Installation Routine

Version 1·2

Designer **Craig Ward (100554.2072@Compuserve)**
Date **1/8/95**

Function **Dynamic Installation routine for MS-Windows**

****

My Terms

This product has been released into the public domain, **FREE!** I want no money in return, and you are welcome to make as many copies, and distribute them, as you like. **However,** I would appreciate some feedback, especially bugs or imporovements that you think that I should make.

History

This product was designed to facilitate installation in a MS-Windows environment. The push to start the project came when I needed my own installation utility for my own applicaitons (such as Paradox or Delphi app's). Since I can remember not too long ago trying to find a **FREE** installation routine on forums, and to no avail, I decided to design this one in such a way that it was dynamic and free for all to use and proliferate.

Files

The utility has several key files which **must be distributed** with the setup.exe file, and **must be located on the A: Drive**. These are:

Setup.Exe
Setup.Ini
Inst.Bat
Inst.Pif

These files have the following responsibilities:

Setup.Exe - This is of course the exe file that the user will run.
Setup.Ini -This file is where the developer's run-time preferences are stored (this is explained more later)
Inst.Bat -This file is called by Inst.Pif. This stores the file-copying command line.
Inst.Pif - This is the pif file called by Setup.Exe, which runs the batch file.

Important: *The application copies all .pif,.ini,.bat files from the setup disk to the windows directory when it opens. This means that when you have downloaded the programme, place it all on a floppy disk (it must be drive a!) and then run.*

Setup.ini

This file contains all of the run-time preferences. Each one is explained here:

[DRIVE] - the amount of required space for the programme (**in bytes!!!!**). Note: the installation utility checks the c: drive.

[DESKTOP] - title of the form, note to the form, and the bitmap used on the form. Note that you can't put a bitmap on the background form.

[TARGETDIR] - the directory you want created (setup.exe will prompt the user if it finds that the directory already exists)

[#DISKS] - the number of disks that your programme is on. Quite simply, setup.exe runs a small For.... loop which increments for the number of disks that you use.

[PG] - the title of the program group that you want your application to be stored in.

[PI] - first you must specify how many items that will be stored in the above mentioed program group (again, that For... loop), and then specify the title and command line for each. **Get this right, or who knows what may happen!** An improvement to the installation utility, is that you can now specify the icon for each program item (note: you must specify the index of the icon if the icons you are calling from are contained in an EXE or DLL, otherwise just zero if it is an ICO file).

Therefore, a line in the INI file must read (where n=the number of the program item):

[**n**]=**exe path <switches>,program title,ico path,ico index**

Thus a typical line in the INI file would look like this:

```
1=c:\myApp\myApp.exe,MyApps Title,c:\myApp\myApp.ico,0
```

[README] - the line for your readme file.

[EXECUTABLE] - the name of a file that you may wish to execute once the setup file has finised copying disks. For instance, you may want to unzip or expand the files that were on your disks. I have included an example (within which notice the use of the -o switch for pkunzip: this overwrites existing files). **Note: this file should really be copied over onto the user's hard-drive, which is what the setup programme will do if it is on the same disk at the start of the programme.**

Inst.Bat

I've found that a good command line is:

xcopy a: c:\<your target directory> /Y /E

This is because xcopy, with the /E switch, creates any subdirectories (and copies files from) that may be on your floppies, whilst the switch /Y will overrwrite existing files. Obviously, you are having to mention your target directory!

Inst.Pif

This has to point at inst.bat, and since I've configured it to run hidden, you're better off not tampering wth it.

Conclusions

I hope that it helps. I've started using it, and since I designed it to be so dynamic, I don't have to compile it each time that I want to install an application somwhere. **Have fun and don't forget to give me some feedback!**

"Remember, the force is with you..."