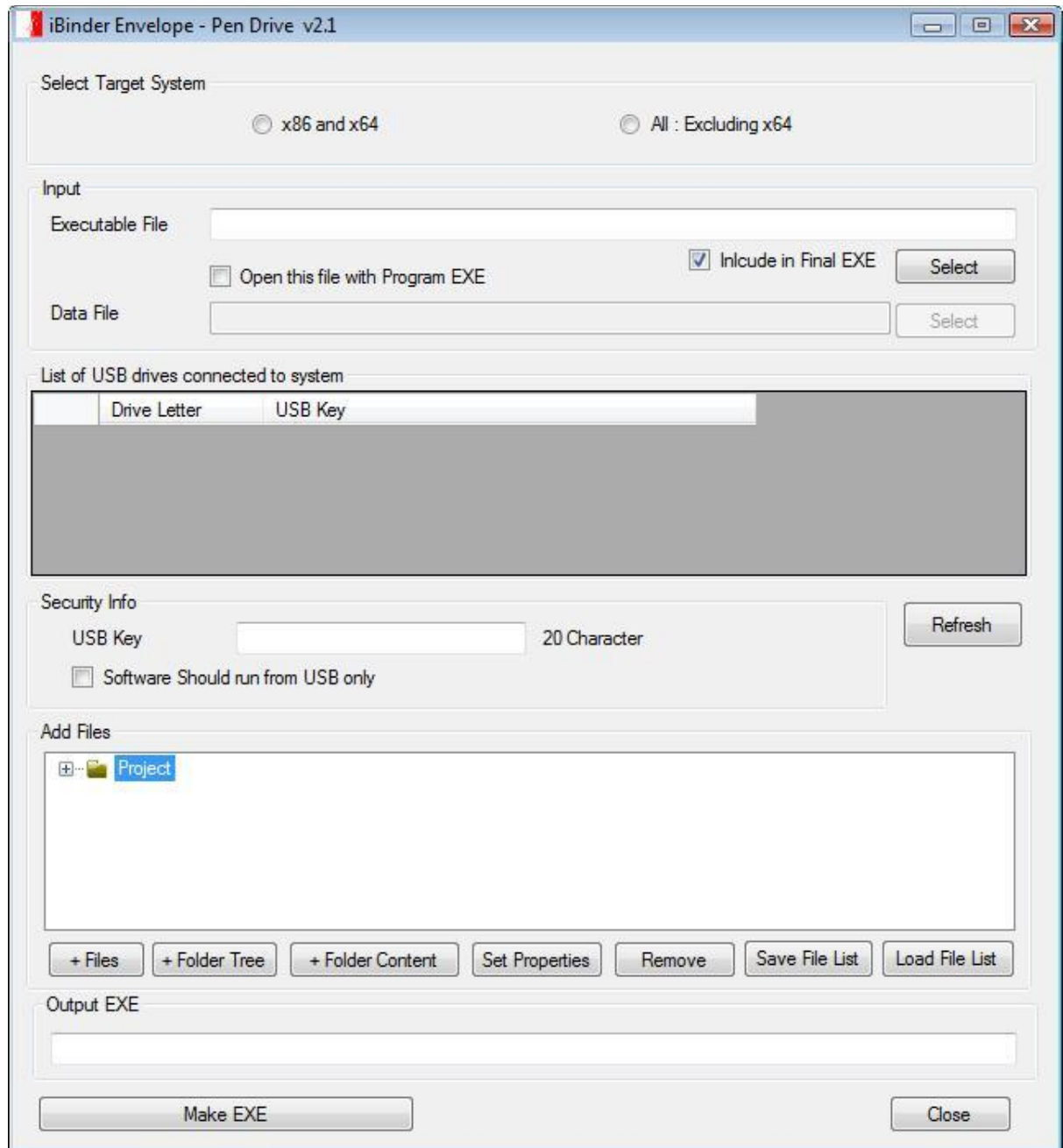
**Standard:** Allows creating single protected EXE from an executable, [optional] a data file and any number of other files like plug-in, DLL, COM, OCX etc.

### **3. Pre-Requisites**

The enveloper requires Microsoft Dot Net Framework 2.0 or 4.0

The final EXE which is distributed to end user also requires Microsoft Dot Net Framework 2.0 or 4.0

## 4. Application user Interface



**Make sure that your File path, File name and USB key should not have any Blank space.**

## 5. Target System

Technology has given us many good things but also created complexity for programmers. A program designed for 64 bit hardware do not work on a 32 bit hardware or even a Intel hardware program do not work [or do not work efficiently] on other hardware architecture like AMD.

Our envelope will allow you to select the target system for your final executable which you will be distributing. If your existing files are designed for x86, select the same here. Please note that this will not change the target system of your existing exe but will simply create a new executable for the selected system.



- 1. X86 and x64**  
Select this option if client Machine has Intel based CPU or compatible.
- 2. All : Excluding x64**  
Select this option if client Machine has non-Intel compatible CPU (for example: AMD Processor)

## 6. Input Files

The first file must be an executable file. The envelope will run this file only. Other files will depend on this first executable and can be used from inside this application or as a link.

### Guidance for Including “Executable File” in “Final EXE”:

- If the executable file is part of windows system, do not include it in the final EXE like “notepad.exe”
- If you are binding a data file and your main executable is not owned by you or you do not have distribution rights, do not include executable in final EXE.

### Data File

Data file, if added will be opened as a command line operation. Your main program [executable] should support command line operation to open the file. For example:

```
C:\notepad.exe readme.txt
```

The above command will run notepad and open readme.txt file.

The screenshot shows a software interface with the following elements:

- Input Executable File:** A text field for entering the name of the executable file.
- Data File:** A text field for entering the name of a data file.
- Include in Final EXE:** A checked checkbox with a 'Select' button next to it.
- Process Data file with EXE:** A checkbox.

Three callout boxes provide instructions:

- 1.** Uncheck this checkbox if you don't want to bind the Executable EXE in Output EXE **Otherwise** keep it checked.
- 2.** We can Choose Executable File with User Interface by Pressing on the Select button.
- 3.** You can also type the name of the Executable File name [if “Include in Final EXE” is unchecked”] Like: Notepad.exe  
Remember that file name is case sensitive for enveloper.



Input

Executable File: Notepad.exe

Include in Final EXE [Select]

Protect data file with EXE

Data File: [Select]

Select the data files you want to protect with this EXE by first clicking on the check box as pointed.

Input

Executable File: Notepad.exe

Include in Final EXE [Select...]

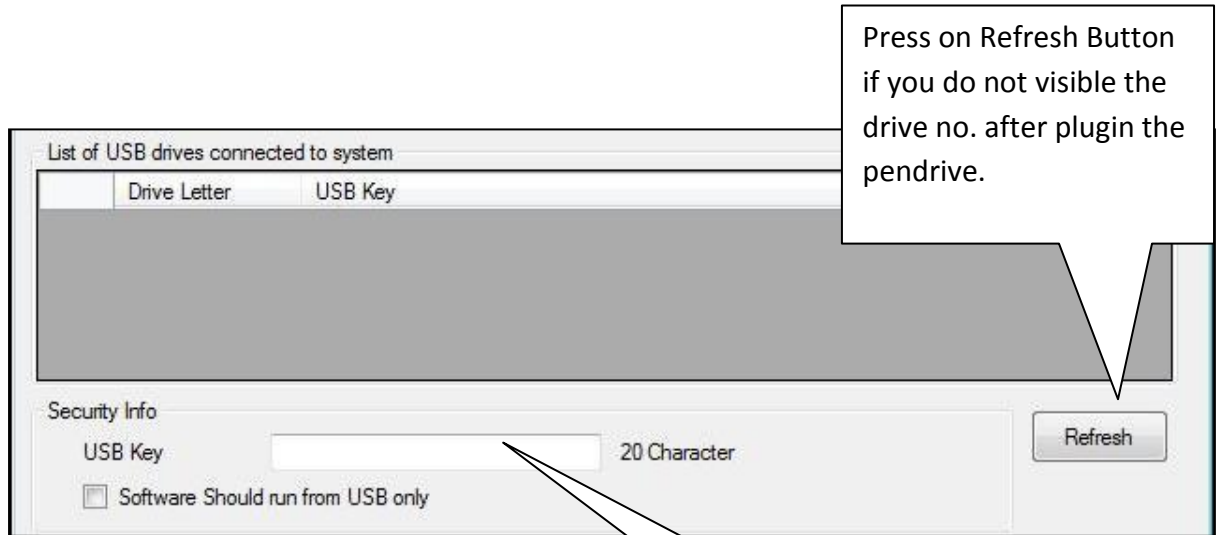
Protect data file with EXE

Data File: C:\demo\testing.txt [Select...]

After pressing on the select button open a new dialog, and then select the data file.

## 7. Security Information

This information depends upon the security hardware. This will ensure verification as per data specified by you.



Or  
Simply type the exact drive no. of the drive for which you want to make the enveloped exe.

Select the Drive no. of the drive which will be displayed that act as a USB key.

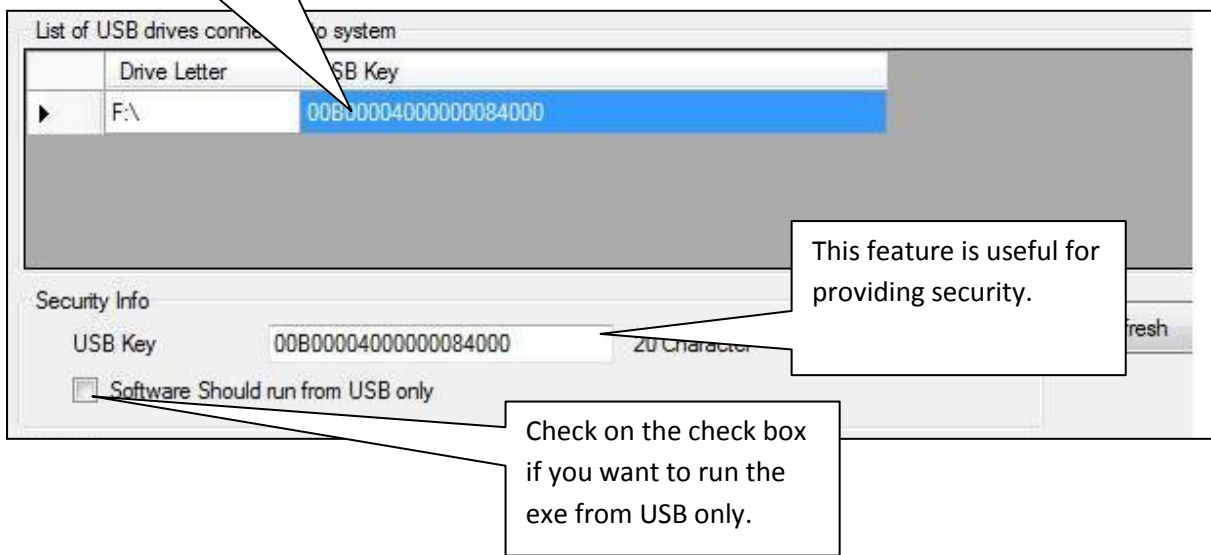


Image showing filled data

## 8. Add Files

This field allows you to add and remove multiple files, folders, contents inside the folder and also to set the properties of the files and folders.

- To add a file, click on "**+ Files**".
- To add a folder along with its contents, click on "**+ Folder Tree**".
- To add only the contents of the folder, click on "**+ Folder Content**".
- To remove a file or any folder, select the particular file or folder, and then click on "**Remove**".
- To change the file or folder properties select the particular file or folder and click on "**Set Properties**", a "**File Properties**" editor dialog will appear. [**Set Properties** is explained later]

If you want to use list of files a number of times then add the files once and then save the file list by clicking on "**Save File List**", it will save the entire file list in a "**.ibinder**" file.

To use this saved file click on "**Load File List**" which will open a new dialog then choose the saved file. This will load the files again. Remember that the ibinder file keeps the original location of the added files intact and hence "**Load File List**" will work only for the same project or for simple modifications on the same project on same system.

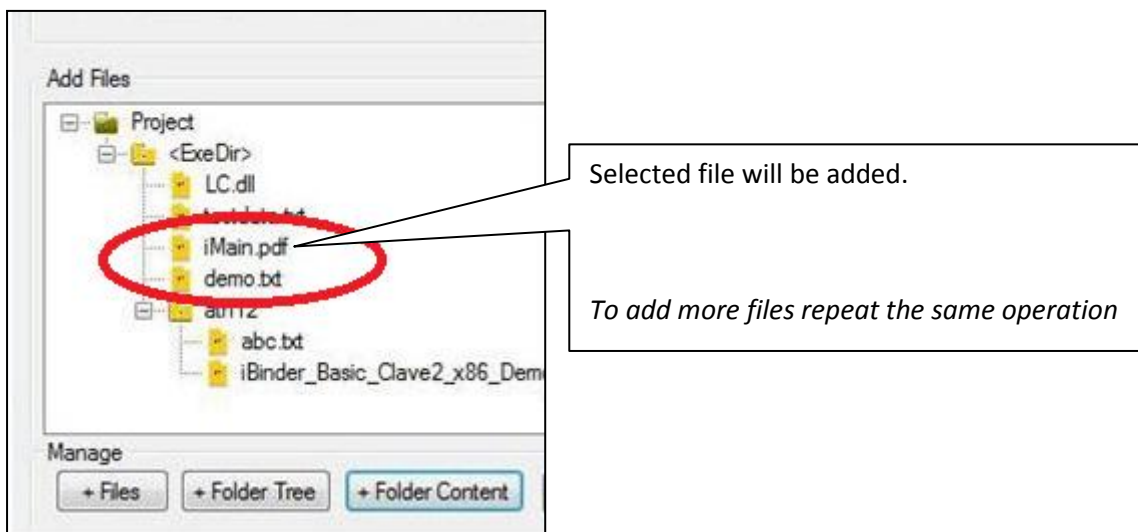
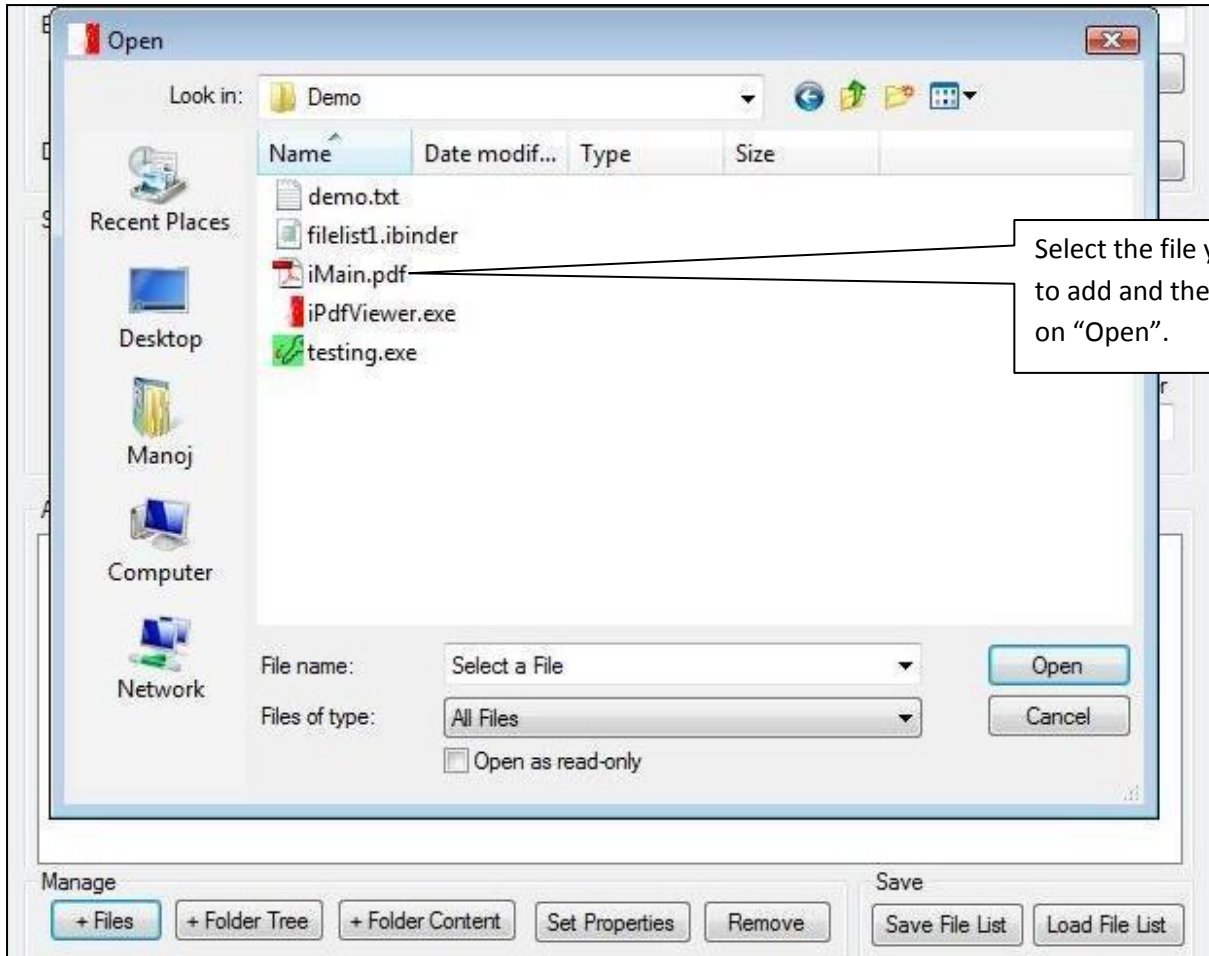
**For fast and proper binding, copy all files you want to add in the EXE in the enveloper folder keeping desired directory structure.**

## 8a. Adding Files

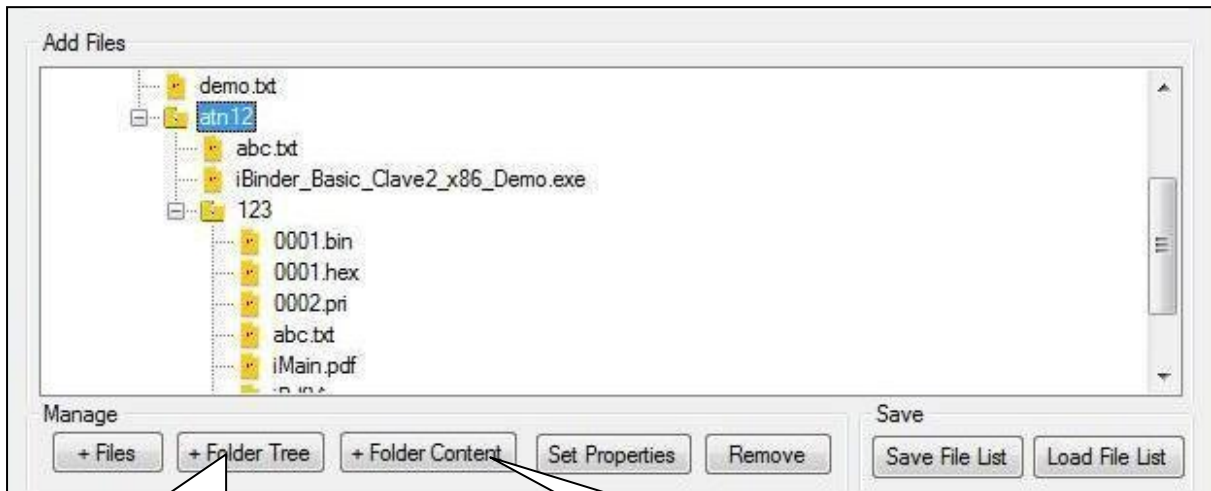
To add a file, click on "+ Files".



In order to add files click on the "+ Files" which will open a new dialog.

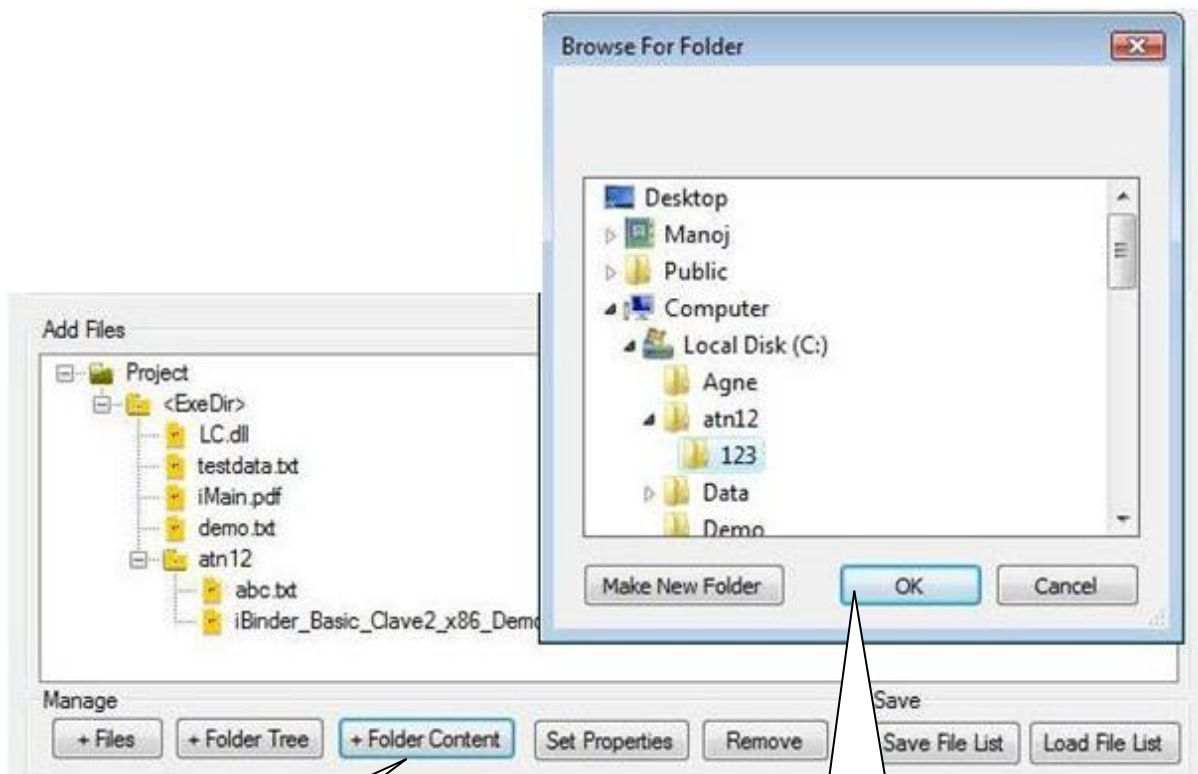


## 8b. Adding Folders



You can add folder and its contents by clicking on the  
“+ Folder Tree”

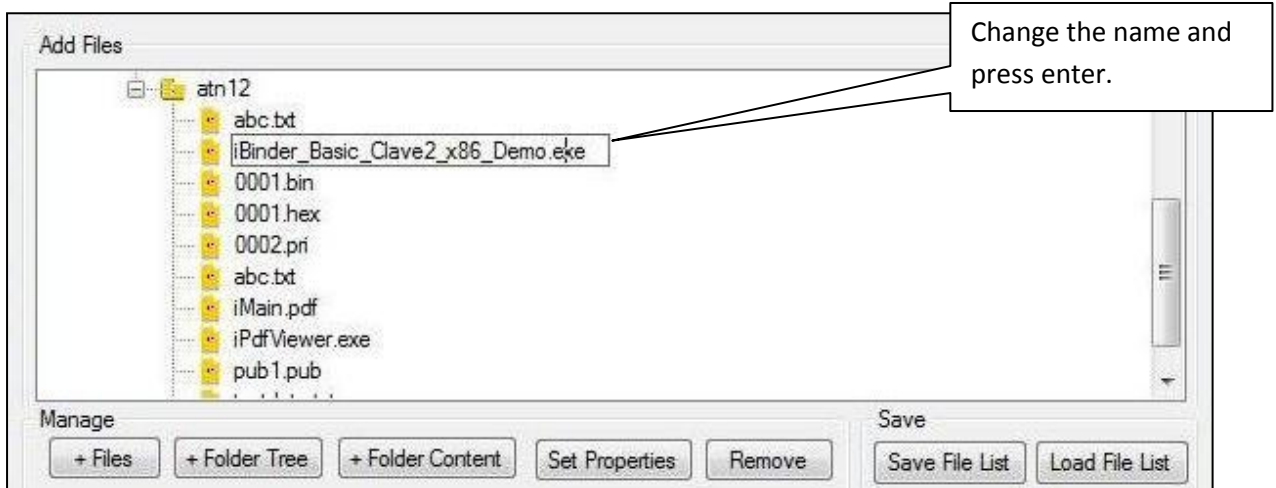
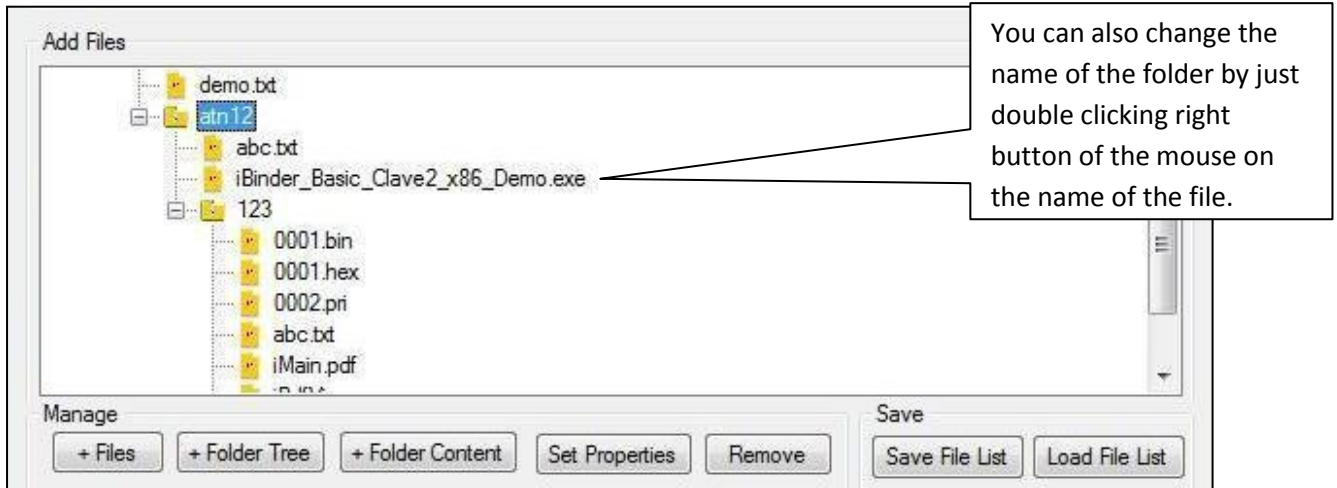
You can add contents of the folder [and not the folder itself] by clicking on the  
“+ Folder Content”



1. Press on “+Folder Content” which will open this dialog.

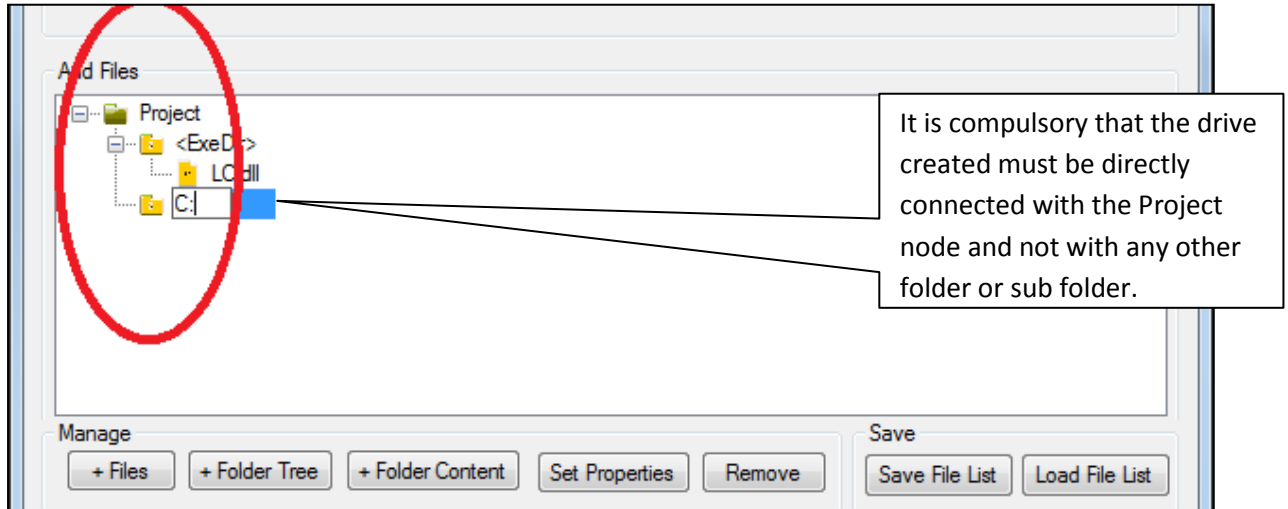
2. Browse the folder whose content you want to add, and then press OK.

## 8c. Renaming files and folders



## 8d. Adding Drives

To add a “drive”, simply add a folder on “**Project**” node using “+Folder Tree” and change it to “drive letter” like “C:”.



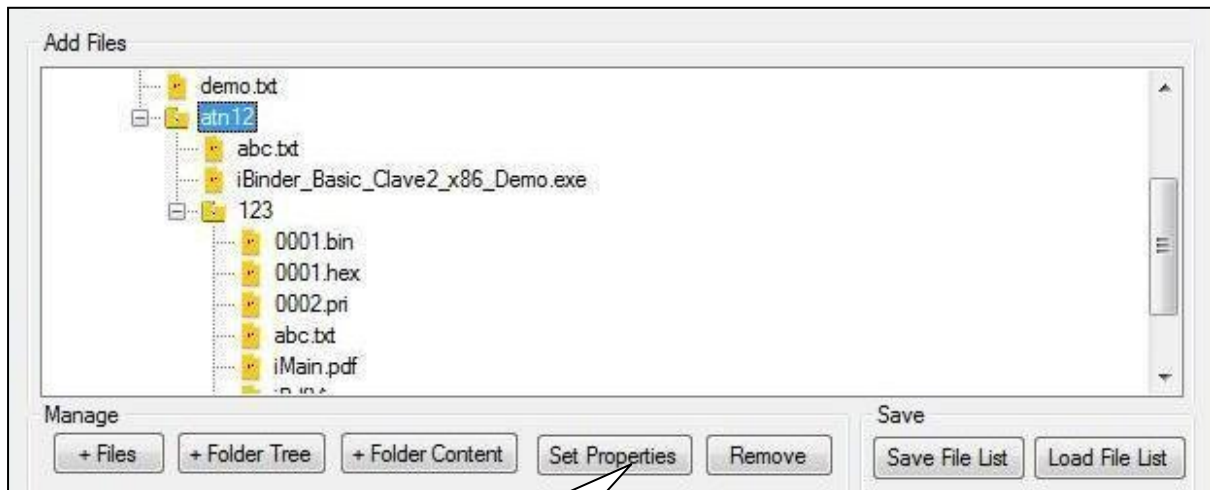
If you want to add a new directory structure starting from a drive letter, treat the folder as drive. Add the folder using “+Folder Tree” and change the name of the folder with the alphabet along with colon (:) like “C:”

Make sure that the folder which you want to treat as drive is on “**Project**” node in tree view.

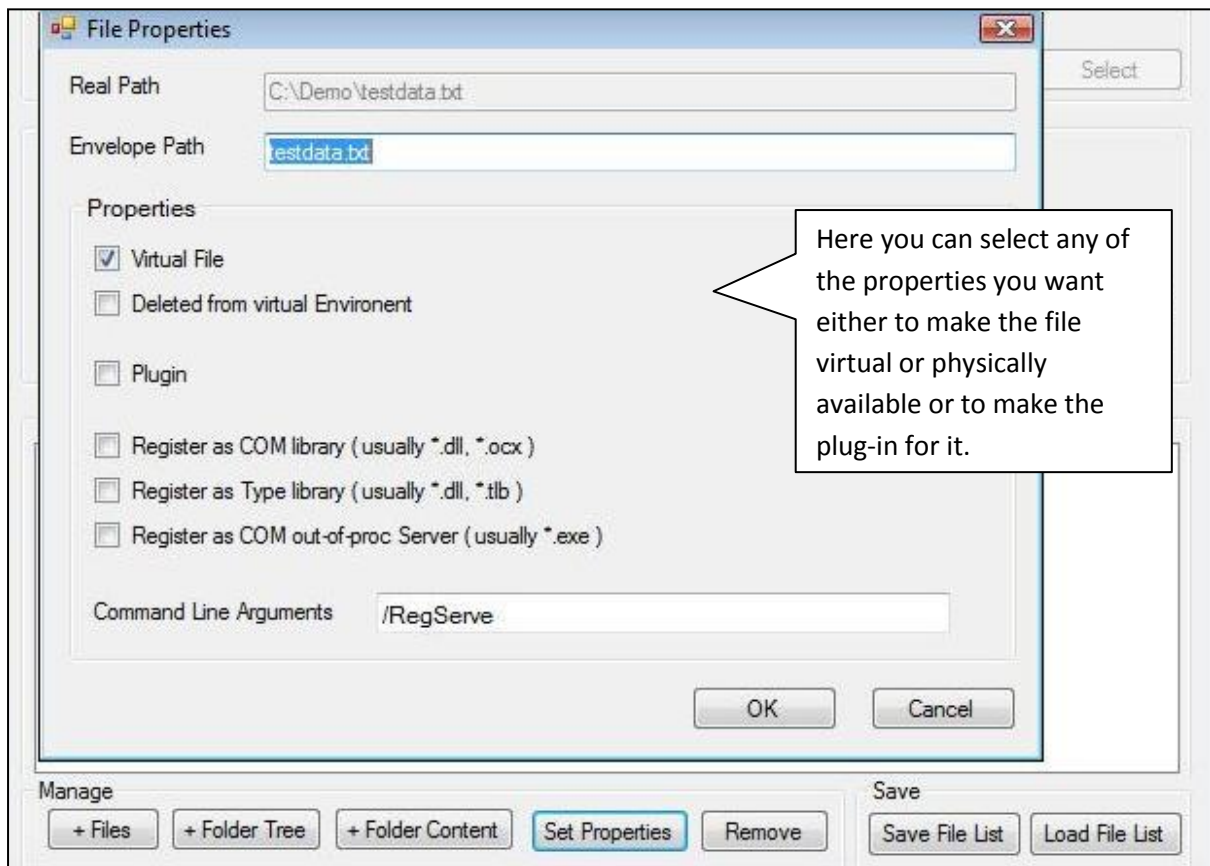


## 8e. Setting File Properties

For setting the properties of the file or folder, click on “**Set Properties**” and do the following.



You can set the properties of the file or folder by selecting the particular file or folder and press on Set Properties.



Here you can select any of the properties you want either to make the file virtual or physically available or to make the plug-in for it.

### **Virtual File:**

If you do not want file to be accessible to normal user, make it as virtual through clicking on the virtual property of that file. It will be accessible only to your application. In some cases, if sub process [second application running from the envelope] is not able to access your file, uncheck it.

To mark the file as virtual, select the "**Virtual File**" checkbox. Virtual file is a file which will be embedded in the resulting exe. The path where the virtual file will be created when the application starts is specified in the virtual path field. It can be (and normally is) a relative path.

### **Deleted from virtual Environment:**

Check if file is not to be used by any additional or sub process [second application running from the envelope]

### **Plug-in:**

An individual category of files is plug-in. They are DLLs that expand the functionality of the application being created.

To add the plug-in, add it by clicking on the "**Select**" button. Plug-in can also be embedded in the exe to be generated or, they can be stored along with it.

### **Register as COM Library:**

COM/ActiveX components are traditionally registered in the system registry to allow the corresponding Windows API functions find the modules (DLL or EXE) that contain the code of the components.

A major problem that occurs with the registration is the lack of user privileges, necessary for writing to the registry.

Also, when creating portable applications, it is important to be able to run applications without prior installation.

When adding a new file to the project, you can specify that it's an ActiveX / COM library, and it needs to be registered in the virtual registry when the application starts.

For example, suppose your application plays flash movies. Then you simply add the flash.ocx file and specify that this file is virtual, "check" "Register as COM Library [dll/ocx]" checkbox - and your application no longer depends on whether or not the client computer has a flash player installed.

Why does that happen? The thing is that when the application starts, it calls the function for registering the specified ActiveX at the same time, for the time the function is running a special flag remains set, and all changes are saved in the virtual environment. In other words, the real registry doesn't get modified, and the rest of the application's code runs just as if the ActiveX was actually installed in the system.

And what's especially important is that the modification of the original exe isn't required.

**Register as Type Library:**

If your file is a type library and requires registration before use, check this and your application will run as if type library is registered

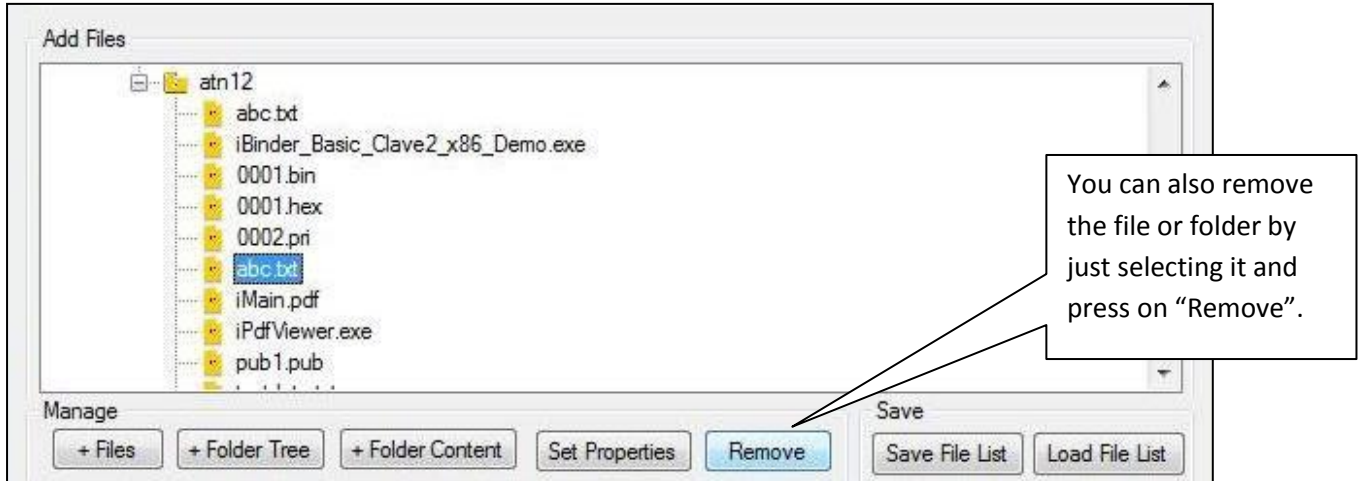
**Register as COM out-of-proc server:**

In-process COM servers (in a DLL) are normally more efficient than out-of-process servers (in an EXE, locally or on a remote machine) because data marshalling and transferring is bypassed since the client and server live in the same address space. As an example, high-performance DirectX components are always in the in-process servers. However, there are occasions that an out-of-process COM server is preferred, even locally.

If your file is an in-process COM server, select this option.

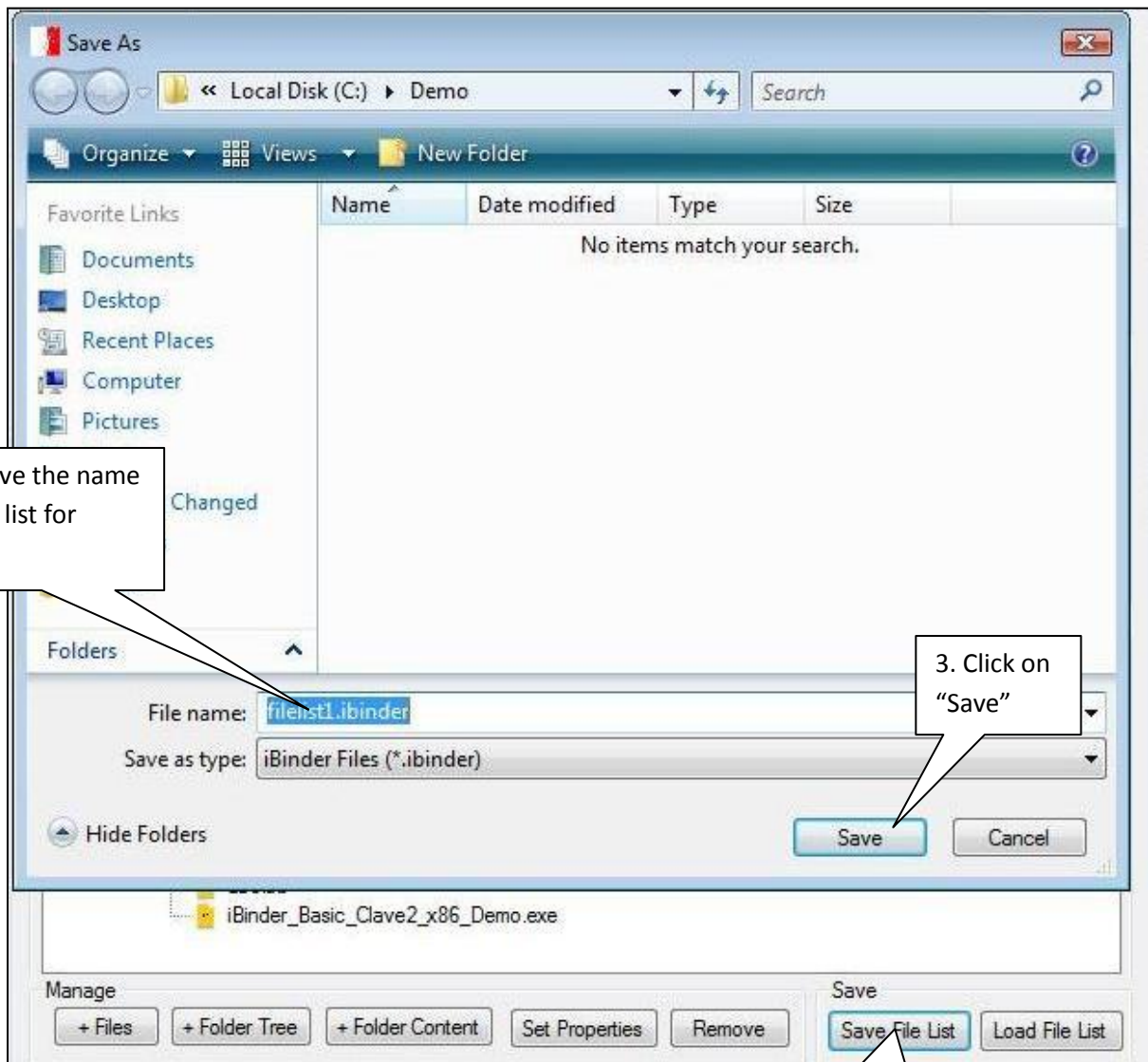
## 8f. Remove a file or folder

To remove a file or folder select it and click **“Remove”**.



## 8g. Save File List

If you want to use list of files a number of times then add the files once and then save the file list by clicking on “**Save File List**”, it will save the entire file list in a “.ibinder” file.



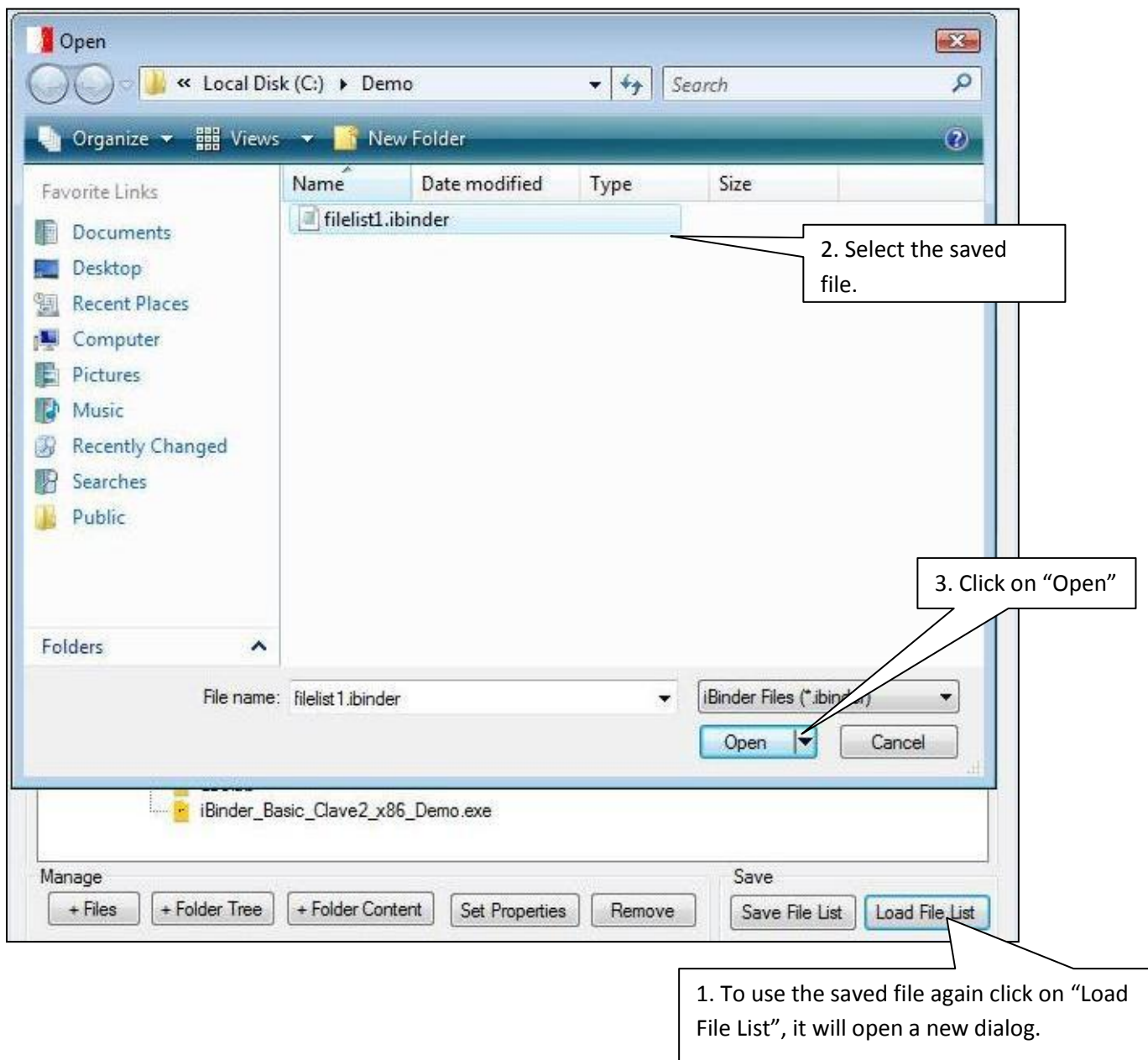
2. Then give the name of the file list for saving.

3. Click on “Save”

1. To avoid including the same files again and again you can save the file list for future use by clicking on the “Save File List”, then it will open a new dialog.

## 8h. Load File List

To use this saved file click on **“Load File List”** which will open a new dialog then choose the saved file. This will load the files again. Remember that the ibinder file keeps the original location of the added files intact and hence **“Load File List”** will work only for the same project or for simple modifications on the same project on same system.

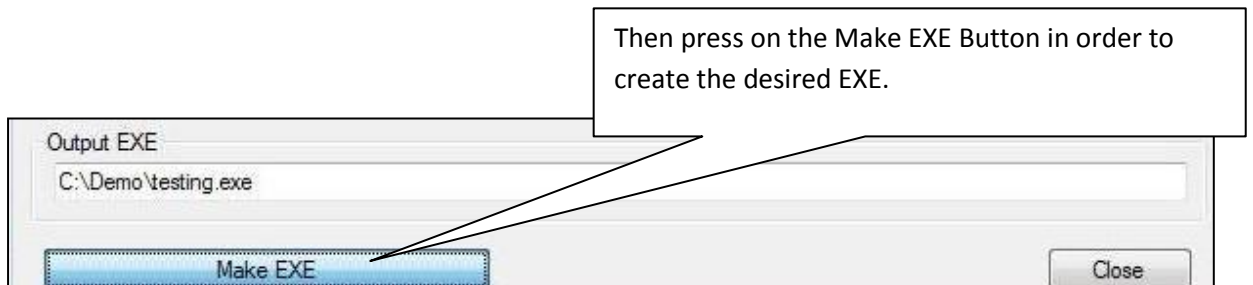
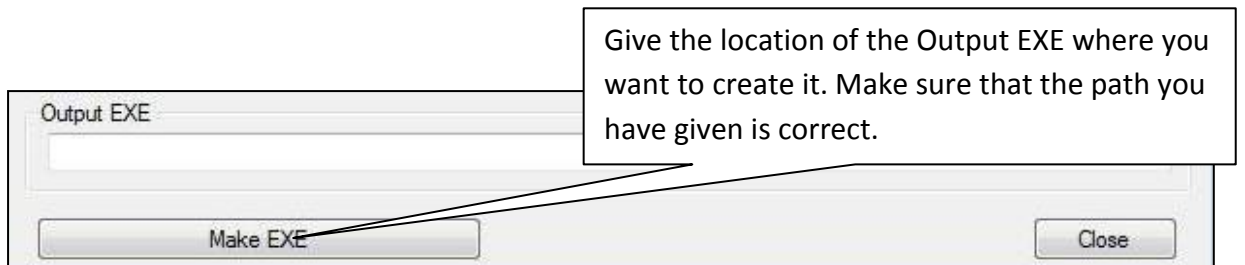


## 9. Output File

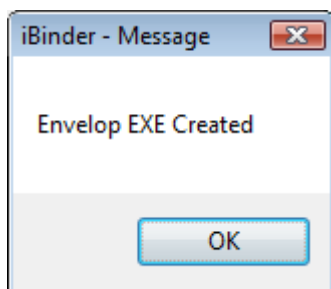
This is your final single EXE including all the files you have selected to add.

Specify the proper file location.

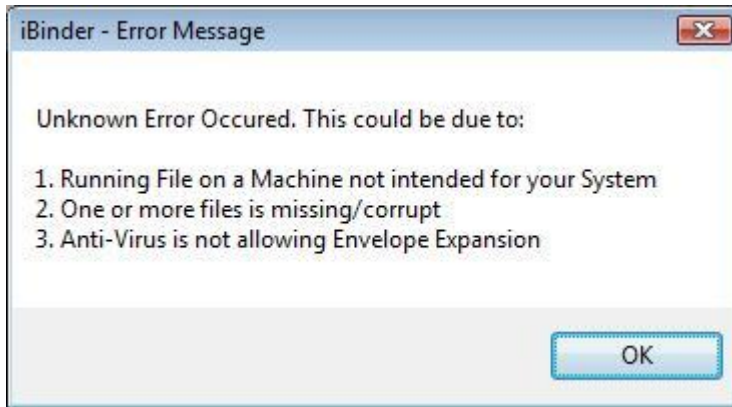
This file should be given to end user.



- Run the generated Enveloped EXE by just double clicking it, this EXE will work only with the particular Clave2 dongle having unique identity for providing security.



## 10. Error Messages while creating envelope



During envelope creation, you may encounter above error message. The resolution could be as follows:

1. Use `_x86` enveloper on Intel CPU based machines including 64 bit systems and `_all` enveloper on other CPU.
2. Make sure that files you have added are available and they are not corrupt
3. Disable anti-virus and try again
4. Copy the Program EXE in the enveloper folder [or vice versa] and try.
5. Before adding file, copy them in enveloper folder [keeping desired directory structure].



## 11. Error Messages while running enveloped EXE



If the user has not attached the vender supplied pen drive/flash drive while executing the EXE, this error messages may appear indicating failure of verification.

**Please contact us for purchasing SenseLock Dongle:**

**Innovative Matrix Softech Pvt. Ltd.**

Door No. 7005, 7th Floor, Emerald House  
1-7-264 Kalasiguda  
Near Paradise Circle  
Sarojini Devi Road  
Secunderabad - 500 003  
**India.**

Phone: 040 32518777

Mobile: 91.93485 18777

Email: [mail@isohelpline.com](mailto:mail@isohelpline.com)

Website: [www.isohelpline.com](http://www.isohelpline.com)