

*"There are two types of computer users:
those who backup,
and those who WILL..."*

EVACopy

by

Or Ben Shabat

Version 5.2

<u>Page</u>	<u>Topic</u>
2	Brief Method
3	Example of How EVACopy Works Restoration Procedures
4	The Settings File and Command-Line Parameters Examples of Command-Line Parameters Exit Codes The Log File
5	Advanced use of EVACopy
6	EVACopy.ini Explained
10	The Resident Feature
13	EVAConfig: Configuration Tool
15	TLPD: Too Long Paths Detector
17	LaRVe: Restore Tool
18	MIAfind: Missing-In-Action Finder
19	Note: EVACopy vs. volume snapshots Note: EVACopy vs. incremental and differential backup Note: EVACopy vs. repositories
20	Note: EVACopy and Linux Note: EVACopy source code Note: EVACopy tray icon Note: When NOT to use EVACopy
21	Best Practices for Effective and Fault-Free Use
22	Free, open-source, and all that other stuff Your Feedback is Valuable !

Brief

EVACopy - stands for EVACuate and Copy - is a backup program that has *versioning* and *easy restoration* in mind.

EVACopy is an open-source project, hosted at <http://evacopy.sourceforge.net>

A lot has been said about the importance of keeping backups, but *versioning* means that apart from an exact copy of your recent contents, you also keep all earlier versions of files and folders that were modified or deleted.

This is very useful when you want to put your hands on a few-days-old copy of a document, but you are horrified to find that the backup – you so carefully perform every day – was overwriting the older copy with the latest one...

The common solution to this problem is to use several media units (like tapes, CD's or DVD's), each one holding a complete copy of your entire contents, in a specific point in time. For this solution to be effective, it must be applied at least daily. This method is very wasteful, due to the fact that backing-up casual changes take far less space and time than a complete backup.

Another solution involves using Windows (Vista and later) or Linux (LVM) volume snapshot feature, but this solution has its own issues – see later in this manual.

Method

Before backing-up your files from the source to the destination, EVACopy compares the destination to the source. Those files that have changed in the source since the last backup (and therefore are going to be removed or overwritten in the destination), are evacuated to a dedicated folder (the evacuation folder), leaving the destination ready to receive the latest files from the source.

This dedicated folder is hidden by default (so you don't confuse it with your latest files). Its name is `_EVAC` by default. The underscore is used to distinguish it from other folders, and to show it first when sorted by name.

The files in the evacuation folder keep their directory structure, meaning you'll find them in the same subfolder as in the source. They are also stored with a notation that indicates the time they were evacuated from the destination, so you can easily sort them out by time to retrieve them later. That notation is the name of the evacuation folder, which is written in the format `YYYY-MM-DD-hh-mm` (year,month,day,hour,minute).

Example of How EVAcopy Works

- The source folder is "C:\User\Docs". The destination folder is "F:\Backup\Docs"
- The file "C:\User\Docs\Accounting\Income\January.xls" was backup-up yesterday, and the yesterday copy is "F:\Backup\Docs\Accounting\Income\January.xls"
- Today, the file "C:\User\Docs\Accounting\Income\January.xls" was updated.
- Today is January 23rd, 2007. The time is 5:10 pm. EVAcopy creates the path "F:\Backup\Docs_EVAC\2007-01-23-17-10".
- EVAcopy detects that the file "January.xls" was modified recently, and moves the file: from "F:\Backup\Docs\Accounting\Income", to "F:\Backup\Docs_EVAC\2007-01-23-17-10\Accounting\Income".
The yesterday copy is now:
"F:\Backup\Docs_EVAC\2007-01-23-17-10\Accounting\Income\January.xls".
- After the evacuation, the folder "F:\Backup\Docs\Accounting\Income" does not contain any file named "January.xls".
- After the evacuation of all modified files was completed, EVAcopy copies the updated file: "C:\User\Docs\Accounting\Income\January.xls"
To: "F:\Backup\Docs\Accounting\Income\January.xls",
resulting in an identical contents of both source and destination folders,
plus the earlier version of the file,
as: "F:\Backup\Docs_EVAC\2007-01-23-17-10\Accounting\Income\January.xls"

Restoration Procedures

Very few backup programs think about "the day after". What happens when you actually need to restore your files? With most professional (and costly) software and hardware, the restoration process is quite tedious. With EVAcopy it's as simple as it can be.

- To restore the entire source, simply copy the contents of the destination, excluding the evacuation folder.
- To restore a specific file that was modified or deleted, you can either browse for it in the evacuation folder (enable browsing hidden files), or preferably search it (include hidden files and folders). When searched, you get all earlier versions of the file, sorted by date. Choose the last one (or earlier, if you wish) and copy it back to the source. You can even open the files directly from the evacuation folder, to help you review and select the file to restore.
- To restore a folder, or in case you can't remember the name of the file but you know in which folder it was, follow the same procedure.
- You can also use LaRve – a dedicated restore tool (see later in this manual). LaRve can list files in any point in time in which EVAcopy performed a backup, and also list all versions of a specific file.

For example:

You want to restore the earlier version of "January.xls" (as in the last example). Search for it. The search result will show it in two locations:

- 1) "F:\Backup\Docs\Accounting\Income\January.xls"
- 2) "F:\Backup\Docs_EVAC\2007-01-23-17-10\Accounting\Income\January.xls"

The 1st is the most recent file. The 2nd is the file before it was modified at January 23rd, 2007, at 5:10 pm.

Choose the copy to restore, and simply copy it to the source folder.

Note: You can – and are encouraged to – set up a second line of defense. For example, home users can burn the entire destination folder (including the evacuation folder) to a CD or DVD every week or so. Corporate users can use the conventional backup method (most organizations use one) to backup the destination folder instead of (or along-side with) the source.

The Settings File and Command-Line Parameters

EVACopy settings are stored in a simple text file in standard "ini" format. This file is named EVACopy.ini and resides in the same folder as the program itself (EVACopy.exe).

To use another settings file, run EVACopy with a single command-line parameter. That parameter must be the name (either absolute or relative) of a settings file. If that file is missing or invalid, EVACopy tries to use the default settings file.

If any of the settings in the file is missing or invalid, or if the entire file is missing or invalid, these are the default settings you should be aware of:

- Name of the evacuation folder is _EVAC and is hidden;
- System files are excluded from the backup; only changes and errors are logged.

You can specify multiple pairs of source-destination in the settings file (but not in the command-line). They are carried-out sequentially. See next pages for more details.

Examples of Command-Line Parameters

```
evacopy.exe
    does what "evacopy.ini" says
evacopy.exe c:\MyFiles f:\backup\c\MyFiles
    uses the specified paths for source and destination. the rest is set by "evacopy.ini"
evacopy.exe "c:\My Files" "f:\backup\c\My Files"
    same as above, but using long file names
evacopy.exe my-settings.ini
    does what "my-settings.ini" says. if "my-settings.ini" is missing, "evacopy.ini" is used.
```

Exit Codes

EVACopy returns exit codes you can use when running from a batch file.

0 = Program completed successfully.

1 = Problems with parameters. Result: Program terminated without doing anything.

2 = Can't create or write to the destination folder. Result: Program terminated without doing anything.

3 = Can't create or write to a subfolder in the destination folder, probably due to a too long path name or locked files. Result: Program partially completed.

4 to 8 = reserved.

9 = Program terminated by user. Result: Program partially completed.

The Log File

EVACopy produces a log file, with a name containing the time of backup, similar to the name of the evacuation folder, and located next to it. It's a simple text file, meant to be used for troubleshooting. For the earlier example, the log file would be:

```
"F:\Backup\Docs\_EVAC\EVACopy-log-2007-01-23-17-10.txt"
```

By default, the log file lists only changes and errors. It can be very detailed (see settings file), but then it bloats. However, it is recommended that you go through the **debug** mode just once (test it on a small source folder), to get yourself familiar with it. That way, should something go wrong, you could trace the problem more easily.

To detect errors, search for the string "ERROR!" in the log file. As a system administrator, you can do the same for all files named "EVACopy-log-*.txt" located in your backup media, to search for errors for all of your users or workstations at minimal time.

EVACopy can also produce a main log, where only main events like "application start", "job start", "job done", etc. are recorded. This log can be used to gather usage statistics, for single workstation as well as for several. See "MIAfind" later in this manual.

Advanced use of EVACopy

Like any other program that uses command-line parameters, EVACopy can accept standard environment variables (like %COMPUTERNAME%). But the standard variables are not enough, so in the settings file you can use them as well as the following variables, which the program itself defines:

Environment variables for program paths:

%exeDrive% - drive letter (with trailing colon) of the drive you run EVACopy from.
%exePath% - the path you run EVACopy from.
%WorkingDir% - the current working directory.

Environment variables for the current user:

%Desktop%
%MyDocuments%
%Favorites%
%LocalSettings%
%LocalAppData% (Application Data under Local Settings)

Environment variables for all users:

%AllUsersDesktop%
%AllUsersDocuments%
%AllUsersFavorites%
%AllUsersAppData%

Note: The Application Data and profile paths of the current user are already stored in %APPDATA% and %USERPROFILE%. So is the profile path of "All Users", which is %ALLUSERSPROFILE%. The variables specified above are complementary to the variables already exist in the system (Windows NT and later).

For example: You run EVACopy from your disk-on-key because you use it to backup several computers. These settings backup "My Documents" and "Desktop" of the current user on every computer to a folder on the disk-on-key you run EVACopy from:

```
[PathsX]
%MyDocuments%|%exeDrive%\backup\%COMPUTERNAME%\My Documents
%Desktop%|%exeDrive%\backup\%COMPUTERNAME%\Desktop
```

You can use the settings files in conjunction with the command-line parameters and the program exit codes to make a more elaborate use of it via batch files. For example, If you prefer using a faster, more robust or more advanced copier than EVACopy (in this example I'll use robocopy), then you would like EVACopy to perform only the evacuation. Modify the settings file: in section [Mode], set Copy=no (see next page for details). Use the following batch file:

```
start /wait evacopy.exe "D:\Users Files" "F:\backup\Users Files"
if not errorlevel 0 goto Error
robocopy "D:\Users Files" "F:\backup\Users Files" /s /r:0
goto End
:Error
echo an error occurred in the evacuation process! backup aborted.
:End
```

A smart combination of the settings file, the command-line and the exit codes is probably enough for any backup job you may need to perform.

EVACopy.ini Explained

The settings file is not quite a standard .ini file. It has sections, and the sections are standard except of two, [PathPredef] and [PathsX]. They are all explained here.

Section [General]

The `Version=5` line indicates that the file is compatible with EVACopy v5.x – meaning that newer versions of EVACopy (up until futuristic v6.0) don't need to be re-configured.

Section [PathPredef]

This section specifies predefines paths for use in next section. It is empty by default.

Syntax

`/n|path`

`n` is a digit 1-9, `path` is a path name (absolute or relative).

For example:

`/1|P:\backup`

In this example, when `/1` appears in the next section, it is substituted by `P:\backup`

Section [PathsX]

This section specifies paths of the source and destination paths of each backup pair, a descriptive name for each backup pair, exclusion rules and idle settings.

It may seem complicated, but it's actually quite simple. Take a close look at the examples in the next page.

Syntax

`name|src|dst||Xname|/Xattr|>XsizeMax|<XsizeMin|idle:`

The line is composed of two halves, separated by `||` (two vertical line characters).

The 1st half specifies (optional) name and (mandatory) source and destination paths for a backup pair.

The 2nd half is completely optional, and if omitted, the `||` can be omitted too. This half specifies exclusion rules and idle settings that apply for this pair only. For common exclusion rules (that apply to all pairs) - see below.

Each half is composed of arguments, separated by `|` (a single vertical line character).

Syntax Explanation

1st half:

`name`: A descriptive name of the source-destination pair.

`src` and `dst`: These paths may be absolute or relative (relative to the working directory).

2nd half:

`Xname`: Exclude files and folders with specific names and extensions, separated by a semicolon. Wildcards are currently not supported. The only exception exists for extensions (in this format: *.ext). For example: `temp;*.tmp;pagefile.sys;*.swf`

All instances of a specified name are excluded, unless you specify a path of a specific instance, relative to `src`. For example: `\downloads\temp` will exclude only the file or folder named `temp` in the folder `downloads` in the `src`, while in the previous example, all files and folders named `temp` are excluded, wherever they reside.

- This is not the case for extensions. i.e. `\downloads*.tmp` is invalid, and ignored.
- Absolute paths are also invalid. i.e. `C:\downloads\temp` is ignored.

`/Xattr`: Exclude files and folders with specific attributes. Options are any combination of the letters AHRS (Archive, Hidden, Read-only, System).

Example: use `/HS` to exclude Hidden files and System files.

>XsizeMax: Exclude files over the specified size. Size in KB, can't be with decimal point.

Example: use >200000 to exclude files over 200MB.

<XsizeMin: Exclude files under the specified size. Size in KB, can't be with decimal point.

Example: use <10 to exclude files under 10KB

|idle: This is not an exclusion rule; it tells EVACopy if it should remain resident. See the topic about the Resident feature later on.

Note that in the 2nd half, the order of the arguments has no importance. This is not the case in the 1st half, where the only mandatory arguments are *src* and *dst*, so some other valid formats are:

<i>src</i> <i>dst</i>	name and all exclusion rules are omitted
<i>src</i> <i>dst</i> <i>Xname</i> / <i>Xattr</i> < <i>XsizeMin</i>	only name and <i>XsizeMax</i> are omitted
name <i>src</i> <i>dst</i> / <i>Xattr</i> > <i>XsizeMax</i>	<i>Xname</i> and <i>XsizeMin</i> are omitted

However, there is one exception to this:

If both *src* and *dst* are omitted (i.e. the line begins with | |), then the exclusion rules are applied to all pairs specified.

Examples

C:\User\My Documents | P:\backup

only *src* and *dst* are specified

Mr. Anderson's files | C:\User\My Documents | P:\backup

name is also specified

Mr. Anderson's files | C:\User\My Documents | P:\backup | | *.tmp

also exclude *.tmp files

C:\User\My Documents | P:\backup | | /H | >150000

no name, exclude hidden files and files greater than 150MB

C:\User\My Documents | P:\backup | | /HR | >150000

also exclude read-only files

C:\User\My Documents | P:\backup | | downloads; *.tmp | /HR | >150000

also exclude any file or folder named "downloads" and files with "tmp" extension

C:\User\My Documents | P:\backup | | <10

exclude only files smaller than 10KB

Auto-completion

In case *dst* ends with / or // (single or double slash character), it is auto-completed as following:

/ is auto-completed to the last subfolder of the *src* path.

For example:

C:\user\My Documents | P:\backup/

is auto-completed to:

C:\user\My Documents | P:\backup\My Documents

// is auto-completed to the full path of the *src* path (with drive letter colon removed). For example:

C:\user\My Documents | P:\backup//

is auto-completed to:

C:\user\My Documents | P:\backup\C\user\My Documents

This also applies when using environments variables. For example:

(assuming user name is "Neo" and Neo's documents folder is in its default Windows XP path)

%MyDocuments% | P:\backup/

is auto-completed (after the custom environment variable is replaced with its value) to:

C:\Documents and Settings\Neo\My Documents | P:\backup\My Documents

Similarly,

%MyDocuments% | P:\backup//

is auto-completed to:

C:\ ... | P:\backup\C\Documents and Settings\Neo\My Documents

Substitutions with predefined paths

In case `src` or `dst` includes `/1` to `/9`, they are substituted with paths that were predefined in the `[PathPredef]` section.

For example, add this line to the `[PathPredef]` section:

```
/1|P:\backup
```

This line defines that `/1` is substituted with `P:\backup`

For example:

```
C:\user\My Documents|/1
```

is substituted with:

```
C:\user\My Documents|P:\backup
```

You can also add `/` and `//` to `/1`, for example:

```
C:\user\My Documents|/1/
```

is substituted and auto-completed to:

```
C:\user\My Documents|P:\backup\My Documents
```

You can also predefine this combination.

For example, add this line to the `[PathPredef]` section:

```
/2|P:\backup//
```

In this case,

```
C:\user\My Documents|/2
```

is auto-completed to:

```
C:\user\My Documents|P:\backup\C\user\My Documents
```

But take care not to use `/` and `//` twice, for example:

```
C:\user\My Documents|/2//
```

will result in an error, because `/2` already includes `//`

You can use environment variables here also, For example: `/2|%exeDrive%\backup//`

Notes

1) About environments variables:

`src` and `dst` can use default and custom environments variables.

Default environments variables include `%COMPUTERNAME%`, `%windir%`, etc. – Depending on your version of Windows. Custom environments variables include `%MyDocuments%`, `%Desktop%`, `%Favorites%` etc.

2) About command-line parameters vs. settings file:

By default, no `src` and `dst` are specified, and they are retrieved from the command-line parameters. This means that if they are omitted from the command-line, and still unspecified in the file, then EVACopy will ask you to configure them. If `src` and `dst` are specified in both command-line parameters and in this file, the command-line parameters are used.

3) About syntax:

Do NOT enclose long path names in double-quotes (""), also don't use trailing backslash (\).

4) About exclusion rules:

When multiple exclusion rules are specified - e.g. `|/S|>150000` - a file must comply with any of the rules in order to be excluded. e.g. `|/S|>150000` will exclude a file if it is EITHER a System file OR greater than 150MB (or both).

Currently there's no option for AND exclusion rules, e.g. to exclude a file ONLY if it's BOTH a System file AND greater than 150MB.

Default value

The default value is a single line:

```
||/S
```

(two vertical line characters, followed by a backslash and the letter S)

This line specifies common exclusion rules. By default, System files are excluded – this deals with those pesky "desktop.ini" and "thumbs.db" files, and others. These exclusion rules are applied to all pairs specified in the next lines, regardless of specific exclusion rules you may set. You may change this line, For example:

```
||/S|*.tmp;temp|>200000
```

Section [Evac]

The `PathName=` line specifies the name of the evacuation folder (located inside the destination folder). If you change it in the settings file (after some evacuations were already performed), remember to change it in the directory tree itself as well!

Section [Mode]

The `AcceptRedirections=` line (valid only in default EVACopy.ini) tells the program if it should be redirected to this file when the settings file specified as a command-line parameter doesn't exist.

The `NotifyRedirections=` line tells the program if the user should be notified about it.

The `UseProgressMeter=` line tells the program if it should display a file count in the tray icon tooltip. This indicator shows the user if the program hangs (never happened).

The `Detect=` line specifies the methods of detecting file changes. Options:

`Detect=time` (default) checks the file modified time.

`Detect=size` checks the file size.

A size change implies time change, because you simply can't change the file size without changing the time. However, it's possible to change the time without changing the content of the file – in which case the file doesn't require backup. Therefore the time check is more comprehensive, but may result in cluttering.

The `Log=` line specifies what is logged in the log file (Errors are always logged). Options:

`Log=C` (default) or `Log=changes` logs files and folders that were evacuated and/or copied.

`Log=X` or `Log=exclusions` also logs exclusions rules and excluded files and folders.

`Log=F` or `Log=full` also logs skipped files and all files and folders encountered.

`Log=D` or `Log=debug` adds human-readable remarks to the full log, and logs the exclusion rules and events in detail. Use it for troubleshooting only, because it may result in a bloated log file.

The `Copy=` line tells EVACopy if it should perform the copy phase (after the evacuation phase was completed) or not. See page 5 for an example.

The `CopyEvenIfEvacHadErrors=` line is valid only if `Copy=yes`, it tells EVACopy what to do in case the evacuation phase encountered errors.

Section [UponCompletion]

The `ShowErrorsList=` line, if set to `no`, disables the errors list summary, which is displayed if errors occurred during the backup. The errors list is also saved to a file in the temporary folder, by the name "EVACopy-ErrorsList-*[descriptive name]*.txt" (if descriptive name was not specified, the source path is used).

The `UserNote=` line, if set to `no`, disables the user note dialog box.

The `UserNotePrefix=` line sets a prefix to the user note. By default it's "before", because the user note refers to the evacuated files – just before they were removed or modified.

The `UserNoteAppendToFolder=` line, if set to `no`, appends the user note to the log file name only. By default it's appended to both the log file name and the evacuation folder (unless nothing was evacuated and the evacuation folder was removed).

The `Hidden=` line tells EVACopy if the evacuation folder should be hidden or not. By default it is hidden, to distinguish it from the other files and folders in the destination folder.

The `RemoveUselessLog=` line, if set to `no`, keeps log files that have no entries. By default, if no errors occurred and no changes were made, and the logging mode is not set to anything more detailed than "changes", then the log file is useless, and therefore is removed.

The `NotifyIfLogRemoved=` line, if set to `no`, disables the user notification of the log file being removed.

The `Action=` line tells EVACopy what to do when the job was completed. By default it's nothing, but it can be a file to execute, or any power-management action: shutdown, reboot, standby, hibernate, logoff, or lock.

The `ActionEvenWithErrors=` line is ignored if `Action=none`, otherwise it tells EVACopy if it should perform the specified action even if errors were encountered.

Section [Resident]

This section sets options for the behavior of the Resident feature, described next.

The Resident Feature

The Resident feature allows EVACopy to remain resident (as a tray icon) and to automatically initiate a backup job when the user is idle for some time (default is one minute). Via the tray icon menu you can also initiate a backup job manually, browse folders, review and restore earlier versions of files, change program settings, etc.

To use the Resident feature, edit the settings file (EVACopy.ini). Add `idle:y` to the end of the source|destination line. Remember to add the two vertical lines if no exclusion are specified.

From now on, you need to start EVACopy just once, and it will remain resident. If you wish to make EVACopy start automatically when Windows starts, move EVACopy shortcut from the desktop (or from wherever it is) to the Startup folder in the Windows Start Menu.

Idle Options

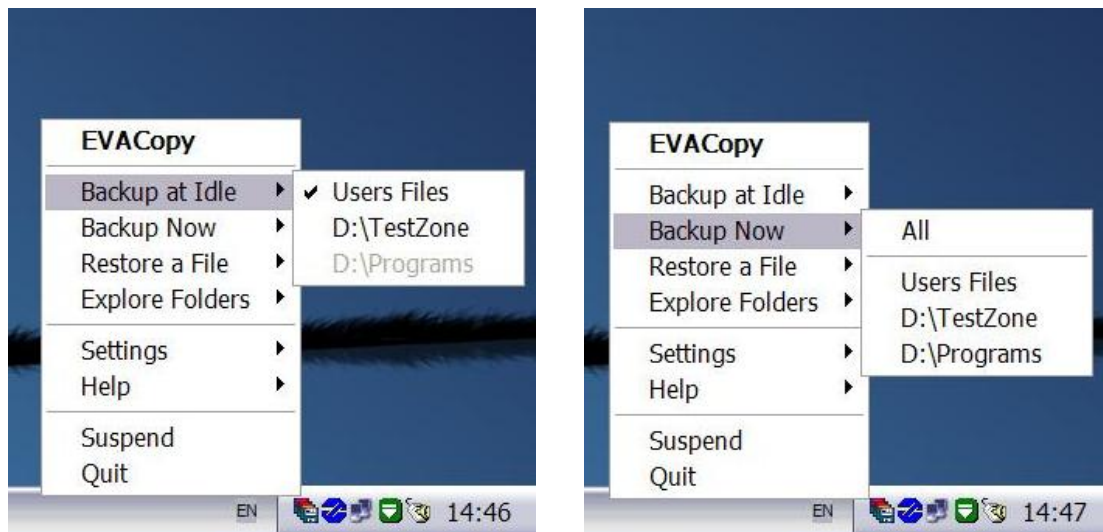
To get resident, `idle:` must be specified for at least one source|destination pair. The `idle` parameter accepts the following arguments:

`y` (or `yes`, not case-sensitive): 'idle' is active for this pair, but can be toggled off by the user.

`n`, omitted or otherwise: 'idle' is not active for this pair, but can be toggled on by the user.

`x` : 'idle' is inactive for this pair, and can't be toggled on by the user (disabled).

EVACopy Tray Menu



Clicking the headline does the same as the Help>About : It pops up an About box.

In the 'Backup at Idle' submenu, active pairs are checked, and Inactive pairs are unchecked. Clicking any pair toggles the 'idle' status on and off for that pair. Disabled pairs are grayed out, and you can't click them.

The 'Backup Now' submenu (or menu item, if only one pair is defined) enables you to manually initiate a backup job, either of a selected pair or of all pairs. Disabled pairs can be initiated too. While a backup job is in progress, the tray icon blinks (this behavior can be altered), and the tray menu is unresponsive to the user. Instead, the Emergency Break is operational - see later on.

The 'Restore a File' submenu (or menu item, if only one pair is defined) launches LaRve – a tool used to list files and their earlier versions, and restore them. See next topic for details. If the submenu is grayed, it means that LaRve.exe is not found.

The 'Explore Folders' submenu enables you to explore the folders associated with every pair – the Source, the Destination and the Evacuation.

The "Settings" submenu allows you to:

- 1) "Configure" EVACopy – launch EVAConfig.exe, which is EVACopy configuration utility. If this item is grayed, it means that EVAConfig.exe is not found.
- 2) "Edit" the settings file directly, using Windows default text editor.
- 3) "Reload" the settings after they were modified in either way.

The "Help" submenu is meant for displaying help (yet to be written), this user's manual, news from EVACopy website (<http://evacopy.sourceforge.net>), and information about the program.

You can suspend EVACopy with the "Suspend" item; the program is still running, but will not automatically initiate anything. However, you can still use the other menu items. Note: after "Reload", the program is still suspended.

You can also "Quit" EVACopy completely.

Note: when resident, EVACopy tray icon is used to launch the tray menu, and to show (in a tool tip) the current status of the program. When a backup job is in progress, the menu becomes inaccessible, and the icon's functionality changes – see page 19.

The [Resident] section in the settings file

This section sets options for the program when it remains resident. It is used only if `idle` is enabled for at least one pair.

`Language=` sets the language of the tray menu, messages, tooltips, etc. The argument is a 3-letter abbreviation of the language, default is blank for English. If invalid, then default English is used. Otherwise, a language file "`Lang-xxx.ini`" is used, where `xxx` is the argument specified.

For example, `Language=heb` will make EVA Copy use the file "`Lang-heb.ini`" – the Hebrew translation file. If the file is missing, default English is used. You can use the empty language file (named "`Lang-.ini`") to make your own translation (or add the `sys.admin` remark to the "About" box).

Creating your own translation file for another language is very easy:

- 1) Rename the "`lang-.ini`" file to "`lang-xxx.ini`", `xxx` being your language 3-letter abbreviation. e.g. for an Italian translation. rename it to "`Lang-ita.ini`"
- 2) Edit the file using any text editor. The file contents are self-explanatory.
- 3) Edit the main settings file "`EVA Copy.ini`":
 - a. Change the `Language` argument to (in this example) `Language=ita`
 - b. If you intend to use your own language for user's notes, in section `[UponCompletion]` change the string of `UserNotePrefix` to your language translation of the English word "before".
 - c. Similarly, in section `[Resident]` change the string of `BOSNote` to your language translation of the English expression "program start".

Next in the `[Resident]` section, `IdleTime` is the number of minutes the computer should be idle before an automatic backup job is initiated. Default is one minute.

`BlinkIcon` specifies if the tray icon should be blinking while a backup job is in progress.

`BOS` (Backup On Startup) specifies the pair(s) for which a backup job should be initiated when the program starts. Options are (`C` is the default):

`A`: backup jobs are initiated for All pairs.

`N`: No backup jobs are initiated.

`C`: backup jobs are initiated only for Checked pairs (with `idle:y` in the `[PathsX]` section).

`U`: backup jobs are initiated only for Unchecked pairs (with `idle:n` in the `[PathsX]` section).

`BOSNote` is the note that is attached to the jobs that are initiated at startup, according to the `BOS` argument.

The next three arguments specify if some tray menu items should be visible to the user. By default, the entire menu is visible; however, for a system administrator, it may be more comfortable if the regular user has no such a direct access to, for example, the Settings file. You can disable the following submenus: "Explore Folders", "Settings", and "Help".

Automatic Backup Jobs

Automatic backup jobs (that are initiated when idle time is reached) use somewhat different settings than those used by user-initiated backup jobs, regardless of what is specified in the settings file: • no action upon completion • no user note • no errors list • useless log file is removed; user is not notified if so happens.

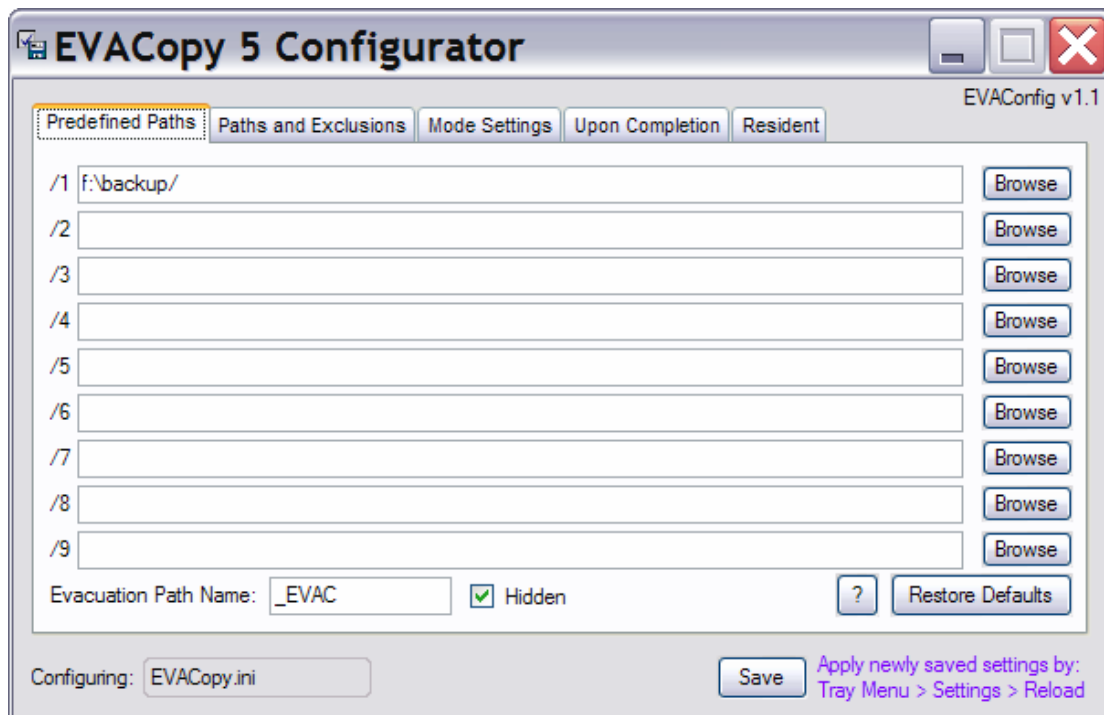
User-initiated backup jobs also have no action upon completion, but the rest is determined by the settings file. That's except "Backup Now → All", which is – in order to avoid interrupting a set of jobs – treated as an auto-initiated job.

EVAConfig: Configuration Tool

EVAConfig is EVACopy configuration tool, which enables you to modify settings files in a graphic user interface. EVAConfig can be launched in 3 ways:

- 1) Via the tray menu "Settings → Configure", to modify the currently used settings.
- 2) Independently (double-click EVAConfig.exe), in which case you need to select a settings file to modify.
- 3) When you launch EVACopy for the first time – or any other time when the paths are not configured – you are presented with the option to launch EVAConfig.

EVAConfig is an exact match for the settings file, where every section is displayed in a tab. The first tab is for the [PathPredef] section:



It's actually quite self-explanatory. You can either type-in the predefined paths, or browse for them.

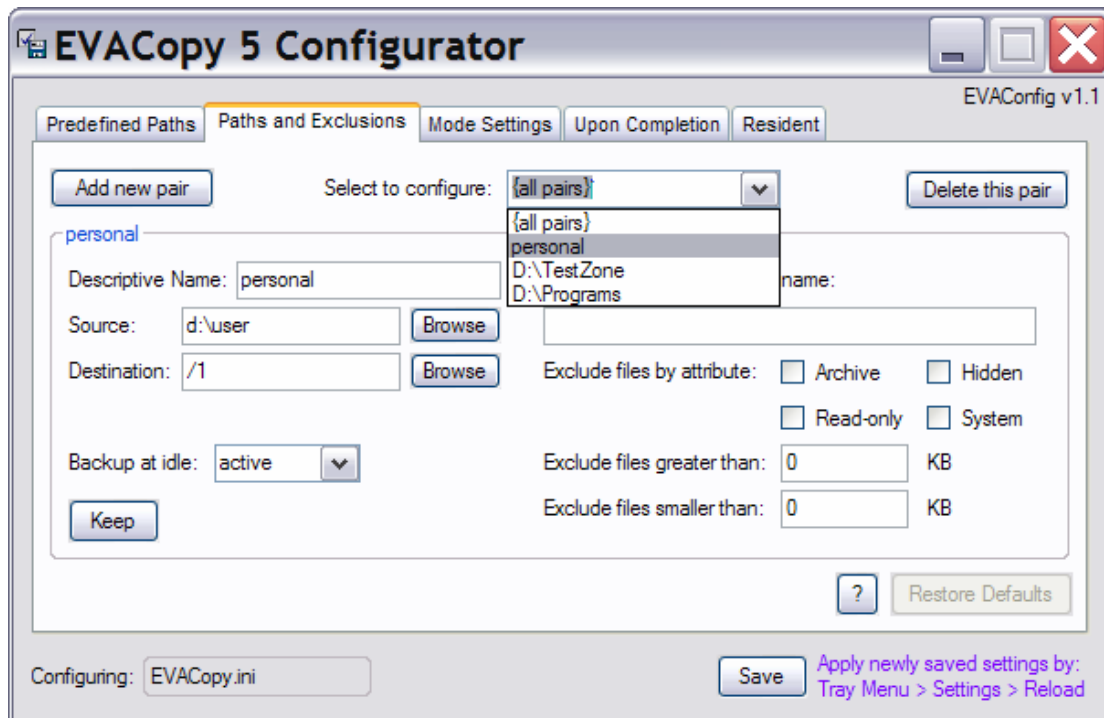
Note: you can hover the cursor over some items and get tips. For example, if you hover over the settings file name (the lower-left corner), the tip shows the file's full path.

Note: you can click the [?] button to get some help. This button is available in most tabs.

IMPORTANT: EVAConfig is completely independent of the currently running instance of EVACopy! Therefore – note the comment in purple in the lower-right corner – when configuration changes are made, to apply them you **MUST** revert to EVACopy tray menu and reload the settings, or re-launch EVACopy.

Also note that you do not have to close EVAConfig in order to apply the changes to EVACopy. EVAConfig can remain open as you change your settings and reload them, until the settings are satisfactory.

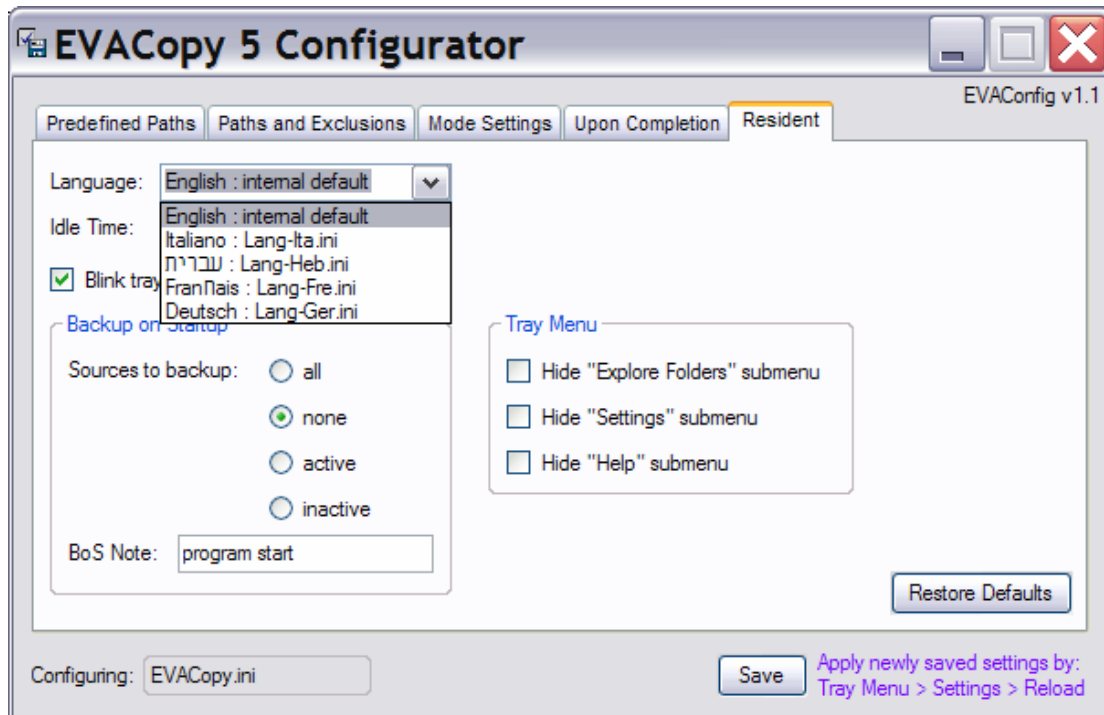
The second tab is for the [PathsX] section. Here you can add, modify and delete your backup pairs, as well as setting common exclusion rules:



IMPORTANT: to keep your changes, you must click "Keep" before you go about making changes to other settings, or else your changes will be lost !

The third and fourth tabs deal with [Mode] and [UponCompletion] sections, respectively.

The fifth tab handles the resident feature of EVACopy, i.e. what to do when EVACopy remains resident.



The language can be selected from a drop-down list. This list is built by EVAConfig according to the files named "Lang-??? .ini", which must already exist.

The language files are standard .ini files, where the first section is:

```
[General]
LanguageName=
```

This is where the name of the language is specified, and this is the name displayed in the list.

TLPD: Too Long Paths detector

EVACopy includes a limited method of avoiding too long paths, and warn the user if any exist. For a more comprehensive method to detect these issues, this additional tool is introduced.

TLPD was developed as a complementary product for EVACopy, but it's very useful as a standalone program as well. TLPD has its own sourceforge page, at:

<http://sourceforge.net/projects/tlpd/>

TLPD scans the directory tree, looking for files and folders with full path name longer than a specified threshold. The reason for this is the inability of Windows operating systems to handle path names longer than 255 characters. Some programs fail to work correctly with them, and TLPD is used to detect them for you, so you can move or rename them to shorter paths.

Standard Use

TLPD is included in the EVACopy archive. Start TLPD.exe, and then you are asked for the path to check and for the threshold to use. With these parameters the program gets to work. This may take a while, as TLPD traverses the directory tree. Finally, TLPD displays a log with a list of files and folders with too long path names.

The log is saved in the temporary folder by the name "TLPD-log.txt", so you can review it later.

Advanced Use

You can start TLPD with command-line parameters, the 1st is the path to check, and the 2nd is the threshold. For example:

```
TLPD.exe c: 200
```

This is intended for automated silent operation. This means that the log file is not presented to the user, only saved to disk. You can integrate TLPD into a script in this way, and make use of the exit codes described next.

You may also pass asterisk (*) as the first parameter (instead of a specific path), and TLPD will scan all drives (local and network) in your system. For example:

```
TLPD.exe * 200
```

If the command-line parameters are misconfigured, the program ignores them and switches to standard mode.

Tray Icon

TLPD displays a tray icon, indicating that the program is in progress. If you click it, the program is paused, and a menu appears where you can choose to resume or to terminate, or to leave it paused. When the program is paused, the icon blinks into a red X sign.

In silent mode, the tray icon is not displayed. However, you can see that the program is in progress by the TLPD.exe process in Windows Task Manager, and also terminate it from there. Remember that the silent mode is intended for advanced users.

Exit Codes

TLPD returns the following exit codes:

0 = Program completed successfully, too long paths were not found.

1 = Error in path parameter.

2 = Error in threshold parameter.

3 = Program completed successfully, too long paths were found!

9 = Program Aborted by user, by pressing "Cancel" when asked for path or threshold.

Open Source

TLPD is quite simple, written as an Autoit v3.3.0.0 script. The script includes the pseudo-code and remarks.

Enjoy!

LaRVe: Restore Tool

Brief

LaRVe (**L**ist **a**nd **R**estore **V**ersions) is a complementary utility for EVACopy.

LaRVe replaces FileDigger and extends your selective restore capability, beyond what Windows built-in search engine can offer. LaRVe is now the preferred restore method for backups made by EVACopy.

Purpose

The easiest restore procedure is simple - use Windows built-in search engine to search through the evacuation folder. The search results window lists all earlier versions of the file(s) you searched for, you only need to select the one you want.

However, sometimes it's not enough. Let's say you have some files by the same name in different folders, e.g. "..\john\letter.doc" & "..\smith\letter.doc". In this case, searching for earlier versions of "letter.doc" produces results for both john's and smith's files (and maybe others), while you are probably interested in john's letter only.

Windows can't search for all files "..\john\letter.doc", but LaRVe can.

LaRVe also displays the results in a much clearer view, and enables additional actions to be performed (like one-click restore).

How To Use

The best way to launch LaRVe is from EVACopy tray menu (Restore a File → [pair name]).

You can also launch LaRVe independently (LaRVe.exe), but then you need to point it yourself to the destination folder. Also, one-click restore is unavailable.

LaRVe then displays the recent backup - a list of files in the destination.

The list is sorted alphabetically. You can select the desired file, and:

- 1) Click [History] to list earlier versions of the selected file. LaRVe displays a list of earlier versions, of which you can select the file to restore.
- 2) Click [Open] to view the file's contents (and then save it somewhere else, or copy parts of it to the clipboard). Note: if you do not select any file and click [Open], then the folder itself opens in Windows Explorer.
- 3) Click [Copy] to copy the file to the clipboard, and paste it somewhere else.
- 4) Click [Compare] to compare the file to its copy in the source. "Compare" is currently unavailable - it will be implemented in the future. In any case, like "Restore", "Compare" is also unavailable if you launched LaRVe independently.
- 5) Click [Restore] to restore the file back to the source, overwriting the file by the same name, if exists.

If the file you want doesn't exist in the destination, it means that EVACopy performed a backup job after the file was deleted from the source, therefore the file was evacuated.

In this case, you can use the [<Back] and [Forth>] buttons to display lists of files evacuated in the past backup jobs. Click [<Back] until you observe the file you want to restore. Select the file, and select one of the options described above.

Click [Recent] to return to the recent backup, wherever you are in the history.

Click [Done] to return to the files list. When the files list is displayed, click [Done] to quit LaRVe.

MIFind: Missing-In-Action Finder

Brief

MIFind is a complementary utility for EVACopy. When EVACopy is deployed over multiple networked workstations, MIFind is used for alerting if any of the workstations did not perform a backup job recently.

MIFind uses the "main log" feature, introduced in EVACopy 5.2. This feature enables the creation of a main log file, where EVACopy is logging main events like "application start", "job start", "job done", etc. When multiple instances of EVACopy (running on multiple workstations) are configured to append to the same main log file, MIFind can analyze this log, compare to a list of active workstations, and alert if any workstation is missing.

How To Use

Step 1: Configure EVACopy on every workstation to use a main log file:

Create a file "MainLog.ini" in the same path where EVACopy.exe resides.
The file content should be as follows:

```
[General]
Path= (path of the main log file)
```

For example, the content of the included file "MainLog-example.ini" can be used when all workstations launch EVACopy from the same path, and have the permission to append to the main log file:

```
[General]
Path=%exePath%
```

Step 2: prepare a list of active workstations:

Create a file "MIAhosts.txt", in every line put a hostname of a workstation running EVACopy.

Step 3: Launch MIFind:

Launch MIFind.exe, it will read the hosts list from "MIAhosts.txt" and the main log file from the path specified in "MainLog.ini". MIFind will then create a report file "MIArep.txt", located in the same path (if appropriate permissions exist) or in your temporary folder (%temp%). The report details the count, hostnames and percentage of the workstations that did not perform a backup in the last 24 hours.

Any GUI messages you see have a timeout of 10 seconds, so MIFind will not hang when launched by a scheduled task, and you can review the report at any time.

To check for missing workstations in the last N hours, launch MIFind.exe with N as a command-line parameter.

You can include MIFind in a batch file, and operate on the exit code MIFind provides:

Exit code 0: all workstations are accounted for.

Exit code 1: some workstations are not accounted for.

Exit code 2: input error (main log file not found, hosts file not found, etc.)

Note: EVACopy vs. volume snapshots

Operating systems that support volume snapshots (most Linux distros, Windows Vista and later) are becoming more common for the personal computer. This is after a long period of time in which it was a feature of server operating systems only. The volume snapshot feature does have a side-effect of keeping earlier versions of files, mainly for locked files prevention. However, using volume snapshots does present some difficulties:

- For personal computers: The earlier versions are kept on the source. If you come to need your backup, most likely it's because the source was corrupted, so the volume snapshot is practically useless to you. In addition, using volume snapshots requires a knowledgeable user to configure and maintain.
- For enterprise computers (backing up to a volume-snapshots-enabled server): It's up to your system administrator to decide when to perform the backup. Usually performed once a day (just before the formal backup procedure), it lacks the ability to be activated by you, the user, specifically for your files. With EVACopy, you can backup your contents any time, in a matter of seconds, always keeping earlier versions.
- EVACopy can backup your files to any storage location - be it a file server, your office's NAS, a simple home network share, FTP-drive, USB disk drive, your personal disk-on-key, etc. No specially configured server is required, no specific operating system, no extra measures. Just use it.

However, the volume snapshots restoration interface (i.e. Windows Shadow Copy Client) is quite effective. EVACopy has learnt a lot from it. Users who are accustomed to using it at work, will find EVACopy restoration method very familiar.

Note: EVACopy vs. incremental and differential backup

There are two common types of backup that may seem similar to what EVACopy does. The incremental and the differential backup methods are very much the same, so we won't discuss the differences between the two here. But they both are completely unlike EVACopy.

Incremental and differential backup both overwrite changed files on the destination, and do not delete files from the destination. This has two drawbacks: the destination gets cluttered with junk (old files that were moved or deleted), and earlier versions don't get saved anywhere. This is exactly what EVACopy strives to prevent.

There's yet another genre of programs, called "synchronization" (rather than "backup"). These programs offer features like removing deleted files, reverse backup, combining source and destination folders, etc. But the few that I know to offer any efficient feature to preserve earlier versions are poorly designed, hard to deploy and configure, difficult to use, and costly.

Note: EVACopy vs. repositories

There is another technology for versioning, that relies on *repositories*. A repository ("repo" for short) is a folder located on a server (local or web) containing your files, so you don't have any files on your workstation. Instead, you have a *repository client* program installed. When you work on a file, the client retrieves the file from the repo. When you finish with the file and save it, the client saves it back in a way that the earlier version is preserved.

This method requires a repository server and client software installed on the workstations. This makes it a too complex solution, implemented mostly by software developers. The advantage of it is that it enables several users to work on the same file simultaneously, and being all the files located in a single repo, they are easier to manage.

Note: EVACopy and Linux

EVACopy is designed for the Windows operating systems. Linux has plenty of scripting tools to achieve the same goal, and the average Linux user is by far more knowledgeable than the average Windows user, to say nothing of the Linux system administrator. I presume Linux users can take care of their own backup methods. In any case, the source code is (at least, it used to be) quite simple. So if you have some scripting skills and you think EVACopy for Linux can be useful, feel free to write it down. Let me know how it works out...

Note: EVACopy source code

EVACopy is written as an AutoIt v3.3.0.0 script. It's full of remarks. Enjoy!

Note: EVACopy tray icon

The tray icon appears when EVACopy starts, and disappears when EVACopy ends. That's how you know when EVACopy is at work. A tool-tip that shows on mouse hover tells you what phase EVACopy is in (Starting, EVAC or Copy). Other than that, the tray icon can be used as:

Emergency Break

Double-click the tray icon to pull the emergency break and terminate EVACopy while running. This results in an exit code 9, and the action upon completion – if defined – is disabled. You are also notified what are the paths EVACopy was using, and you can observe the log file later on.

Note: When NOT to use EVACopy

In general, the contents in your computer divide into main three categories: *operating system*, *installed programs* and *personal contents*.

Don't use EVACopy to backup your operating system or installed programs

It's useless, considering the circumstances in which you would want to restore them. These contents are best handled by a disk/partition imaging software, like Ghost, Acronis, CloneZilla, etc. If no drive image is available, then the "restoration" is usually performed by reinstallation.

Avoid using EVACopy for very large and unchanging files

For example, don't use EVACopy to backup your media library (songs and movies) – media files very seldom change, and their history is useless. These contents require a one-time backup, so use regular CD's or DVD's.

Use EVACopy with caution for large data files

Some programs store all their data compressed in a single large file, rather than in separate files. The most notorious example is Microsoft Outlook, with its bloating PST files. EVACopy can't isolate changes inside files (like messages received, moved or deleted inside a PST file). This makes EVACopy the least-worst backup solution – because neither can any other program, and still earlier versions are kept. There are some workarounds to this, e.g. by using several PST files, or weekly removing earlier versions to external media. It's up to you – or your system administrator – to decide the best course of action.

To sum up

EVACopy is intended for backing-up folders of personal contents. These usually include many relatively small files, when some files are being created, updated or removed on an hourly to daily basis by the users themselves. For examples:

- Your work-related or personal documents and files, pictures etc.
- Files of a web site that you build and maintain
- Code of a program that you develop or test

As you probably already figured out, the operating system and the installed programs are most definitely NOT personal contents. By the way, **it is very important to make the distinction** between (for example) Microsoft Word as a **program** composed of files, and Word **documents** (*.doc files), which are personal. Even your media library, although connected with your personal preferences, are not personal (That's unless you are a musician, and you record your own songs – then they are personal contents indeed).

Best Practices for Effective and Fault-Free Use

- Avoid using very long names for your files and folders. This makes the full path name very long, and it may result in too long path names of files in subfolders. Windows can't handle path names longer than 256 characters (Windows XP and later).
- Avoid using UNC paths (i.e. \\server\share\folder), for the same reason. Use mapped network drive instead.
- Make sure you have full writing permissions to the destination folder.
- Be sure to close all programs that may be using a file in the source folder. If a file is used, it may be locked and EVACopy can't handle it. e.g. if your source includes an Outlook PST file, close Outlook. In contrast, you can leave your internet browser open, as well as any other program that does not use files from the source. Microsoft Office documents usually can be copied even when open, but if in doubt, close them.
- If you move (or rename) a file or folder within the source, EVACopy will treat it as deleted (from its original location) and therefore evacuate it. Also, EVACopy will see it as new (in its new location), and therefore copy it. This results in a waste of space in the evacuation folder, and a waste of time copying it again to the destination folder. To avoid this, try to remember that when you move or rename a large file or folder within the source, *replicate the same action in the destination*.
- EVACopy is a one-way backup. Do NOT make any changes to the destination folder, because they will not be replicated back to the source.
- If you are working on a long document or code, create a shortcut to EVACopy in the Quick Launch toolbar. After you make a major change to the document and save it, click the shortcut for quick backup (no need to close the document). Actually, this is very handy: While writing EVACopy, I could revert to earlier code when a dead-end came up, because I found myself using it on its own files...

Free, open-source, and all that other stuff

- EVACopy is free for any use: personal, commercial, or otherwise. No shareware, no demo, no paid "pro" version. FREE.
- "EVACopy" is NOT a registered trademark; it's just a nice name I came up with. Someone else may use this name in the future for some other product, software or service; I can't do anything about that, and I have no such intention. I trust you to know which is which.
- EVACopy is portable. Not so much in a sense that it can run from your disk-on-key, but it means that no setup is required, no administrative privileges, no messing around with system files or the registry.
- No need for Java, .NET, or any other system component preinstalled.
- EVACopy was developed and used under Windows XP, but it should work fine under any version of Windows later than 3.11. If you're having trouble with that (and you're sure it's not any trivial issue, like permissions and stuff), let me know.
- No internet connection is needed, and EVACopy will make no attempt to connect to the internet. This also means that EVACopy does NOT check for updates. Hopefully, there won't be many – I strive to make it perfect in the first attempt. Well, I said that when the first version was released. Now we're on the 16th (!!!) Release... So keep an eye on the website every now and then, just in case.
- EVACopy is GPL. There's only the obvious no-liability thing (which means EVACopy can NOT be liable for any damage you may have inflicted upon your system). There is no nagging and useless EULA, and no lawyers were involved here. How bored do you think I am? ☺

Your Feedback is Valuable !

Visit <http://evacopy.sourceforge.net> for news and updates, and contribute your own remarks, wishes and bug alerts.

I hope you'll enjoy using EVACopy at least as much as I enjoyed writing it !