

TShFileOp

For DELPHI 2 and 3 or C++ BUILDER

Michel BURDIN - August 97

Did you know that there is a very simple way to copy, move delete or rename files from a Delphi or C++ Builder application?

To achieve that, the **ShFileOperation** function should be used, which is part of Windows 95 or NT4. If you are courageous, first read the Windows' API documentation. If you want to use it quickly, just put a **TShFileOp** on a form, set a few properties and call the **Execute** method...

There are many other advantages:

- During long operations, a progress window will be displayed, showing an animation where files fly from a folder to another.
- Deleted files can be sent to the recycle bin instead of being remove from disk.
- Dialog boxes are displayed every time files are about to be overwritten or deleted.
- As all dialog boxes are part of windows' environment, they are displayed in the end user's language. So, you do not have to bother with translation.

There is just one restriction:

- The component will work only with Windows 95 or NT 4. It cannot work with NT 3.5 nor with Windows 3.1 + Win32s.

TShFileOp can be used in Delphi 2, Delphi 3 or C++ Builder.

The developement of TShFileOp is based on the article "Probing into the Shell's File Operations" in "The Unofficial Newsletter of Delphi Users - Issue #19 - January 1997". Unfortunately, the example described in this article did not work very well and I had to get more information in the Windows API documentation. Also, I prefered to implement it as an easy-to-use component.

DISTRIBUTION:

This component is freeware (this does not mean "public domain"), as well as its source files. If you enhance it in any way, please send a mail to **mburdin@hol.fr** (or Compuserve 101774,626) so that I can get informed.

This component must not be sold nor used as a commercial argument. If you plan to distribute it freely, establish a link to my WEB page instead of uploading the ZIP file, because it is the best way to make sure that the most recent version is always available:

http://ourworld.compuserve.com/homepages/m_burdin

Anyway, if you prefer to upload it on a forum or a ftp, you are responsible to maintain it up to date. You must consider that it would be absurd to let old versions be available while I am spending time to fix bugs or to enhance the component.

IMPORTANT: considering that operations performed by this component will affect existing files, it can involve disastrous situations if you do not take care. Consequently, you will use it at your own risks and I will NOT be responsible for any damage. Do not install TShFileOp component if you do not agree.

INSTALLATION:

TShFileOp can be installed as any other component. First copy ShFileOp.dcu and ShFileOp.dcr in the LIB subdirectory (or any other subdirectory where you usually place components files) and make a backup copy of Cmplib32.dcl (Delphi 2). Then run Delphi and install TShFileOp from the "Install components" command in the IDE. Once the VCL has been recompiled, the component can be found in the "Samples" page of the palette.

GETTING STARTED:

A sample application is provided to explain how to use TShFileOp. The source code includes comments that explain every important operation.

The component must be dropped on a form or a data module, exactly as an OpenFileDialog component. In the object inspector, a few properties can be found, but no events. Anyway, most properties are usually set by code, except the name.

Every time you need to manipulate files, you must follow these steps:

- 1. Specify the source file(s):** every file must be added to the **FilesList** property (an integrated TStringList). The best way to achieve this is to provide a standard OpenFileDialog box allowing to select multiple files, and then Assign the Files property of the dialog box to the FilesList property of TShFileOp. Note that even if just one file is concerned, it must be added to FilesList.
- 2. Specify the destination:** this can be either a directory, or a filename. Of course, if you plan to copy or move several files together, the destination should be a directory. If the files have to be deleted, no destination is necessary. If you want to rename a file, the new filename must be specified in the **Destination** property.
- 3. Choose the action to perform** using the **Operation** property. It can take one of the following predefined values: **shCopy** (default), **shMove**, **shDelete** or **shRename**. If the operation is shDelete, you can send the files to the recycle bin by adding the **shAllowUndo** option in the **Options** property (it is by default). If it is not, the files will be physically deleted. Also, this option allows the user to undo the last operation (if possible) by typing Ctrl-Z in the desktop.
- 4. Call the Execute method:** This call will actually perform the action. If it takes more than 2 seconds, the progress window will be displayed (except if the **shSilent** option has been included). If some files are about to be overwritten, the user will be prompted for confirmation (except if the **shNoConfirmation** option has been included).

5. See what happened: Once the operation is completed, the **Execute** method returns a boolean value: True if everything was OK, False if an error occurred. Note that it returns True even if the user cancelled the operation (this is not an error). In this case, the **Aborted** property will be set to True, which allows you to take appropriate decision in the following code. Note too that the **FilesList** property is automatically cleared every time the **Execute** method terminates and whatever happened. If you need to use the same files for several operations, you must set the **KeepFilesList** property to True before calling the **Execute** method.

REFERENCE:

Aborted property (boolean, read only, runtime only) : informs if the user cancelled or not the current operation, for example by clicking on the Cancel button in a confirmation dialog box. Note that when an operation is cancelled, the **Execute** method returns True anyway.

Destination property (String) : specifies either a destination directory or a filename. When a **shCopy** or **shMove** operation concerns several files, the destination must be a directory. When you want to rename a file, the destination must be the new filename. The destination is ignored for a **shDelete** operation.

Execute method (boolean) : performs the action by calling the **ShFileOperation** API function. After the operation is completed, the method returns True if everything was OK (even if the user cancelled the operation) or False if an error occurred. The **FilesList** property is cleared (but not destroyed) whatever happened, except if the **KeepFilesList** has been set to True before calling the method.

FilesList property (TStringList) : this list must contain one or more files before the **Execute** method can be called. These files are the source files for **shCopy** or **shMove** operations or the files to be deleted or sent to the recycle bin. If the operation is **shRename**, you can only rename one file at once, so only this file will be included in the **FilesList** property. Note that **FilesList** is emptied (but not destroyed) everytime the **Execute** method terminates the operation, except if **KeepFilesList** has been set to True before calling the method. Furthermore, the **FilesList** property can be manipulated as any other **TStringList**, except being destroyed (this is performed automatically by the component). You can use the **Add**, **Move**, **Delete**, **Clear** methods or assign the contents of another list, search an item using **IndexOf**, etc.

KeepFilesList property (boolean) : set this property to True if you do not want the **Execute** method to empty the **FilesList** property every time it is called. This can be useful if you want to copy the same files several times or if you prefer to clear the list yourself after having examined the return value of the **Execute** method. This property is only efficient if it has been set before calling the **Execute** method (it can also be specified once for all).

Operation property (TShOp) : this property determines the operation to perform. It can take one of the following predefined values : **shCopy** (default), **shDelete**, **shMove** or **shRename**. **TShOp** is an enumerated type .

Options property (TshFlag) : is a set of one or more options that influence the behaviour of the Execute method:

ShAllowUndo : if included, will allow the user to undo the last operation by typing Ctrl-Z. It allows too to send deleted files to the recycle bin instead of destroying them.

ShFilesOnly : the operation will be performed only if wildchars (*.*) are specified in FilesList.

ShNoConfirmation : confirmation dialog boxes will not be displayed (the same as answering by "Yes to all").

ShNoConfirmMkDir : no confirmation is asked when a new directory needs to be created.

ShRenameOnCollision : files that already exist will be automatically renamed (" Copy of ... ") without confirmation.

ShSilent : the progress window will not be displayed.

ShSimpleProgress : the progress window is displayed, but without the filenames.

Title property (String) : this property can be used, if the ShSimpleProgress option is included, to specify the title of the progress window.

CONTACTING THE AUTHOR:

- Send me a mail to **mburdin@hol.fr** (or 101774,626 if you are a Compuserve member). I'll be happy, because I experienced that freeware authors are often ignored. Just tell me if you find the component useful, if it fits your needs, if you have any suggestion or if you noticed some bugs.
- You can consult my web page at **http://ourworld.compuserve.com/homepages/m_burdin**. It's in french, but you will find the most recent version (if any), which filename is Shfileop.zip.
- You can contact me by postal mail at the following address:

Michel BURDIN
30, rue de la fontaine Soyer
21850 SAINT-APOLLINAIRE
France

It would be nice if somebody could help me to rewrite this documentation in good english (as you can guess, this is not my native language!). Please send me this file after corrections by Email so that I can distribute it to other users.