PFS-Defrag

COLLABORATORS			
	TITLE : PFS-Defrag		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		August 8, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

PFS-Defrag

1.1 PFS-Defrag - Welcome

```
                PFS-Defrag 1.0
© 2000 Keith Halstead
pfsdefrag@halsteadk.freeserve.co.uk
http://www.halsteadk.freeserve.co.uk/pfsdefrag/

** IMPORTANT: **

Please read all sections of the documentation before using PFS-Defrag.


                 I AGREE - take me to the contents
                (If you choose not to try PFS-Defrag then please
                let me know
                 why.)
```

1.2 PFS-Defrag - Contents

```
                PFS-Defrag 1.0
© 2000 Keith Halstead
pfsdefrag@halsteadk.freeserve.co.uk
http://www.halsteadk.freeserve.co.uk/pfsdefrag/


                 WHAT IS IT?

                 LEGAL

                 ACKNOWLEDGEMENTS

                 REQUIREMENTS

                 INSTALLATION

                 CONFIGURATION
```

1.3 PFS-Defrag - What is it?

WHAT IS IT?

PFS-Defrag is a program to defragment PFS-II and PFS3 partitions in a safe, high-level way. It provides lots of feedback to the user, a high level of intelligence and gives the user plenty of control. DOpus is highly recommended as you will get progress bars and greater opportunity to abort, although it is not required.

Like all other PFS defragmenters, it runs the DiskValid program supplied with PFS to find the fragmented files. These are then copied somewhere else (no free space check is carried out yet) and then copied back over the originals. This is the standard way of attempting to defragment PFS disks. PFS-Defrag manages this process in a very safe way and also gives you plenty of control over it.

You will probably need to run the program twice per disk. Very large files are difficult to defragment and it is probably not worth defragmenting files >10MB unless you have very fast disks or lots of patience. Also, the more free space you have on the partition you are defragmenting, the more effective it is likely to be.

PFS-Defrag also makes 4 log files in RAM: which provide various pieces of information:

```
   dvoutput      - full output from Diskvalid
   dverroroutput - just the errors/notes that Diskvalid found
   fragoutput    - list of fragmented files and information about them
   defragoutput  - log of activities during the defragmentation stage
```

If you want to e-mail me about a problem you are having with PFS-Defrag please ensure you attach these files to your message. See the

Contacts

section.

This is quite a basic synopsis of what PFS-Defrag does, really it does a lot more. See the Walk-through section for further details, but first read the

Legal

,

Requirements

,

Installation

,

Configuration
and
Usage
sections.

Return to Contents

Read Legal Notes

1.4 PFS-Defrag - Legal

LEGAL

This software is distributed as freeware. However I would appreciate an email from you with any comments or suggestions or just, "I use it and I like it!".

This software is not public domain, i.e. I retain full copyright. It can only be distributed as a complete unchanged package and no money may be charged for it. It may be distributed on Aminet CD's and Amigactive CoverCD's without further permission as long as all files are included unchanged. Public domain libraries may not distribute it alone on a floppy disk due to its small size.

Due to the nature of this software you must use it at your own risk. You can examine the source code of the program to be happy that it will not damage your disks' contents. I have tested all parts of the program on my system and not had any significant problems, but please read the "SAFETY NOTES" section below. The program has also been tested extensively by several beta-testers over the last couple of months.

If you don't trust it, don't use it! (I am quite happy to use PFS-Defrag on all my PFS partitions without making backups first.)

Return to Contents

Read Acknowledgements

1.5 PFS-Defrag - Acknowledgements

ACKNOWLEDGEMENTS

Many thanks to all brave(?) beta testers and supporters of PFS-Defrag:

Aad Teijl
Bill Eaves
Geoffrey Webb
Harry Runge
Henrik Andreasen
James Hays
John E Lord
Jonathan Hart

Kevin Tiernan
Malcolm Harnden
Mark Bond
Michael Johnstone
Michael O'Hara
Neil Bothwick
Paul Brazier
Richard Lane
Roy Bartsch
Svein Ingve Wik

Return to Contents

Read Requirements

1.6 PFS-Defrag - Requirements

REQUIREMENTS

1. PFS-II or PFS 3 already installed on at least one partition/disk
2. Diskvalid
3. rexxreqtools.library
4. rexxtricks.library

The two add-on ARexx libraries are both available from Aminet or from many magazine coverdiscs (check their libs/ drawers). They are also available for download from the PFS-Defrag web-site as complete distribution archives.

Return to Contents

Read Installation Instructions

1.7 PFS-Defrag - Installation

INSTALLATION

1. Ensure RexxMast is running and you have the two required libraries installed in LIBS:

2. Put the script where you like on your system.

3. Configure PFS-Defrag to suit your system.

Return to Contents

Read Configuration

1.8 PFS-Defrag - Configuration

CONFIGURATION

There are a number of items which you can configure to customize the operation of PFS-Defrag. This process will become more automated in future versions of PFS-Defrag. Some of the variables MUST be configured or at least checked by you to ensure that PFS-Defrag can find the necessary files. If you need any help with this then please feel free to
drop me a line
.

For now you will need to edit the script in a text editor. Take care that your text editor will not truncate any long lines in the file. Do not edit anything before "/**** START OF USER-CONFIGURABLE SECTION ****/" or after "/***** END OF USER-CONFIGURABLE SECTION *****/". Only edit the values of the variables, i.e. the text enclosed in ' ' marks. Ensure that the "/***** END ..." line is still the 31st line after editing the script.

Here is a description of all the configuration items.

dfpath - this is the full path to the PFS-Defrag script file on your system. You MUST edit this unless you create a path identical to the default value.

dvpath - this is the full path to the Diskvalid program on your system that is supplied with PFS 2 and 3. If this program is stored somewhere in your default system path (i.e. in a directory specified with the "path" command in your user-startup or startup-sequence files like "c:") then you can leave this as just 'diskvalid'.

defaultdest - PFS-Defrag defragments your files by copying them to a different location and then copying them back over the originals. You can either specify a suitable directory on the command line (see
Usage
section) or
PFS-Defrag will display a directory requester for you to specify one. You can specify a default location for this directory requester to display by using this variable. So if you leave it at the default value of 'RAM:' then the directory requester will open showing RAM:. This makes it quick to always use the same destination (just press OK to close the directory requester) if you do not want to specify it on the command line.

textviewer - specify the path to your favourite text viewer program or leave it at the default value to use Multiview. If PFS-Defrag and Diskvalid detect errors on your disk you will be able to view the Diskvalid output with this program before proceeding with the defragmentation.

YAM users only:

maxyamfolder - this variable controls the maximum number of messages in a YAM folder to automatically defragment a YAM message. See the
Walk-through

section for more about this. (Don't forget to read the
Usage
section!)
If you're uncertain, leave the value at its default setting and you won't have any undue problems.

yampath - a (partial) path to your YAM messages. When PFS-Defrag is checking to see if a file could be a YAM message it checks to see if this text is contained in the path to the file. The default value is suitable for most people as other checks are made - the only exception is if you have called your main YAM folder something else. It does not make any difference where you have assigned YAM:.

Return to Contents

Read Usage Instructions

1.9 PFS-Defrag - Usage

USAGE

It is recommended that you quit YAM before running this program. For other programs that may access your disk, bare in mind that Diskvalid will make the partition invisible to the system while it runs.

The easiest way to run the program is by double-clicking its icon although there are others that may suit you better, especially if you use DOpus.

From a shell window type the following (replace "[path]" with the path to the program on your system):
```
rx [path]PFS-Defrag
```
A volume requester will appear asking you for a volume to defragment. There are also two optional arguments:
```
rx [path]PFS-Defrag <drive> <destdrawer>
```

If you specify a drive to defragment (or any path on that drive) then PFS-Defrag will just get on and defragment that drive. If you specify a "destdrawer" then PFS-Defrag will attempt to use that path to temporarily copy fragmented files to. Note that if you want to use "destdrawer" you must also specify a "drive". If you don't specify the "destdrawer" argument then a directory requester will pop up if there is any defragmentation to be done. (So don't worry too much about specifying any arguments as you will get requesters for them.)

The recommended way is via DOpus. Install the PFS filetype supplied and edit it so that the paths of programs are suitable for your system. Right-click on a PFS disk and select "Defragment".

Alternatively make a suitable filetype to match a (PFS) disk/partition and call the program with the following (replace "[path]"!):
```
ARexx    [path]PFS-Defrag {Qa}
```
Ensure the "Output window" and "Window close button" flags are selected.

Or if you always want to use RAM: as a default destination (ensure you have

enough for the largest file you will defragment) then use:
   ARexx   [path]PFS-Defrag {Qa} RAM:

Or you could just run the program without any arguments and a volume requester will pop up asking you for a disk to defragment.
   ARexx   [path]PFS-Defrag
Ensure the "Output window" and "Window close button" flags are selected.

The main advantage of using DOpus with this program is that DOpus users will get a progress indicator and the ability to abort in the middle of the defragmentation process. However, most of the requesters can be cancelled and doing so quits the program so there is plenty of opportunity to abort before defragmentation starts for other users. PFS-Defrag will take advantage of DOpus in this way if it is running, whether or not you choose to use DOpus to start it.

Return to Contents

Read Walk-through Description

1.10 PFS-Defrag - Walk-through

WALK-THROUGH

Here's what happens when you run PFS-Defrag and what each of the requesters does when you give certain answers.

A less technical, illustrated guide is also provided on the PFS-Defrag

web-site
, which you may want to look at first.

1. If you don't specify a volume to defragment then a volume requester will pop up asking you for one. If you select a non-PFS volume then Diskvalid and PFS-Defrag will detect this, tell you, and quit. Depending on the exact type of non-PFS device, Diskvalid will get to a different stage but the end result will be the same. Try running it on RAM: and CD0:!

2. Diskvalid (supplied with PFS2/3) is run in "analysis" mode on the volume you specified. PFS-Defrag will watch the output and detect errors and fragmented files. At this point DOpus users will also get a nice progress indicator. A full log is made at "RAM:dvoutput" and a log of errors and notes is made at "RAM:dverroroutput".

3. If errors or notes were found then this is reported. You can choose to view a log of these or the full output of Diskvalid before proceeding. Otherwise you can continue anyway (this should be fine if only notes were found) or quit. After viewing a log you can choose to continue or quit. Logs are viewed with the text viewer you specify in the "textviewer"

user-variable
.

4. If no fragmented files were found then PFS-Defrag informs you and quits.

5. Otherwise PFS-Defrag will gather information about the fragmented files (size, protection bits, etc.). You will get a requester telling you how many fragmented files were encountered, the total size of them, the total number of fragments and the largest single fragmented file. You can continue to the next stage or abort PFS-Defrag. A log of this information is made in "RAM:fragoutput".

6. If you didn't specify a valid destination drawer (see Usage section) then a directory requester will pop up asking you to choose one. It will open showing the user-specified "defaultdest" directory. Cancelling this requester quits PFS-Defrag.

   In future versions, at this stage PFS-Defrag will hopefully check the available free space on the destination drawer. For now just choose a location where there is sufficient free space for the largest fragmented file (as reported at stage 5).

7. When the destination drawer is confirmed you can choose the size of the largest file to defragment. Files larger than a few megabytes take several passes to defragment (ie. you will need to run PFS-Defrag several times). Ensure you do not specify a number larger than the available free space for your destination drawer as PFS-Defrag does not yet check this. Cancelling this requester quits PFS-Defrag. Entering "0" will cause PFS-Defrag to ignore the size of the files and attempt to defragment all of them.

8. At last the defragmentation can start. A further subdirectory is made for safety in the destination drawer you chose. The name of this has a "random" element using the time(s) function of ARexx. A log file for the defragmentation process is also opened at "RAM:defragoutput" which will contain most of the actions carried out during step 9.

9. Operation on each fragmented file:

   a) Check it is not a YAM message file.

      If a YAM message file is copied, moved or otherwise operated on without using YAM, then YAM will need to rebuild the folder containing that message when it next starts up. This may take a while on folders with lots (hundreds) of messages. However, this is just a matter of waiting around; you won't lose any e-mails. Indeed, as you are a PFS owner, this process is much quicker than with FFS. :) The same applies to the ".index" folder index files and PFS-Defrag treats these in the same way as message files.

      There is less of a problem with messages in small folders so you can specify a maximum limit in the user-variable "maxyamfolder". If the folder containing this message has less messages than "maxyamfolder" then defragmentation will proceed automatically. 100 is a reasonable value for this variable.

PFS-Defrag checks whether the file has a path which contains the text in the "yampath"
user-variable
. If so, it then checks whether the filename is either 9 characters long with a "." as the 6th character or if it is named ".index". If all this is true then the file is assumed to be a potential YAM message or index file.

If a YAM message or index file has been found, a requester will appear giving you four choices:

1. "Defragment" - defragments this message file only; requester will reappear if there is another one to defragment
2. "Defragment All" - defragments all YAM message files regardless of the size of the folder (i.e. doesn't perform these checks again)
3. "Skip" - skips defragmentation of this file
4. "Skip All" - skips defragmentation of any YAM message in a folder with more messages than "maxyamfolder" in it from now on

The "All" options only apply to this occurrence of PFS-Defrag running.

If you use this requester then check the "defragoutput" log file after PFS-Defrag has finished to see PFS-Defrag's "thought processes" for YAM files.

If you choose to defragment the YAM file it is then treated like any other fragmented file:

b) Temporarily change the protection bits of the fragmented file to "rwd" to ensure there will be no problems copying or overwriting it.

c) Copy the fragmented file to the temporary subdirectory. It is given a number as a name so there is no need to check the allowable filename length (unless you are defragmenting more than 999,999,999,999,999,999,999,999,999,999 files - that is untested :)).

d) Copy the fragmented file back over its original location. (Thanks to PFS, your original file is still there until the disk is validated, provided you have sufficient free space. Even if the computer crashed while doing this and their WAS a problem with the original file, the file would still be recoverable from the temporary location.)

e) Delete the copy of the fragmented file from the temporary subdirectory.

f) Change the protection bits of the defragmented file back to their original settings.

g) Repeat steps a-f for all fragmented files smaller than the maximum size you specified.

10. Remove temporary subdirectory and close the defragmentation log file.

11. Report the number of originally fragmented files and the number of files that PFS-Defrag attempted to defragment and that it skipped, either because they were too big or they were YAM files. You can choose to run PFS-Defrag again on this partition, again on a partition of your choice or just quit.

Note that if you run PFS-Defrag again from this requester (i.e. without quitting and restarting) then there may be no output in the window although it will otherwise work correctly and new log files will be produced (overwriting the old ones). This problem will be sorted out in the next public release version. (This feature was added quite late by popular demand of the beta-testers but to solve this small problem requires some work which would delay public release further.) It does not affect normal usage or re-running the program just once from this requester.

Note that at none of these stages is a "delete" or "move" command carried out on any of your original files.

Return to Contents

Read Safety Notes

1.11 PFS-Defrag - Safety Notes

SAFETY NOTES

This program should be safe. Ahem.

As far as I know, the worst that could happen is that PFS-Defrag will not copy the file back to the correct place. However the original would never have been deleted so you just have two copies of the file. This hasn't happened to me yet! To test this, carefully watch the free space on the partition you are defragmenting before and after you run PFS-Defrag. It should be the same afterwards, or certainly not LESS!

The program should be completely safe with paths and filenames with spaces in them. I haven't tested other "restricted" punctuation (and probably won't). Fragmented files with a " or ' in their name are likely to cause problems (but these are illegal characters anyway). Fragmented files with brackets ( ) in their names WILL cause some errors to pop up - PFS-Defrag won't break but those files will not be defragmented.

There will be problems if you defragment a file like a keyfile which "breaks" when you copy it (although these probably won't end up fragmented anyway). Please submit the names of any files you know of that do this to me. I will add a routine to prohibit defragmenting any files that have these problems.

If you want to fragment some files for testing purposes then you could try packing and unpacking largish files (especially if you do a few of them). Using the XPK RAKE packer seems to work particularly well and is very fast. Another good way is to download some files using several resumes (get HTTPResume and keep hitting "Abort" and "Start"). Voyager and IBrowse cache files also tend to get very fragmented, as do email message files.

If you take a look at the source code please note that I haven't attempted to make it very efficient yet.

If you want the ultimate guarantee then I can tell you I am quite happy to run PFS-Defrag on any of my partitions without making a backup first. I never release software that I am unhappy about using myself. That said, I cannot

be responsible for any freak damage that occurs to your system.

Return to Contents

View Program History

1.12 PFS-Defrag - History

```
                HISTORY

1.0   * first public release (25-Mar-00)
      - added a new log file for defragmentation process at RAM:defragoutput
      - some fixes/improvements to progress indicator
      - fixed bug in error handling if you attempted to defragment a non-PFS
        drive like CD0:
      - finally displays sizes in MB to 1 decimal place using trunc()
      - some more checks for most serious/common Diskvalid error reports added
      - added icon for easy running from Workbench
      - documentation completely updated for public release version
      -
               support web-site
                now online!

0.9ß  * fifth beta pre-release (16-Mar-00)
      - more checks for Diskvalid error reports added
      - now keeps track of number of errors and notes in progress requester
        as Diskvalid runs, and number of defragmented and skipped files -
        Opus Magellan II only
      - fixed YAM .index file finding
      - report number of skipped/defragmented files at end with an option to
        run PFS-Defrag again on the same or a different disk
      - docs NOT updated for this release

0.8ß  * fourth beta pre-release (11-Mar-00)
      - fixed defragmenting of YAM .index files - now treated like a message
      - added "yampath" variable to specify partial location of YAM messages
      - correct handling of any protection bits
      - added more advanced error/note handling
      - full and error/note logs from Diskvalid made in RAM: - can be viewed
        in PFS-Defrag if errors/notes found before defragmenting
      - updated docs, particularly installation and walkthrough sections
      - last minute! - also logs fragmented file info in RAM:fragoutput

0.7ß  * third beta pre-release (19-Feb-00)
      - converted script to use rexxreqtools for all requesters (DOpus users
        still get a nice progress indicator and can abort during operation):
        no reason now to make the script DOpus-only! :)
      - fix for when file size cannot be found by the script for some reason
        (will skip that file)
      - now finds "error:" as well as "ERROR:"
      - added "defaultdest" variable to specify default location of temporary
        directory requester (defaults to RAM:)
      - added check for defragmenting YAM messages in large folders
      - changed copying routine so there is no need for filename length
```

```
        checking
      - improved progress bar indicator to not include the size of files that
        won't be defragmented (larger than maximum) - better indication of
        completion
      - documentation is now more detailed, including a full walk-through of
        the program's operations, and in AmigaGuide® format

0.6ß  * second beta pre-release (07-Feb-00)
      - fixed bug where archive protection bit wasn't put back if it existed
      - uses ARexx delete() function instead of c:delete now
      - changed some progress requester text to reduce chances of truncation
      - can specify any path to defragment that disk
      - all checking of whether device is PFS is now done by DiskValid
      - changed display in MB so numbers are rounded up - no more "0MB"
      - does not rely on presence of setfnsize for PFS2 users
      - can force it to use a temporary location on a disk with shorter max
        filename lengths to avoid infinite looping of the program (esp. PFS2
        users)
      - implemented abort button (aborts when current file has finished
        defragmenting, can't abort while Diskvalid is running)

0.5ß  * first beta pre-release (05-Feb-00)
```

Return to Contents

View Future Ideas

1.13 PFS-Defrag - Future

```
                FUTURE

      - make configuration easy
      - add free space checking
      - put code on a strict diet to reduce current bloatedness
      - and I have a long list of other things but I can't tell you yet...
```

Return to Contents

View Contact/Support Information

1.14 PFS-Defrag - Contact/Support

```
                CONTACT/SUPPORT INFORMATION

I hope you find this useful!  Please e-mail me with your comments, reports or
suggestions.  Please also pay the website a visit some time.

If you are reporting a bug then please e-mail me the 4 log files produced in
RAM: (attach them to your mail as 4 separate files or archive them into one
lha file please).  They are named "dvoutput", "dverroroutput", "fragoutput"
```

and "defragoutput". If you can provide a log of all the output that appears in the window then that is very useful too.

Keith Halstead
pfsdefrag@halsteadk.freeserve.co.uk
http://www.halsteadk.freeserve.co.uk/pfsdefrag/

Get Opus Cut'n'Paste from
http://www.halsteadk.freeserve.co.uk/cutnpaste/

Return to Contents