

---

---

## NSCOPYFILE

---

---

By: Bill van Melle (vanMelle@Xerox.com)

The module NSCOPYFILE modifies COPYFILE so that if both the source and destination files are on NS file servers, the copying is done by an NSFiling-specific copy routine. This routine copies *all* attributes of the source, including non-standard ones, such as those used by Viewpoint. Thus, you can safely copy Viewpoint files from inside Lisp without losing information. In addition, if the copy is from an NS file server to itself, the copy is performed by the server itself, which is considerably faster than shipping the file over the Ethernet.

In addition, you can also copy entire directories, by specifying directory names as the source and destination, e.g.,

```
(COPYFILE "{FS:}<Carstairs>Lisp>" "{FS:}<Calvin>Lisp>Current>")
```

The destination directory must not already exist, since this operation creates an entirely new directory, whose contents are a copy of all the source directory's offspring, to all levels. If the destination directory happens to exist but has no children, it is considered vestigial and is quietly deleted first (Lisp usually suppresses such directories when performing a directory enumeration).

You can also use RENAMEFILE in the same manner to either rename a directory, or to move an entire directory and its descendents to a new node in the file server's hierarchy, or to a new server altogether. You must, of course, have access rights to delete the source directory *and* all its children, and the destination must be on an NS file server.

A word about protection: when a file is copied or moved, the new file is given "defaulted" access rights, i.e., its protection is set as specified by its new parent (sub)directory, just as if you had created the file afresh by any other means. Thus, if the original file happened to have its own explicit protection, that protection is ignored. When copying or moving an entire directory, only the top-level directory receives default protection, so if any individual descendent file had non-default protection, that protection is copied verbatim. This can lead to confusion—you may want to use the NSPROTECTION module to change the new directory's descendents to default protection. See the documentation of NSPROTECTION for more discussion about protection issues. Note that if a file/directory is renamed within the same parent directory, the operation is considered merely "changing the name", and its protection is left unchanged.

Note: If you are using the FILEWATCH module, be aware that files being copied between NS servers do not appear (because the files are not opened by the normal Lisp file system).