

TShFileOp

Pour DELPHI 2 et 3 ou C++ BUILDER

Michel BURDIN - août 97

" Le saviez-vous? On peut copier, déplacer, supprimer ou renommer des fichiers d'une manière très simple dans les applications tournant dans Windows 95...

- Non?

- Si !

- Mais alors, comment faire ? "

...

En principe, on devrait utiliser la fonction **ShFileOperation** contenue dans Windows. Si vous êtes courageux, consultez la documentation de l'API de Windows. Si vous préférez l'utiliser tout de suite, placez simplement un composant **TShFileOp** sur une feuille, réglez quelques propriétés et appelez la méthode **Execute**...

C'est aussi simple que cela, mais il y a aussi de nombreux avantages:

- Pour les opérations longues, la fenêtre de progression bien connue sera affichée (vous savez, celle qui affiche une animation avec des feuilles passant d'un dossier à un autre).
- Les fichiers supprimés peuvent être envoyés dans la corbeille au lieu d'être détruits.
- Des boîtes de dialogue sont affichées à chaque fois qu'un fichier doit être écrasé, supprimé, etc.
- Comme toutes les boîtes de dialogues proviennent de Windows, il sont affichés dans la langue de l'utilisateur. Donc pas besoin de se soucier de la traduction.

Il y a juste une restriction:

- Pour le moment, ça ne fonctionnera que sous Windows 95. Si quelqu'un a l'occasion de tester sous NT 4 et si ça marche, tenez moi au courant.

TShFileOp peut être utilisé dans Delphi 2, Delphi 3 ou C++ Builder.

Le développement du composant TShFileOp est inspiré de l'article "Probing into the Shell's File Operations" dans "The Unofficial Newsletter of Delphi Users - Issue #19 - January 1997".

Malheureusement, l'exemple qui y est décrit ne fonctionne pas correctement et il m'a donc fallu consulter la documentation de l'API de Windows pour compléter mon information et implémenter tout ça sous la forme d'un composant simple à utiliser.

UTILISATION, DISTRIBUTION:

Ce composant est un freeware (attention: je n'ai pas dit "domaine public"), ainsi que ses sources. Si vous l'améliorez d'une manière ou d'une autre, tenez-moi au courant (envoyez-moi un message à **mburdin@hol.fr** ou Compuserve 101774,626).

Ce composant ne doit être ni vendu, ni utilisé comme argument commercial. Si vous voulez le distribuer gratuitement, au lieu de charger le fichier ZIP sur votre page, faites un lien vers ma page WEB où il y aura toujours la plus récente version:

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

Si vous voulez malgré tout le mettre sur un forum ou un ftp, vous avez l'entière responsabilité d'en assurer les mises à jour, sachant qu'il est absurde que d'antiques versions traînent ça et là, alors que les plus récentes existent pour corriger les bugs (sinon, ça ne sert à rien que je les fasse évoluer, c'est évident).

IMPORTANT: dans la mesure où les opérations effectuées avec ce composant affectent des fichiers existants, cela peut conduire à des situations désastreuses si vous ne prenez pas les précautions d'usage. Par conséquent, vous l'utiliserez sous votre entière responsabilité et sans mettre en cause celle de quiconque suite à des dégâts éventuels. N'installez pas le composant si vous n'êtes pas d'accord avec ce principe.

INSTALLATION:

TShFileOp s'installe comme n'importe quel composant. D'abord, copiez ShFileOp.dcu et ShFileOp.dcr dans le sous-répertoire LIB et faites un backup de Cmplib32.dcl (Delphi 2). Ensuite, installez le composant ShFileOp en utilisant la commande "Installer des composants" dans l'EDI. Une fois la VCL recompilée, le composant pourra être trouvé dans la page "Samples" de la palette.

POUR COMMENCER:

J'ai fourni une application d'exemple pour expliquer l'utilisation du composant TShFileOp. Le code source est commenté pour que vous compreniez bien chaque étape.

Le composant doit être déposé sur une feuille ou sur un module de données, exactement comme un composant OpenFileDialog. Dans l'inspecteur d'objet, vous verrez quelques propriétés et aucun évènement. De toute façon, la plupart des propriétés sont spécifiées dans le code, sauf le nom.

Lorsque vous voulez manipuler des fichiers, vous devez suivre les étapes suivantes:

- 1. Spécifier le(s) fichier(s) à manipuler:** chaque fichier doit être ajouté dans la propriété **FilesList** (une TStringList incorporée). La meilleure façon de le faire est de fournir une boîte de dialogue standard d'ouverture de fichier permettant de faire une sélection multiple, puis d'assigner la propriété Files de la boîte de dialogue à la propriété FilesList de TShFileOp. Notez que même si un seul fichier est concerné, il doit faire partie de la liste FilesList (où il sera alors le seul élément).
- 2. Spécifier la destination:** ce peut être soit un répertoire, soit un nom de fichier. Bien entendu, si vous voulez copier ou déplacer plusieurs fichiers, la destination sera un répertoire. Si les fichiers sources doivent être supprimés, aucune destination n'est nécessaire. Si vous voulez renommer un fichier, vous devez spécifier le nouveau nom dans la propriété **Destination**.
- 3. Choisir l'opération à réaliser** à l'aide de la propriété **Operation**. Elle peut prendre l'une des

valeurs prédéfinies suivantes: **shCopy** (défaut), **shMove**, **shDelete** ou **shRename**. Si l'opération est **shDelete**, vous pouvez envoyer les fichiers à la corbeille en ajoutant l'option **shAllowUndo** dans la propriété **Options** (elle l'est par défaut). Si elle n'y est pas, les fichiers seront réellement supprimés. Ceci permet également à l'utilisateur d'annuler une autre opération (si possible) en tapant Ctrl-Z dans le bureau.

4. Appeler la méthode Execute: c'est elle qui déclenche l'opération. Si elle prend plus de 2 secondes, la fenêtre de progression est affichée (sauf si vous avez inclus l'option **shSilent**). Si des fichiers sont sur le point d'être écrasés, une boîte de dialogue demande à l'utilisateur s'il veut continuer ou annuler l'opération (sauf si vous avez inclus l'option **shNoConfirmation**).

5. Voir se qui s'est passé: quand l'opération est terminée, la méthode **Execute** retourne une valeur: True si tout s'est bien passé, False s'il y a eu une erreur. Notez qu'elle retourne True même si l'utilisateur a interrompu l'opération (ce n'est pas une erreur). Dans ce cas, la propriété **Aborted** contiendra True, ce qui vous permet de prendre une décision en conséquence. Notez aussi que la propriété **FilesList** est automatiquement vidée à chaque fois que la méthode **Execute** se termine et quoi qu'il se soit passé. Si vous voulez utiliser les mêmes fichiers pour une autre opération, mettez la propriété **KeepFilesList** à True avant d'appeler la méthode **Execute**.

REFERENCE:

Aborted propriété (boolean, read only, runtime only) : informe si l'utilisateur a ou non annulé l'opération pendant son déroulement, par exemple en cliquant sur Annuler lors d'une confirmation. Notez que quand une opération est annulée, la méthode **Execute** retourne quand-même True.

Destination propriété (AnsiString) : spécifie un répertoire ou un fichier de destination. Quand une opération **shCopy** ou **shMove** concerne plusieurs fichiers, la destination doit être un répertoire. Lorsque vous renommez un fichier, la destination doit être le nouveau nom du fichier. La destination est ignorée lors d'une opération **shDelete**.

Execute méthode (boolean) : déclenche l'opération en appelant la fonction **ShFileOperation**. Une fois l'opération terminée, la méthode retourne True si tout s'est bien passé (même si l'utilisateur a interrompu l'opération) ou False si une erreur s'est produite. La propriété **FilesList** est vidée de son contenu, sauf si **KeepFilesList** a été mise à True avant d'appeler la méthode.

FilesList propriété (TstringList) : cette liste doit contenir un ou plusieurs noms de fichiers avant d'appeler la méthode **Execute**. Ce sont les fichiers sources des opérations **shCopy** ou **shMove** ou les fichiers à supprimer ou à envoyer à la corbeille. Si l'opération est **shRename**, vous devez manipuler un seul fichier à la fois et n'ajouter qu'un seul nom de fichier dans la propriété **FilesList**. Notez que **FilesList** est vidée (mais pas détruite) lorsque **Execute** termine l'opération, sauf si **KeepFilesList** a été mise à True avant d'appeler la méthode. Par ailleurs, la propriété **FilesList** peut être manipulée comme n'importe quelle **TStringList**, sauf être créée ou détruite

(c'est fait automatiquement par le composant). Vous pouvez utiliser les méthodes Add, Move, Delete, Clear des listes ou assigner le contenu d'une autre liste, rechercher un élément avec Indexof, etc.

KeepFilesList propriété (boolean) : mettez cette propriété à True si vous ne voulez pas que la méthode Execute vide la propriété FilesList. Ce peut être utile si vous voulez copier la même série de fichiers plusieurs fois ou si vous préférez vider la liste vous-même après avoir examiné la valeur de retour de la méthode Execute. Cette propriété n'a d'effet que si elle a été spécifiée avant l'appel de la méthode (elle peut aussi être spécifiée une fois pour toutes).

Operation propriété (TshOp) : cette propriété détermine l'opération à réaliser. Elle peut prendre l'une des valeurs prédéfinies suivantes : shCopy (défaut), shDelete, shMove ou shRename. Le type TShOp est une énumération.

Options propriété (TshFlag) : un ensemble d'une ou plusieurs options qui influent sur le déroulement de la méthode Execute:

ShAllowUndo : si elle est incluse, elle permet à l'utilisateur d'annuler l'opération avec Ctrl-Z. Elle permet également d'envoyer les fichiers à supprimer à la corbeille au lieu de les détruire.

ShFilesOnly : n'effectue l'opération que sur les fichiers lorsqu'un joker (*.*) est spécifié.

ShNoConfirmation : les boîtes de confirmation ne sont pas affichées (équivalent à répondre " Oui pour tous ").

ShNoConfirmMkDir : ne demande pas confirmation pour la création d'un nouveau répertoire si c'est nécessaire pour l'opération.

ShRenameOnCollision : renomme automatiquement les fichiers qui existent déjà (" Copie de ... ") sans demander de confirmation.

ShSilent : n'affiche pas la fenêtre de progression.

ShSimpleProgress : affiche la fenêtre de progression, mais sans les noms des fichiers.

Title propriété (AnsiString) : cette propriété peut être utilisée si l'option ShSimpleProgress est incluse, afin de spécifier le titre de la fenêtre de progression.

POUR CONTACTER L'AUTEUR:

- Envoyez moi un message à **mburdin@hol.fr** (ou 101774,626 si vous êtes abonné à Compuserve). Ça me fera plaisir, car les auteurs de freewares sont généralement laissés à l'abandon dès lors qu'on imagine s'être approprié leurs créations (c'est typiquement français, d'ailleurs).
- Vous pouvez consulter ma page web à **http://ourworld.compuserve.com/homepages/m_burdin**. Vous y trouverez la plus récente version (s'il y en a).
- Vous pouvez aussi me contacter par fax ou téléphone au (33) 03 80 78 08 87 ou par courrier à l'adresse suivante

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

France