Backup Dialog
Version 3.57

Designer **Craig Ward, 100554.2072@compuserve.com**
Date **6/2/96**

Function

Dialog box which will backup files from source directory to destination directory.

Update

A new facility which enables the user to only copy newer files (this has required an amendment to the HideBackupDlg call).

****

My Terms

This product has been released into the public domain, FREE! I want no money in return, and you are welcome to make as many copies, and distribute them, as you like. However, I would appreciate some feedback, especially bugs or imporovements that you think that I should make.

Use

There are several calls that the DLL possesses. These are explained below:

ShowBackupDlg(pSource,pDestination,pHelp,pCompat: pChar; bOverwritePrompt: boolean);
- opens the dialog box. This proc copies the contents of a whole directory that conforms to the mask passed over.

HideBackupDlg(pSource,pDestination,pHelp,pCompat: pChar; bOverwritePrompt, bCopyNew: boolean);
- loads, but does not open dialog box, and executes the backup automatically, then closes.This proc copies the contents of a whole directory that conforms to the mask passed over.

CopyIndivFile(pSourceFile,pDestinationFile: pChar);
- copies the source file to the target file.

ExpandIndivFile(pSourceFile,pDestinationFile: pChar);
- copies the source file to the target file, and expands the file if it was compressed (using MS-COMPRESS.EXE). It will replace the extension if it ends in an underscore.

, where,

pSource	- source directory
pDestination	- destination directory
pHelp	- application help file (the DLL uses 102 help context)
pCompat	- file mask

bOverwritePrompt - if true, will prompt for overwrites
bCopyNew - if true, will only copy newer files
pSourceFile - source file
pDestinationFile - destination file

Examples

For example (in Delphi):

```
procedure TForm1.Button1Click(Sender: TObject);
var
hDLL: THandle;
CallDLL: procedure (pSrce, pDest, pHelp, pCompat: pChar; bOverwritePrompt: boolean);
begin
@CallDLL := nil;
hDLL := LoadLibrary('cw_bck.dll'); {will work if dll is in the path}
if hDLL > HINSTANCE_ERROR then
 try
  @CallDLL := GetProcAddress(hDLL,'ShowBackupDlg');
  if Assigned(CallDLL) then
   CallDLL('c:\myApp\tables','c:\myApp\backup','c:\myApp\myApp.hlp','*.*',true)
 finally
  FreeLibrary(hDLL);
 end;
end;
```

Files

Store the DLL in the app's directory, or the Windows directory, whichever you prefer.

Conclusions

I hope that it helps. Have fun and don't forget to give me some feedback!

"...the truth is out there..."