

SWBBack

Introduction

SWBBack is an ActiveX component that allows you to add data backup and restore capabilities to your Visual Basic 6.0 applications. The control allows you to specify the backup folder (which can be either a removable drive letter, or a directory on your hard drive), as well as a list of files to be included or excluded in the backup or restore process.

[Performing a Backup](#)

[Performing a Restore](#)

Properties

[BackupDirFile](#)

[BackupFolder](#)

[EraseBackupFolder](#)

[RestoreFolder](#)

[AvgCompression](#)

Methods

[AddIncludeFileSpec](#)

[AddExcludeFileSpec](#)

[ClearIncludeSpecs](#)

[ClearExcludeSpecs](#)

[BackupFiles](#)

[RestoreFiles](#)

[GetBackupFileList](#)

Events

[BeforeProcessing](#)

[BeforeDiskChange](#)

[BeforeDeleteBackupFolder](#)

[AfterDeleteBackupFolder](#)

[AfterDiskChange](#)

[OnFileChange](#)

[OnFileProcessing](#)

[AfterProcessing](#)

Performing a Backup

To allow your application to perform a backup process, your application must do the following:

- 1 Call the `ClearIncludeSpec` and `ClearExcludeSpec` methods to clear the list of files to be included or excluded from the backup process.
- 2 Call the `AddIncludeSpec` method to add to the list of files to be backed up.
- 3 Optionally call the `AddExcludeSpec` method to add to the list of files to be excluded from the backup.
- 4 Set the `BackupFolder` property to the location where the backup data file will be stored. This will usually be the root directory of some removable drive. For example, this property will commonly be set to "A:" if backing up to the user's floppy disk drive. PLEASE READ THE IMPORTANT NOTE REGARDING THIS PROPERTY IN THE TOPIC `BackupFolder`.
- 5 Call `BackupFiles` to begin the backup process.

These are the minimum steps required to perform a backup within your application. You can optionally trap some or all of the events to allow real-time reporting as the process progresses.

Performing a Restore

To allow your application to perform a restore process, your application must do the following:

- 1 Call the `ClearIncludeSpec` and `ClearExcludeSpec` methods to clear the list of files to be included or excluded from the restore process.
- 2 Call the `AddIncludeSpec` method to add to the list of files to be restored.
- 3 Optionally call the `AddExcludeSpec` method to add to the list of files to be excluded from the restore.
- 4 Set the `BackupFolder` property to the location where the backup data file can be found. This will usually be the root directory of some removable drive. For example, this property will commonly be set to "A:" if the backup was previously made to the user's floppy disk drive.
- 5 Set the `RestoreFolder` property to the new "root" directory where the files should be restored.
- 6 Call `RestoreFiles` to begin the restore process.

These are the minimum steps required to perform a restore within your application. You can optionally trap some or all of the events to allow real-time reporting as the process progresses.

Property BackupDirFile

Type: String

Access: Read-Only

Description:

Returns the path and filename of the temporary file used to store the file directory while the backup or restore process is in progress. This file will normally be stored in the directory defined as the Windows "Temp" directory according to the system it is running on.

Property BackupFolder

Type: String

Access: Read/Write

Description:

Sets or returns the directory name where the backup file is to be stored. This will normally be a removable drive letter ("A:"), but could also be the name of an existing directory on the hard drive. The backup data will be stored at this location in a file called "data0001.dat". If additional disks are needed to contain all of the backup data, subsequent data files will be numbered sequentially (i.e. "data0002.dat", etc).

The control automatically detects when there is no space remaining in the BackupFolder, and prompts for another disk when this occurs.

IMPORTANT: IF THE "EraseBackupFolder" PROPERTY IS TRUE, ALL FILES AND FOLDERS UNDER THE BACKUPFOLDER LOCATION WILL BE DELETED WHEN THE BACKUP BEGINS. IF YOU SPECIFY A FOLDER ON THE HARD DRIVE, ALL FILES AND DIRECTORIES UNDER THAT FOLDER WILL BE DELETED AND REMOVED.

Property EraseBackupFolder

Type: Boolean

Access: Read/Write

Description:

Determines whether or not the contents of the backup media in AND below the folder specified in the BackupFolder property will be permanently deleted prior to writing the new backup information to that location.

IMPORTANT NOTE: If this property is set to True, all files and subfolders in AND below the directory specified by the BackupFolder property WILL BE DELETED. It is recommended that this property be set to True when backing up to removable media, and set to False when backing up to a fixed (hard) drive, unless you are sure you want to delete all information contained within the BackupFolder directory.

Property RestoreFolder

Type: String

Access: Read/Write

Description:

Sets or returns the directory name of the "root" folder where the restored files will be copied to. When files are backed up, their original relative path is also stored with the backup information. Upon restoration, the files that are restored are stored in their original relative paths. However, you can specify that the restored files are stored under an alternate root. The original relative paths for each file will be created under that root if they do not already exist.

For example, a file may have been backed up from "C:\My Files\Docs\Report.doc". To restore the file to its original location, specify a RestoreFolder of "C:\". To restore to a different location, you might specify a RestoreFolder of "C:\RestoredFiles". This would cause the file to be restored under the path "C:\RestoredFiles\My Files\Docs\Report.doc". This is useful when you do not want to overwrite the existing copy of a file, perhaps to allow for comparison after restoring the file.

Property AvgCompression

Type: Double

Access: Read-Only

Description:

Returns the average percentage of compression that has occurred during the current backup process. This property does not return a meaningful value during the restore process and therefore should not be used during a restore process. This number represents the amount of data compression that has occurred for all files up to the point the value was retrieved.

During the backup or restore process, you can retrieve the value of this property during the OnFileProcessing event to provide real-time compression percentages in your application's user-interface.

To obtain the total compression percentage for the entire backup process, you can also use the CompressionPct argument that is passed to the AfterProcessing event. That value will contain the total compression percentage for the entire backup process.

Method AddIncludeFileSpec

AddIncludeFileSpec (FileMask as String, Optional FileAttributes as Long = FATTR_ANYFILE, Optional ProcessSubDirs as Boolean = True)

Description:

Adds a file specification to the collection of file specifications that represent the files to be *included* as part of a subsequent backup or restore process. Files existing in the specified path will be processed (backed up or restored), provided they match the filemask at the end of the path specification, and their attributes are specified in the FileAttributes bitmask, and depending upon the setting of the ProcessSubDirs indicator.

Files that exist below the specified directory will be processed if the ProcessSubDirs indicator is set to True, and they match the filemask and the attributes specified.

FileAttributes and ProcessSubDirs arguments do not need to be specified.

FileMask String The complete path and file mask of the files to be processed.

FileAttributes Long The combination of file attributes of files to be processed. The allowable file attributes are:

vbNormal	0
vbReadOnly	1
vbHidden	2
vbSystem	4
vbArchive	32

The default (if not specified), FATTR_ANYFILE is equal to 39, and represent all possible file attribute settings.

ProcessSubDirs Boolean Indicates whether or not matching files existing under this directory will be processed.

Method AddExcludeFileSpec

AddIncludeFileSpec (FileMask as String, Optional FileAttributes as Long = FATTR_ANYFILE, Optional ProcessSubDirs as Boolean = True)

Description:

Adds a file specification to the collection of file specifications that represent the files to be *excluded* as part of a subsequent backup or restore process. Files existing in the specified path will be excluded, provided they match the filemask at the end of the path specification, and their attributes are specified in the FileAttributes bitmask, and depending upon the setting of the ProcessSubDirs indicator.

Files that exist below the specified directory will also be excluded if the ProcessSubDirs indicator is set to True, and they match the filemask and the attributes specified.

FileAttributes and ProcessSubDirs arguments do not need to be specified.

FileMask String The complete path and file mask of the files to be excluded.

FileAttributes Long The combination of file attributes of files to be excluded. The allowable file attributes are:

vbNormal	0
vbReadOnly	1
vbHidden	2
vbSystem	4
vbArchive	32

The default (if not specified), FATTR_ANYFILE is equal to 39, and represent all possible file attribute settings.

ProcessSubDirs Boolean Indicates whether or not matching files existing under this directory will be excluded.

Method ClearIncludeSpecs

`ClearIncludeSpecs ()`

Description:

Clears the collection of file specifications that were added using the AddIncludeFileSpec method.

Method ClearExcludeSpecs

`ClearExcludeSpecs ()`

Description:

Clears the collection of file specifications that were added using the [AddExcludeFileSpec](#) method.

Method BackupFiles

BackupFiles ()

Description:

Starts the backup process. All files that match at least one of the specifications added using the AddIncludeFileSpecs method will be backed up, provided that they match none of the specifications added using the AddExcludeFileSpecs method.

Prior to calling this method, the application should set the BackupFolder property to the location where the backup information will be stored.

The application should also call AddIncludeFileSpecs and AddExcludeFileSpecs to create the list of files to be processed prior to initiating the backup process.

Method RestoreFiles

RestoreFiles ()

Description:

Starts the restore process. All files that match at least one of the specifications added using the AddIncludeFileSpecs method will be restored, provided that they match none of the specifications added using the AddExcludeFileSpecs method.

Prior to calling this method, the application should set the BackupFolder property to the location where the backup information can be found. It should also set the RestoreFolder property to the location where the restored data should be stored. Files that are restored will be stored in their original relative paths, however, these paths will be relative to the new "root" specified by RestoreFolder.

The application should also call AddIncludeFileSpecs and AddExcludeFileSpecs to create the list of files to be processed prior to initiating the backup process.

Method GetBackupFileList

`GetBackupFileList ()`

Description:

Returns an array of strings that represent all the files contained in the backup datafile(s). Note that this list does not exclude any files that may match the Exclude File specs. It simply returns a complete list of all files that were backed up.

Prior to calling this method, the application should set the BackupFolder property to the location where the backup data was stored.

Event BeforeProcessing

BeforeProcessing(ByVal TotalFiles As Double, ByVal TotalSize As Double,
Cancel As Boolean)

Description:

Occurs after the list of files to be processed has been created, but before the first disk is prompted for.

TotalFiles The total number of files to be processed. This number takes into account files that will be included or excluded by the Include/Exclude file specs.

TotalSize The total byte size of the files to be processed. This number takes into account files that will be included or excluded by the Include/Exclude file specs.

Cancel This value can be set to True by the calling application to cancel the backup or restore process at this stage.

Event BeforeDiskChange

`BeforeDiskChange (ByVal NextDisk As Integer)`

Description:

Occurs immediately before the first, or next disk is prompted for during a backup or restore process.

NextDisk The number of the disk that is about to be requested.

Event BeforeDeleteBackupFolder

BeforeDeleteBackupFolder ()

Description:

Occurs immediately before the control starts to delete the directory tree starting at the folder specified by the BackupFolder property. Because this process may take several seconds to perform depending on the existing data at that location, this event allows the calling application to post a message to the user regarding this process.

This event is only fired during a backup process.

Event AfterDeleteBackupFolder

AfterDeleteBackupFolder ()

Description:

Occurs immediately after the control finishes deleting the directory tree starting at the folder specified by the BackupFolder property.

This event is only fired during a backup process.

Event AfterDiskChange

`AfterDiskChange (ByVal NextDisk As Integer, ByVal Cancel As Boolean)`

Description:

Occurs immediately after the first, or next disk is prompted for during a backup or restore process. This event is fired only after the correct disk is inserted, or the user clicks Cancel when prompted for the correct disk.

NextDisk The number of the disk that was just requested.

Cancel Informs the calling application whether or not the user clicked Cancel rather than inserting the requested disk.

Event OnFileChange

```
Public Event OnFileChange (ByVal FileName As String, ByVal FileSize As Double)
```

Description:

Occurs immediately before a file is about to be stored in or extracted from the backup data file.

FileName The complete path and filename of the file about to be processed.

FileSize The size (in bytes) of the file about to be processed.

This event is fired before the file is processed to allow the calling application update its display.

Event OnFileProcessing

```
Public Event OnFileProcessing(ByVal FileName As String, ByVal FileSize As Double, ByVal FileBytesProcessed As Double, ByVal TotalBytesProcessed As Double, Cancel As Boolean)
```

Description:

Occurs after each buffer is written to the backup data file during a backup process, or written to the restored file during a restore process.

FileName The complete path and filename of the file currently being processed.

FileSize The size (in bytes) of the file currently being processed.

FileBytesProcessed The number of bytes processed already for this file. This number, when divided by the FileSize argument, provides the percentage complete for the current file.

TotalBytesProcessed The number of bytes processed for all files during this backup or restore process. This number, when divided by the TotalSize argument provided by the BeforeProcessing event, provides the percentage complete for the entire backup or restore process.

Event AfterProcessing

AfterProcessing(ByVal TotalFiles As Double, ByVal TotalSize As Double,
ByVal TotalDisks As Integer, ByVal CompressionPct As Double)

Description:

Occurs after the last file has been backed up or restored. In the case of a backup process, this event occurs after the last file is backed up, but before the backup process prompts for the first diskette (to store the backup directory information).

TotalFiles The total number of files to be processed. This number takes into account files that will be included or excluded by the Include/Exclude file specs.

TotalSize The total byte size of the files to be processed. This number takes into account files that will be included or excluded by the Include/Exclude file specs.

TotalDisks The total number of disks used to store the backup data. This number will represent the last disk called for during a restore process.

CompressionPct The percentage of compression achieved for the entire backup. This value will not have any meaning after a restore process.

