

## **About Dialog**

### **Purpose**

The **About Dialog** displays version and copyright information about this version of the Setup Builder application.

### **Available from**

- Help/About Setup Builder menu option

## Backdrop Display Dialog

### Purpose

The **Backdrop Display Dialog** enables the user to set up the criteria for displaying a backdrop during an install procedure.

### Available from

- Project/Backdrop Display menu option
- Fifth toolbar button

### Comments

The dialog contains the following fields:

### Selection

This list is used to select the settings on the right of the dialog you wish to view or change

### Backdrop Bitmaps

The backdrop display shown depends on the size of the screen on the machine on which the installation suite is running.

Setup Builder currently supports three screen sizes (640x480, 800x600 and 1024x768) for which you can specify individual bitmaps. Press the browse button to search for a bitmap currently installed on your machine to be used as a backdrop bitmap.

If you specify 'None' in the Appearance field for a specific screen resolution, you will not be able to specify a bitmap for that screen resolution.

The 'Default Bitmap Resolution' is supplied in order that an installation procedure can handle unrecognised screen sizes. This setting will be used as a 'last resort' bitmap. It is recommended that either the default shaded blue backdrop is used ie no bitmap file name or a centred or a tiled bitmap is used. It is not recommended that a 'straight' full screen bitmap is used because it may not fit all screen sizes. Similarly, a 'Stretched' bitmap might look strange on some screens.

Note that any bitmaps you select for backdrops will be automatically copied onto your first install suite diskette during the project build process.

### Appearance

This field controls how the backdrop will be displayed for each screen size. The appearance may be:

- None, in which case no backdrop will be displayed
- Default, in which case the default shaded blue backdrop will be displayed
- Centre on Screen, in which case the bitmap will be centred on the screen
- Tile on Screen, in which case the bitmap will be tiled on the screen
- Stretch to fit Screen, in which case the bitmap will be stretched to fit the screen

### Text and Size

The backdrop display may have four items of text drawn on it. These are used for displaying the application name and copyright messages for example. The text appears in one of four positions on the backdrop:

- Top Left
- Bottom Left
- Top Right
- Bottom Right

The text may contain leading/trailing spaces in order to move it relative to the appropriate edge of the screen.

The text may contain multiple lines in order to display multiple line messages on the backdrop display: press Ctrl+Enter to start a new line.

The size field specifies the size of the font (in points) to use to display the text.

**Text Colour**

Each of the four backdrop text items may be displayed in different colours. The Red, Green and Blue fields enable you to specify the colour to use to display the text.

Please note that not all colours are supported by all monitors, so it is advisable to only use the standard 'pure' colours otherwise your installation procedure might not appear correctly.

Press 'Preview' to view a sample of how the backdrop will appear.

Press 'Ok' to save any changes, or 'Cancel' to exit without saving any changes.

## **Browse Bitmap Dialog**

### **Purpose**

The **Browse Bitmap Dialog** enables the user to search for a bitmap file to be used on a backdrop display.

### **Available from**

- 'Browse' buttons on Backdrop Display Dialog

### **Comments**

Select the drive/directory by clicking on the appropriate fields. Enter the name of the required bitmap file and press 'Ok' to select it, otherwise, press 'Cancel' to exit without selecting a file.

## Build Project Dialog

### Purpose

The **Build Project Dialog** enables you to select the disk drive and directory on which you require your setup disk suite to be built.

### Available from

- Project/Build menu option
- 'Ok' button on the Optional Installation Files Dialog.

### Comments

The default location is that which was used for the last build process. Enter a new location if required.

Press the 'Ok' button to select the drive and proceed to build the project, or 'Cancel' to exit without building the project

### Notes

The name of the directory entered need not already exist since Setup Builder will automatically create it. The directory may also be several levels of subdirectory down - Setup Builder will perform a multi-level directory create to create a tree of directories.

If a single drive specification ( eg a:) is entered, the directory used will be the root (and not the current directory on a:) since a '\' character is always appended to the directory entered by the user. If you want another directory, then it must be specified in full.

**Warning: It is important to note that Setup Builder will delete all files in the specified directory before it copies the installation suite files to that directory.**

## **Cannot Create Project Error**

An error has occurred while attempting to create a new project file.

### **Possible causes:**

An attempt to create a new project on a write-protected diskette

An attempt to create a new project in the same location and with the same name as the currently open project

An attempt to overwrite an existing project file which is write-protected

An attempt to create a new project on a diskette which has not been formatted or has serious read/write failures - either reformat the diskette or discard it

## **Cannot Open Project Error**

This error occurs when Setup Builder has failed to successfully open a project file.

### **Possible causes**

An attempt has been made to open a Project which does not exist. If the project appears in the most recently used project list at the bottom of the Project menu then it is likely that the project file has been deleted

A general failure to read a disk

An attempt has been made to open a project file on a disk which is write-protected. Setup Builder MUST have read-write access to project files in order to successfully open them

## Commonly Reported Problems and Questions Asked

Following is a list of reported problems and commonly asked questions about Setup Builder together with the solutions.

### **'This application uses CTL3DV2.DLL, which has not been correctly installed'**

Setup Builder uses CTL3DV2.DLL to provide a '3d' appearance to all controls in dialogs. CTL3DV2.DLL is a Microsoft product and expects to be installed in the \WINDOWS\SYSTEM directory. If it is not, then the above error results. The Setup program for Setup Builder installs CTL3DV2.DLL in \WINDOWS\SYSTEM. If you receive this error then another copy of the .DLL must have been found, probably in your \WINDOWS directory or a directory in your path. It is suggested that you only have one copy of CTL3DV2.DLL on your machine to avoid this error. Note that the application can be run perfectly successfully even if the above message does appear.

### **'This software has not been installed correctly. Please re-run the installation procedure'**

Setup Builder uses a technique of coded serial numbers to determine the expiry date of the Shareware Edition. It is possible that you have installed the application perfectly correctly and you still receive this error when you run the application. In this case, it is due to a bug in version 3.06. You will also receive this error if you tamper with certain entries in the .INI file. The solution is to delete \WINDOWS\BUILDER.INI and re-install the software. Better still, obtain the latest version of the software since this bug has now been fixed.

### **C:\WINDOWS\DOSEXEC.BAT Cannot find file. Check to ensure the path and filename are correct**

Setup Builder uses the Microsoft COMPRESS.EXE program to compress files. It creates a batch file (DOSEXEC.BAT) in the WINDOWS directory which contains a call to COMPRESS and then uses a Windows .PIF file (DOSEXEC.PIF - also in the WINDOWS directory) to run the batch file. The above error occurs whenever Windows has not been installed in the C:\WINDOWS directory, for example on a network drive or simply another directory eg C:\WIN31. To correct the problem you must use the Windows PifEdit program to modify DOSEXEC.PIF. You should modify the 'Program Filename' to point to the directory in which Windows has been installed. A typical example modification might be:

```
G:\WIN31\DOSEXEC.BAT
```

You should then save the changed .PIF file. COMPRESS.EXE must also be in a directory pointed to by your PATH= in your AUTOEXEC.BAT. The above should resolve the problem.

### **When I run my setup procedure and select 'de-install' I get a message 'The XYZ software has not been correctly installed. The De-installation procedure is unable to de-install the XYZ software.' and I am unable to de-install my software which I have only just installed**

There are a number of reasons as to why a de-installation procedure fails with this message:

The most common reason is due to the name of the application .INI file being incorrect. When an



item of software is installed, the setup procedure stores the directory name in which the software was installed in the application .INI file. At the same time, a file called DEINST.INF is created in the application installation directory which stores information on what files were installed and how to de-install them. DEINST.INF is used to de-install your software and is effectively pointed to by an entry in the application .INI file. Every application installed by Setup Builder must therefore have its own .INI file.

You can specify the name of the application .INI file in the Disk/Script File section of the Project/Attributes dialog. By default, the .INI file defaults to %Application% (which is the value set in the Application Name field in the Application section of the Project/Attributes dialog) which is the name of the application.

If you change the application name to a name longer than 8 characters or place spaces or invalid file name characters in the application name, this will cause the .INI file to have an invalid file name which causes the installation directory of your software not to be saved in the .INI file (the .INI file can't be created with an invalid file name) which in turn causes the de-install to fail with the above message because the DEINST.INF file cannot be found.

The solution is to change the name of the .INI file: eg to MYAPP.INI This will solve the problem, but you will need to do a re-build of the project. Having done this, you can then set up the .INI file manually and place an entry in it:

```
[Install]
Path=C:
```

You can then run your de-install.

If the DEINST.INF file is deleted this will also result in the above message.

## **File Compression Errors**

One of the following file compression errors may occur:

### **Unable to find DOSEXEC.PIF**

File compression requires the use of DOSEXEC.PIF. This error occurs when this file cannot be found to perform file compression

### **Source not found**

This error occurs when a file which does not exist is included in a project.

This would happen when a project is created and the source file is later deleted without removing it from the project.

### **Compression Failure**

This error usually occurs when there is not enough space on the target disk, the disk is write protected, the file attempting to be created is write protected or the COMPRESS.EXE program fails.

### **Unable to find COMPRESS.EXE**

This error occurs when the application is unable to find the COMPRESS.EXE program. It must be in the application directory or in a directory in your PATH

### **Copying a file onto itself**

This error occurs when the file specified in the project has the same name as a file on the setup disk being created.

This error would only occur if a setup disk was being created from the WINDOWS directory since this is where temporary compressed files are stored prior to copying to the setup disk

## **Import 3rd Party Components Dialog**

### **Purpose**

The **Import 3rd Party Components Dialog** enables the user to automatically import files into a project which are part of a third party product

### **Available from**

- File/Import 3rd Party Components menu option

### **Comments**

Select the components you require to be imported into your project by selecting one or more of the items listed and then press the 'Ok' button to add them to your project.

Press 'Cancel' to exit the dialog without adding any extra files to your project.

### **Notes**

Third party components are those such as dynamic link libraries and support files supplied by other vendors, for example, the Visual Basic run-time file VBRUN300.DLL and the dynamic link libraries which make up Access to support a Visual Basic application are such third party components.

All the components supported by Setup Builder are listed in the IMPORT.INI file which may be found in your WINDOWS directory.

If you wish, you may add further components to this file but you must comply with the file format documented in IMPORT.INI.

IMPORT.INI will be upgraded in future versions of Setup Builder to support further third party products.

## **Stop Build Confirmation**



















This message appears when the user has selected to stop a build process from continuing.

Select 'Yes' to stop the build process.  
Select 'No' to continue the build process.

## Contents for Setup Builder Help

Setup Builder is a utility program for creating and maintaining Windows-hosted application installation procedures.

To learn how to use Help, press the F1 key.

-  [What is Setup Builder ?](#)
  
-  [Menu Options](#)
-  [Main Setup Builder Window](#)
-  [Setup Builder Errors](#)
-  [Commonly Reported Problems and Questions Asked](#)
  
-  [Technical Notes](#)
-  [Further Technical Information](#)
-  [Setup Script Help](#)
  
-  [Quick Start to creating an Install Disk Suite](#)
-  [Sample Projects/Files](#)
  
-  [Other products available](#)
-  [What is Shareware ? - Description and Disclaimer](#)
-  [Payment for and Registration of Software](#)
-  [Software Price List](#)
-  [Contacting the Author](#)
  
-  [Application Limitations & Limitations on use](#)
-  [Other Setup Builder Users](#)
  
-  [Help File Copyright](#)

## File Copy Errors

One of the following file copying errors may occur:

### **The specified source file '..' could not be found"**

This error occurs when a file which does not exist is included in a project.

This would happen when a project is created and the source file is later deleted without removing it from the project.

### **An error occured while creating '..'. Try removing write-protection from this file/disk.**

This error usually occurs when there is not enough space on the target disk, the disk is write protected or the file attempting to be created is write protected.

### **There is not enough space on the target disk to copy '..'. Try using another diskette.**

This error occurs when there is not enough space on the target drive for the specified file.

Setup Builder will not normally report this error as it normally asks for another diskette when this error occurs.

However, this error may occur at the end of building a setup procedure when a temporary copy of the setup script file is copied from the project (.SPJ) file directory to the first diskette in the install suite.

Please see the [Technical Notes Setup Script diskette space](#) section for information on how to reserve more space for the script file.

### **You cannot copy '..' onto itself - the source and target file names must not be the same.**

This error occurs when the file specified in the project has the same name as a file on the setup disk being created.

This error would only occur if a setup disk was being created on the same disk as that containing all the files making up the project.

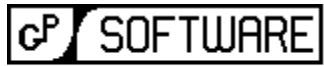
Since Setup Builder always creates setup procedures in the root directory of a disk, the source project files would have to be in the root of the same disk on which the setup procedure was being created.

### **An error occured while reading '..'.**

This error occurs when Setup Builder has failed to read a block from a file. It normally only occurs when there is disk corruption.

### **An error occured while writing '..'.**

This error occurs when Setup Builder has failed to write a block to a file. It normally only occurs when there is disk corruption.



**GRAHAM PLOWMAN SOFTWARE**

This Windows Help file was written by Graham Plowman  
using Help Builder Version 1.09.001 and refers to:

Setup Builder Version 4.01.002 / 16/04/96

Copyright 1993 - 1996 G.Plowman

## **Create Dependencies Warning**

This warning occurs when SetupBuilder identifies that the project file (.SPJ) which is being opened has been moved to another directory which is different from that in which the project was originally created.

SetupBuilder assumes that if a project file has been moved, then it is likely that the component files have also been moved. Since the names of these files are held in the project file, they must be amended to reflect the new project directory.

If the user selects the Yes button, the dependencies will be rebuilt.

If the user selects the No button, dependencies will not be rebuilt but the project file will be amended to register the fact that it has been moved, but not the topic files.

Normally, if you copy all the component files of a project from one directory to another then you should answer 'Yes' to rebuilding dependencies.

If you just move the project (.SPJ) file then you should answer 'No'



## **Delete File**

You have just selected to delete the currently selected file(s) from the list of files in the project.

The current dialog is asking you to confirm that you do in fact wish to remove the selected file(s) from the list.

Answer 'Yes' to remove the file(s) or 'No' to keep the file(s).

You can add the file(s) back in to the project at a later time via the Project/Edit dialog, but you will need to re-enter the File/Attributes for each file deleted.

## **Edit Project Dialog**

### **Purpose**

The **Edit Project Dialog** enables you to select all of the files which you require to be included on your setup distribution disk(s) and therefore in the project.

### **Available from**

- Project/Edit menu option
- 'Ok' button on the [New Project Dialog](#).

### **Comments**

The top-left list box shows all of the files in the current directory - the current directory is displayed above the list box.

To include a file in the project, double-click on the required file and it will appear in the list box at the bottom left.

Note that Setup Builder will not allow you to include a file twice.

The top-right list box shows the current drives and directories. Click on these as required to change drives/directories.

The bottom-left list box shows all of the files which have been selected for inclusion in the project. To remove a file from the project, double-click on the file in this list.

The 'Add All' button will add all of the files shown in the top-left list to the bottom-left list, whereas 'Remove All' does the opposite.

Press the 'Ok' button to save the list, or 'Cancel' to exit without saving the list.

### **Notes**

Any new files added to a project will have default file attributes so you should adjust these via the File/Attributes menu option.

If the Shareware Edition of Setup Builder is being used, the number of files allowed in a project will be limited to 25.

The Professional Edition will allow up to 1024 files.

**The selected script file is already encrypted Warning**  
**The selected script file is already un-encrypted Warning**

These warnings occur when an attempt is made to:

- Encrypt an already encrypted script file
- Un-encrypt a script file which is not encrypted

When these messages appear, no further action is performed on your script file and it will be left intact.

## Encrypt/Unencrypt Script File Dialog

### Purpose

The **Encrypt/Unencrypt Script File Dialog** enables you to encrypt or un-encrypt a script file created by Setup Builder or to encrypt scripts that you have written manually to be supplied on your distribution disks

### Available from

- File/Encrypt / Un-encrypt Script menu option

### Comments

Enter the name of the script file to be encrypted or un-encrypted in the **File Name** field or press the **Browse** button to search for the required file.

Select the appropriate operation by clicking the **Encrypt Script File** button or the **Un-encrypt Script File** button. Then press the **Ok** button to commence the conversion.

### Notes

Setup Builder will not allow you to encrypt a script file which is already encrypted and similarly, it will not allow you to un-encrypt a script file which is not encrypted.

The script file interpreter automatically recognises encrypted script files and reads them accordingly. There is no need for any extra switches/parameters to tell it whether a script is encrypted or not.

The conversion process actually replaces your original script file with the newly converted version so you don't end up with two files: your original file and the newly converted version. The new file is created as a temporary file on your hard disk. It is then copied over your original file. The temporary file is deleted afterwards.

## Setup Builder Errors

Create Dependencies

Too many files selected for project

Failed to open the selected file

File already exists in the project

File copying errors

The selected script file is already encrypted

The selected script file is already un-encrypted

Unable to create project file

Unable to open selected project file

Unrecognised project (.PRJ) file

Unrecognised project (.PRJ) file version

## **Failed to execute program error**

This error occurs when Setup Builder has failed to execute the script interpreter for previewing displays or testing scripts.

### **Possible causes**

There is not enough memory available to run the interpreter program. Close down some other currently running applications to free up memory.

## **File already exists in the project Warning**

This warning occurs when a Visual Basic .MAK project file is being imported into a Setup Builder project or a file is being pasted from the Clipboard. Setup Builder has identified that an attempt is being made to add a file to the current project when it already exists in the project.

A file cannot be added to a Setup Builder project more than once.

## File Attributes Dialog

### Purpose

The **File Attributes Dialog** enables the user to make selections about how a file is to be installed during the install procedure. It is also possible to specify the same selections for a number of files at the same time: see the notes at the bottom of this page.

### Available from

- File/Attributes menu option
- Fourth toolbar button
- Double clicking on the file in the main file list

### Comments

The dialog contains the following fields:

### Attributes

This list is used to select the settings on the right of the dialog you wish to view or change

### Target Location

This specifies the directory where the file is to be installed. It must NOT contain a file name and may optionally end with a back slash.

You can place a directory name in this field, but it is advisable to take advantage of the Setup system's ability to use variables in which case you may place the following in the field:

**%InstallPath%** This will cause the file to be installed in the directory which the user enters when prompted to enter a path in which to install files during the installation process.

**%WindowsDirectory%** This will cause the file to be installed in the Windows directory.

**%SystemDirectory%** This will cause the file to be installed in the Windows system directory.

### Copy Comment

This is the message which will appear at the top of the gauge dialog for the file when file copying is in progress in the installation procedure.

### File Usage

This specifies how a file is to be copied and may be one of the following:

'Do Not Copy File' simply does not install the file at all. It is the programmers' responsibility to write some 'User defined' code to install these files as required. Often, this facility is used to supply files on install diskettes as extra components for the user to copy manually when required.

'Display in gauge' is the default. This causes the file to appear in the gauge and to be installed on the user's machine.

### Check if in use

Causes a check to see if the file (normally executables/DLLs etc) is in use by Windows. If it is, an error message will appear and the install procedure will be terminated.

### Confirmation

This option enables you to specify the action to be taken during installation if a file already exists. The action may be:

'No overwrite confirmation' - the file will be installed anyway whether it already exists or not

'Confirm overwrite if file exists' - displays the overwrite message below if the file already exists

'Confirm overwriting newer file only' - checks to see if the file already exists and displays the overwrite message below to ask the user to confirm overwriting the file. The confirmation will only appear if the file being installed is older than the one to be overwritten. If the file being installed is newer than the existing file then the existing file will be overwritten automatically with no overwrite message. This enables files to be automatically upgraded where necessary.



If an overwrite confirmation has been configured and the file does not already exist on the user's machine, the file will be automatically installed with no message displayed.

### **Overwrite Message**

If you require a specific message for an overwrite confirmation, complete this field.

Note that a \$ sign in the Overwrite message field signifies substitution for the file name.

If this field is left blank, a default message will be displayed which shows the names, dates, times and lengths of the two files being compared:

Setup is attempting to overwrite:

C:\TEMP\TEXT2.TXT  
(24/6/95 11:24 47 bytes)

with:

A:\TEST\TEXT2.TXT  
(24/6/95 11:10 47 bytes)

Are you sure you wish to overwrite this file ?

### **File Compression**

Setup Builder enables you to place files on your setup disks in compressed form. The compression process uses the standard Microsoft LZExpand method.

It is suggested that you compress files which are bigger than 200K, but not those below about 2K since the latter will not actually save any diskette space because the smallest allocateable block on a diskette is usually 2K. This is however at the user's discretion.

Check the checkbox to compress the file or leave it blank if the file is not to be compressed.

If compression is selected, another field will appear next to the check box. This field contains the file extension to be used for the compressed file when it is placed on your setup diskette. By default, Setup Builder uses the Microsoft convention of placing a '\_' as the last character (eg .EXE becomes .EX\_ and .A becomes .A\_), however, you may type a different extension if required, but it must not be the same as the extension of the original file.

### **Optional File Inclusion**

This check box marks a file to appear in the Optional Installation Files Dialog when a project is built, giving you the opportunity to select it for inclusion in the project at build-time.

### **Add to Registry**

This check box enables a file to be registered in the Shared File Registry which is used to control the deletion of shared files during de-installation.

Please see the **Register** function in the script language help for a detailed description of how the Registry operates and how it should be used.

Please note that the file Registry facility is only activated if you have selected the 'de-install' option in the Project Attributes dialog.

### **Program Manager Icon**

The check box specifies whether an icon is to be created in a Program Manager group for the file. If you check this, you should enter the Icon Text which will appear below the icon in Program Manager and any parameters which the file may need if it is an executable.

You may also optionally specify the name of a .EXE or .DLL file from which to extract the icon to use in Program Manager. The index field enables the number of the icon to be specified. The first icon in a file is zero, the second one and so on.

If the icon file is left blank, the name of the file being installed will be used as a default. The default icon is zero. It is possible to specify the icon index and no icon file: the icon will be taken from the file being installed.

The Working Directory field may be optionally left blank, but if supplied, should contain the name of the directory you wish to be made current for the icon when the icon is run. The directory of

the file being installed is taken as the default.

It is recommended that any directory specifications are given in terms of the installation directory, for example: %InstallPath%

Please see the Setup Script file for further information on the script language and variables.

### **Target File Protection**

It is possible to write-protect files after they have been installed and similarly, a re-installation needs to be able to unprotect them before installation. Use the check boxes as required.

Press 'Ok' to save any changes, or 'Cancel' to exit without saving any changes.

### **Notes**

Using the **File Attributes** dialog it is possible to assign the same settings to a number of files in the project at the same time. To do this, click on the required files in the Setup Builder main window which are to have the same attributes, then select 'File/Attributes' from the menu.

When you press 'Ok' in the **File Attributes** dialog, all of the selected files will be given the same attributes as the file which was selected last in the list - the file with the dotted lines around it in the Setup Builder main window.

The following attributes will be made the same:

- Target Location
- Copy Comment
- File Usage
- Overwriting Message
- File Protection

## The IMPORT.INI file

The IMPORT.INI file is supplied with Setup Builder and resides in the WINDOWS directory.

It is used by Setup Builder to list third party software vendor's products and the supporting files used by those products to provide a quick and easy way to import 3rd party products into a Setup Builder project for distribution with your application.

Typically, all Visual Basic applications must be supplied with VBRUN300.DLL and MSAFINX.DLL (The latter is necessary to stop the notorious 'File not found error'. It contains the iif() function and various date and financial functions). These files are listed in IMPORT.INI as Visual Basic 3.0 Runtime support files. By selecting 'Visual Basic 3.0 Runtime' files in the Import 3rd Party Components Dialog, the user can quickly and easily import the appropriate files into their project.

If your Windows installation is in a different location to C:\WINDOWS or any of the component files listed in IMPORT.INI are in different locations, you will need to edit IMPORT.INI to point to the appropriate locations of the files on your hard disk.

IMPORT.INI is a standard windows .INI file. Each section heading lists the name of the third party component and this is displayed in the Import 3rd Party Components Dialog. Under each section is a list of each of the files which make up that 3rd party component. The format of the individual file items must conform to the format documented in the IMPORT.INI file, otherwise Setup Builder will not read them correctly and may import unwanted files into your project.

## **Invalid directory Error**

This error occurs when an invalid directory name has been entered.

## Application Limitations

Setup Builder currently has the following limitations, which will be modified in future releases:

- 25 files per project (Shareware Edition), 1024 (Professional Edition)
- Setup Builder has only been tested under Windows 3.1 and 3.11. It has not been tested under OS/2, Windows NT or Windows 95.

### Limitations on Use

Users of the Shareware Edition must **not** distribute the diskettes produced by Setup Builder, however users of the Professional Edition may do so freely.

There is no fee payable for distributing large numbers of installation procedures, however, the author does request that if you use Setup Builder for distributing other software for monetary gain that you mention Setup Builder and the author's [contact details](#) in your supporting documentation or help files.

The privilege of this lack of distribution fee is on condition that you do not sell on Setup Builder or any of its components for monetary gain. You may supply INST.EXE and a script file with your applications freely but you must not charge for them in your overall software charge and you must have registered your copy of Setup Builder.

## **Save Before Continue ?**

The option you have just selected will cause loss of changes which have been made to the project since it was last saved.

The current dialog is asking you to confirm what you wish to do:

**Yes** Will save the project and continue with the option selected

**No** Will continue with the option selected without saving the changes and therefore losing them

**Cancel** Will not save the changes, but it will not continue with the selected option either.

## Main Setup Builder Window

The main Setup Builder window appears when the application is first started up and remains on the screen until the application is either minimised or closed.

When a Setup project is open, the main window displays a list of all the files currently selected for inclusion in the project.

The list is split into a number of columns:

### File Name

Specifies the name of a file to be copied onto the Setup diskettes when the project is built.

### Prot

When a file is being installed from a setup suite, this option specifies the target file protection. '.' means none. A 'B' may appear meaning that the file will be unprotected before it is overwritten. An 'A' means it will be protected after it has been installed.

### Usage

The Usage section contains several columns in the order which follows. The meanings of the column entries are:

- CU Check if in use (ie check if Windows is currently running it)  
This option is used to specify what to do if the file is currently being run by Windows when the install procedure is attempting to install it. Installing a file which is in use by Windows (.EXE, .DLL etc) often causes application crashes.  
If the file is in use, the user will be informed in a 'graceful' fashion and the install procedure will terminate.
- CO Confirm Overwrite (ie check if file exists before installing it and ask the user to confirm overwriting if it does exist)
- DG Display in gauge. The default option which simply copies the file in without any checking and displays it in the progress bar during installation.
- C Compression. Setup builder can compress files before placing them on installation diskettes. A file marked with this option will be compressed. Note that this option requires the Microsoft COMPRESS.EXE program to operate.  
When running the installation procedure, the file will be automatically un-compressed.  
Note that un-compressing does not require COMPRESS.EXE.
- O Optionally Included File. At build time a dialog appears from which all the files marked 'O' can be selected for inclusion or omitted from a setup procedure. This enables you to maintain one project but optionally supply certain files.
- R Registry. File will be included in the Shared File Registry.

If all options are blank, the file will be copied to an install diskette, but it will not be installed when the install procedure is run. This facility is useful for supplying supplementary files which the user can copy off of install disks if they require them.

### Icon

This column contains a 'Y' if an icon is to be created on a program manager group for the file. In this case, the icon text will also appear next to it.

An 'N' means that no icon will be created.

At the top of the Setup Builder main window is the menu and a toolbar. Toolbar options may be selected by simply pressing the appropriate button with the mouse. All of the toolbar options are duplicated on pulldown menu options.

At the bottom of the Setup Builder main window is the status bar. It displays messages about Setup Builder's current activity. It also displays a brief description of menu options when they are highlighted on a pulldown menu.

Both the toolbar and status bars may be removed/activated by selecting the appropriate option from the **View** menu.

To close the Setup Builder main window and therefore exit the application, either press Ctrl+F4 or use the mouse to double click on the system menu icon or select the **File/Exit** menu option.



## Menu Options

The Setup Builder program has the following menu options:

- File Options relating to files selected in the project
- Edit Options relating to cutting, pasting and selection of files in the project
- View Options relating to the Setup Builder main window appearance
- Project Options relating to a Project
- Help Options relating to help

### File Menu

#### Attributes

This option enables the user to specify how a particular file in the project is to be installed. This includes the directory, whether an icon is to be created and overwrite/usage checks.

See [File Attributes Dialog](#)

#### Delete

This option enables a file to be deleted from the list of files in the application main window.

#### Encrypt/Un-encrypt Script

Enables the user to encrypt or un-encrypt manually created script files.

#### Exit

Exits the Setup Builder application. You will be asked to save the project if it has been changed in any way since it was last saved.

### Edit Menu

#### Cut

Cuts the selected files from the list of files in the main window to the application clipboard

#### Copy

Copies the selected files from the list of files in the main window to the application clipboard

#### Paste

Pastes files from the application clipboard into the list of files in the main window

#### Select All

Selects all files in the application main window

#### Unselect All

Unselects all files in the application main window

### View Menu

#### Toolbar

Toggles on and off the display of the tool bar. This is the row of icons at the top of the Setup Builder main window.

#### Status Bar

Toggles on and off the display of the status bar. This is the bar at the bottom of the Setup Builder main window which displays various messages about progress of the currently selected facility within Setup Builder.

### Project Menu

#### New

Enables a new project file to be created.

See [New Project Dialog](#)

Open

Opens an existing project file on the disk.

See [Open Project Dialog](#)

Save

Saves the currently open project to disk.

Save As

Saves the currently open project under a new name.

See [Save Project As Dialog](#)

Close

Closes the currently open project file. You will be asked to save the project if it has been changed in any way since it was last saved.

Edit

Enables the user to select files for inclusion in an installation disk suite.

See [Edit Project Dialog](#)

Import VB Project

Enables a Visual Basic project to be imported into the current Setup Builder project. The import process includes all files listed in the VB .MAK file into the Setup Builder project.

See [Import VB Project Dialog](#)

Import 3rd Party Components

Enables third party components to be imported into the current project. Third party components are those such as run-time dynamic link libraries and their support files.

See [Import 3rd Party Components Dialog](#)

Backdrop Display

Enables the user to configure/maintain a backdrop display for an installation

See [Backdrop Display Dialog](#)

Program Manager Group

Enables the user to specify how the setup procedure should make a Windows Program Manager group.

See [Program Manager Group Dialog](#)

Attributes

Enables the user to specify attributes about the project.

See [Project Attributes Dialog](#)

Dialog Text

Enables the user to specify the text in each of the dialogs which will appear during an installation procedure.

See [Project Dialogs Dialog](#)

User Code

Enables the user to enter user-defined code at various stages of a setup procedure.

See [User Defined Code Dialog](#)

Build Setup Procedure

Builds the currently open project.

Test Setup Procedure

Tests the currently open project by running it.

## **Help Menu**

Index

Shows the main index for help on Setup Builder.

Search for Help On

Enables the user to search for help on a specific topic.

Using Help

Provides help on how to use the help system.

Setup Script Help

Displays the Setup Script language help file.

About Setup Builder

Displays version and copyright information about this version of Setup Builder.

## **Failed to open the selected file Error**

This error occurs when an attempt has been made to open a file, normally for encryption. The error means that the file either does not exist or it cannot be opened for reading and writing

## Optional Installation Files Dialog

### Purpose

The **Optional Installation Files Dialog** enables the user to build a project and have the ability to include or omit files as required. This means that you don't need to maintain multiple projects if the only difference between them is that they install a few slightly different files.

### Available from

- Project/Build menu option when some files in the project are flagged as being optionally included on install disks.

### Comments

Note that this dialog does not appear when Project/Build Setup Procedure is selected if no files in the project have been marked for optional inclusion.

The dialog contains two list boxes:

The top list shows all files in the project which may be optionally included on your setup suite disks, but are currently not selected for inclusion.

The lower list shows all files in the project which may be optionally included on your setup suite disks and are currently selected for inclusion.

The 'Add All' button will move all files in the top list to the bottom list - ie select them all for inclusion on your setup disks.

The 'Remove All' button will move all files in the bottom list to the top list - ie omit them all from inclusion on your setup disks.

Press the 'Ok' button to save the changes or 'Cancel' to exit and stop the build process.

Pressing the 'Ok' button will then cause the [Build Project Dialog](#) to appear.

## **Other Users of Setup Builder**

Setup Builder is also used by the following organisations:

- Intel Corporation
- Texas State Auditors Office
- Cinema Magnetique Communication SA, Paris
- FSI GmbH, Kuessaberg, Germany
- Idaho State Library
- Moore Data Management, Minneapolis
- Dun & Bradstreet SA, Mexico
- British Royal Air Force
- Fujitsu Australia, Melbourne, Victoria
- American Airlines
- UK Nirex Limited, Harwell, England
- Canada Life, Co. Dublin, Ireland
- Philips Dictation Systems, Vienna, Austria
- University of Nottingham, England
- BP Oil Deutschland GmbH
- DataCard Canada Inc, Mississauga
- Airbus Industrie, Blagnac, France

## **Payment for and Registration of Software**

If you wish to continue to make use of the Setup Builder software you are expected to pay a registration fee.

The registration fee for the Setup Builder software may be found in the [Software Price List](#) section of this help file.

Compuserve also has a facility for payment for shareware software which is ideal for overseas users. Essentially, a shareware author registers their software in a database and then the facility transfers payment from one user's account to another.

You can pay for Setup Builder via this facility via (GO) SWREG at any time. Select the 'Register Shareware' option. The registration number of this software is 4153.

Your registration fee entitles you to use this software on a single computer for an unlimited period and to make as many copies as you wish for backup purposes.

Upon receipt of your registration fee you will receive a disk (mail orders only) containing the latest version of the software and advice on how to remove the 'shareware warning' dialog, stop the one month disabling feature and release the various shareware restrictions in the software. You will also receive notice of updates to the software. Free technical support is available to registered users.

See Also: [Contacting the Author](#), [What is Shareware ? - Description and Disclaimer](#), [Software Price List](#)

## **Program Manager Group Dialog**

### **Purpose**

The **Program Manager Group Dialog** enables the user to specify how a Windows Program Manager group is to be created after installation of all the files in the install suite.

### **Available from**

- Project/Program Manager Group menu option
- Sixth toolbar button

### **Comments**

The dialog contains the following fields:

### **Program Manager Group**

This enables the user to specify how the program manager group will be created. It may be:

'No Group' in which case no Group file will be created

'Make Group' in which case a new group will be created.

'Use Existing Group File' in which case a new group will be created from an existing group file which you may have specified as a file to be installed.

### **Give User Option of building Group**

Check this field if you wish the setup procedure to give the user the option of creating a group. If this is not checked and 'Make Group' above has been selected, the group will be created without giving the user any option.

### **Group Caption/File Name**

If 'Make Group' is selected above, you should place the caption text of the new group in this field.

If 'Use Existing Group File' is selected above, you should enter the file name of the group file to be used. It may contain variables (eg %InstallPath%)

### **Group File Protection**

A group file may be write protected. This option enables you to protect the group file after installation so that no one can change it or to unprotect it before a re-installation.

Press 'Ok' to save any changes, or 'Cancel' to exit without saving any changes.



## **Save Project As Dialog**

### **Purpose**

The **Save Project As Dialog** enables a project file to be saved under another name.

### **Available from**

- Project/Save As menu option

### **Comments**

Select the drive/directory as required by clicking on the appropriate fields. Enter the new name of the project and press 'Ok' to save it.

Note: Saving a project under a new name will release the old file so that another user on a network can open it.

## Project Attributes Dialog

### Purpose

The **Project Attributes Dialog** enables the user to specify some basic attributes about the way in which the project is to operate during the installation process.

### Available from

- Project/Attributes menu option
- Seventh toolbar button

### Comments

The dialog contains the following fields:

#### Application Name

This field will be the contents of the %Application% variable during project installation. Enter the name of the application in this field. (eg Setup Builder)

### Comments

These are comments you wish to place at the top of the Setup script file and each line should be preceded with //

Normally you would place a message here of what the Setup procedure is going to install - the name of the software/version number/date etc.

#### Display 'Welcome' Dialog

To be 'friendly' to the user, an installation procedure can display a welcome dialog. This field enables you to specify whether this is required.

#### Prompt for Installation Path

The default installation path (below) is the normal directory in which your application would be installed.

Using the prompt field, you may also specify whether you wish to give the user the choice of changing this at install time - it is best to do so.

The text typed in this field will form the %InstallPath% and %InstallDrive% variables.

Note that this dialog will not appear during an installation if Setup identifies that the application has already been installed. You must de-install the application first then install it again to be able to place the application in a different directory.

Setup will not allow you to install two versions of an application in different directories

#### Display 'Licensing' Dialog

If you wish to license your software to a specific user and/or company, you may use this option to display a dialog which will ask for this information.

The text entered will be written to a .INI file which is specified in the 'Licensing .INI file' field below.

Suggestion: Licensing can be implemented in your software by reading the .INI file to display the text in the About dialog of your application.

Licensing information is placed in the Licensing .INI file section:

```
[License]
```

```
UserName=Graham Plowman
```

```
Company=My Company Name Ltd etc
```

#### De-Installation Option

Setup Builder can give the user the ability to provide the user with the option to install or de-install a piece of software.

Check this field to give the user the ability to de-install your application.

Note that Setup Builder automatically creates the script code to de-install your application. This code deletes all files installed (subject to the Registry), the application .INI file and the application directory. You can add extra code in the 'Application De-Installation' section of the User Code

Dialog to perform extra special de-installation such as removal of directories underneath the application directory which were created at install time.

### **Diskette Label Prefix**

This is the first 6 characters of the label to be placed on each diskette of the installation disk suite. The installation procedure uses this to ascertain whether the user has inserted the correct diskette in the drive.

The disks are numbered: LABEL01, LABEL02 etc where LABEL is the text you enter in this field.

### **Script File Name**

The default is usually sufficient and must be used for installation diskette suites.

See the Technical notes for more information on this field and how a Setup procedure works.

### **Encrypt Script File**

Script files may be encrypted. Normally they are ASCII human-readable files, but if you require some security so that they are not readable, this field gives that option.

### **Default Installation Path**

The default installation path is the directory in which your application will be installed. Even if you do not enable the user to change this using the 'prompt for installation path' above, you must specify this field.

The installation procedure will create this directory if it does not exist, but will proceed normally if it does exist.

### **Reserve Space**

When an installation disk suite has been completed, Setup Builder writes the appropriate script file on the first diskette. This field enables you to specify how much space to reserve for this file. See Technical Notes **Setup Script diskette space** for more information on this field.

### **Space Required**

This is the amount of disk space required for your software to be installed on a machine. The Project/Edit dialog normally calculates the correct amount of space required, but by pressing the **Calc** button you can re-calculate the space required.

Setup Builder calculates the space required by adding the size of all the files in your project together and then adding an extra margin for de-compression temporary files (normally the size of your largest compressed file).

### **Split Files Larger than**

In order to make efficient use of diskettes in an install disk suite, Setup Builder will split large files across one or more diskettes. A large file is considered as any file that won't fit on a diskette either because the diskette is full or the file is larger than the diskette size.

In some situations, you may not want files to be split. In this case you might want a file to simply be placed on the next diskette.

This field enables the user to specify the 'threshold' file size: given a file which will not fit on a diskette, files smaller than the threshold will be placed on a completely new diskette. Those larger than the threshold will be split across diskettes.

It is recommended that the threshold is between 50 and 100K. It is not worth splitting files smaller than this.

### **Application .INI file**

This is the name of the .INI file to which to write the licensing information entered in the 'Licensing' dialog. If the 'Licensing' dialog has been selected you should enter this field. If you do not, the name will default to the %Application% name .INI file. For example, if %Application% held 'test' then %IniFile% would hold 'test.ini'. However, this will fail if %Application% holds multiple words or is longer than 8 characters so it is in your interest to set the Application .INI file field if you wish to implement licensing!

The Application .INI file is also used to store installation information of where an application was installed. This is so that the de-install process knows where to de-install the application from.

**Copy setup.exe to diskette**

Check this field to have the build process place setup.exe on the first diskette.  
This is the program the user can run to initiate the installation suite.  
You must check this field or the install.exe field or both. You cannot leave both unchecked

**Copy install.exe to diskette**

Check this field to have the build process place install.exe on the first diskette  
This is the program the user can run to initiate the installation suite.  
You must check this field or the setup.exe field or both. You cannot leave both unchecked

**Copy lzexpand.exe to diskette**

Check this field to have the build process place lzexpand.exe on the first diskette.  
This file will automatically upgrade older versions found for the de-compression process

Press 'Ok' to save any changes, or 'Cancel' to exit without saving any changes.

## Project Dialog Text dialog

### Purpose

The **Project Dialogs Text dialog** enables the user to specify the text in each of the dialogs which will appear during an installation procedure.

### Available from

- Project/Dialogs menu option
- Eighth toolbar button

### Comments

You should find the defaults sufficient.

You may place the following embeded variable names in any of the fields:

%Application% for the application name

%Caption% for the normal dialog caption (unlikely to be useful)

%InstallPath% for the directory name where the application is to be installed

%InstallDrive% for the drive letter where the appliaction is to be installed

The 'Language' field may be used to control which language the installation procedure will be created in.

Press the 'Defaults' button to set the text for all the fields to their default values.

Changing the language and pressing 'Defaults' will cause the fields to be set to the defaults for the selected language.

Press 'Ok' to save any changes, or 'Cancel' to exit without saving any changes.

## **Import VB Project Dialog**

### **Purpose**

The **Import VB Project Dialog** enables a Visual Basic .MAK project file to be imported into a Setup Builder project.

### **Available from**

- Import VB button on Project/Edit dialog

### **Comments**

Select the drive/directory by clicking on the appropriate fields. Enter the name of the required project and press 'Ok' to open it.

## **New Project Dialog**

### **Purpose**

The **New Project Dialog** enables a new project file to be created.

### **Available from**

- Project/New menu option
- First toolbar button

### **Comments**

Select the drive/directory as required by clicking on the appropriate fields. Enter the name of the new project and press 'Ok' to create it.

Note: If you currently have a project open which has been changed since it was last saved, you will be asked if you wish to save it before a new project is created.

A newly created project is automatically locked such that another user cannot open the project or overwrite it in a network situation.

## **Open Project Dialog**

### **Purpose**

The **Open Project Dialog** enables an existing project file on the disk to be opened.

### **Available from**

- Project/Open menu option
- Second toolbar button

### **Comments**

Select the drive/directory by clicking on the appropriate fields. Enter the name of the required project and press 'Ok' to open it.

Note: If you currently have a project open which has been changed since it was last saved, you will be asked if you wish to save it before another project is opened.

Setup Builder keeps .SPJ project files open for the duration the user has the project 'open'. This is to maintain network awareness and stops two people updating the same project at the same time.



## Quick Start to creating an Install Disk Suite

This topic should give you a good understanding of the process of creating an installation procedure.

Before you can install any software or create any installation diskettes, you need to decide what files you want to install on your client's machine and where those files will be installed on that machine. It is also a good idea to have some idea of how you want your installation procedure to look.

You can then proceed to the first step:

### Creating a Project

In order to use Setup Builder to create any installation diskettes, you must create a project as the first step.

A project contains all the information about an installation suite necessary to build diskettes for an application. It includes the names of the files and all the attributes you set or select.

To create a project, select the Project/New menu option to obtain the New Project dialog. Enter the name of the file you wish your project to be called and press the **Ok** button. The new project will then be created and the Edit Project dialog will appear. This dialog enables you to select all the files you wish to be placed on your install diskettes. You should select all .COM, .EXE, .DLL, .VBX, .OCX and data files etc required by your application.

Note that you do not need to include CTL3DV2.DLL as Setup will automatically install this if it does not exist on the client's machine or if your version is newer than theirs. Setup requires this file in order to operate itself.

The Edit Project dialog is also available from the Project/Edit menu option, so it is possible to add more files to your project later if required.

Press **Ok**. Your project has now been created.

### Setting File Attributes

Now that you have created your project and selected the files you want to install, you need to make selections about how you want the files to be installed, for example, the directory in which to install them and whether they will have icons on Program Manager.

To set the attributes of a file you can double click on it in the Setup Builder main application window. Alternatively, you can also select the File/Attributes menu option to obtain the File Attribute dialog.

The File Attributes dialog appears showing the 'File Copying' information.

The Target Location is the directory in which the file is to be installed. By default, this appears as %InstallPath%. This represents a variable name used by the underlying Setup Script interpreter at run time. The variable holds the directory entered by the user in the 'Installation Path' dialog. It is not recommended that you 'hard code' the directory here because some users may not have the same drives that you have. You can however, place the file in a subdirectory of the application directory, so for example, if the user had entered C:\TEMP\ as the installation directory and you wanted the file to appear in C:\TEMP\WORK, then you would place %InstallPath%\WORK in the Target Location field. The installation procedure automatically adds a trailing \ character to the path name entered by the user in the 'Installation Path' dialog.

The Copy Comment appears as a message on the screen when the file is being installed to tell the user what is being installed.

File Usage determines whether a file will be displayed in the progress gauge at install time. If it is not displayed, the file will not be copied and it becomes the user's responsibility to copy the file from the diskette manually or the developer's responsibility to manually write 'user code' in the Setup project to perform the action. This enables you to supply sample files on your install disks without actually installing them.

Confirmation determines what Setup should do at install time if the file already exists on the target machine. This may be:

No overwrite confirmation - install the file regardless

Confirm overwriting if file exists - If the file already exists, confirm overwriting it

Confirm Overwriting newer file only - If the file exists, confirm overwriting it if the existing file is

newer than the one being installed

The overwrite message will appear when either of the 'Confirm Overwriting' options are selected and the file being installed has been found to already exist.

You can optionally compress files. By convention, compressed files have a different file extension. It is recommended that you use the default supplied, but you can change it if required. It must not be the same as the original file.

Optional File Inclusion enables you to maintain one project instead of several when you require more than one installation configuration - when each configuration contains different files. Making a file optional will cause a dialog to appear at 'Build' time for you to select the optional files to be included on the diskette.

The File Attributes dialog also enables you to specify whether a file should have an icon on Program Manager. Check the 'Make Icon' field to have an icon created.

The Icon Text will appear on the Program Manager group (discussed later on this page).

You can have parameters to the file, such as a file name if the icon is an executable.

The icon file is normally left blank, but if supplied should refer to a .EXE file containing icons such as PROGMAN.EXE

If the icon file is specified, the the Icon Index should also be specified. Executables contain multiple icons in a list - you see this in Program Manager when you select 'Change Icon'. The first is number zero, the second one and so on. Enter the number of the icon you require.

Icons may optionally have a Working Directory. The default as in Program Manager, is the directory of the executable file.

It is also possible to set the read only attribute of files installed using the File Protection options. This feature is not often used, although an 'unprotect' before copy will ensure against an installation procedure failing due to read-only files.

Having set the above attributes for one file, you then need to repeat the exercise for all the other files in your project. If you have several files which have the same attributes (other than file name of course) you can select them all by single-clicking on them and then select the File/Attributes menu option. In this situation, the files will all be set to have the same attributes as the last file you selected - the one with the dotted line around it in the list. Press **Ok** in the File Attributes dialog to save the attributes to all the selected files.

### **Creating a Backdrop Display**

This is a feature which is becoming very common and adds a very 'professional' looking touch to your installation - first impressions count etc

The backdrop display appears on the screen from the start of an installation. When you installed Setup Builder, the backdrop was the blue shaded background.

To set up a backdrop for your application, select the Project/Backdrop Display menu option to obtain the Backdrop Display dialog.

Because different machines have different screen sizes, your installation procedure must cope with this. This is why Setup Builder gives you the facility to select different bitmaps for different screen sizes and a 'default' selection for unrecognised screen sizes.

Here are some suggested approaches to backdrops:

Select the appearance to 'None' for all screen sizes if you don't want a backdrop.

Select the appearance to 'Default' for all screen sizes if you just want the default shaded blue backdrop. This is the recommended approach if you cannot decide!

Select the appearance to 'Centred on Screen' and select a small bitmap (such as your company logo) for all screen sizes.

Select the appearance to 'Tile on Screen' and select a small bitmap (such as your company logo) for all screen sizes.

Note that any bitmaps selected will be automatically placed on the first diskette of your installation suite at 'Build' time, so you do not need to include them in your project, however if

you do, it does not matter.

In addition to the backdrop display and bitmap, you can also place text at each corner of the screen. You can select the font size and colour. It is recommended that you at least use the Top Left Text to display your product name. See the Setup Builder installation and the sample file SETUP.SPJ for examples on this.

Press **Ok** to save the backdrop information.

### **Creating a Program Manager Group**

You can have your installation procedure create a Program Manager group when your software has been installed. Indeed you should do this if you have selected any of your files to have Program Manager icons.

To set up the group, select the Project/Program Manager Group to obtain the Program Manager Group dialog.

Click on 'Make Group' to have a group created. The title of the group will be that specified in the Group Caption field.

It is recommended that you give the client the option of creating a group: check the 'Give user option' field. If they select 'No' at install time, no icons or Program Manager group will be created. Please note that Setup Builder, unlike many other install utilities, will recognise if a program manager group has already been created for your application and it doesn't create another group with the same name if you install your software twice.

Similarly, Setup Builder does not create duplicate icons in a group. The Group Caption and Icon Text are used as identifiers in order to determine this.

If you wish to supply a ready made group file in your installation procedure, check 'Use Existing Group File' and place the name of the file in the Group Caption/File Name field. In this situation, you can also use the Group File Protection options. They are not valid if you have selected 'Make Group' above. You will need to include the group file in your project and set its Target Location to %WindowsDirectory% in the File Attributes dialog.

Note that when icons are created they are created on the currently active Program manager Group. If you select 'No Group' the icons could appear on any group! Creating a group automatically makes the new group the current group so you do not need to worry about selecting the current group.

Press **Ok** to save the Program Manager Group information.

### **Setting the Application Name and Install Attributes**

In order for an install suite to be created, a few settings must be set which describe the install script file.

Select the Project/Attributes menu option to obtain the Project Attributes dialog.

The application name is the name of your application, for example, Setup Builder. This text will be placed in the %Application% variable for use by the underlying script program.

Comments are script language comments you wish to appear at the top of the script file.

Normally, this will detail the application and version etc. To create new lines, press Ctrl+Enter. All lines must start with //

There are a number of default dialogs which the user can be prompted with during an installation. Normally, the 'Welcome' dialog and the 'Installation Path' dialog are used. The text shown in these dialogs will be discussed later on this page.

The default disk label prefix should be sufficient. The script file name must be SETUP.INF for automated installations.

Set the Default Installation Path to the directory of where your software is to be installed. This will appear as the default in the 'Installation Path' dialog at install time.

The reserve space default and split files size should be sufficient. Refer to the help on the Project Attributes dialog for more information on these.

The Licensing .INI file is used if you have selected to display the 'Licensing Dialog'. This dialog requests the user to enter their name and company name and stores them in the .INI file

specified in this field. By default the field contains %Application%.INI so be sure to set this field or keep the Application Name to 8 or less characters.

Press **Ok** to save the project attributes information.

### **Setting Dialog Text**

An installation procedure contains a number of standard dialogs throughout its progress. It is possible to set the text displayed in these dialogs by selecting the Project/Dialog Text menu option to obtain the Dialog Text dialog.

You should find the defaults sufficient, but if not you can change them. Note that you can embed %Application% in the text to refer to the application name or %InstallPath% for the installation directory.

Multiple lines can be specified by inserting | (vertical bar) characters.

Press **Ok** to save the project dialog text.

### **Creating the Disks**

If you have followed this page from the top, you should now be ready to create your installation diskettes.

Insert a formatted diskette into your diskette drive and press the 'Build' toolbar button or select the Project/Build menu option.

If you have marked some of your files as optional, the Optional Files dialog will appear to enable you to select the files to be placed on your diskette. Press **Ok** to continue or **Cancel** to stop at this stage.

The Build Project dialog will then appear to ask you which drive/directory to create your installation procedure. Enter the required location and press **Ok** to continue.

Press **Ok** again to continue.

Setup Builder will then proceed to 'clean' the requested directory of files and create your installation suite within it. It will automatically ask you for more diskettes if a diskette becomes full. Large files will automatically be split across multiple diskettes if required and files will also be compressed if this option has been selected. Note that compression requires the Microsoft COMPRESS.EXE/.TXT program to be in a directory specified by your PATH statement.

Once all files have been copied, Setup Builder automatically creates an install script and may ask you for the first disk again since this file is always placed on the first diskette. (If the script file will not fit, you need to increase the reserve space mentioned above).

Setup Builder automatically places all the component files it needs for its own use on your diskettes, so you don't need to worry about this.

### **Afterwards...**

Once you have created your diskettes and install suite, it is strongly recommended that you test it on a variety of machines, ideally ones that don't have your software or any of its components already installed. You can test the installation on your machine by pressing the 'Run' toolbar button in Setup Builder, or alternatively, the 'Install' icon in the Setup Builder Program Manager group.

It is suggested that you diskcopy the diskettes created by Setup Builder in order to duplicate them.

It is possible to copy install diskettes into one directory on a hard disk or network provided there are no compressed or split files present. The install will operate as normal from a hard disk and it recognises that it doesn't need to ask for extra diskettes! This is possible because diskettes are not labelled using DOS labels. Each diskette contains a label file instead.

This concludes the Quick Start to creating an Install Disk Suite. By now you should have a good understanding of how to create an installation procedure using Setup Builder.

If you have any further questions, please don't hesitate to [contact the author](#).

## Shared File Registry

Often during an installation of some software, it is common to place files in the \WINDOWS or \WINDOWS\SYSTEM directory. Sometimes, the files placed in these directories are used by more than one application, for example, .DLL files. At some point, it is likely that one or more of the applications sharing these common files will be de-installed, causing a potential problem with files that have been installed for shared use, especially if the shared files are deleted.

To resolve the problem of shared files being de-installed, Setup Builder supports a 'shared file registry' system. In simple terms, the registry is just a list of files with a count of the number of applications using them. As applications are installed, the counts are increased and when applications are de-installed, the counts are decreased. When counts reach zero, files are deleted.

The registry itself is simply the REGISTRY.INI file in your \WINDOWS directory. The file contains a list of all the files which have been registered as shared, together with a count of the number of applications which are currently installed which use the files.

When a file is registered, it may already exist in the registry (another application is already using it), in which case its count is simply increased by one. If it isn't already registered, the file is added to the registry with a count of 1. Before a file is registered, the Register function checks to see if the file exists. In this way, files which have already been installed by other applications which are not in the registry can be accounted for in the registry, so it is possible that if you Register a file, its count may start at 2 if the file already existed.

To integrate with this facility, Setup Builder creates script code which registers a file before it is actually installed/copied.

The Setup Builder de-install facility integrates with the shared file registry.

When a piece of software is de-installed, files are usually simply deleted, however, files which have been registered in the registry must only be deleted if their share count reaches zero.

Therefore, shared files must be removed with the **UnRegister** function and not the **Delete** function. UnRegister retrieves the share count for a file and decreases it, writing it back to the registry. The file is not deleted until the share count reaches zero (ie all applications using the file have been de-installed) and then the registry entry for the file is removed - the file is no longer registered as being shared.

The registry functions do not support wild card file names.

### Example

```
Register("C:\WINDOWS\SYSTEM\CTL3DV2.DLL")
```

```
' Install the file/overwrite check etc
```

```
STOP
```

```
' De-Install
```

```
UnRegister("C:\WINDOWS\SYSTEM\CTL3DV2.DLL")
```

## Sample Projects/Files

Setup Builder has been supplied with some sample files demonstrating many of the features supported by the product.

It is suggested that you view both the project files and the script files in order to gain a greater understanding of how Setup Builder operates.

The following samples are supplied:

- SETUP.SPJ
- SAMPLE1.SPJ

### **SETUP.SPJ**

This project is supplied in order that you can create more Setup Builder installation disks to pass on to friends/relatives/colleagues etc.

When you build the project, it will create a diskette with files on which is the same as that you first used to install Setup Builder.

### **SAMPLE1.SPJ**

This sample project is supplied to demonstrate many of the features supported by Setup Builder. It is a ready made example which can be compiled immediately (assuming you installed Setup Builder in the default C:\SETUP directory).

The project is designed to create an installation diskette containing your AUTOEXEC.BAT, CONFIG.SYS, and the Notepad, Cardfile and Calculator programs supplied with windows. When the install suite from this project is run, the files will all be installed in the C:\TEST directory. Various options have been set for the different files:

#### **AUTOEXEC.BAT**

This is a standard installation of a file.

#### **CONFIG.SYS**

This file has the 'Confirm Overwrite' option set which means that if the file already exists when you are installing, a message will ask for confirmation of overwriting the existing file with the new file.

#### **NOTEPAD.EXE**

This file has the 'Confirm Usage' attribute set. This means that if the notepad.exe file is already running under Windows at installation time, the installation will stop and the file will not be installed.

The file is also an optionally included file, meaning that you can select whether to include it on your installation diskette at 'Build' time.

It will also be installed in a new program manager group with a parameter file C:\AUTOEXEC.BAT. Its icon will be set to the fourth icon in progman.exe. (Icons are zero subscripted in executeables, so the first is zero, the second one and so on)

The working directory will be set to C:\TEMP

#### **CARDFILE.EXE**

This is a standard installation file which has an icon created on Program Manager.

#### **CALC.EXE**

This is a standard installation file which has an icon created on Program Manager. The icon is the first (zero) in CALC.EXE. This is in fact the default icon, so the icon file could in fact be omitted in this instance.

## **Backdrop Display**

The SAMPLE1.SPJ project has the backdrop display configured as follows:

The bitmap displayed will be as follows:

- 640 x 480 screen resolution: arches.bmp, centred on screen
- 800 x 600 resolution: no backdrop at all
- 1024 x 768 resolution: default shaded blue backdrop
- Default resolution: leaves.bmp stretched to fit the screen. This will appear if the screen resolution is one other than the three above

All four backdrop text items have been configured to demonstrate their appearance.

### **Program Manager Group**

This has been configured to create a group called %Application%. At run time, the %Application% value represents a program variable and its value is that placed in the 'Project Attributes' dialog 'Application Name' field.

### **Project Attributes**

The application name has been set up. This value is placed in the %Application% variable at run time and may be placed in any string for substitution according to the standards of the Setup Script language.

The comments specified will appear at the beginning of the script file.

The 'Welcome', 'Installation Path', 'Licensing' and 'De-Installation' dialogs have all been configured to appear.

The project uses the default Disk/Script file information, the only difference being the 'Default Installation Path'. This is the path name the user will be prompted with in the 'Installation Path' dialog as a default location to install the software.

### **Project Dialog Text**

This has all been left as the defaults.

### **User Code**

This has been configured to demonstrate a number of features.

The initialisation code retrieves the location of where the software was installed ready for the De-Installation option.

The Pre-File copying code has been set up to save the location of where the software has been installed for later use by the De-Install option. It also writes the version number of the application for use by potential future installation procedures and creates a sub-directory of the installation directory for use by the application, although for this demonstration, no files are actually placed in the directory.

Post Program Manager Group creation has been set up to manually create another Icon.

Installation Completion has been set up to ask you if you wish to view your AUTOEXEC.BAT file and shows it using notepad if the user selects 'Yes'. Normally, as in the Setup Builder installation procedure, this feature would be used to display installation notes etc.

The Application De-Installation code has been set up to remove the application from your hard disk. Setup Builder does not automatically produce code to do this because it cannot know what files to de-install especially as some may or may not have been installed given the flexibility of the installation procedure.

It is suggested that you 'Build' this project and view the SETUP.INF file produced to gain a greater understanding of how Setup Builder operates.

### **SAMPLE2.INF/.INI**

This script file demonstrates how INST.EXE, the interpreter module supplied with Setup Builder,

can be used to write small 'programs'.

You can add SAMPLE2.INF on to the Program Manager StartUp group. It allows you to choose which programs you want to run when you first start up Windows instead of running all of them as would normally occur.

SAMPLE2.INF contains comments/information on how it should be used. You can view the file using any ASCII file editor, such as notepad, for example.

### **DIR.INF**

This script file demonstrates setting up a user defined dialog which will present the user with several fields, asking the path names in which to store the individual components of an application.

### **OPTIONS.INF**

This script file demonstrates setting up a user defined dialog which effectively provides a 'menu' for selecting different facilities from a setup procedure. It presents the user with several radio buttons from which a selection is made and the Ok button is pressed to activate the selection.



## Select Script File Dialog

### Purpose

The **Select Script File Dialog** enables the user to browse for a script file to be encrypted or un-encrypted.

### Available from

- **Browse** button on the [Encrypt/Unencrypt Script File Dialog](#)

### Comments

Select the drive/directory by clicking on the appropriate fields. Enter the name of the required script file and press 'Ok' to select it.

## **Failed to create directory Error**

This error occurs when Setup Builder has failed to create the directory which was specified for the location to create the new installation procedure.

### **Possible causes**

An invalid directory name has been specified

An invalid drive letter was specified

An attempt has been made to create an installation procedure on a disk which is write-protected. Setup Builder MUST have read-write access to the target diskette

## Technical Notes

### File locking/sharing

It is advisable to install SHARE.EXE to maintain file locking on project files on your local drive(s). Network drives are normally configured for this anyway.

### Windows Version

Setup and Setup Builder will only operate under Windows 3.10

### How the Setup Procedure operates

When a Setup procedure is run, the user executes SETUP.EXE, usually from a diskette. This program copies the files INST.EXE, SETUP.INF and DIBAPI.DLL to your WINDOWS\TEMP directory (which is created if it does not exist) which must have write access.

The INST.EXE program is then run on your hard drive by the SETUP.EXE program and passed the SETUP.INF file name as a parameter script file to be interpreted (run). This is because diskettes may be changed during the installation process.

SETUP.EXE only recognises SETUP.INF which is why your script file should always be named SETUP.INF for an installation diskette suite.

However, if you create script files to be run from your hard disk via the File Manager Association with .INF files, the script file may have any name since INST.EXE will accept any file name as a parameter script file to be run.

At the end of running a script, the CLEANUP.EXE program is run from the last disk. This removes the setup procedure and all of its components from your hard disk, leaving it tidy.

### Setup Script diskette space

When Setup Builder creates the first diskette of an installation suite, it reserves a portion of the disk by creating a dummy script file. As the build proceeds, files are copied on to subsequent diskettes as required. At the end, Setup Builder creates a temporary copy of the Setup script file in the same directory as the project (.SPJ) file and then requests the first diskette again in order to copy the Script file onto it, overwriting the dummy file. This process is used because the script cannot be created until Setup Builder knows which disk each file has been copied to. This will not be until all files have been copied.

The amount of space reserved by default is 24K. Copying of the script file to the diskette may cause a diskette space error if there has not been enough diskette space reserved for the script file.

In this situation, you need to change the reserved space. To do this, check the size of the temporary copy of the script .INF file which is created in the same directory as the .SPJ project file (using File Manager or similar product) and adjust the 'Reserve Space' field in the Project Attributes Dialog to show a number which represents the number of K Bytes you wish to reserve (default is 24, reserving 24576 bytes = 24K)

### Program Manager Group file protection

Setup Builder offers the ability to set the write protection flag of a program manager group file to stop a user from changing the group or deleting items from it.

The default for group creation is to create a new group with same name as that of the application.

Protection of the group file uses the 'Group Caption/File Name' field in the Program Manager Group Dialog to obtain the name of the file to adjust. If this is the default (%Application%) and 'Make Group' is selected, then the name of the file obtained will be the value in the 'Application Name' field of the Project Attributes Dialog. In this case, the application name must be one word and less than 8 characters long otherwise group file protection may not work.

Similarly, when Program Manager creates groups given a caption, it often appends a numeric to the caption to obtain a name which is different from an existing name. This means that Setup cannot guarantee successful implementation of group file protection when the group is not created from an existing file.

For best results where group file protection is required it is best to create the group from a file using 'Use existing group file' in the Program Manager Group Dialog and placing the actual file name of the group file in the 'Group Caption/File Name' field. The group file itself may be installed

as part of your installation process either into your own directory (best) or the Windows directory (risk of overwriting something with the same name).

**Backdrop Bitmaps**

Backdrop bitmap files are copied by the script into the WINDOWS\TEMP directory when the script first runs so that they may be used at any time during an install script. The script automatically deletes them just before it finishes.

## Test Project Dialog

### Purpose

The **Test Project Dialog** enables you to select the disk drive and directory from which you wish to run a setup disk suite in order to test it.

### Available from

- Project/Test menu option
- Toolbar button

### Comments

The default location is that which was used for the last build process. Enter a new location if required.

Press the 'Ok' button to select the drive/directory and proceed to run the setup process, or 'Cancel' to exit without running the process

### Notes

If a single drive specification ( eg a:) is entered, the directory used will be the root (and not the current directory on a:) since a '\' character is always appended to the directory entered by the user. If you want another directory, then it must be specified in full.

## **Too Many Files Error**

Setup Builder has a limit of 25 files per project for the Shareware Edition and 1024 files per project for the Professional Edition.

This error occurs when the user has attempted to exceed this limit by adding more files.

## **Unrecognised Project ID Error**

Setup Builder project files have a file identifier encoded in them.

An attempt has been made to open a project file which has an unrecognised file identifier. This means that Setup Builder does not recognise the selected file as a Setup Builder project file. You should select another file or if the file really was a project file, it has probably become corrupted, in which case you should recreate the file.

## **Unrecognised Project Version Error**

Setup Builder project files have a file version number encoded in them in order that future versions of Setup Builder can recognise files for upgrade/conversion purposes.

An attempt has been made to open a project file which has an unrecognised version number. Setup Builder should only give this error when an attempt is made to open a file which is not a project file, but just happens to have the correct file identifier - a rare situation. Setup Builder recognises all previous file versions and automatically converts them to the latest format for the program release when they are saved. This will make files non-backward compatible.



## User Code Dialog

### Purpose

The **User Code Dialog** enables the user to manually enter specific code at various stages of the installation process.

### Available from

- Project/User Code menu option
- Ninth toolbar button

### Comments

You should refer to the Setup Script Help to see the commands and standards used.

The User Code dialog enables user-defined code to be entered at several stages of the installation procedure.

Select the required code from the drop down list at the top of the dialog:

### Initialisation code

This code is executed immediately at the start of a script, just after the default variables have been set up. You might want to use this to determine whether your application has been previously installed and exit so that it is not installed again. You may even require a different installation procedure to be executed in this event.

### Pre file copying code

This code is executed before any files are copied (in the gauge). You might wish to use this to delete any unwanted files for example.

### Post file copying code

This code is executed after files have been copied (in the gauge). You may wish to use this to ask the user questions (via the `MessageBox` function) and to manually copy any files previously specified as 'Do not copy file'.

### Post PM group creation

This code is executed after Program Manager group creation. This would normally be used if you wanted to create icons for which no files have been copied (eg `NOTEPAD.EXE`). In this instance it is likely that none of the project files would have 'Make Icon' selected.

### Installation Completion

This code is executed right at the end of a Setup script after all copying and installation and after the 'Installation Successful' message box.

This section can be used to provide features such as starting `NOTEPAD.EXE` to view a `README.TXT` file after installation or even starting the application just installed.

No code is executed after this code in the script file.

### Application de-installation

This code is only included in a setup script if the 'de-installation' option in the [Project Attributes dialog](#) is checked.

The de-installation code created by Setup Builder asks the user to confirm de-installation. It then deletes all files which were installed by the install procedure. The user-defined code for de-installation is then executed, followed by the automatic deletion of the application directory, `.INI` file and Program Manager group. A message then appears to advise the user that de-installation has been completed.

Under most situations, you should not require any user code to de-install an application, however if you create sub-directories of the application directory or any extra `.INI` files, then these must be deleted by manual code. Setup Builder will not remove directories other than the main application directory and even this won't be removed if it is not empty or there are further sub-directories. Setup Builder will ensure that files installed in sub-directories are deleted provided it

installed them in the first place. If not, you must delete them with manual script code.

**:USER1-6**

This code is placed towards the end of the setup script and may be branched to by any of the other user code sections using a GOTO script statement, for example:

```
GOTO :USER1
```

These user code sections are ideal for placing script for different types of installations. At the end of one of these user sections, you should place a GOTO :END statement, otherwise the running of your script will pass into the next user code section - which you may actually want it to do.

It is recommended that you place some text in all the user code sections, build your setup disk suite and then view the .INF script file to understand the positioning of the user code sections.

Select some text and press 'Cut' to cut the text to the Windows Clipboard.  
Select some text and press 'Copy' to copy the text to the Windows Clipboard.  
Press 'Paste' to paste text from the clipboard into the user code.

Press 'Ok' to save any changes, or 'Cancel' to exit without saving any changes to any of the user code sections.

## Disk Required

Setup Builder requires a new disk to be inserted in your disk drive. This may be because it requires the first disk of an installation suite or because it has copied as many files onto the current installation disk as possible and there is now no more space on it, so another diskette is required.

Insert the diskette with the number requested into the drive and press 'Ok' to continue building the installation disk suite or 'Cancel' to stop.

### NOTES:

Upon copying all files onto your installation diskettes, Setup Builder may ask for the first diskette (disk #1) again. This is because the script file is always placed on the first diskette of an installation suite. Please also see [Technical Notes](#) **Setup Script Diskette Space** for more information.

**As part of its disk 'tidy' procedure, Setup Builder deletes all files in the specified directory of new diskettes inserted during the project building procedure.**

Setup Builder does not format diskettes for you so you should ensure that all installation suite diskettes are pre-formatted.

Setup Builder will **not** split large files across multiple diskettes. Therefore, if a file is larger than your largest diskette, Setup Builder may keep on asking for new disks and never place anything on them. This is a limitation of Setup Builder.

## **What is Setup Builder ?**

Setup Builder is a utility program for creating Windows Hosted setup procedures for installing software under Microsoft Windows.

## **Contacting the Author**

Graham Plowman can be contacted by post at the following addresses:

PO Box 1124, Manly 2095, NSW, Australia

or

'Fiddlers Rest', Gaveston Hall Drive, Nuthurst, Horsham, West Sussex, RH13 6RG, England

All post sent to the British address will be forwarded to Graham at the Australian address, so please allow for a slight delay.

Graham can also be contacted on:

Compuserve ID: 100105,536

Internet: 100105.536@compuserve.com

Internet: gplowman@ozemail.com.au

## Integrity Checking

This application has an automatic self-checking feature to advise of corruption to the executable file.

The main purpose of this is to enable the program to identify whether it has been interfered with, either by a binary file editor or a computer virus.

Note that attempts to change icons and messages etc using products such as AppStudio or Resource Workshop will render this application unuseable since they will cause a failure of the check.

If the application executable is corrupted in any way, the following message will appear:

Integrity Check Violation Error: Integrity has been violated!

If this message does appear, you should re-install this application after checking the reason for it occurring.

Note that all of the author's software is Virus Checked before it is distributed.

Other integrity-related messages are possible. If you need assistance, please [contact the author](#).



GRAHAM PLOWMAN SOFTWARE

## Price List for Software

Graham Plowman, PO Box 1124, Manly 2095, NSW, AUSTRALIA or  
'Fiddlers Rest', Gaveston Hall Drive, Nuthurst, Horsham, West Sussex, RH13 6RG, ENGLAND

Prod No	Product	Price AU\$	Price UK£	CIS Price US\$	CIS Shipping US\$	CIS Total US\$	CIS SWREG ID
#00	HelpBuilder	\$80.00	£40.00	\$64.00	\$10.00	\$74.00	#4152
	HelpBuilder Upgrade	\$40.00	£20.00	\$32.00	\$5.00	\$37.00	#5193
	HelpBuilder User Manual	\$15.00	£7.00	\$10.00	\$8.00	\$18.00	#5401
#01	Setup Builder	\$80.00	£40.00	\$64.00	\$10.00	\$74.00	#4153
	Setup Builder Upgrade	\$40.00	£20.00	\$32.00	\$5.00	\$37.00	#5194
#03	HelpBuilder SpellChecker	\$50.00	£25.00	\$40.00	\$6.00	\$46.00	#10705
#04	SpellChecker (Developer)	\$80.00	£40.00	\$64.00	\$10.00	\$74.00	#10521
	Distribution License (per copy)	\$30.00	£15.00	\$24.00	\$4.00	\$28.00	#11064
	Postage, packing & handling (orders received by post)						
	Per disk (in Australia)	\$5.00					
	Per disk (outside Australia)	\$10.00	£5.00				
	Per manual	\$10.00	£5.00				
	Any number of user license numbers	\$5.00	£2.50				

All mail orders must be paid by cheque or bankers draft, made out to G.Plowman either in Australian Dollars (please post to Australian address) and drawn on an Australian bank or in UK Pounds Sterling (please post to UK address) drawn on a British bank. Credit card payments cannot be accepted.

Postage is only applicable to mail orders. Payments/orders made by post will receive a disk containing the requested software. Unless you state otherwise, it will be assumed that 3 1/2 inch disks are to be sent. The postage charge is applicable per disk ordered. Where a postal order is only for user license numbers (ie no disk required), the user license number postage charge applies for the whole order.

Payments may also be made via the Compuserve SWREG (GO SWREG) facility (Register Shareware option). SWREG will debit a payment from your Compuserve account and credit it to the author's Compuserve account.

If payment is made via Compuserve you will receive an acknowledgement via E-Mail, a license number and unlimited time usage.

Payment must be made in full before goods are delivered. Registrations cannot be made over the phone with a cheque to follow.

All registrations receive free technical support.

If you register by post and you have an E-Mail address (either Internet or Compuserve), please advise the author of that E-Mail address to enable easier and quicker notifications of upgrades and new versions.

Please note that if you obtained your copy of this software from a Shareware library and paid a fee at the same time, this fee is NOT a registration fee, it is purely a charge by the Shareware library to cover their costs in producing the diskette(s) for you, plus of course their profit. To register your copy you must send the appropriate registration fee to the above address.

It is a condition of registration that you accept that you have fully tried and tested this software

and found that it meets your requirements to your satisfaction. Please note that once a license number has been given, no refund will be possible.

Prices are correct as at 14/4/96. All prices and specifications are subject to change without notice.  
E.& O.E.



## Products Available from the Same Author

The following products are all available from the same author:

### Help Builder

HelpBuilder is a Microsoft Windows 3.1 application for creating and maintaining Windows Help files and is designed for developers wishing to create Windows Help files to distribute with their products and for anyone wishing to create stand-alone Windows Help files.

This help file was created using Help Builder.

The application is not an 'add-on' to a commercially available word processor, rather, a Multiple Document Interface (MDI) application integrated with the Microsoft HC31.EXE help compiler (or equivalent compatible eg HCP.EXE)(HelpBuilder is configureable). Since it is likely that anyone using this product is likely to be an applications developer using Visual Basic, Microsoft C/C++ or Borland C++ they will already have HC31.EXE as it is supplied with these products.

HelpBuilder does not 'shell' out to DOS to compile files but provides a complete integrated Windows hosted compilation process.

### Windows Help and Help Compiler Features Supported

- Topic pages
- Links and 'popup' windows to other topics, including optional 'picture hotspots', macros and 'windows'
- Browse sequence support
- Ability to place bitmaps in help files
- Fonts, colours, bold and italic text
- Header files
- Context identifiers
- Locking of and changing colour of top of Windows Help page
- Help file start up macros
- Topic start up macros
- Changing of 'locked' and client 'area' colours in Windows Help
- Compiler configuration - warning levels, CD ROM, compression etc
- Configuration of 'windows'
- Build tags
- Optional add-on spelling checker

### Product Features

- Full keyboard support
- Projects to group individual topic files for a specific application
- Topics held in separate files in order that they may be shared between projects
- Toolbar selection of many facilities and status bars
- Error checking on project building
- Self-checking executable - to detect corruption/tampering eg Virus
- Project Wizard to create a basic project
- Application Installation and De-installation procedure
- Printing of topics and projects
- Fully integrated and context sensitive Windows Help file including 'How to..' section
- Can create Help files for VB, C/C++, PowerBuilder etc - any language supporting Windows Help and passing of context numbers
- Ability to insert RTF commands
- Network locking on project and topic files
- Automatic creation of .HPJ and .RTF files

- Sample demonstration help file project
- Multi-tasking during compilation - no 'DOS' windows!
- Helpfiles 'testable' from within HelpBuilder
- Configurable compiler - via application .INI file
- Topic locator
- Importing of Norton Guides help projects
- In-built text macros

## HelpBuilder SpellChecker

HelpBuilder SpellChecker is an addon Dynamic Link Library for HelpBuilder which provides spell checking of help topics. It requires HelpBuilder version 1.09.001 or greater.

### HelpBuilder SpellChecker Features

- Small, self contained .DLL, requiring no third party products
- Supports dictionaries in different languages (ie different dictionary files)
- Words can be added to dictionary (extendable dictionary)
- Ignore word option during spell checking
- Ignore all occurrences of a word option during spell checking
- Prompts with alternative spellings and suggestions
- Online help

## Setup Builder

Setup Builder is a Windows 3.1 application for creating and maintaining Windows Hosted installation procedures and is aimed at developers who wish to provide that 'professional' touch to the installation of their software.

The installation procedure for this piece of software was created using Setup Builder.

The interpreter part of the product can also be used to run Windows Hosted DOS-like batch files using the script language supported.

Forget about distributing Microsoft's 0.5 Mb offering just to install a 100-200K executable! This product uses less than 300K of diskette space and is highly tailorable and doesn't require the user to get involved with C/C++, DLL's or Visual Basic in order to create install disks.

Setup Builder is not 'tied' into any one development tool: it can be used with any development tool you may choose.

### Setup Builder Application Features

- Setup projects to which files can be added/removed
- Automatic creation of installation disks, requesting extra disks as required
- Automatic splitting of large files across diskettes (including compressed files)
- Automatic creation of install script code
- Automatic creation of de-install script code
- Ability for user to place 'user-defined' code in install procedure if required
- Windows hosted with toolbar and status bar
- All files can have attributes set: target location, overwrite checks, in use checks, file compression, create program manager icons etc
- Ability to create program manager groups and icons
- Up to 1024 files in an installation (Professional Edition)

- Optional encryption of install scripts (so that users can't change them)
- Context sensitive, comprehensive Windows Help file
- Network locking on project files
- Visual Basic .MAK project import facility
- 3rd party product import facility eg VB run-time, Access etc (Configurable)
- Ability to change text on all dialogs used in an installation (also allows different languages - defaults to English, French, German, Italian or Spanish)

**Interpreter Supported Features** (Interpreter is placed on install disks to 'run' install procedures)

- Script language (similar to DOS batch files) supporting if and goto statements, string and numeric variables
- Windows extensions - predefined dialogs (user definable text), message boxes, creation of program manager icons/groups and ability to delete them
- User defined dialogs with controls
- Comprehensive Windows Help file
- String and numeric manipulation (left, right, mid, +, -, \*, /)
- Copying of files
- Backdrop bitmaps during installation
- Optional fuel gauge during copying
- Date and time functions
- DOS file management - create/delete/renaming of directories, disk space and file presence/overwrite/version checks
- Windows .INI file management
- File management - create/delete/renaming/copying/date & time setting, reading and writing of ASCII files
- Nested script files - call a script from another script
- External calls to run other Windows applications
- Shared file registry (stops un-installing of files shared by multiple applications)
- English, French, German, Italian and Spanish language support on dialogs

## SpellChecker

The Developer Edition of SpellChecker is a Windows 3.1 addon Dynamic Link Library which can be built into your applications to provide a spell checking facility which would otherwise normally require the use of DDE/OLE communications with products such as Microsoft Word. SpellChecker is an integrated .DLL, which is compact in size and is easy to build into your applications quickly. It is supplied complete with a help file which documents how to use and program it. SpellChecker may be built into any application constructed using a development tool which supports .DLL API calls such as C/C++, Visual Basic etc

### SpellChecker Features

- Small, self contained .DLL, requiring no third party products
- Simple to build in to applications via .DLL API calls (documented).
- Supports dictionaries in different languages (ie different dictionary files)
- Words can be added to dictionary (extendable dictionary)
- Ignore word option during spell checking
- Ignore all occurrences of a word option during spell checking
- Prompts with alternative spellings and suggestions
- API call to spell check a Windows Edit field
- API call to spell check a string buffer

- API call to spell check an ASCII file
- API call to spell check an individual word
- Online help - runtime (separate file)
- Online help - development/programming information (separate file)

## **Assistance/Further Information**

If you require any assistance or further information on any of the above products, please contact the author at the address in the [Contacting the Author](#) section of this help file.

## **What is Shareware ?**

### **Definition of Shareware**

Shareware distribution gives users a chance to try software before buying it. If you try a Shareware program and continue using it, you are expected to register. Individual programs differ on details: some request registration while others require it, some specify a maximum trial period. With registration, you get anything from the simple right to continue using the software to an updated program with printed manual.

Copyright laws apply to both Shareware and commercial software, and the copyright holder retains all rights. Shareware authors are accomplished programmers, just like commercial authors, and the programs are of comparable quality. (In both cases, there are good programs and bad ones!) The main difference is in the method of distribution. The author specifically grants the right to copy and distribute the software, either to all and sundry or to a specific group. For example, some authors require written permission before a commercial disk vendor may copy their Shareware.

Shareware is a distribution method, not a type of software. You should find software that suits your needs and pocketbook, whether it's commercial or Shareware. The Shareware system makes fitting your needs easier, because you can try before you buy. And because the overhead is low, prices are low also. Shareware has the ultimate money-back guarantee: if you don't use the product or it doesn't do what you want it to do or you don't like it, then you don't pay for it.

### **Reasons for Registering**

- All the 'unregistered copy' messages will disappear
- All the 'limiting' features of the Shareware version will be removed
- You will receive notifications when the next major versions become available
- You will receive free technical support
- You will automatically become a registered user of future versions with no more fees to pay (subject to major versions)

### **Grant**

Graham Plowman grants you a non-exclusive license to use this software free of charge for a one month evaluation period after which you must register (pay) for a copy of this software if you wish to continue to use it. Upon registration you are granted a license to use this software for an unlimited time period.

You may not modify, translate, reverse engineer, decompile, disassemble or create derivative works based on this software except where this documentation specifically states otherwise.

### **Title**

Title, ownership rights and intellectual property rights in and to this software shall remain in Graham Plowman. This software is protected by international copyright treaties. Your license does not give you any rights to title, ownership or copyright.

### **Disclaimer of Warranty**

Users of this software must accept this disclaimer of warranty:

"This software is supplied 'as is' without warranty of any kind. The author disclaims all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The entire risk as to the quality and performance of this software is borne by you.

The author assumes no liability for damages, direct or consequential, which may result from the use of this software."

## **Limitation of Liability**

UNDER NO CIRCUMSTANCES AND UNDER NO LEGAL THEORY, TORT, CONTRACT OR OTHERWISE, SHALL GRAHAM PLOWMAN BE LIABLE TO YOU OR ANY OTHER PERSON FOR ANY INDIRECT, SPECIAL OR INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF GOODWILL, WORK STOPPAGE, COMPUTER FAILURE OR MALFUNCTION, OR ANY AND ALL OTHER COMMERCIAL DAMAGES OR LOSSES. IN NO EVENT WILL GRAHAM PLOWMAN BE LIABLE FOR DAMAGES IN EXCESS OF THE LIST PRICE OF A LICENSE FOR THIS SOFTWARE, EVEN IF GRAHAM PLOWMAN SHALL HAVE BEEN INFORMED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY.

## **Termination**

Your license will terminate automatically if you fail to comply with the limitations described. On termination you must destroy all copies of the software.

## **Miscellaneous**

This software is a "shareware program" and is provided at no charge to the user for evaluation. Feel free to share it with your friends, but please do not give it away altered or as part of another system. The idea of "user-supported" software is to provide personal computer users with quality software without high prices, and yet to provide incentive for programmers to continue to develop new products. If you find this program useful and you wish to continue to use it after a reasonable trial period, you must make a registration payment. The registration fee will license one copy for use on any one computer at any one time. You must treat this software just like a book. An example is that this software may be used by any number of people and may be freely moved from one computer location to another, so long as there is no possibility of it being used at one location while it's being used at another.

Just as a book cannot be read by two different people at the same time.

Commercial Users should register as soon as possible and not let the one month expiry feature stop the program and cause problems with meeting project deadlines: 'urgent' registrations are still subject to the speed of E-Mail systems and the postal system - please don't leave registration to the last minute. You are not allowed to distribute the output of the Shareware Edition of this software for financial gain. Various limiting features have been built into the Shareware Edition. Site-License arrangements can be made by contacting the author.

Anyone distributing this software for any kind of remuneration must first contact the author. In this way the distributor can be kept up-to-date with the latest version of this software.

You are encouraged to pass a copy of this software to your friends for evaluation. Please encourage them to register their copy if they find that they can use it. All registered users will receive a copy of the latest version of this program, a license number, free technical support and notifications of upgrades.

It is a condition of registration that you accept that you have fully tried and tested this software and found that it meets your requirements to your satisfaction. Please note that once a license

number has been given, no refund will be possible.

Graham Plowman

## **Further Technical Information**

Further technical information about GP Software products may be obtained from the GP Software Technical Support help file which may be found in the Compuserve WinShare forum. Simply GO WINSHARE and select 'Library Search'. Enter 100105,536 as the 'Contributor' and then press 'Search'. Click on the required item(s) and press 'Retrieve' to download them.

The GP Software Technical Support help file is a file containing technical information (Technical Reports - TR's) about GP Software products which is not supplied in the help files supplied with the products.

It supplies information on technical issues and how to resolve them as well as reported faults, their current state of resolution and work-arounds etc.

There is also technical information about how various parts of the applications work.

This file will be continually updated as the products are developed and as information becomes available.

Please note that as faults are identified and fixed, the fixes will automatically appear in future versions of the software, normally the next release.





