WinSetup 2.0

Powerful and Easy
Installation Program
For Microsoft Windows 3.1

Idea and documentation:
Benny Nissen

Programming:
Benny Nissen

BeastWare:
Svenstrupvejen 8
DK-4130 Viby Sj.
Danmark
Phone +45 4239-8410
E-mail : beast@ruc.dk

CONTENTS

Please take time to read the entire documentation before using this software. Printout the documentation to make it easier to read. (41 pages).

COPYRIGHT

BeastWare
Svenstrupvejen 8, DK-4130 Viby Sj. Danmark
Phone: +45 4239-8410
E-mail: beast@ruc.dk
µ §

BeastWare
Svenstrupvejen 8, DK-4130 Viby Sj. Danmark
Phone: +45 4239-8410
E-mail: beast@ruc.dk
µ §

Overview

Thank you for examining BeastWare's WinSetup product. Version 2.0 is an attempt to bring a more professional and stable Installation Package to the general public. WinSetup is a well-designed, easy-to-use installation utility that rivals or surpasses other commercial installation programs. The intention is to make it easy for the user to include a powerful installation program for the Windows environment. WinSetup can be used to install any product to the Windows environment, including other software, presentations, price lists, etc. All you need, besides WinSetup, is an ASCII / ANSI editor.

The benefits of this software can be summarized as the following.

Easy to understand and use for all users.
Easy configuration.
Easy to change installation language.
Fast and safe installation.
Small size of program file.
Capable of expanding compressed source files.
Capable of modifying Windows initialization files.
Capable of updating Windows registration database.
Capable of installing items in Windows shell (Program Manager).
Capable of installing from different disks or directories.
Desktop bitmap.
Shifting bitmaps during file copy.
Selective installation (installation options).
File version or date stamp checking.
File assembling from multiple volumes
Complete help system.
Attractive window layout.
Multitasking during installation.
Progress meter both in minimized and normal window status.
Space checking on destination drive.
Low price.

To keep the program small and easy to use, the following limitations have been necessary.

No modification of autoexec.bat and config.sys
Unable to restart Windows to replace system files.

BeastWare
Svenstrupvejen 8, DK-4130 Viby Sj. Danmark
Phone: +45 4239-8410
E-mail: beast@ruc.dk
µ §

It is easy to modify the program to include these listed limitations if you acquire the source code. See chapter "Registration form" for more information. The program is designed to install 95 % of the Window products on the marked today.

WinSetup is controlled completely by a single INI-style configuration file. Nearly all the parameters available in the file are optional. You can run a installation so simple it contains only a few brief lines or you can go for the giant multiple disk installation.

"Why do I need an automated installation program ?", you may ask.

First of all, not all people know how to install a program manually, neither in Program Manager, nor in the file system. Some programs, like control panel programs, libraries, drivers etc. may need a special installation procedure to function properly. And then, using an installation program is more convenient for the end user.

"But isn't it just more trouble to include an installation program ?" (size etc.).

No, not at all, the program file is small and it is very easy to configure the program to install your product. You may get your product out to the people who don't have a lot of computer experience. WinSetup is very inexpensive. Using WinSetup for your product installations will ensure that the process is safe and efficient, and that the installation meets Windows standards. WinSetup gives you what you need in an installation utility with very little effort.

The program has been written because of the lack of a powerful but easy and small installation program for Windows. During the development of another program we needed an installation program, and found that we didn't wanted to use Microsoft's setup toolkit, because of it's complicated configuration (takes a lot of time to understand). We didn't find another installation program we liked, so we had to write one ourselves. WinSetup is the result. It is developed with special care devoted to program stability. A crash during installation gives very bad impression of the product to install. WinSetup is the second version, the first version has never been released as a commercial product.

WinSetup is not free, it is released as Shareware. See chapter "Shareware conditions and registration" for more information. If a distributed product is released as a Public Domain or Freeware version, (no expenses, in any way, for the end user) WinSetup can be used free of change, and distributed with the released product (not documentation). The only obligation is to send a note either by mail or e-mail to the copyright holders with the name of the product to install, including your company and / or name plus

address.

You can always get the latest version of WinSetup, from one of the following internet locations.

ftp.cica.indiana.edu (USA)
nic.funet.fi or ftp.funet.fi (Europe)
mac-dev.ruc.dk (Denmark) Path: \winsetup\wsetupXX.zip

Thanks to all the users and beta testers who have helped me shape WinSetup into what it is today. Some of the really neat features of WinSetup has came from ideas thrown out to me by others. Special thanks to Claus Castle Andersen, Janusz Kuzminski and Iben Kofoed for assisting during development of WinSetup. WinSetup 2.0 have been tested by 40 people all over the world for a couple of months.

Benefits of registration
- you will be able to use WinSetup to install your commercial product
- you will get the latest version of WinSetup
- you will get help if you have problems with WinSetup.
- you will be on our mailing list
- you will get upgrades at a low price
- you will get free upgrades, to correct errors.
- you will support the Shareware idea.

Program presentation

The program flow is very simple, It is separated into seven steps / procedures which are performed in turn during a typical installation :

1 : Install initialization.
2 : Specifying options (end user task).
3 : File copy / expansion.
4 : Registration (merge .REG file into REG.DAT).
5 : Initialization file update.
6 : Shell communication (Program Manager).
7 : Program execution.

The seven steps performed by the program are reflected in the configuration file for the program. See Appendix A for more information.

First a window states that the program is initializing and a meter is indicating that

something is going on. During this step the program copy different files to the Windows temporary directory. This makes it possible to access them if the end user change disk during installation.

A main window makes it possible to specify destination directory and installation options. See chapter "Installation options". When the end user presses the button "Install", the program checks the destination drive for free space. If the space is insufficient the program displays an error message. If the space is sufficient the program tries to change to the directory specified in the "destination" edit control, if the directory doesn't exist, the end user is prompted with a message asking to create the directory. The end user can bypass space check on the destination drive and force installation if there is insufficient space on the destination drive. The installation will continue until there is no more room on the destination disk.

The copy procedure starts. The end user is presented with a window showing the progress of the copy procedure, which file is being copied, the destination, a short description, how many files are to be copied and the current number. It is possible for the end user to press the button "Cancel" to abort the copy procedure after a message for confirmation.
The program can display bitmaps during the copy step. See chapter "Bitmaps"
If the copy procedure finds a file in the destination directory with the same name as the file being copied, it can do a number of actions based on the settings in the configuration file. See Appendix A. The program can find version information in the two files (Version Resource) or compare creation dates. With this information the program can decide to overwrite. There are also settings to always or never overwrite and append to an existing file. The end user can also decide to overwrite, with a dialog box containing version information on both files or just creation dates. The end user has the option to overwrite, not overwrite or cancel the entire installation process. If the product to install is spread over different disks or directories the program can ask for another disk during the copy procedure. It is possible for the end user to bypass a file if WinSetup is unable to locate the file. The end user is warned that bypass of a file, may result in a useless installation. WinSetup can not overwrite Windows system files or other shared files currently in use. If the program encounters a system or shared file during file copy, it ask the end user to close down programs using this file and then continue. The end user can also ignore the file or abort the entire installation. If the copy procedure is running from and to a harddisk with a diskcache program, it is hard to notice a delay in normal operations of the Windows system. If the program is installing from a floppy disk, the Window system can be seriously delayed. If the copy process window is minimized the program displays the dynamic progress meter as the icon.

After all files have been copied, the program can merge a .REG file into the registration database for the Windows system. The end user has no direct influence on this step. See chapter "Registration database" for more information.

Then an update procedure starts updating initializations files in the Window system. The end user has no direct influence on this step.

Next, the program sends DDE messages to the Program Manager (or another shell) to create items in a group, the end user is prompted, and asked if there shall be created any items. If the Program Manager was not running (not the shell) when this procedure started, the installation program will try to close the Program Manager at the end of this step.

At last there is an opportunity to display a message to the end user and execute a program. Then the installation ends.

If everything goes well, a message is displayed stating this. If something goes wrong during the entire installation process, a message is displayed describing the problem, and the rest of the installation process is skipped. The end user is then returned to the first window.

During these seven steps the configuration file is decoded to adjust the programs behavior, the seven steps can be seen graphically as below.

Not all seven steps have to be performed, one or more steps can be omitted. See Appendix A for more information. The program can show a bitmap as a desktop during installation. See chapter "Bitmaps". It is also possible to use a help system during installation. This makes it possible for the end user to get a more complete explanation in different window and dialog boxes. See chapter "Help system".

WinSetup is using a special version of the DOS piece in a Window program. This version can, if executed under DOS, start Windows to continue execution of the installation program. If executed in a DOS box under Windows it will announce that it can not initialize the program in a DOS box, and show the key combinations to get back to Windows.

If you want to make a installation procedure fast, it is now time to take a look at the configuration samples. See Appendix C-E and the files SAMPLE1.INF to SAMPLE3.INF. If you need more information, read the rest of this documentation.

Help system

Included with WinSetup are 3 files that makes it possible to create a help system for the installation.

HELPDEFS.H	**Defines context strings for the help project.**
INSTALL.RTF	**The help text in RTF format.**
INSTALL.HPJ	**The help project file.**

Using a help system in the installation can be of great help to some end users, and is highly recommended. With a help system it is possible for the end user to press F1 or select the help button at almost any place in the installation process.

The help file (INSTALL.HLP) are copied during program initialization, from the source directory where the configuration file INSTALL.INF is, to the Windows temporary directory, so it is possible to access the help file if the end user changes disk.

The help text provided in the INSTALL.RTF file is only a framework, and is not complete. The help text has to be completed with the text you will provide for your installation help system. The file is only a framework because it is impossible to make a text that will be useful for all the possible configurations of the installation program.

It is important to keep the help file small, because this can make the installation program initialize slower. To keep the file small, use maximum compression in your help project and try to avoid using bitmaps in your help file. The program starts slower with a help file because the program copies it to Windows temporary directory during program initialization, this makes it possible to access it, if the end user change disk during the installation procedure.

Use HC31.EXE or HCP.EXE from Microsoft to compile the help project, these help compilers are often shipped with compilers for the Windows environment.

See the Windows documentation for more information on this subject.

Bitmaps

It is possible to use Window bitmaps (BMP or DIB) in two ways. As a screen desktop and as shifting pictures during file copy.

All bitmaps can be compressed by run length encoding (RLE), and can be 1, 4, 8 or 24 bit pictures of any size.
A bitmap window are displayed without a window frame. Make a frame in the bitmap if you want one.

Desktop bitmap

This bitmap can be loaded during start of the program, from the source directory where the configuration file INSTALL.INF is, the bitmap should have the name INSTALL.BMP and can not be compressed with COMPRESS.EXE. See chapter "Compressed files"
It is highly recommended to use RLE compression to minimize disk access, this will make the installation program initialize faster.
The bitmap can be stretched to fit in the desktop, so it is important to keep the bitmap dimensions correct, this means that the length divided by the height equals 1.1/3. A good dimension for the bitmap are 800x600x4 which gives a reasonable resolution on 1024x768 and 640x480 screens and keeps the bitmap file small. The larger a bitmap, the higher the resolution at all screen sizes. If you are using a bitmap without stretching, do not make it larger than 640x480 in order to view the bitmap on standard VGA screens.
Do not make the bitmap file larger than approximately 300 Kbytes to maintain a reasonable startup from a floppy disk.

File copy bitmaps

These bitmaps are copied during program initialization, from the source directory where the configuration file INSTALL.INF is, to the Windows temporary directory, so it is possible to access them if the end user changes disk.
These bitmaps are numbered from 1.BMP to n.BMP, this means that if you have 3 shifting bitmaps they have the names 1.BMP, 2.BMP and 3.BMP.
The bitmaps can be compressed with Compress. See chapter "Compressed files".
It is highly recommended to use RLE compression or compress the files with COMPRESS.EXE to minimize disk access for the bitmap, this will make the program start faster.
Do not make the bitmap files larger then approximately 64 Kbytes and do not use more then 5 bitmaps to maintain a reasonable startup from a floppy disk.
These bitmaps are not stretched, to fit a standard frame, so it is important not to have their length larger then 640 pixels, to make them fit in a standard VGA screen. The height must not exceed 200 pixels to make room for the file copy window. If the height exceeds 200 pixels on standard VGA screens, the file copy window will cover part of the bitmap.
A good dimension for these bitmaps are 300x200x4.
If both a desktop bitmap and file copy bitmaps are used, please note that the

desktop bitmap color palette are the active palette.

Installation options

With WinSetup you can use installation options to let the end user have more influence on the installation procedure. The use of installation options includes specifying the options to use, and to decide if the options shall be specified by check boxes or radio buttons. When using radio buttons it is only possible to select one of the options.

It is not necessary to use different options for the end user. Please don't use this possibility if it is not needed, it will make the configuration file more complicated. A good practice is to make the configuration file first without any options, and then enlarge the file with options afterwards. When you have decided which options to use, modify the configuration file to reflect these changes. See Appendix A and B for more information.

When the end user presses the button "Install", the program creates a string consisting of 0 and 1, that reflects the options selected by the end user.

Program option string => 11

Program option string => 10

The string is then compared with the strings specified for the different parts of the configuration file. A command to copy a file can have the following layout in the configuration file.

42={10}(107698,[r]\prog.exe,[t],Main Program,3)

The string enclosed by the characters '{' and '}' defines the options for which this command shall be performed, this means that only if there is a character "1" in the start of the string, created by the program when the end user press install, can this command be performed.
If there is no option string enclosed by "{}", the command is always performed.
The following table shows the possible results with 2 options.

Configuration file command option strings

		00	10	01	11
Program option strings	00	FALSE	FALSE	FALSE	FALSE
	10	FALSE	TRUE	FALSE	TRUE
	01	FALSE	FALSE	TRUE	TRUE
	11	FALSE	TRUE	TRUE	TRUE

TRUE => Perform the command.
FALSE => Don't perform the command.

Registration database

The registration database is a system-wide source of information about applications. This information is used to support the integration of applications with Windows File Manager and is used by applications that support object linking and embedding (OLE).
For most applications, the developer uses the Windows Registration Editor (REGEDIT.EXE) with the /v parameter to edit the registration database and produce a registration (.REG) file that contains readable text strings corresponding to database entries. After this procedure, it is possible to modify the text strings in the file, so they include the special combination of characters that will be expanded to the end user specified options.

Registration files will be modified by the installation program if they contain the following special combination of characters.

[t] or [T]	**Replaced with end user specified target directory**
[w] or [W]	**Replaced with Windows directory**
[s] or [S]	**Replaced with system directory**
[d] or [D]	**Replaced with Window drive (c:)**
[r] or [R]	**Replaced with last source directory**
[o] or [O]	**Replaced with program option string selected by end user**

BeastWare
Svenstrupvejen 8, DK-4130 Viby Sj. Danmark
Phone: +45 4239-8410
E-mail: beast@ruc.dk
µ §

The .REG file can be merged into the end user's registration database with WinSetup; the procedure is to copy the .REG file to the target directory during the file copy procedure and then specify this copied .REG file as the file to modify during the registration process. See Appendix A.
The registration process will read the file and modify it, by expanding the special characters in each line and then merge it into the registration database by the program "REGEDIT.EXE" or "REGLOAD.EXE". This mean that one of these programs have to be in the windows / system directory or in the DOS search path. To make sure that it is possible to execute one of these programs and merge the developed file into the registration database, REGLOAD.EXE have been supplied with WinSetup. Just copy this program to the system directory during the copy process, remember to use "Version overwrite". See Appendix A for more information.

The following example shows the format of a .REG file that would set up a standard application:

REGEDIT

HKEY_CLASSES_ROOT\Title = Standard Application
HKEY_CLASSES_ROOT\.XXX = Title
HKEY_CLASSES_ROOT\Title\shell\print\command = [t]\prog.exe /p %1
HKEY_CLASSES_ROOT\Title\shell\open\command = [t]\prog.exe %1
HKEY_CLASSES_ROOT\Title\protocol\StdFileEditing\verb\0 = Edit
HKEY_CLASSES_ROOT\Title\protocol\StdFileEditing\server = [t]\prog.exe

The first line of the file must be REGEDIT, as shown. Any subsequent lines that do not begin with HKEY_CLASSES_ROOT are currently treated as comments.

See the Windows documentation for more information on this subject

Compressed files

The Installation program can automatically detect some compressed files and expand them during the copy procedure. Expanded files have to be compressed with Microsoft Compress Version 2.00. The program itself (INSTALL.EXE) and the desktop bitmap (INSTALL.BMP) must never be compressed. All other files can be compressed. The program is using the Lempel-Ziv Encoding functions in the Windows API to expand the source files.

Compress (COMPRESS.EXE) creates compressed versions of one or more files. The resulting files are typically 25 to 45 percent smaller than the original files. Compress is normally included with compilers for the Windows environment.

Command-line syntax for Compress is as follows:

compress [/?][/r] source destination

Following are command-line options and parameters for Compress:

/? **Displays information about how to use Compress.**
/r **Specifies that compressed files should be renamed (required with this installation program).**
source **Specifies the source filename. The name can include a drive letter, a directory path, or both; and it can contain wild cards.**
destination **Specifies the destination. This parameter can consist of a directory (with optional drive letter), a filename, or any combination of the two.**

You have to use the /r option if you intend to use the compressed file with the installation program. The only exceptions is the configuration file (INSTALL.INF), the help file (INSTAL.HLP) and the file copy bitmaps (1.BMP, 2.BMP, etc.) which must have there original file names.
The /r option instructs COMPRESS.EXE to automatically generate the name of the compressed file and more importantly to store the original name in the compressed file.

For example: compress /r *.exe c:\temp

In this case, all files with extension EXE in the current directory will be compressed and the compressed files will be put into c:\temp.
If you don't use /r option, WinSetup will be unable to identify the original file name.

NB: Remember to use the /r option for files to install.
You do not have to compress any files.

Technical comments

This chapter contains technical information that did not fit in elsewhere, this does not mean that the information is not important; it is.

The source code for the program is in ANSI C compiled by a C++ compiler (Borland App. framework 3.1), under the small memory model. The program is using 16 bit Windows API.
The code is written from a scratch, without use of any toolkits.

If you need to change the program code or configuration file and do not have the time or ability to do so, the developers of WinSetup will be able to assist you for a small fee. Please contact the copyright holders.

WinSetup operation

The program loads into memory using the non discardable option, so the program will stay in memory as long as it is active. When the program starts, it copy (expand) the configuration file to the temporary directory specified by the environment variable TEMP (SET TEMP=). During execution the program will read the configuration file from the temporary directory.

If the WinSetup program is initialized without commandline options, the configuration file has to be in the same location as the WinSetup program. It is also possible to specify the location of the configuration file as a command line option to the program.
All other support files (help file and installation bitmaps) have to be in the same location as the configuration file.

The program has been designed to generate understandable error messages in case of error conditions. The program tests the success of almost all calls to Windows and DOS.

All text strings in the program are in the resource of the program file, this makes it easy to change installation language, by using a resource editor. All the resources can be changed by registered developers. Do not change the special control characters in the text strings, this mean words beginning with '%'. Adding words beginning with '%' to the strings can make the program and the system crash. Only change plain text.

At the end of program execution there is a possibility to run another program, this can be used to update DOS system files (AUTOEXEC.BAT and CONFIG.SYS), restart Windows to replace system files, execute the installed program or just view a readme file.

There is no special procedure to let the end user enter user name and registration number.
There are a number of solutions to this.

1 : The installed program can ask for it at the first execution.
2 : Run a program at the end of installation to ask the end user for this information
3 : Get the source code and modify the program to do this task.

WinSetup can create many subdirectories in one command, if the subdirectories are in the same path. For example if you need to create c:\temp\temp2 and neither temp nor temp2 exists, WinSetup can create both directories in one command call.

It is always a good practice to include a removal program / procedure with your product, especially if you place files in the windows or system directory and modify the registration database or system initialization files. WinSetup is incapable of removing your installed program. You can make a removal procedure by creating an initialization file with the install information during installation (target directory, files copied etc). You can then make a program to read this initialization file to remove the product.

There may be files which are too large to fit onto distribution disks. This is not a problem, if you can split the files up into chunks small enough to fit. WinSetup provides a way to re-assemble the chunks back into one big file. See Appendix A for more information. For splitting files, you are on your own. So long as the files are cleanly split, WinSetup will simply append them together. If you split them using some tools that adds re-assembly information to the chunks, WinSetup can only re-assemble them with the information still embedded. If your file was compressed before splitting, expansion will fail.

WinSetup limitations

WinSetup require MS Windows 3.1 running under DOS (3.3 or later) file system. The program has been developed for MS Windows 3.1 (how many are actually running 3.0 in 1994 ?) and will probably not execute under MS windows 3.0. This choice was made to avoid the use of a number of DLL's and a more complicated code.

The hard / software demands for the program are minor, so if Windows 3.1 is already running, this program should also be able to run. The only consideration is use of large bitmaps that can give rise to memory problems, on low memory computers. The execution continues even if the program cannot display bitmaps because of low memory. If you use bitmaps there should also be at least a VGA screen adapter in the computer.

The free disk space test, performed before the file copy procedure, only check disk

space on one drive. With the installation program it is possible to copy files to different drives. This means that only installation files that have the same drive destination as the specified destination directory will have their file size added together and the space requirement checked on the destination drive.
This can be a problem if installing files to the windows or system directory, and the destination drive is different from the windows or system drive. The space for the file installed in the windows or system directory is not checked. To help on this problem, please specify the default destination drive to the same as the Windows drive, This will not solve the problem but, if the end user does not change the drive, the space check will be for all the files including the file to be installed in the windows or system directory.

The program is not capable of doing things like !

1 : *If this file exist*
2 : *copy the file*
3 : *If the file don't exist*
4 : *copy another file*

This kind of intelligent behavior would have made the configuration file a lot more complicated. To achieve this behavior, get the source code for the program and code this behavior.

WinSetup should accept long file and directory names (longer then 8+3), but it has not been tested. The program only accepts a path smaller then 80 characters.

There have been no special considerations for networking during the development of WinSetup, but it is possible to install over a network if it is possible to use normal copy commands over the network and Windows is logged on to the network.

The DDE commands to the Windows shell are sent to the shell with almost no modifications and checks for correct syntax. This means that any error in these commands doesn't give an error message and the command is presumably not performed. The correctness of this configuration file section is very important.

The program assumes that there is enough place for the INSTALL.INF and possibly INSTALL.HLP and bitmap files in the Windows temporary directory at initialization. If the program cannot find a place for INSTALL.INF in this directory, the program will have to quit after a message. The other support files allow the installation to continue in case there is no room for them in the temporary directory

Upcoming implementations

WinSetup is under constant development. At this point, the following things are planned :

1 : Implement 3D controls by means of Microsoft's CTRL3D library for Windows. (almost finished)
2 : More sophisticated DDE communication with the Windows shell. (place software in existing groups)
3 : A small version specific for BBS use, without bitmap and registration support. (size 25 Kbytes)
4 : Utility program to generate configurations files from a directory structure and to check options in the configuration file.

These implementations are scheduled to be released before end of 1994, not necessarily at the same time, and are free of change for registered users.

Currently the following possibilities are investigated.

1 : Improved support of network.
2 : Replace system files.
3 : Modify DOS configuration files.
4 : Use of a DLL to display a dialog box for option selection.
5 : Procedures to check for prior versions of a installed product and upgrade it.
6 : Compression / expansion with code from the Info-zip group.
7 : Removal procedure / program for installation.
8 : AND / OR logic when modifying system.

If there is anything else missing in this product, please let us now, so we can decide if it is a feature we want to implement in future versions.

Support

Support is only available trough e-mail / mail, we will try to answer all questions and solve any problem with the software, but can not guarantee any solution. Support is only available for registered users. Please make sure that the question is not answered in this documentation before sending any mail.

When reporting a problem, at least supply the following information.

BeastWare
Svenstrupvejen 8, DK-4130 Viby Sj. Danmark
Phone: +45 4239-8410
E-mail: beast@ruc.dk
µ §

1 : The hard and software environment (include AUTOEXEC.BAT, CONFIG.SYS, WIN.INI, SYSTEM.INI)
2 : The version of WinSetup software (See about in system menu).
3 : The INSTALL.INF file.

Please direct all questions to these addresses. Try e-mail first if possible.

E-mail :
beast@ruc.dk (Internet)

Mail :
BeastWare.
Benny Nissen,
Svenstrupvejen 8,
4130 Viby Sj.,
Denmark.
Mrk. WinSetup Support.

Known errors

At this point there are not reported any errors in the software, and there is no knowledge of any other problems. If you in any way discover problems with this software, please report it, it is the only way that we can improve this product, to run without problems in all Windows environments.

WinSetup has been tested on the following Window platforms (also many non US / UK versions).

MS Windows 3.1 and 3.11
MS Windows 3.11 with WIN32s API.
MS Windows for Workgroups 3.11
IBM OS/2 2.1
MS Windows NT 3.1 and 3.5 and some beta versions
Beta versions of Chicago.
Apple PowerMac running Windows emulation (SoftWindows).

The program has been tested with the following programs as the Windows shell.

MS Program Manager 3.1 and 3.11
HP Dashboard 1.0.

Norton Desktop for Windows 2.0.
PC Tools for Windows 1.0 and 2.0

Shareware conditions and registration

You should carefully read the following terms and conditions before using this software. Use of this software indicates your acceptance of these terms and conditions. If you don't agree with them, don't use this software.

Shareware Version

You are hereby allowed to: use the Shareware Version of WinSetup make as many copies of the Shareware version of this software and documentation as you wish; give exact copies of the original Shareware version to anyone; and distribute the Shareware version of the software and documentation in its unmodified form via electronic means. There is no charge for any of the above.

WinSetup may be used personally or in a business, to evaluate the software in a 31 days trial period.

If a distributed product is released as a Public Domain or Freeware version, (no expenses, in any way, for the end user) WinSetup can be used free of change, and distributed with the released product (not documentation). The only obligation is to send a note either by mail or e-mail to the copyright holders with the name of the product to install, including your company and / or name plus address.

All commercial use of this Shareware Version is specifically prohibited without prior written permission. You may not modify or dis-assemble WinSetup , nor distribute any modified or dis-assembled versions of WinSetup

This version of WinSetup may be freely copied and distributed in unmodified form but may not be sold. You are specifically prohibited from charging, or requesting donations for more than the cost of shipping and handling, for any such copies, however made; and from distributing the software and / or documentation with other commercial products; and from using the software personally or internally in a business over 31 days without prior written permission.

BeastWare
Svenstrupvejen 8, DK-4130 Viby Sj. Danmark
Phone: +45 4239-8410
E-mail: beast@ruc.dk
µ §

Registration

Basic registration allows the company or the person stated in the Registration Name section of the registration form to distribute the software (not documentation) with their own commercial products, or use it internally in a business. It also allow changes to the program resource to meet special requirements

Full registration include Basic registration but allow the company or the person stated in the Registration Name section of the registration form to use the source code in their own commercial products and / or make changes to the source code to meet special requirements. The original source code is still copyrighted by the copyright holders for WinSetup.

Registered users upgrade of WinSetup cost for major changes one third of a complete registration, minor changes are free (except shipping).

The author reserves the right to change pricing at any time.

It is not allowed to use WinSetup, the code or part of the code in software that is released as a commercial product to compete with WinSetup (The copyright holders can have financial loss from this release). This means that you may not use the code or part of the code to make an installation program that you wish to sell as a commercial product. You may not dis-assemble WinSetup , nor distribute any modified or dis-assembled versions of WinSetup.

Disclaimer of Warranty

This software and the accompanying files are distributed "as is" and without warranties as to performance of merchant liability or any other warranties whether expressed or implied. Because of the various hardware and software environments into which WinSetup may be put, no warranty of fitness for a particular purpose is offered. In no event shall the author be liable for any damages whatsoever arising out of the use of or inability to use this product, even if the author has been advised of the possibility of such damages. The entire risk arising out of the use or performance of this software and documentation remains with you. No claims are made regarding the accuracy of this program.

Good data processing procedure dictates that any program be thoroughly tested with non-critical data before relying on it. The user must assume the entire risk of using the program.

BeastWare
Svenstrupvejen 8, DK-4130 Viby Sj. Danmark
Phone: +45 4239-8410
E-mail: beast@ruc.dk
µ §

BeastWare
Svenstrupvejen 8, DK-4130 Viby Sj. Danmark
Phone: +45 4239-8410
E-mail: beast@ruc.dk
µ §

Registration form

WinSetup 2.0

Basic registration (includes latest version):	$ 80,00 or DKr. 500,00	( )
Full registration with source code (Borland C 3.1):	$ 320,00 orDKr.2000,00	( )
Upgrade from Basic to Full registration:	$240,00 orDKr.1500,00	( )
Overseas air-mail shipping (other than Europe):	ADD:$ 8,00 or DKr. 50,00	[ ]
Europe mail shipping:	ADD:$ 4,00 or DKr. 25,00	[ ]
Internet e-mail shipping (free of charge):		[ ]

Currency: US dollars [ ] DK kroner [ ] Sub-total: _____

Enclosed is a international check or money order for total of:
_____

Registration Name
Company / Person (required):
_____

Name:
_____

Title:
_____

Address:
_____

_____

City, State, ZIP:
_____

Country :
_____

Phone (Fax) :
_____

BeastWare
Svenstrupvejen 8, DK-4130 Viby Sj. Danmark
Phone: +45 4239-8410
E-mail: beast@ruc.dk
µ §

E-mail :

_____

I have read and agree to abide by the license agreement (Shareware conditions and registration)

Signature

_____

Expect 4 weeks delivery time. Diskette size. 3 1/2" DD [ ] 5 1/4" HD [ ]

How did you learn about WinSetup?
[ ] Searched through the index of a ftp site [ ] Found it on CompuServe
[ ] Recommended by a friend/colleague [ ] Found it on a BBS
[ ] Found it on a CD or catalogue
[ ] Others. Please state:

_____

Please make sure the check is valid in Denmark and payable to:
Benny Nissen, Svenstrupvejen 8, 4130 Viby Sj. Denmark.

BeastWare
Svenstrupvejen 8, DK-4130 Viby Sj. Danmark
Phone: +45 4239-8410
E-mail: beast@ruc.dk
µ §

Appendix A : Configuration file format

The configuration file for WinSetup is a normal Windows initialization (INI) file, which you may already have experience with.

The configuration file INSTALL.INF is a special ASCII file that contains entry-value pairs representing run-time options for the installation program. The configuration file can be compressed, the program will expand it before use. See chapter "Compressed files" for more information.

To make comments in the file add a semicolon (;) in front of each comment line.

An initialization file is separated into sections. Sections in the initialization file have the following form:

```
[section]
entry=string value
    --
    --
   etc.
```

The configuration file is separated into seven sections, that reflect the seven steps in the program. See chapter "Program presentation".

Basic:	**General entries used when initializing program.**
MainWindow:	**Options to select and anything seen in the first window.**
FileCopy:	**All about file copying.**
RegistrationFiles:	**Files to update registration database with.**
IniModifications:	**How to modify initialization files in the Windows environment.**
PMOptions:	**DDE commands to Program Manager.**
ExecuteAfterInstall:	**Program to execute after installation, and its behavior.**

The following characters have a specific use. They can not be used as normal text characters. These characters represent control codes in the configuration file.

{ and }	**Define start and end of configuration file option strings.**
(and)	**Define start and end of command string.**
[and]	**Define start and end of a special replace character, or define**

sections.

,	**Defines separation of parameters in a command string.**
=	**The first '=' in a line defines separation of entry and string value.**
"	**Reserved for future use.**
TAB	**Can not be used in a initialization file (Indicate end of line).**

All characters accept '[', ']' and TAB can be used in entry RunText and WindowText.

Strings will be modified by the installation program if they contain the following special combination of characters.

[t] or [T]	**Replaced with end user specified target directory**
[w] or [W]	**Replaced with Windows directory**
[s] or [S]	**Replaced with system directory**
[d] or [D]	**Replaced with Windows drive (c:)**
[r] or [R]	**Replaced with last source (read) directory**
[o] or [O]	**Replaced with program option string selected by end user**
[n] or [N]	**Replaced with new line, only in entry RunText and WindowText**
[q] or [Q]	**Replaced with tabulation, only in entry RunText and WindowText**

No string can be more then 400 characters long, if longer an error message is generated.
Please remember that strings can be made longer by special control codes.

The program does not use pixel units to make windows and buttons larger (see Basic section and MainWindow section) instead the program is converting the device independent value specified to a device dependent value based on settings in the current Windows environment. This guarantees that the text shown in windows is always shown completely. If the value was pixel units, then you suddenly would not be able to see some part of the text after you shifted from small to large fonts in your Windows environment.

The rest of this chapter describes each entry in detail, the parameters between '[' and ']' define parameters that are an opportunity and not required.

Basic section

- ***ShowDesktop=(Number[,Text,Typeface,Size[,Bold[,Italic[,Alignment[,Color]]]]])***

Use a bitmap window as installation background, this requires that INSTALL.BMP is at the same location as INSTALL.INF during install initialization. Following are the parameters for this command:

Number : Type of desktop bitmap.
0 : Don't use a bitmap desktop. Default value.
1 : Uses the AND operator to combine eliminated lines with the remaining lines when removing information from the picture. This mode preserves black pixels at the expense of colored or white pixels. This mode creates a full screen bitmap desktop.
2 : Uses the OR operator to combine eliminated lines with the remaining lines when removing information from the picture. This mode preserves colored or white pixels at the expense of black pixels. This mode creates a full screen bitmap desktop.
3 : Deletes the eliminated lines. Information in the eliminated lines is not preserved when removing information from the picture. This mode is useful for photo realistic bitmaps. This mode creates a full screen bitmap desktop.
4 : This mode creates a screen desktop with the size of the bitmap. The bitmap is never stretched to fit a specific screen size. Shows the bitmap in its normal state.

Text : Text string to display on desktop bitmap. Text is automatically wrapped to next line if longer then desktop width. Max. 400 characters.

Typeface : Text string specifying the font to show text with. Max. 400 characters.

Size : Numeric value specifying the size of the font to show text with. Max. unsigned 16 bit value

Bold : 0 = Non bold font style. Default value.
1 = Bold font style.

Italic : 0 = Non italic font style. Default value.
1 = Italic font style.

Alignment : 0 = Left aligned desktop text. Default value.
1 = Center aligned desktop text.

Color : 0 = White desktop text. Default value.
1 = Black desktop text.

•*ShowHelp=Number*

Show help buttons in windows, this requires that INSTALL.HLP is at the same location as INSTALL.INF during initialization. Following are the parameters for this command:

Number : 0 = No help. Default value.
1 = Show help.

•*DebugMode=Number*

Perform a fake installation without modifying the target drive (Nothing is altered). Windows will show more information about the installation process. Only useful for debugging purposes. Following are the parameters for this command:

Number : 0 = Normal mode. Default value.
1 = In Debug mode.

•*ExtraButtonWidth=Number*

Make buttons wider than normal. (make room for more text) Following are the parameters for this command:

Number : 0 to n units wider. Not pixel units. Max. unsigned 16 bit value.

MainWindow section

•*WindowTitleText=Text*

Window title bar text used in the main window. Following are the parameters for this command:

Text : A text string to show in title bar of main window. Max. 64 characters.

•*WindowText=Text*

Text to show / display in the main window. Following are the parameters for this command:

Text : A text string to show at the top of the main window. Max. 400

characters.

•*ExtraWindowWidth=Number*

Make main window wider than normal. (make room for more text) Following are the parameters for this command:

Number : 0 to n units wider. Not pixel units. Max. unsigned 16 bit value.

•*ExtraWindowHight=Number*

Make main window higher than normal. (make room for more text) Following are the parameters for this command:

Number : 0 to n units higher. Not pixel units. Max. unsigned 16 bit value.

•*ShowSourceDir=Number*

Show the source directory edit control. Following are the parameters for this command:

Number : 0 = Don't show control. Default value.
1 = Show control.

•*HideBrowseButton=Number*

Hide the directory browse buttons. Following are the parameters for this command:

Number : 0 = Show button. Default value.
1 = Hide button.

•*StartDestDir=DirectoryString*

The default directory at startup, shown in the destination edit control. Following are the parameters for this command:

DirectoryString : A directory string following DOS convention (c:\dir). Max. 80 characters.

•*ShowAsRadio=Number*

Type of option boxes (radio or check boxes) Following are the parameters for this command:

Number : 0 = Show as check boxes. Default value.
1 = Show as radio boxes.

•*Number=(OptionName[,Checked])*

Variable number of options commands (there can be many of these types of entries). Shown in the main window, to let the end user select options. The entire command must never exceed 400 characters. Following are the parameters for this command:

Number : 1 to 20 (Options shown in numeric order)

OptionName : Name of option. shown in main window.

Checked : Check status of option on startup.
0 = Unchecked. Default value.
1 = Checked.

FileCopy section

•*TotalPictures=Number*

Total number of bitmaps to change between during file copy step. Require that this number of bitmaps is at the same location as INSTALL.INF. ([number].BMP) Following are the parameters for this command:

Number : 0 to n. Max. unsigned 16 bit value (Have to be lower or equal to the actual number of bitmap files).

•*MinOptions={OptionString}*

The minimum options for the installation. If this entry exist, there will be 2 buttons created in the main window to select minimum and normal options. Normal options are the options selected at runtime. You set normal options when specifying options for the installation in section MainWindow. (You specify if an options shall be on or not when

the main windows is shown)

OptionString : Option string consisting of 0 or 1. Used to set options in main window. Read from left to right. Max. 20 characters.
0 = Set option off.
1 = Set option on.

•*ExtraBytes=Number*

Extra bytes free to allow installation, in addition to what the installed files occupy on disk. (This is required if there has to be extra file expansion etc. after installation) Following are the parameters for this command:

Number : 0 to n. Max. signed 32 bit value.

•*Number={OptionString}*
(FileSize,FileName,TargetDir[,Description[,OverwriteAction[,DiskName[,Attributes]]]])

Variable number of file copy / expand command. (there can be many of these types of entries) The entire command must never exceed 400 characters. Following are the parameters for this command:

Number : 1 to n. Max. unsigned 16 bit value (Commands performed in numeric order).

OptionString : Option string consisting of 0 or 1. Compared with option string selected by the end user in main window, to decide if entire copy command should be performed. A match of 1 in same position from the 2 option strings will perform the copy command. Can be omitted, then the copy command is always performed. Max. 20 characters.

FileSize : Expanded file size for the file to copy. Used to calculate space needed for installation and to adjust progress bar in file copy window. Max. signed 32 bit value.

FileName : File Name with full path and extension to copy / expand to destination directory.

TargetDir : Directory to copy / expand file to. If the directory does not exist, it is

created automatically. Max. 66 characters.

Description : Short text description of file. Shown in file copy window. Max. 64 characters.

OverwriteAction : Action to take if destination file already exist in destination directory.
0 Prompt end user to decide if the file shall be overwritten. Default action. (Dialog with version information or creation dates from both files).
1 Always overwrite destination file.
2 Never overwrite destination file (Skip current installation file).
3 Compare version information in both files. Overwrite if able to find version information in both files, destination version is equal or lower than source version and language / codepage is the same. If unable to find version information for both files the OverwriteAction is 0 (prompt end user).
4 Compare file dates for the files. Overwrite if destination creation date is equal or lower then source creation date. Compare year, month, day, time, min. but not sec.
5 Append source file to end of destination file. For this action to succeed the destination file must exist.

DiskName : Name of disk with file. DiskName is compared from one file copy command to the next. If any changes the program ask for the disk with name DiskName. The first file copy command (Number = 1) specifies the name of the disk with the installation program (current disk). Max. 64 characters.

Attributes : A string specifying the attributes to set in destination file. WARNING: The installation program is not capable of overwriting a file with one of these attributes. Max. 3 characters.
R or r : Set file attribute read-only.
H or H : Set file attribute hidden.
S or s : Set file attribute system.

RegistrationFiles section

•*Number={OptionString}(FileName)*

Variable number of file names to modify with installation options and merge into registration database for the Windows environment. (there can be many of these types of entries) The entire command must never exceed 400 characters. Following are the

parameters for this command:

Number : 1 to n. Max. unsigned 16 bit value. (Registration performed in numeric order)

OptionString : Option string consisting of 0 or 1. Compared with option string selected by the end user in main window, to decide if this registration should be performed. A match of 1 in same position from the 2 option strings will perform the registration. Can be omitted, then the registration is always performed. Max. 20 characters.

FileName : Specify the path and filename with extension to the .REG file to modify with installation options and merge into registration database for the Windows environment. Max. 80 characters.

IniModifications section

•*Number={OptionString}(INIFile,Section[,Entry[,StringValue]])*

Variable number of initialization file modifications (there can be many of these types of entries). The entire command must never exceed 400 characters. Following are the parameters for this command:

Number : 1 to n. Max. unsigned 16 bit value. (Modifications performed in numeric order)

OptionString : Option string consisting of 0 or 1. Compared with option string selected by the end user in main window, to decide if this modification should be performed. A match of 1 in same position from the 2 option strings will perform the modification. Can be omitted, then the modification is always performed. Max. 20 characters.

INIFile : Specify the path and filename with extension of the initialization file to modify. If INIFile does not contain a fully qualified path and filename for the file, the installation program searches the Windows directory for the file. If the file does not exist, the file is created in the Windows directory. If INIFile contains a fully qualified path and filename and the file does not exist, the file is created. The specified directory must already exist. Max. 80 characters.

Section : If the section does not exist, it is created. The name of the section is case-insensitive; the string may be any combination of uppercase and lowercase letters.

Entry : If the entry does not exist in the specified section, it is created. If this parameter or entry is omitted, the entire section, specified by the Section parameter, including all entries within the section, is deleted.

StringValue : If this string is omitted the entry specified by the Entry parameter is deleted.

PMOptions section

•*PMGroup=(GroupName[,GroupPath])*

The PMGroup entry instructs Program Manager to create a new group or activate an existing group window. This is only done if there are any Program Manager commands. Following are the parameters for this command:

GroupName Identifies the group to be created. If a group already exists with the name specified by GroupName, Program Manager activates the group window.

GroupPath Specifies the path of the group file. If your application does not supply this parameter, Windows uses a default filename for the group in the Windows directory. Max. 80 characters.

•*Number={OptionString}*
(CmdLine[,Name[,IconPath[,IconIndex[,xPos,yPos[,DefDir[,HotKey,
[,Minimize]]]]]])

Variable number of Program Manager commands (there can be many of these types of entries). An entry instructs Program Manager to add an icon to an existing group. Following are the parameters for this command:

Number : 1 to n. Max. unsigned 16 bit value. (PM commands performed in numeric order)

OptionString : Option string consisting of 0 or 1. Compared with option string selected by the end user in main window, to decide if this PM command should be performed. A match of 1 in same position from the 2 option strings will perform the command. Can be omitted, then the command is always performed. Max. 20 characters

CmdLine: Specifies full command line required to execute the application. At minimum, this string is the name of the executable file for the

application. It can also include full path of the application and any parameters required by the application. Max. 80 characters.

Name: Specifies the title that is displayed below the icon in the group window.

IconPath: Identifies the filename for the icon to be displayed in the group window. This file can be either a Windows executable file or an icon file. If the IconPath parameter is not specified, Program Manager uses the first icon in the file specified by the CmdLine parameter if that file is an executable file. If CmdLine specifies an associated file, Program Manager uses the first icon of the associated executable file. The association is taken from the registration database. If CmdLine specifies neither an executable file nor an associated executable file, Program Manager uses a default icon. Max. 80 characters.

IconIndex: Specifies the index of the icon in the file identified by the IconPath parameter. The IconIndex parameter is a number. PROGMAN.EXE contains five built-in icons that can be used for non-Windows programs. IconIndex starts with 0.

xPos: Specifies the horizontal position of the icon in the group window. This parameter is a number. Use both the xPos and yPos parameters to specify the position of the icon. If you do not specify the position, Program Manager places the icon in the next available space.

yPos: Specifies the vertical position of the icon in the group window. This parameter is a number. Use both the xPos and yPos parameters to specify the position of the icon. If you do not specify the position, Program Manager places the icon in the next available space.

DefDir: Specifies the name of the default (or working) directory. If this option is not specified, Program Manager uses the path to the program as the default directory. Max. 66 characters.

HotKey: Identifies a hot (or shortcut) key that is specified by the user.

Minimize: Specifies whether an application window should be minimized when it is first displayed. 1 = True, 0 = False.

ExecuteAfterInstall section

- *OptionsForExec={OptionString}*

The options for which to execute the program and display RunText in a dialog box. If this option string is decoded falsely at runtime, the other entries in this section have no meaning. Following are the parameters for this command:

OptionString : Option string consisting of 0 or 1. Compared with option string selected by end user in main window, to decide if this section has meaning. A match of 1 in same position from the 2 option strings will perform this section. Can be omitted, then this section is always performed. Max. 20 characters.

•*HideNormalComplete=Number*

Controls whether to show the normal installation complete message box. Only after a successful installation. Following are the parameters for this command:

Number : 0 = Show dialog box. Default value.
1 = Don't show dialog box.

•*YNButtons=Number*

Controls whether to display YES/NO buttons or OK button in the run dialog box. Only after a successful installation. Following are the parameters for this command:

Number : 0 = Show OK button. Default value.
1 = Show YES/NO buttons..

•*RunText=Text*

Text to display in the run dialog box. Only after a successful installation. If no text is specified, no run dialog box is displayed. Following are the parameters for this command:

Text : A text string shown in the run dialog box. Max. 400 characters.

•*RunProgram=CommandLine*

A command line to execute after successful installation and the end user select YES or OK in run dialog box or no RunText is specified. Following are the parameters for this

command:

CommandLine : A DOS command with full path and filename with extension. Can include any number of parameters. Max. 80 characters.

•*ShowDesktopDuringExec=Number*

Tracks execution of program to show desktop bitmap during execution. After execution the desktop bitmap is destroyed. If there is no desktop bitmap, this entry has no meaning. Only functional for true MS Window compatible programs (no DOS programs), that pop up a window. Following are the parameters for this command:

Number : 0 = Don't show desktop bitmap. Default value.
1 = Show desktop bitmap.

Appendix B : Basic steps to create an installation program

Following these guidelines to create an installation program will ensure that the process is performed quickly and without problems.

Basic components of WinSetup

The WinSetup software includes the following basic components:
The main program, INSTALL.EXE, which copies the configuration file (INSTALL.INF) and other supporting files to a temporary directory on the end user's hard disk and then launches the configuration file. When the installation is complete, INSTALL.EXE removes the temporary files.
The configuration file INSTALL.INF that is used when WinSetup is installed to your harddisk. Can be used as a starting point for creating your own configuration file.
Sample configuration files (SAMPLE1.INF, SAMPLE2.INF, and SAMPLE3.INF), which you can use as a starting point for creating your own configuration file.
Sample registration file SAMPLE.REG which you can use as a starting point for creating your own registration file.
A registration loader from Microsoft REGLOAD.EXE to make sure that it is possible to update the registration database in the Windows environment.
A help file framework (INSTALL.HLP, INSTALL.RTF, INSTALL.HPJ and HELPDEFS.H), which you can use as a starting point for a help system.
Bitmaps (INSTALL.BMP, 1.BMP and 2.BMP) that are used when the software is

installed to your harddisk. Can be used freely.

Use these components as described in the next section to create your own Installation program. Using WinSetup for your product installations will ensure that the process is safe and efficient, and that the installation meets Windows programming standards.

Steps for creating an configuration file

Use the following procedure to create a configuration file for your product. Not all procedures are necessarily needed to install your product. The remainder of this chapter discusses each step in detail.

To create an WinSetup configuration file:

1 : Identify the files that will be installed for your product.
2 : Design the directory structure for those files.
3 : Identify the values / strings needed to update the registration file and create the file.
4 : Identify the changes needed to initialization files in the Window environment.
5 : Modify one sample configuration file so that it will install your product.
6 : Test the configuration file.
7 : Design the options you will need for the installation.
8 : Modify the help framework files to create your own help system.
9 : Create installation bitmaps used during installation.
10 : Compress source files.
11 : Modify the configuration file so it include options, bitmaps and help system.
12 : Test the configuration file

Step 1: Identify the files that will be installed for your product
Before you start modifying sample files, it's a good idea to make a list of the files you will need to install. For each file that you will install, answer the following questions:
1 : Is this file unique for your product, is it a shared resource, or is it a system resource? For example, a shared resource could be a language dictionary used by more than one product of your company. A system file that is currently in use cannot be copied onto the end user's system.
2 : Can the end user decide whether to install this file? For example, is the installation of tutorial files optional? If so, you'll need to design options that asks the end user to choose the files to install.
3 : Does the file contain version information.

4 : If an older version of the file already exists, should you overwrite it.
5: Note the file size for the expanded file.
6 : Note a small description of the file.
Beside each filename on your list, make notations indicating the answers to these questions. These notations will help you later when you design options and set the properties for each file in the configuration file.

Step 2: Design the directory structure for the installable files
Take the time now to sketch out the directory structure for your product by organizing the installable files into categories that make sense. For example, you might put all computer-based training files in one sub directory and documentation files in another. You may need to place some files in the Window directory. On the other hand, one directory (with no sub directories) may suffice for a product that has only a few files.

Step 3: Identify strings needed to update registration file.
If you need to update the registration database this is the right time to make the registration file. Remember to take the directory structure into account. Can the end user decide whether to update the registration database ? For example, if the end user does not install some files, then the update is obsolete. See chapter "Registration database" for more information.

Step 4: Identify changes to initialization files.
If you need to modify any initialization files, please identify the needed changes. Can the end user decide whether to modify a file? For example, if the end user does not install some files, then the modifications is obsolete.

Step 5: Modify one sample configuration file to create your own configuration file.
WinSetup contains three sample configuration files (SAMPLE1.INF, SAMPLE2.INF, and SAMPLE3.INF). The sample .INF files contain calls to Installation procedures that you would typically use to install your product. The sample .INF files describe the installation for WinSetup. Each sample installs a slightly different type of product: One installs a set of files with no special requirements; one uses more complicated options and installs several sets of files based on the end user's choices; and one installs files using all the possibilities in the configuration file. Use these samples as a starting point for creating your own configuration file.
For more information about each of these files, see Appendix A to D.

Step 6: Test your installation script.
Once you've created your own configuration file, test it thoroughly. Copy the install and source files to one directory on your harddisk. Test the file under a

variety of situations. Check the results by verifying that the files were copied to their appropriate directories and that files were updated correctly.

Step 7: Design Options.
Identify the options you will need for your installation program. What input does the end user provide during installation? For example, can the end user decide not to install some of the product files?
Make a rough sketch of the options and identify the controls (check or radio boxes) that will be needed. For more information about this process, see Chapter "Installation options"

Step 8: Create help system.
If you decide to include a help system for your installation, use a word processor that can read the RTF format and modify / complete INSTALL.RTF with the text for your help system.
Then compile the help system and test it without INSTALL.EXE, to make sure there are no errors in the help file. See Chapter "Help system"

Step 9: Create bitmaps.
Create bitmaps used in the installation process, by using a bitmap paint program capable of saving in Windows BMP / DIB format. Please select a program capable of saving with RLE compression. See chapter "Bitmaps"

Step 10 : Compress source files to save space on your installation disks. See chapter "Compressed files" for more information.

Step 11: Modify your configuration file.
Modify your configuration file to include options, help system, compressed files and bitmaps. See Appendix A and chapter "Installation options"

Step 12: Test your installation script.
Once you've modified your own configuration file, test it thoroughly. Copy the install products and source files to different floppy disks. Test the file under a variety of situations and computer configurations. Check the results by verifying that the files were copied to their appropriate directories and that files were updated correctly.

Finishing Things Up:
The last step in creating your installation might mean ZIPping them up for electronic distribution or loading them onto CD-ROM. Most of the rules for disk distribution apply for ZIP file distribution. One major difference is that the end user must UNZIP all files before beginning the installation process. You may

compress files that you distribute in a ZIP file, but you should try creating installations with compressed and uncompressed files to see which produces the smaller ZIP file. Sometimes compression's schemes work against each other. When a file is compressed the first time, it is generally smaller than if it gets compressed a second time.

Appendix C : Configuration sample 1

A very simple sample, showing the installation of a text editor EDITOR.EXE. The installation updates WIN.INI so that filenames with the extension .TXT are related to the editor.
The installation creates a group in the Windows shell, with the editor as a item. You can not directly test this configuration file, because you don't have the source files. Modify the sample to install some of your own files. Look at SAMPLE1.INF for more information.

Appendix D : Configuration sample 2

This is a configuration file for the installation of WinSetup. Rename SAMPLE2.INF to INSTALL.INF to test the configuration file and it's behavior. This configuration file shows some possibilities of a configuration file, including the use of options and a help system. The sample does not use any compressed files. Look at SAMPLE2.INF for more information.

Appendix E : Configuration sample 3

This is an extended configuration file for the installation of WinSetup. Rename SAMPLE3.INF to INSTALL.INF to test the configuration file and it's behavior. The configuration file shows the use of almost all possibilities of a configuration file, including the use of different installation disks. The sample does not use any compressed files. Look at SAMPLE3.INF for more information.

BeastWare
Svenstrupvejen 8, DK-4130 Viby Sj. Danmark
Phone: +45 4239-8410
E-mail: beast@ruc.dk
µ §