

COPYFILES

Stored on: Library>CopyFiles.Tedit, .Press, Interpress
Last Modified: 8-Oct-84
By: Sybalsky, Masinter

This package makes it easy to copy or move groups of files from one place to another.

(COPYFILES *source destination options*)

Copies the files designated by *source* to the place designated by *destination*. *source* is a pattern such as given to DIRECTORY or DIR; it can also be a list of file names. *destination* is either a directory name, or a file-name pattern, with a 1-1 match of "*"s in *to* to "*"s in *source*. (The number of *s in each *source* pattern needs to match the number of *s in each *destination* pattern.) The argument *options* is a (list of) options (if you have only one and its an atom, you can supply it as an atom), as follows:

1. You can specify how verbose COPYFILES is about what it is doing:

QUIET don't print anything while working.

(OUTPUT file)

print output on file. (OUTPUT T) is the default.

TERSE only print a "." for each file moved/copied.

2. you can specify whether COPYFILES should ask before each transfer. Default is not to ask.

ASK ask each time before moving/copying a file (default is to not ask).

(ASK N) Ask, with default to No after DWIMWAIT seconds.

(ASK Y) Ask, with default to Yes after DWIMWAIT seconds.

3. COPYFILES normally uses COPYFILE to create a new file. It also usually only copies the "highest version", and creates a new version at the destination. Alternatively, you can specify any of the following:

RENAME or MOVE

use RENAMEFILE instead of COPYFILE, i.e., the source is deleted afterwards.

ALLVERSIONS

Copy all versions, and preserve version numbers.

REPLACE

If a file by the same name exists on the destination, overwrite it (don't create a new version)

4. After COPYFILES gets done, it can be instructed to delete some files afterward:

PURGE

This involves a separate pass (afterwards): any file on the *destination* which doesn't have a counterpart on the *source* is deleted.

PURGESOURCE

converse of PURGE (and used by it): if the file is on the source and not on the destination, delete it.

COPYFILES normally compares the creation dates of the file on the source and any matching file on the destination to determine whether it is necessary to copy. The following options are mutually exclusive:

- > copy only when a file by the same name but an earlier creation date exists on the destination.
- >A Similar to >, but also copy if the file doesn't exist on the destination, i.e., > ALWAYS.
- # Copy only when a file by the same name but a different creation date exists on the destination.
- #A Similar to #, but also copy if the file doesn't exist on the destination, i.e. > ALWAYS.
- =A Copy if there isn't a file of the same name on the destination; if there is, leave it, but if the file on the source doesn't exist at all on the destination, *do* copy it.

Not all combinations of options make sense; for example, ALLVERSIONS probably doesn't work right with any date comparison algorithms.

Examples:

```
(COPYFILES '{ERIS}<MASINTER>*.MAIL '{PHYLUM}<MASINTER>OLD-*.MAIL)
```

will copy the any mail file on {Eris}<Masinter> to {Phylum}<Masinter>, renaming FOO.MAIL to OLD-FOO.MAIL.

```
(COPYFILES '{ERIS}<MASINTER>*.MAIL '{PHYLUM}<MASINTER>OLD-*.MAIL 'RENAME)
```

will use RENAMEFILE instead.

```
(COPYFILES '{{DSK}TEST {DSK}WEST) '{PHYLUM}<MYDIR>)
```

will move the files TEST and WEST from {DSK} to {PHYLUM}<MYDIR>.

```
(COPYFILES '{PHYLUM}<LISPARS>*.AR '{PHYLEX:}<LISP>ARS> '>A)
```

will copy all of the ARs on {PHYLUM}<LISPARS> to the Phylex NS file server; if any are already there, it won't bother copying them.

```
(COPYFILES '{PHYLUM}<LISPARS>AR.INDEX '{DSK}AR.INDEX '(>A REPLACE))
```

will copy the AR index to {DSK}, replacing any older version that is already there.

```
COPYFILES({DSK}*.; {FLOPPY}) will copy all files on {DSK} with no extension to {FLOPPY}.
```

```
(COPYFILES '{ERIS}<LISPUSERS> '{PHYLUM}<LISPUSERS> '#A PURGE))
```

will make {Phylum}<LISPUSERS> look like {ERIS}<LISPUSERS>; bringing over any file that isn't already on Phylum and then deleting the ones that were on {PHYLUM} and aren't on {ERIS} any more.