

SortCopy

Thomas Richter

COLLABORATORS

	<i>TITLE :</i> SortCopy		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Thomas Richter	August 10, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	SortCopy	1
1.1	SortCopy Guide	1
1.2	The THOR-Software Licence	2
1.3	About SortCopy	2
1.4	Installing SortCopy	3
1.5	Command line options	3
1.6	Backup Examples	5
1.7	Warning and Error Messages	5
1.8	Thank you, folks	7
1.9	The history of SortCopy	7

Chapter 1

SortCopy

1.1 SortCopy Guide

SortCopy Guide

Guide Version 1.14 SortCopy Version 1.34

Table of Contents

I. **The Licence**

Read This First!

II. **What is it: Overview**

What it does...

III. **Installation**

How to install SortCopy.

IV. **Command Line Options**

How to invoke and use SortCopy.

V. **Examples**

Command line examples for easy backup.

VI. **Warning and Error Messages**

...and what they mean.

VII. **Thanks**

Special "thank you"'s go to...

VIII. **History**

What happened before...

© THOR-Software

Thomas Richter

Rühmkorffstraße 10A

12209 Berlin

Germany

E-Mail: thor@einstein.math.tu-berlin.de

WWW: <http://www.math.tu-berlin.de/~thor/thor/index.html>

SortCopy is FREEWARE and copyrighted © 1993-2000 by Thomas Richter. No commercial use without permission of the author. Read the **licence** !

1.2 The THOR-Software Licence

The THOR-Software Licence (v2, 24th June 1998)

This Licence applies to the computer programs known as "SortCopy" and the "SortCopy.guide". The "Program", below, refers to such program. The "Archive" refers to the package of distribution, as prepared by the author of the Program, Thomas Richter. Each licensee is addressed as "you".

The Program and the data in the archive are freely distributable under the restrictions stated below, but are also Copyright (c) Thomas Richter.

Distribution of the Program, the Archive and the data in the Archive by a commercial organization without written permission from the author to any third party is prohibited if any payment is made in connection with such distribution, whether directly (as in payment for a copy of the Program) or indirectly (as in payment for some service related to the Program, or payment for some product or service that includes a copy of the Program "without charge"; these are only examples, and not an exhaustive enumeration of prohibited activities).

However, the following methods of distribution involving payment shall not in and of themselves be a violation of this restriction:

(i) Posting the Program on a public access information storage and retrieval service for which a fee is received for retrieving information (such as an on-line service), provided that the fee is not content-dependent (i.e., the fee would be the same for retrieving the same volume of information consisting of random data).

(ii) Distributing the Program on a CD-ROM, provided that

a) the Archive is reproduced entirely and verbatim on such CD-ROM, including especially this licence agreement;

b) the CD-ROM is made available to the public for a nominal fee only,

c) a copy of the CD is made available to the author for free except for shipment costs, and

d) provided further that all information on such CD-ROM is redistributable for non-commercial purposes without charge.

Redistribution of a modified version of the Archive, the Program or the contents of the Archive is prohibited in any way, by any organization, regardless whether commercial or non-commercial. Everything must be kept together, in original and unmodified form.

Limitations.

THE PROGRAM IS PROVIDED TO YOU "AS IS", WITHOUT WARRANTY. THERE IS NO WARRANTY FOR THE PROGRAM, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IF YOU DO NOT ACCEPT THIS LICENCE, YOU MUST DELETE THE PROGRAM, THE ARCHIVE AND ALL DATA OF THIS ARCHIVE FROM YOUR STORAGE SYSTEM. YOU ACCEPT THIS LICENCE BY USING OR REDISTRIBUTING THE PROGRAM.

Thomas Richter

1.3 About SortCopy

SortCopy is a tiny shell-only backup program, it copies one or multiple files and directories, like the usual "copy" command does.

This is absolutely nothing new, but unlike other copy tools, this one tries to copy links as well. If the link source and the link destination is copied, "SortCopy" will erase the link destination and re-create the link to the copy of the link source. This works for both, SoftLinks and HardLinks, as well as links to directories as long as the directory and the link to the directory is copied.

As a tiny extra, it keeps the destination directories alphabetically sorted.

SortCopy comes with a lot of backup-related options not available with "copy". You may choose to copy only files with the "a" (archive bit) reset, only files with a newer creation date than in the backup, remove obsolete files from the backup or set the archive bit of all copied files.

All this makes "SortCopy" useful for a very simple backup procedure, as it re-creates a one to one image of the source directory on a file system level. It doesn't write cryptic binary backup files, it doesn't pack files in any way, so restoring the backup works even without SortCopy.

"Simplest is safest and best!"

WARNING: Even though "softlinks" are partially integrated into the OS, this does not mean that they are fully working! I found that even the latest FFS version, V43.19, has problems with them and may invalidate your HD. This is NOT a SortCopy problem at all. So "use softlinks at your own risk"!

1.4 Installing SortCopy

There's not too much that needs to be said about installation - as SortCopy doesn't require any additional files, copy it wherever you want to keep it. The "C:" directory is probably a good choice.

If you like to keep this guide, copy it to a safe place - that's all.

1.5 Command line options

SortCopy comes with a rather big bunch of command line options, here's the command template:

```
SortCopy FROM/A,TO/A,QT=QUIET/S,QTR=QUIETER/S,SETA=SETARCHIVE/S,NONA=NONAONLY/S,OVR=OVERWRITE/S,DEL=DELETETECOPIED/S,FORCE/S,UPD=UPDATE/S,PRB=PURGEBACKUP/S,PRO=PURGEONLY/S,BUFSIZE/K/N
```

Now, what does all this mean:

FROM: Source specifications.

Wildcards are allowed here. Unlike older versions of SortCopy, the 1.20 release copies also files to the destination.

If the FROM argument is the name of a directory and not a wildcard, SortCopy will NOT create a destination directory of the same name, but copy the contents of this directory to the destination location. This is similar to the logic of the usual "copy" command and backwards compatible to the older releases.

A tiny warning about links: Both types of links are not resolved "across patterns on the root level" - that is, if the pattern "foo?" matches the link "foo1" to "foo2", both at the root level of the directory to be copied, SortCopy won't be able to rebuild this link.

The solution to this restriction is simple: Just copy the complete directory that contains both, the link and the link destination.

TO: Destination of the copy operation.

No wildcards allowed. The argument must match a valid directory and not a file, obviously. If the destination directory does not exist, SortCopy will create one for you.

QUIET: Suppresses printing the file names being copied.

QUIETER: Suppresses all messages except errors and warnings.

SETARCHIVE: Set the archive bits of the copied files afterwards.

This option, together with the next, NONAONLY, is very useful to create incremental backups. Once SortCopy has set the archive bit of all copied files, creating or modifying files will clear their archive flag afterwards. SortCopy can be instructed to copy only these - not yet backup'd - new files.

NONAONLY: Copy only files with the archive bit cleared.

This is the second option useful for incremental backups. If the archive flags are always set during a backup with SETARCHIVE, the new and changed files will have their archive bit cleared. Giving "NONAONLY" as an option will thus instruct SortCopy to copy only new or modified files.

Directories are looked in, regardless of their archive bit. If they contain any file with an unset archive bit, this file is of course copied as well.

OVERWRITE: Overwrite destination files in any case.

Tell SortCopy to overwrite all files in the destination, regardless whether they are write protected or not. You should specify this option if you're using SortCopy as backup utility, or SortCopy won't be able to keep the write protected files up to date - simply because it won't copy a newer version on top of the backup otherwise.

Another side effect of this option is that SortCopy will replace directory entries even if they are of different type, i.e. it will automatically erase a directory of the same name if a file has to be copied on top of it, or it will erase a file to be able to replace it with a directory. Usually, SortCopy will fail in case the directory structure has changed in this way.

LEAVEEMPTY: Don't delete empty directories in the destination.

Instruct SortCopy even to keep empty directories in the backup. The default operation is to delete an empty backup directory - because it is not needed.

DELETECOPIED: Delete files after copying them.

Perform a "move" operation: SortCopy copies the specified pattern in a first step and erases the source afterwards.

WARNING: This operation is dangerous!

SortCopy is smart enough to cancel the deletion process if any trouble occurs during backup - the source is only erased if every single file in the source directory could be backup'd.

FORCE: Delete also files that are protected from deletion.

This is the "OVERWRITE" equivalent for file deletion. If this argument is present, SortCopy will even erase write protected files; either as part of the "purge" process or during erasing the source with "DELETECOPIED".

This option should be used in conjunction with DELETECOPIED, PURGEBACKUP or PURGEONLY.

UPDATE: Update only files with newer creation date.

This is another way of managing incremental backups: SortCopy will only copy those files to the destination that are not "up to date" - that is, differ either by creation date or file size. SortCopy will not, however, perform a byte-by-byte comparison to find out if the files are really identical; this operation would be much too slow.

PURGEBACKUP: Remove old files from the backup.

SortCopy will delete files from the destination tree that are no longer present in the source. This operation is useful to keep the backup consistent with the source - it would accumulate all the old garbage otherwise.

REMARK: If even write protected obsolete files should be deleted, specify the "FORCE" option as well.

PURGEONLY: Remove only old files, but do not copy.

This option is similar to **PURGEBACKUP** except that no copy operation is performed. The destination is scanned for old files to be deleted.

It's again worth to consider the "FORCE" keyword to remove even write protected obsolete files from the destination.

BUFSIZE: Specify the buffer/chunk size for the asynchronous copy mode.

This specifies the buffer size for the asynchronous copy mode. Source data will be written in chunks of this number of bytes, and written back as soon as the data arrives while new data is read in. Read and write operations will overlap for optimal performance and dependent on the speed of the source and destination, SortCopy might hold more than one buffer in memory.

Unfortunately, tests have shown that the gain of parallel IO is minimal on my system, so the chunk size is by default as large as possible. Thus, without this option, parallel IO is effectively disabled. Try which buffersize works best for your system.

1.6 Backup Examples

Here's the first example how I use SortCopy. This is actually a small script, taken from my S: drawer:

```
SortCopy FROM SCSI: TO DH3:SCSI QUIET SETARCHIVE NONAONLY OVERWRITE
```

Copy all files from my HD (called SCSI) to DH3, into a directory called "SCSI".

Be QUIET - don't print too much information, print only directory names and not file names.

SETARCHIVE and NONAONLY are used to perform a consistent incremental backup, OVERWRITE is given even to overwrite write-protected files in the destination.

The example above has the drawback that it keeps even old files in the destination that are no longer present in the source. The following command line would do that for you:

```
SortCopy FROM SCSI: TO DH3:SCSI QUIET SETARCHIVE NONAONLY OVERWRITE PURGEBACKUP FORCE
```

In addition to the line above, SortCopy is instructed to remove obsolete files from the destination ("PURGEBACKUP"), even if they are write protected ("FORCE").

The following line restores a backup back to my HD:

```
SortCopy FROM DH3:SCSI to SCSI: QUIET OVERWRITE
```

It doesn't set any archive bits in the backup, given here as "FROM" argument. Why should it?

The following command line will "cleanup" my backup by removing all old files, even if they are write or delete-protected:

```
SortCopy FROM SCSI: to DH3:SCSI PURGEONLY FORCE
```

1.7 Warning and Error Messages

Here are a couple of warnings SortCopy may spit out and you may wonder what they mean. This list is not complete, I removed all the "obvious" examples.

Increase the stack size with "STACK" and retry.

SortCopy run out of stack space. This happens only if the directory you want to copy is extremely deep - SortCopy is actually very careful about its stack usage. I guess several hundred levels are needed to produce this warning!

You might then either try to increase the stack size a bit, like with the command

Stack 8192

or should probably check your HD with a HD tool like DiskSalv - something might be broken and caused SortCopy to run into an infinite recursion.

Destination "." is not a directory.

You specified either a file as destination - SortCopy expects a directory to put the archive into - or the backup contains a file that has the same name as directory in the source, at the same relative position.

Consider, for example, the directory "bla" contains an entry "foo" both in the source and the destination, but the source "foo" is a directory and the destination "foo" is a plain file.

To get rid of the warning in this case, use the "OVERWRITE" option to tell SortCopy to delete the destination file.

"OVERWRITE" won't help in the opposite situation: A destination directory needs to be replaced by a file. SortCopy will now warn you like this:

Can't create the destination file "...": object already exists

and will abort copying this file. I guess it's better to leave the backup directory alone and to check manually if it contains something of worth. If not, delete it by hand and restart the backup process.

Warning: The name "." of the link is already in use.

SortCopy found a link to a directory in the source, but a file/directory or link of the same name exists already in the backup directory.

SortCopy will not try to resolve this link because it's not clear where the destination link is going to, neither if it is a link at all.

TIP: Try to avoid links to directories, they are good for surprises!

Can't create dummy link directory ".."

SortCopy doesn't copy links to directories immediately. Instead, it creates an empty "dummy" file as place-holder and deletes this place holder afterwards when the actual linkage is performed. This message tells you that the dummy file can't be created, for whatever reason. SortCopy will nevertheless try again to resolve the link later on.

"Can't resolve link ".."

SortCopy found a link in the source directory, but was unable to find out where this link is going to. This might happen with some softlinks because their current implementation in the OS (not in SortCopy, though) is rather "flaky".

TIP: Avoid softlinks, try to use hardlinks.

Warning: Link destination of ".." not copied, using the original path.

This warning message is generated during the last step of the copy process, the reconstruction of the links on the backup media.

SortCopy was unable to find the object a copied link is going to in the backup directory. But since both the backup directory and the dereferenced object are both on the same volume, SortCopy creates a link to the same object the source link is going to.

That might require some explanation: Consider two files, "org" and "link", where "link" is a hard link pointing to "org". If "org" and "link" sit in different directories and you create a backup of the "link" home directory, SortCopy won't obviously find the "org" file in the backup just made. Instead, it will create a link in the destination that goes to "org", too.

That works, however, only if the source and destination directory reside on the same volume.

Warning: Link destination of ".." not present, using a copy instead.

This is a similar warning message that can show up when SortCopy is building the links.

SortCopy found that it needs to create a link, but the object the link is going to was not found on the destination directory. SortCopy will instead not create a link but keep a copy of the linked object on the destination.

Again, some background information: The situation is again the same as above, with two files "org" and "link" pointing to "org". If only "link" is backup'd, SortCopy won't find "org" on the destination. If the source and the destination volume do not coincide, it can't even create a link to the same object, but keeps a copy of "org" on the destination volume.

The difference between this warning and the last is that the destination and source volumes coincide for the former warning, but not for this one.

Warning: The archived directory link ".." was left out.

SortCopy couldn't create a link to a directory, but since this directory was archived anyways, it was left out.

The situation is as above, except that the links are going to directories, not files. SortCopy doesn't try to copy the complete directory contents in this case, but seeks other solutions. The simplest is if the linked directory is archived anyways, so a link is not really required.

Warning: Created an empty directory for ".." instead.

SortCopy couldn't create a link to a directory, probably because the linked object hasn't been copied to the backup directory. SortCopy created an empty directory as destination.

This is again quite the same situation, with links between directories. But even worse, the linked directory in question was neither archived, nor included in the backup. SortCopy created an empty directory in the destination as "replacement".

This might be not fully satisfactory, but copying the complete directory might be a bit dangerous as it could easily fill up the destination or create an infinite recursion. If this warning is generated and you really want to keep a backup of this link, copy the contents manually from the source to the directory SortCopy prepared for you.

TIP: As I said: Try to avoid links to directories.

1.8 Thank you, folks

Special thanks go to....

- Will Stanton for his useful ideas and improvements about this tiny tool, which is getting less and less tiny for that reason. (-:

Special thanks does NOT go to...

Commodore Amiga for their @^&%! - computer (adverb censored). This ugly thing still fails to work correctly. Now my Chip Mem starts to get "creative" (bits start flipping). It's a CREATIVE computer, you know...)-:

Viscorp for their GREAT support.... my computer is still to big for a set-top box (-;

1.9 The history of SortCopy

New in release 1.06:

-Added the UPDATE command line option. See above for details.

New in release 1.07:

-SortCopy will now create a destination directory if it isn't already present. - The archive bit messages are now a bit shorter.

New in release 1.20:

Quite a lot: (Thus, the immense increase in the version number)

- Added the PURGEBACKUP and PURGEONLY flags. - SortCopy can now copy files as well. - Links will be resolved across different wild card matches. This did not work before. - Rewrote major parts of this tool, code cleanup and minor bug fixes. - SortCopy is now "pure", i.e. can be made resident.

New in release 1.21:

- SortCopy handles now safely the situation where a file in the destination must be replaced by a directory from the source of the same name. - Introduced short-cuts for most command line options. - Wrote this guide.

New in release 1.22:

- SortCopy 1.21 showed a wrong error message in case the wildcard parsing failed. Fixed.

New in release 1.23:

- Fixed a bug in the directory scanner that could cause enforcer hits in case of a failure. - The destination directory is no longer created in case the source is not readable.

New in release 1.24:

- An obsolete directory gets deleted if a file of the same name is present in the source and the "OVERWRITE" keyword is given on the command line. Please note that SortCopy will NOT overwrite files by directories or directories by files if OVERWRITE is not given. - The meaning of the UPDATE option changed a bit. The file dates have no longer to be identical, but the backup must be older than the original to be replaced by a newer version.

New in release 1.25:

- Fixed a bug in the "OVERWRITE" keyword.

New in release 1.26:

- Fixed again some bugs in the link resolve routine, and a bug in the directory scanner.

New in release 1.27:

- Only cosmetical changes.

New in release 1.28:

- SortCopy returned a wrong error message in case the source pattern did not match any file.

New in release 1.30:

- SortCopy uses now optionally an asynchronous copy mechanism, it will read and write data in parallel.

New in release 1.31:

- SortCopy forgot to generate an error requester in case the asynchronous copy operation failed. Fixed.

- Included the AWPipe/Rexx based SortCopy GUI by Niels Goers, requires Os 3.5, Reaction and the AWPipe 2.22 or better. Thanks, Niels! AWPipe can be found here: <http://www.btinternet.com/~bsteers/>

New in release 1.32: OK, this is now really the last release.

- I found a fatal bug in the copy routine, it failed to copy files larger than the available memory. Fixed.

- Included another GUI, thanks Alex!

New in release 1.33: Sigh. Still not the last one.

- Found again a fatal bug in the copy routine which did not handle ^C properly and could hang. *sigh* Fixed.

New in release 1.34:

- Found still another bug in the copy routine. It might have run into a hangup condition in case a file of length 0 was copied, and it still might have hung under certain other conditions, including a ^C.
