

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

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 in order to determine the expiry date of a review (unregistered) copy of the software.

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 reinstall the software the next day. Better still, obtain the latest version of the software since this bug has now been fixed.

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



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
-  Setup Script Help

-  Other products available
-  What is Shareware ? - Description and Disclaimer
-  Contacting the Author

-  Help File Copyright

File Copy Errors

One of the following file copying errors may occur:

General copying error

This is a general file copying error and usually occurs when a read or write to a diskette has failed.

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.

Unable to create target

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.

No space on target

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.

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 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.

This Windows Help file was written by Graham Plowman using HelpBuilder Version 1.07 and refers to:

Setup Builder Version 3.07 / 9-4-1995

Copyright 1993, 1994, 1995 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 an unregistered shareware version of Setup Builder is being used, the number of files allowed in a project will be limited to 25.

Registered versions will allow up to 500 files.

Setup Builder Errors

Unable to create project file

Unable to open selected project file

Unrecognised project (.PRJ) file

Unrecognised project (.PRJ) file version

Too many files selected for project

File copying errors

Create Dependencies

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.

Available from

- File/Attributes menu option
- 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.

'Check if in use' checks with windows 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.

'Confirm overwriting' checks to see if the file already exists (eg a .INI file) and asks the user to confirm overwriting it. If this is selected, you should complete the 'Overwriting message' field.

'Check if in use & confirm overwriting' provides a combination of the two above options.

'Display in gauge' is the default. This causes the file to appear in the gauge during the copying process. Wherever possible, all files should have this option in order that the user receives maximum feedback of progress of the installation process.

Overwrite Message

If you require a specific message for an overwrite confirmation, complete this field. The default should be adequate.

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

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 allocatable 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.

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.

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.

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.

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

Invalid directory Error

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

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 four columns which have entries and meanings as follows:

- | | |
|----|--|
| 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. |
| CO | 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. |

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
- 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.

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.

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](#)

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.

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.

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

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

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.

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 does not create any code to actually de-install your application - you must code this in the User Code Dialog section 'Application De-Installation'

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.

Licensing .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 Licensing .INI file field if you wish to implement licensing!

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

Project Dialogs

Purpose

The **Project Dialogs 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

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 application is to be installed

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

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 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.

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 and SETUP.INF to your Windows directory which must have write access. SETUP.INF is actually renamed in the process to INST.INF so as not to conflict with the Windows Setup configuration program.

The INST.EXE program is then run on your hard drive by the SETUP.EXE program and passed the INST.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 as a parameter the name of the script file to be run.

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 16K. 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 16, reserving 16384 bytes = 16K)

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 same).

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 unregistered shareware versions and 500 files per project for registered versions.

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

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.

User-defined code is entered here to de-install the application. This should include code to delete all files, directories and Program Manager Groups associated with your application.

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

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 address:

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

Graham can also be contacted on Compuserve ID: 100105,536

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 App Studio 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, either directly or via CompuServe ID: 100105,536.

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 aimed at developers wishing to create Windows Help files to distribute with their products.

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)(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'
- 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
- Start up macros
- Changing of 'locked' and client 'area' colours in Windows Help
- Compiler configuration - warning levels, CD ROM, compression etc

Product Features

- Full keyword 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

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 about 80K of diskette space and is highly tailorable and doesn't require the user to get involved with C/C++ or Visual Basic in order to create install disks.

Setup Builder Application Features

- Setup projects to which files can be added/removed
- Automatic creation of installation disks, requesting extra disks as required
- Automatic creation of 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 (requires MS COMPRESS.EXE), create program manager icons etc
- Ability to create program manager groups
- Up to 500 files in an installation
- Optional encryption of install scripts (so that users can't change them)
- Context sensitive, comprehensive Windows Help file
- Network locking on project files

Interpreter Supported Features (this 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
- Comprehensive Windows Help file
- String and numeric manipulation (left, right, mid, +, -, *, /)
- Copying of files
- Optional fuel gauge during copying
- Date and time functions
- DOS file management - create/delete/renaming of directories, disk space and file presence checks
- Windows .INI file management
- File management - create/delete/renaming/copying/date & time setting, reading and writing of ASCII files

Library Manager

Library Manager is a Windows 3.1 version control system for management of software source files and configuration management. It provides all the basic functionality expected of a VCS. Ease of use and simplicity were major design criteria.

The source code for this application is managed using Library Manager.

Library Manager is better than many commercially available Version Control Systems!

Library Manager Features

- MDI User interface displaying current status of all files in a project for ease of use
- Security system to which developers access rights can be configured
- Grouping of files into projects
- Unique use of databases which only records changes to files
- Centralised repository - does not 'protect' a directory like other products which can be 'by-passed' in DOS!
 - Project templates - allows grouping of files so that specific versions of an application can be recreated. Also enables files to be grouped within an application, for example, all the files belonging to a particular sub-system
 - Binary and Ascii files (configurable by file type)
 - Back versions - of both ASCII and binary files
 - File change comments
 - Reports (change comment history, headers, booked files etc) and printing
 - File viewing with no need to book/copy out
 - Network aware/locking - any number of users (subject to license)
 - Multi-tasking - so you can do other things while booking is in progress!

Assistance/Further Information

If you require any assistance or further information on any of my products, I can be contacted at the following address:

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

I can also be contacted on CompuServe ID: 100105,536

Graham Plowman

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.

Disclaimer and Agreement

Users of this software must accept this disclaimer of warranty:

"This software is supplied as is. The author disclaims all warranties, expressed or implied, including, without limitation, the warranties of merchantability and of fitness for any purpose. The author assumes no liability for damages, direct or consequential, which may result from the use of this software."

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 essence 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 of this software must register and pay for their copies within 30 days of first use or their license is withdrawn. Site-License arrangements may be made by [contacting the author](#).

Anyone distributing this software for any kind of remuneration must first [contact the author](#) for authorisation.

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 (CIS: 100105, 536)

