



File Operators ファイル演算子

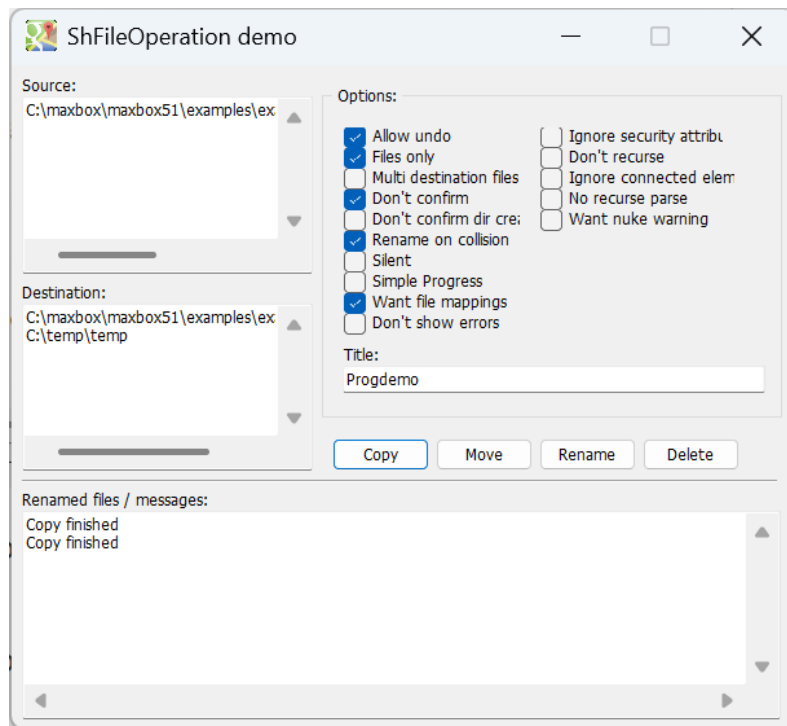
maXbox Starter 154 - Get File Operators.

Source:

[jvcl/jvcl/examples/JvSHFileOperation/JvShFileOperationMainFormU.pas at master · project-jedi/jvcl](#)

Script_: [Download 1425 JvShFileOperationMainFormU1.pas \(maXbox5\)](#)

SHFileOperation copies, moves, renames, or deletes an object in the file system. Instead of performing the action silently as regular file API functions do, SHFileOperation uses the dialog box prompts of the shell. This function is also the proper way to send one or more files to the Recycle Bin instead of deleting them outright. We can use Declare Function SHFileOperation Lib "shell32.dll" Alias "SHFileOperationA" (lpFileOp As Byte) As Long.



App 1425_JvShFileOperationMainFormU1.pas

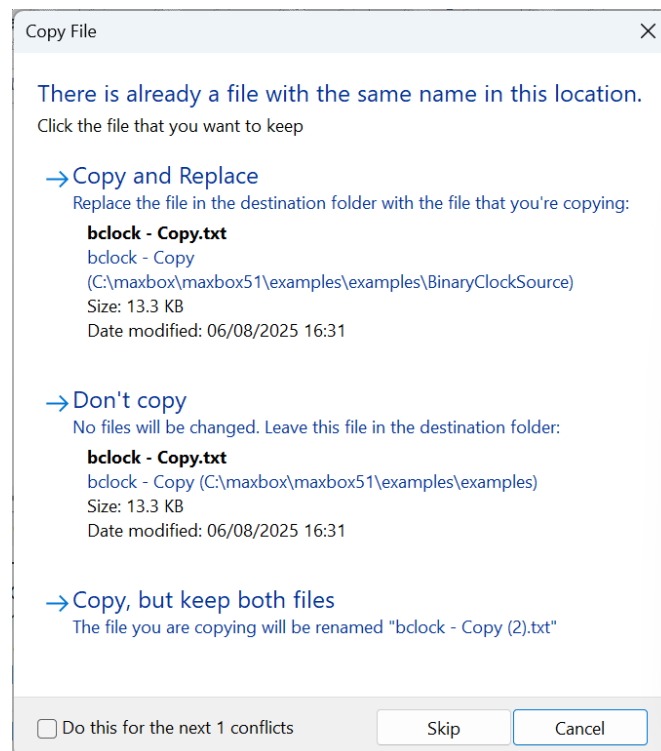
When the user clicks button **Copy**, copies all the txt files inside C:\maxbox\. C:\maxbox\maxbox51\examples\examples\BinaryClockSource*.txt to the Destination C:\maxbox\maxbox51\examples\examples\. Prompt the user as necessary.

```
procedure btnCopyClick(Sender: TObject);
begin
  DoIt(foCopy, 'Copy finished');
end;
```

The SHFileOperation function in Windows Shell API is used to copy, move,

rename, or delete a file system object. Those are the four primary buttons in the app. After the structures have been filled out all we do is fill the SHFILEOPSTRUCT and send it to the function for execution. The function returns zero if successful; otherwise, it returns a nonzero value. It is good practice to examine the value of the fAnyOperationsAborted member of the SHFILEOPSTRUCT. SHFileOperation can return 0 for success if the user cancels the operation.

```
constructor TJvmSHFileOperation.Create(AOwner: TComponent);  
begin  
  inherited Create(AOwner);  
  FSourceFiles := TStringList.Create;  
  FDestFiles := TStringList.Create;  
  FOperation := foCopy;  
  FOptions := [fofAllowUndo, fofFilesOnly];  
end;
```



1425_tutor154copy2025-08-11092442.png

The Destination folder contains the copied files for the first time. A second time the name collision detection operates. Called when a file was renamed (but only if fofRenameOnCollision and fofWantMappingHandle are both True)

Please take following notes:

- Before using the SHFileOperation function, make sure that the operating system has been initialized.
- After using the SHFileOperation function, it is necessary to check the return value to determine if the operation was successful.
- Multiple options can be utilized when setting fileOp.fFlags by connecting them with the logical OR operator (|).
- When setting fileOp.pFrom and fileOp.pTo, it is necessary to use double null characters as separators for multiple files or folders.

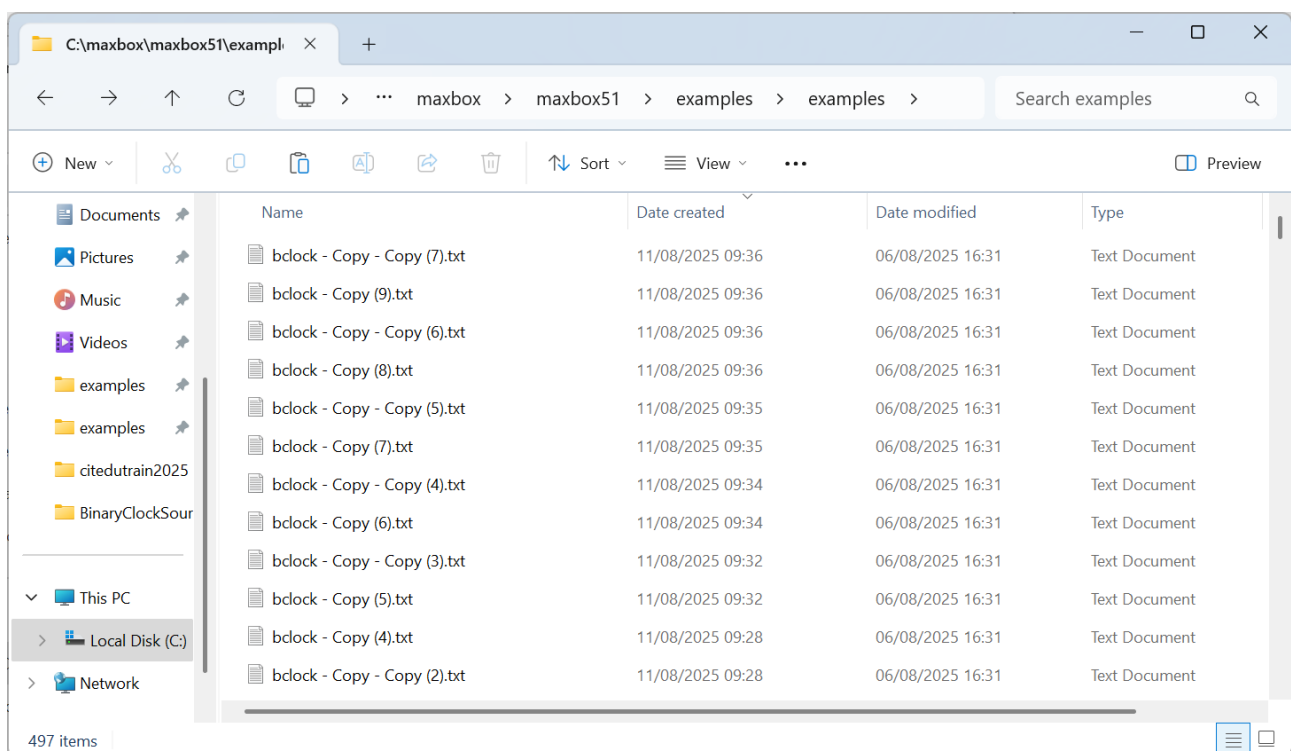
The above is the basic usage of the SHFileOperation function. You can customize the parameters and error handling based on specific needs. To the point 2 check return value:

```
//Options', 'TJvSHFileOptions  //}
if not JvSHFileOperation1.Execute then
    memMessages.Lines.Add(SysErrorMessage(GetLastError))
else
    memMessages.Lines.Add(OkMsg);  //}
```

This name is rebuilt every time 'chkRename' is clicked in options. Because of this, is called rename on collision:

'Gives the file being operated on a new name (such as "Copy #1 of' + '...") '#13#10' in a move, copy, or rename operation if a file of the target name already exists.'

All new renamed files editing should be done in the Destination folder:



```
{** returns True if no error occurred and user didn't abort }
```

```
function TJvSHFileOperation.Execute: Boolean;
const
    AOperation: array [TJvSHFileOpType] of UINT =
        (FO_COPY, FO_DELETE, FO_MOVE, FO_RENAME);
    AOption: array [TJvSHFileOption] of Word =
        (FOF_ALLOWUNDO, FOF_CONFIRM_MOUSE, FOF_FILES_ONLY, FOF_MULTIDESTFILES,
        FOF_NOCONFIRMATION, FOF_NOCONFIRMMKDIR, FOF_RENAMEONCOLLISION,
        FOF_SILENT, FOF_SIMPLEPROGRESS, FOF_WANTMAPPINGHANDLE, FOF_NOERRORUI,
        FOF_NOCOPYSECURITYATTRIBS, FOF_NORECURSION, FOF_NO_CONNECTED_ELEMENTS,
        FOF_NORECURSEREPARSE, FOF_WANTNUKEWARNING);
var
    SFOS: TShFileOpStruct;
    I: TJvSHFileOption; J: Integer; ppFrom, ppTo: string;
    PNameMapping: PSHNameMapping;
```

```

PNameCount: UINT; S,D: string;
begin
  if Length(FSourceFiles.Text) = 0 then
    EJVCLException.CreateRes (@RsENoFilesSpecifiedToTJvSHFileOperatio);

```

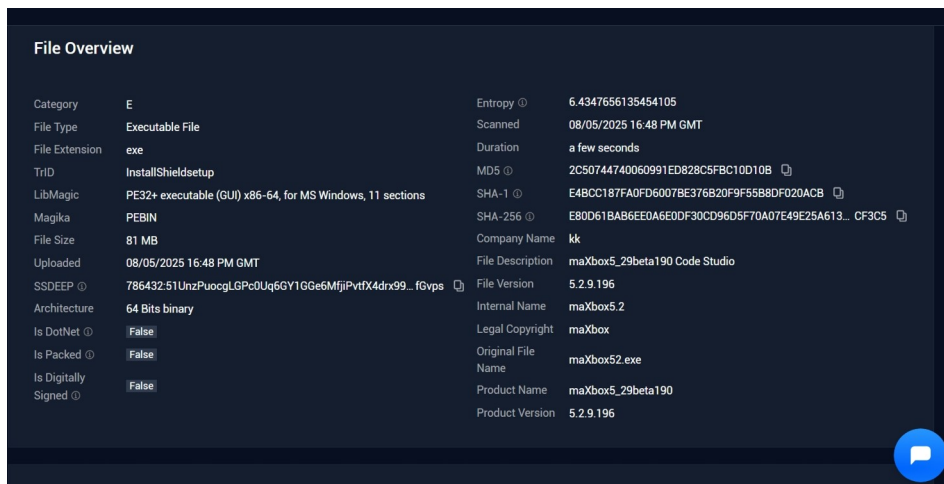
So, basically what we have to do is fill these two lists and pass the structure to SHFileOperation and the function handles all the details. Of course, we have to set appropriate flags to get it working as we like. A typical flag setting would be as follows:

```

procedure DoIt(AType:TjvSHFileOpType; const OKMsg:string);
var AOptions:TJvShFileOptions;
begin
  memMessages.Lines.Clear;
  JvSHFileOperation1.Operation:= AType;
  JvSHFileOperation1.SourceFiles:= memSource.Lines;
  JvSHFileOperation1.DestFiles:= memDest.Lines;
  JvSHFileOperation1.Title:= edTitle.Text;
  AOptions:= [];
  aoptions:= aoptions +[fofConfirmMouse];
  if chkUndo.Checked then
    //Include(AOptions,fofAllowUndo);
    aoptions:= aoptions + [fofAllowUndo];
  if chkFiles.Checked then
    //Include(AOptions,fofFilesOnly);
    aoptions:= aoptions + [fofFilesOnly];
  if chkMulti.Checked then
    //Include(AOptions,fofMultiDestFiles);
    aoptions:= aoptions + [fofMultiDestFiles];

```

Well there is a lot that hasn't been explored by this function plus there is some sort of bug that hasn't been handled while copying folders. The function only copies the files in the source folder to the destination folder and not the whole folder into the destination folder.



References :

[How to use SHFileOperation - Blog - Silicon Cloud](#)

Doc and Tool: [maXbox5 - Manage Files at SourceForge.net](#)

[Release maXbox V5.2.9 · maxkleiner/maXbox5](#)

Max Kleiner 11/08/2025