

# **ProfFileSystem documentation**

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i> ProfFileSystem documentation	
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>
WRITTEN BY		November 23, 2024
		<i>SIGNATURE</i>

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1 ProfFileSystem documentation</b>	<b>1</b>
1.1 ProfFileSystem Manual	1
1.2 distribution	2
1.3 disclaimer	2
1.4 introduction	2
1.5 features	3
1.6 System Requirements	3
1.7 installation	4
1.8 quickstart	4
1.9 install20	4
1.10 install30	4
1.11 turbo	5
1.12 mountlist	5
1.13 usingpfs	5
1.14 format	6
1.15 access	6
1.16 Harddisk use	6
1.17 Harddisk	6
1.18 Harddisk	7
1.19 Harddisk	7
1.20 Harddisk	8
1.21 difference	8
1.22 compatibility	9
1.23 restrictions	9
1.24 faulttolerance	9
1.25 virus	10
1.26 tools	11
1.27 bugs	11
1.28 credits	11
1.29 author	12
1.30 registration	12

# Chapter 1

## ProfFileSystem documentation

### 1.1 ProfFileSystem Manual

Professional File System

Version 9.5 10-10-1994

(c) 1993/94 by Michiel Pelt

Disclaimer and distribution conditions

- Distribution Conditions
- Disclaimer

Introduction

- Introduction
- Features
- System Requirements
- Version History

Installation

- Installation
- Quickstart

Usage

- Using PFS disks
- Using PFS on harddisk

Special features and extras

- Faulttolerance
- Tools included
- Programmers Info

Miscellaneous

---

Repairing virus damage  
Bugs  
Credits  
Author

Registration

Registration  
Registration form

## 1.2 distribution

### Distribution Conditions

This program is copyright (c) 1994 by Michiel Pelt. It may be distributed freely under the condition that no profit is gained from its distribution, nor from any other program distributed in the same package.

Permission is granted to distribute this package by Bulletin Board systems or network sites, under the condition that no fee is charged on downloading it.

All files that are part of this package have to be distributed together and none of them may be changed in any way other than archiving or crunching. The only exception to this rule are the icons, including the tooltypes, which may be changed to accommodate them to the rest of the disk it is distributed on.

This program is distributed as shareware. If you like this product you should register.

## 1.3 disclaimer

### Disclaimer

These programs are provided "as is" without warranty of any kind. All risks involved using these programs are entirely yours. The author assumes no responsibility or liability whatsoever for any damage or dataloss caused by using this package. Although everything has been done to make sure the program is fully functional, the author cannot guarantee this for a 100%.

## 1.4 introduction

---

## Introduction

The standard AmigaDos filesystem (OFS) is not very fast. Commodore has tried to improve the performance of their filesystem a number of times. Workbench 2.0 added the fast file system (FFS), and now, with Workbench 3.0, we have the fast file system with directory caching (FFS-DC). This makes directoryscan faster but writing to disk and directory-operations like delete and rename are slower than ever.

This made me decide to write a filesystem that does better. I used all the knowledge I gained from my computer science study to make it as good as possible.

## 1.5 features

### Features

The professional file system (PFS) has the following features:

High performance

- write: 3-5 times faster
- read: 50% faster
- directoryscan: 10-20 times FFS, 3 times FFS-DC
- delete/rename/protect etc: 10-20 times faster

Other features

- full AmigaDos compatibility
- upto 10% more data on a disk
- better faulttolerance
- parrallel access almost without performance loss
- disks NEVER get invalidated

This is the first version of the filesystem that supports harddisk partitions. Partitions up to 31MB are supported. Both double and high density floppy disks are supported.

## 1.6 System Requirements

### System requirements

PFS will work on any Amiga with kickstart 2.0 or higher. It accepts both double- and highdensity diskdrives and RDB (Ridgit Disk Block) harddisks. The amount of memory PFS uses depends on the contents of the disks present. It will vary between 30 and 60K per floppy-drive and up to 300K for a harddisk partition.

---

## 1.7 installation

Installation  
-----

Quick start  
Installation on Workbench 2.0  
Installation on Workbench 2.1 - 3.1  
Turbo users  
Making and changing mountlist entries  
Installing PFS on harddisk

## 1.8 quickstart

Quick start

If you want to try PFS first, just doubleclick on the "PF0" icon. "PF0:" now refers to PFS disks in drive 0. Now proceed to using PFS disks about formatting and using PFS disks.

## 1.9 install20

Installation on Workbench 2.0

First you have to boot from your standard systemdisk or harddisk. Now execute the "install2.0" script by calling it from the shell or doubleclicking its icon. This script will copy the filesystem to the l: directory of your systemdisk and add two entries to your mountlist, PF0: and PF1:.

To use the filesystem you have to mount it using the mount command:

```
MOUNT pf0:  
MOUNT pf1:
```

"PF0:" and "PF1:" now refer to PFS disks in drive 0 and 1. If you want to have PFS mounted on startup you should add these commands to your startup-sequence.

## 1.10 install30

Installation on Workbench 2.1/3.0

---

First you have to boot from your standard systemdisk or harddisk. Now execute the "install3.0" script by calling it from the shell or doubleclicking its icon. This script will copy the filesystem to the l: directory of your systemdisk and put two files in your SYS:Storage/DOSDrivers directory called PF0 and PF1. Doubleclicking their icons will mount the filesystem on the corresponding drive. "PF0:" and "PF1:" then refer to PFS disks in drive 0 and 1. If you want to have PFS automaticly mounted on startup you can move the PF0 and PF1 files to your Devs:DOSDrivers directory.

## 1.11 turbo

### Turbo users

If you are the proud owner of a 68020+ machine you can, after installing the filesystem normally, replace the 68000 version in the l: directory with the 68020+ version (you can find this in the l drawer). Make sure the name of the filesystem is the same as stated in the mountlist. You can do that by renaming the filesystem in "ProfFileSystem" or by changing the mountlist.

## 1.12 mountlist

### Making and changing mountlistentries

If you have more than two drives, or you can't use the RDB for harddisk installation, you can create your own mountlistentries. Note that the "Buffers" keyword has no meaning with this version of the filesystem. Its merely added for future compatibility. Make sure that you set the "Unit" right. If you are making a harddisk mountlist entry, doublecheck all entries before trying to mount! Please refer to the DOS manual for more details.

## 1.13 usingpfs

### Using PFS disks

-----

Formatting disks  
Accessing disks  
Using PFS on harddisk  
Differences with FFS  
FFS compatibility  
Restrictions

---

## 1.14 format

### Formatting disks

After mounting you can format floppies in PFS format by using the standard `c:format` command. To format a disk named "hello" type:

```
format drive pf0: name hello
```

If the disk has been formatted before, you can use the quick option, even if the previous format was FFS or OFS (this works the other way around too).

```
format drive pf0: name hello quick
```

## 1.15 access

### Accessing disks

Shell use:

PFS disks can be referenced to by diskname (e.g. "hello:") or by mountname (e.g. pf0: or pf1:). Almost all cli-commands work on PFS disks like on any other disk, only faster.

Workbench use:

When a PFS disk is inserted in drive 0 two icons will appear on your workbench: one with label "DF0:PFS" and one with the diskname as label. The first comes from the standard filesystem, the second from the PFS filesystem. You can access the disk through the latter icon. It will behave just like df0:, only faster.

## 1.16 Harddisk use

### Using PFS on harddisk

-----

- Restrictions and requirements
- Preparing installation
- Installing PFS on the partition
- Notes

## 1.17 Harddisk

---

## Restrictions and requirements

PFS 9.5 only supports harddisk partitions of up to 32 MB in size. I know this is a serious limitation; it will be eliminated as soon as possible. It can be installed and automounted in the RidgidBootBlock, but standard mountlists are also supported. Memory usage depends on the size of the partition and the complexity of the directory tree. It will vary between 70K and 150K.

## 1.18 Harddisk

### Preparing installation

Select or make a suitable partition (less than 32 MB). All the data in the selected partition will be destroyed when installing PFS, so make sure you make a backup. PFS must be installed in the l: directory. Copy the version you want (ProfFileSystem or ProfFileSystem68020+) to l: and rename it to 'ProfFileSystem'. Only version 9.0 or higher will work with harddisks!

## 1.19 Harddisk

### Installing PFS on the partition

Now you can install PFS on the RidgidDiskBlock. Make sure you backup the partitions you're going to change! The size of the partition should not exceed 32MB. You can make such a partition with Commodore's HDToolbox, which is supplied with the Workbench software. With the same tool we can install PFS on the partition.

Select 'advanced options' in the the partitioning window. A number of extra buttons appear among which 'add/update...' with which you have to add PFS. Set the dostype field to '0x50465300' , version 9, revision 5 and select ok. After you have done that you can install PFS on the desired partition with the 'change...' button. Use all the default options: automount on, block size 512 (other sizes are not supported), 2 reserved blocks. Now select ok and the partitioning menu will appear again. Go back to the main screen and save the changes.

PFS should be installed on the partition now. After formatting it's available for use. If you've done something wrong and the computer doesn't startup properly you can prevent mounting with the boot menu.

Notes:

- PFS partitions are not bootable, so don't select 'bootable' in HDToolBox.

- Every PFS device (including floppies) will use the version of PFS you have installed in the RDB now. So if there is an update, changing `l:proffilesystem` is not enough, you have to update the RDB version too. This can be done with HDTToolBox too.
- A version for larger partitions is being worked on.

## 1.20 Harddisk

### Notes

During startup the filesystem will be loaded from the `RidgidDiskBlock` and the PFS partition will be mounted. PFS will be in the `FileSystem Resource` list now. This means that any PFS device, including floppies, mounted will invoke the same 'pure' PFS, the one in the resource list, reentrantly.

So if there is an update of PFS and you want to install it, changing `l:proffilesystem` is not enough, you have to update the RDB version too. This can be done with HDTToolBox.

If you don't want PFS to be automounted for some reason (incompatibilities etc.), you can prevent mounting by deselecting the PFS partition with the boot menu.

The stacksize as stated in `mountlists` can now be 600 instead of the 5000 of previous versions.

## 1.21 difference

### Differences with FFS/OFS

When you use the `c:list` command the 'blocks used' information will be too high. The `list` command assumes every file consumes a block of free space to use as headerblock. This filesystem uses blocks much more efficient, however. As a rule of thumb you can say that the real number of blocks used is the given amount minus the number of files listed. You can check this with the "info" command, which does give the correct amount. Directory tools like `Directory Opus` have the same problem with their `spaceleft` figure during a copy session. You'll notice that after copying is finished the figure will go up, back to the correct figure.

You'll probably understand now that there generally goes more on a PFS disk than on a FFS disk. PFS uses a larger part of the blocks for actual data storage.

Programs asking the filesystem information about a pfs disk will be fooled into believing the disk is a FFS disk. I do this to prevent those programs from saying it's not a dos disk without giving the

information requested to the user (e.g. the "info" command).

The "c:AddBuffers" command used on pfs devices won't have any effect on the cache, nor on memory usage. PFS uses a dynamic directory- and allocationblock caching algorithm, without any cachelimits. This won't take too much memory (max about 30K), because the number of directoryblocks on a floppydisk is limited.

## 1.22 compatibility

### AmigaDos compatibility

All Amiga-Dos features are implemented except hardlinks, filechange notification and recordlocks which are not used much anyway. All these features may be implemented in a future version, though. A PFS disk currently cannot be made bootable, so a PFS disk cannot be "installed".

## 1.23 restrictions

### Restrictions

PFS has the standard Amigados restrictions on file and disk naming:

- maximum disknamelength 32
- maximum filenamelength 108
- maximum commentlength 80
- almost unrestricted characteruse in filenames

These restrictions are forced by AmigaDos, they are not inherent to the filesystem. The number of files in a directory or on a disk is restricted by available disk space only.

## 1.24 faulttolerance

### Faulttolerance

-----

If you reset or crash your machine during a write operation a standard AmigaDos disk gets 'invalidated'. When you insert that disk later the disk will be revalidated automaticly. After this the directory is back in the state it was before the writeoperation. This actually is a neat feature, but it has two drawbacks: revalidating takes time and, if you were overwriting a file at the time of the crash, you lose both the original and the new version, since the file has been half overwritten. The directory states falsely that nothing happened, however.

---

With PFS the directory is correct at all times. This means that whenever you remove the disk, reset or crash the system, all the objects (files and directories) listed in the directory will be there as stated in that directory, including the filesize and date. In other words: a PFS disk NEVER gets invalidated. This does not mean you cannot lose any data, however.

This is how it works: when you operate on the disk (creating files and directories, deleting files etc) the filesystem keeps track of all changes to the directory but doesn't write these changes to the disk until the operation is completed. It does write the data but only to clean areas, that is, disk space that wasn't in use the last time the directory was updated. If that's not possible anymore, or you stop writing to the disk for half a second, the directory is updated so that the area occupied by deleted files can be reused without invalidating the disk. So if the machine brakes down at a nasty moment you will probably find the disk in the state it was before you started your writing session. This includes overwritten files: they'll be back in their old state, unless the free disk space didn't allow it (only in that case both versions could be gone).

If you want to make sure you can't lose a file by overwriting it you should keep in mind that the filesystem can only guarantee that if the free disk space is more than the size of the old version + the size of the new version + 5K. Then the directory will be updated after the new version has been written and not between deleting the old and writing the new version.

The directory and allocation tables themselves are also updated in such a way that the disk cannot become invalidated. So even a crash during the update won't invalidate the disk.

An extra advantage of this procedure is that files that are overwritten by a larger version are not fragmented, as in most filesystems. The file will be fragmented only when there is no large enough continuous free segment available. A fragmented file is even defragmented if possible.

Disadvantages are that the freespace on the disk gets more fragmented than strictly necessary and that the directory has to be updated more often. PFS does everything it can to limit freespace-fragmentation as much as possible.

A PFS disk normally cannot get invalidated but it can get corrupted. Nothing can be done against bad tracks on a disk or external programs (like viruses or diskeditors) that mess with the disk. A read or write error makes a disk invalid. There currently is no utility available to recover corrupted disks.

## 1.25 virus

Virus damage and viruskillers

---

Bootblock virusses destroy the bootblock of the disk. For this reason the filesystem doesn't store unrecoverable data in those blocks. It does use the bootblock to recognize a disk as being a PFS disk, so if the bootblock is destroyed, the disk in question won't be recognized anymore. To repair such a disk you simple have to write a PFS bootblock back to it. You can do that by copying a bootblock of an uninfected disk to the disk in question. Most viruskillers have functions to do that. Remember not to change any blocks other than the bootblocks (block 0 and 1).

Most viruskillers have no problem with PFS disks. You should check if your viruskiller accepts uninfected pfsdisks as normal, however.

## 1.26 tools

Tools included  
-----

You can find some tools that can come handy with PFS in the tools directory.

MFS A tool that helps you combine several filesystems (PFS, FFS etc) on a single drive

Diskspare A tool that makes it possible to store more on a single PFS disk.

## 1.27 bugs

Bugs

If you find something that doesn't function properly please let me know.

## 1.28 credits

Credits

Here I would like to thank everyone for their help and support.

Rougier Sylvian and Pierre Carrette for their numerous ideas and support from the very start.

Holger Kruse for his in depth Amiga knowledge.

Pavel Troller for helping develop a multiuser version (not yet finished).

---

Nicola Salmoria for his extensive betatesting and accurate suggestions. And for allowing me to include his excellent tool MFS in the PFS package.

Klaus Deppisch for helping bring his diskspare.device and PFS together.

Andreas Baum for translating the docs to German.

And all registered users for their support.

## 1.29 author

Author

You can reach me at the following address:

Post: Michiel Pelt  
Waalstraat 328  
7523 RS Enschede  
the Netherlands

E-Mail: pelt@cs.utwente.nl

Tel: (0031)-53-337227

## 1.30 registration

Registration

If you like this product you should register. You will be the first to get the next major update.

If you wish to make utilities for PFS, like undelete, optimize or repair, let me know if you need any help. I support all PFS related projects.

The registration fee is as follows:

cash: f30 Dutch Guilders, US \$20, DM 30, £15, FF 100 or the equivalent in another currency.

eurocheque: f30 Dutch Guilders. No other currencies.

money order: f50 Dutch Guilders. No other currencies.

Please sent the completed form to:

Michiel Pelt  
Waalstraat 328  
7523 RS Enschede  
The Netherlands

---

E-Mail address: `pelt@cs.utwente.nl`

All registered users will get the PFS version 10 as soon as its ready. I expect this to be before the end of 1994. If there are any minor updates, they will be published on Aminet.