

ibec_CompressFile

This function allows you to create archives of files and extract them using the `ibec-DecompressFile` function. Archives currently supported by the `ibec_CompressFile` function include the following formats:

ZIP, BZIP, GZIP, JAR, LHA, CAB, TAR, BlackHole.

Syntax

```
function ibec_CompressFile(FileSpec : string; ExcludeFileSpec : string;  
                           ArcType : integer; ArcName : string; Options :  
string;  
                           CallbackBlock : string) : variant;
```

The `ibec_CompressFile` currently returns NULL.

Parameters

FileSpec	A filter to retrieve specific file(s) or a range of files. Wildcard characters (asterisk (*) and question mark (?)) are supported. It can include directory names. Items within FileSpec must be delimited with commas.
ExcludeFileSpec	Defines specific file names or a range of file names (using wildcards) to exclude from being compressed. This parameter has precedence over the FileSpec param. For example, if FileSpec contains a file named file.txt, and ExcludeFileSpec contains a wild card such as *.txt, the value of ExcludeFileSpec overrides the value of FileSpec and the file will not be compressed. Items within ExcludeFileSpec must be delimited with commas.
ArcType	Type of archive. Possible values are: __atBlackHole, __atBZip, __atCab, __atGZip, __atJar, __atLha, __atZip.
ArcName	Defines the file name of the archive to be created or an existing archive to which files are to be added.
Options	List of additional options, must be separated with semicolon. Possible options are: CompressMethod, DeflateType, Password, StoredDirNames, StoreEmptySubdirs, StoreFilesOfType, Action, PartSize, DateAttribute, RecurseDirs. See detailed description of each option below.
CallbackBlock	A call-back [IBEBlock EXECUTE IBEBLOCK IBEBlock]] which will be executed for some events during the compression process. The call-back IBEBlock must have at least one input parameter, which will be used to pass an array of event values. If there is no call-back block use NULL or an empty string as a value of this parameter.

Description of possible options

CompressMethod= Store | Deflate | Fuse | Frozen5 | Frozen6 | MsZip | LZX | Quantum |
Tarred | TarGZip | TarBZip | BZip2

The [default](#) method (if the CompressMethod option is omitted) for each archive type is:

```
__atBlackHole: Fuse
__atZip: Deflate
__atLha: Frozen6
__atCab: MsZip
__atGZip: Deflate
__atTar: Tarred
```

The following is the listing of the value of CompressMethod for each archive type:

```
__atZip: [Store, Deflate]
__atBlackHole: [Store, Fuse]
```

```

__atLha:  [Store, Frozen5, Frozen6]
__atCab:  [Store, MsZip, Lzx, Quantum]
__atGZip: [Deflate ]
__atTar:  [Tarred, TarGZip, TarBZip]

```

DeflateType= Store | Fast | Normal | Best

This defines the setting for archive types which use the Deflate compression method. The default setting is NORMAL. Different settings either increase compression speed but reduce compression ratios, or increase ratios but decrease speed.

Password= <password> : Use the Password option to add encrypted files to a ZIP and lackHole archives or extract encrypted files from ones. If the value of this property is not blank, the value will be used as the password for encryption/decryption.

StoredDirNames= None | Absolute | AbsoluteNoDrv | AbsoluteNoRoot | Relative
| RelativeStoreStart | ExplorerAuto

Use this option to set how directories are to be stored in an archive. The default setting is AbsoluteNoDrv.

Example

```

USING UNC-Pathnames
FileSpec = '//Server/Group11/Emp4129/*.txt';
StoredDirNames  Saved in archive as:
-----
None                proj1.txt
Absolute
//Server/Group11/Emp4129/proj1.txt
AbsoluteNoDrv        /Emp4129/proj1.txt
Relative             proj1.txt (subdirs =
dir/*.txt)
RelativeStoreStart   Emp4129/proj1.txt (subdirs =
mp4129/dir/proj1.txt)
ExplorerAuto         proj1.txt (subdirs =
dir/*.txt)

Using local drives
FileSpec := 'f:\ZipTV\Project1\proj1.exe';
StoredDirNames:  Saved in archive as:
-----
None                proj1.txt
Absolute            f:\ZipTV\Project1\proj1.txt
AbsoluteNoDrv       \ZipTV\Project1\proj1.txt
Relative           proj1.txt (subdirs =
dir\proj1.txt)
RelativeStoreStart   project1\proj1.txt (subdirs =
Project1\dir\*.txt)
ExplorerAuto         proj1.txt (subdirs =

```

dir\proj1.txt)

StoreEmptySubDirs= TRUE | FALSE

When the value of this option is True, empty sub-directories names are stored to the archive. The default setting is True.

StoreFilesOfType=<list_of_file_extensions>

This property contains a listing of file extensions delimited with commas or spaces. Any file whose extension is contained within this list will not be compressed, but stored within the archive during compression.

The default value is .LZH, .PAK, .PK3, .PK_, .RAR, .TAR, .TGZ, .UUE, .UU, .WAR, .XXE, .Z, .ZIP, .ZOO.

Note: The extension separator (dot) character for each extension in the list is mandatory!

Action= Add | Move | Delete | Read

This option defines what action the ibec_CompressFile function is to perform.

Possible values are:

Add	Adds files to an archive if they are found not to already exist in the archive. If they do already exist, a comparison of files date stamp with the date stored in the archive is made. If the date stamps do not match, the file is recompressed, otherwise it is skipped. If the desired CompressMethod of compression is different than the method previously used to compress the file, then the file is recompressed, otherwise it is skipped. If the archive doesn't already exist, it is created and all files matching FileSpec are compressed and added to the archive.
Move	Follows the same convention as Add, but deletes all files on disk that were added to the archive. Files are deleted only after a successful ADD.
Delete	Deletes all files matching FileSpec from an existing archive.
Read	Reserved for future use. The default value for this option is Add.

PartSize=<int_value>[KB|K|MB|M|B] - Use this option to specify the file-size of the output volumes for a multi-volume ZIP archive. For example, PartSize=100MB. PartSize value must be at least 65 KB.

DateAttribute= FileDate | SysDate | MaxFileDate - Use this option to define a file's date when extracted to disk. Use this property to define a file's date to be stored into an archive. Possible values:

- **FileDate** - set the extracted file's date using the date stored in the archive; store the date using the disk file's date being compressed.
- **SysDate** - set the extracted file's date using the systems date/time; store the date using the current system's date.
- **MaxFileDate** - set the extracted file's date using the date from the newest file in the archive; store the date using the newest file's date matching FileSpec.

RecurseDirs - TRUE | FALSE - Use this option to recurse sub-directories for files matching the FileSpec parameter. The default value is False.

OverwriteMode - Skip | Overwrite - Use the OverwriteMode property to either skip or overwrite files

that already exist on disk. The default value is Skip.

ConfirmOverwrites - TRUE | FALSE - The ConfirmOverwrites option is directly related to the OverwriteMode option. The default value is False. This option is reserved for future use, please don't change it yet!

RestoreFileAttributes - TRUE | FALSE - When True, this sets an extracted file's attribute to the setting stored in the compressed header for that file. The default value is True.

UseStoredDirs - TRUE | FALSE - When set to False, this uses the current directory to extract files into, if the TargetDir property is blank. When set to True, the default is the current directory information existing in regard to the internal compressed file. The default value is False.

Example

```
execute ibeblock
as
begin
  cbb = 'execute ibeblock (Vals variant)
  as
  begin
    EventName = Vals['EVENT'];
    Action = Vals['ACTION'];
    File name = Vals['FILE NAME'];
    if (Action = 'COMPRESS') then
      sPref = 'Adding ';
    else
      sPref = 'Extracting ';
    if (EventName = 'FILEBEGIN') then
      ibec_Progress(sPref + File name + '...');
    else if (EventName = 'PROGRESS') then
      begin
        iBytes = Vals['BYFILE'];
        if (ibec_Mod(iBytes, 5) = 0) then
          ibec_Progress(sPref + File name + '... ' +
ibec_Cast(iBytes, __typeString) +

          '%');
        end;
      end;
      ibec_DecodeDate(ibec_Now(), iYear, iMonth, iDay);
      ArcName = 'E:\IBE_' + ibec_Cast(iYear, __typeString) + '_' +
ibec_Cast(iMonth, __typeString)

+
      '_' + ibec_Cast(iDay, __typeString) + '.zip';
      if (ibec_FileExists(ArcName)) then
        begin
          ibec_ShowMessage('Nothing to do.');
```

```
end;

-- Compressing
CompressOptions = 'CompressMethod=Deflate; RecurseDirs=Yes;
DeflateType=Best;
StoredDirs=AbsoluteNoRoot';
FileSpec = 'D:\MyProjects\IBExpert\*.*, D:\MyProjects\IBEScript\*.*,
D:\MyComponents\*.*';
ExcludeSpec = '*.dcu, *.~*, *.bak';
MyVar = ibec_CompressFile(FileSpec, ExcludeSpec, __atZip, ArcName,
CompressOptions, cbb);
end;
```

From:
<http://ibexpert.com/docu/> - IBExpert

Permanent link:
http://ibexpert.com/docu/doku.php?id=05-ibexpert-ibeblock-functions:05-27-miscellaneous-functions:ibec_compressfile

Last update: **2023/07/05 18:38**

