

PowerCache

COLLABORATORS

	<i>TITLE :</i> PowerCache		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		January 5, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	PowerCache	1
1.1	PowerCache Documentation (18-Jul-1993)	1
1.2	Copyrights	3
1.3	Program Disclaimer & Liability	4
1.4	Background Information	5
1.5	Introduction to PowerCache	6
1.6	Overview of Features	7
1.7	Installing PowerCache	8
1.8	Starting PowerCache	9
1.9	Some General Information on Disk Caches	10
1.10	Configuring your Caches	13
1.11	PowerCache in Use	19
1.12	Editing Active Caches	19
1.13	Returning a Cached Unit to Normal	19
1.14	Using Cache Statistics	20
1.15	The Menu System	21
1.16	Stopping PowerCache	23
1.17	Troubleshooting	23
1.18	PowerCache Error Messages	24
1.19	A few Important Notes	28
1.20	Known Bugs	28
1.21	Benchmarks	28
1.22	Using Online Help	32
1.23	The Future	32
1.24	Technical Information	34
1.25	Author's Address	35
1.26	Credits	35
1.27	Shareware notice	36
1.28	How to Register	37
1.29	Keyboard Navigation	38
1.30	Hardware Requirements	39

Chapter 1

PowerCache

1.1 PowerCache Documentation (18-Jul-1993)

Documentation for

PowerCache v37.58

A Flexible and Powerful Disk
Caching System for the Amiga

Written by Michael Berg
Copyright (C) 1993 by Michael Berg
All rights reserved.

This document contains all the information you need to be able to configure and use PowerCache. Please select any one of the following items:

Shareware Notice
About shareware

Registering
How to register

Author's Address
How to reach me

Copyright
You MUST read this

Disclaimer
Important disclaimer note

Background
Some background info

Introduction
Introduction to disk caches

- Overview of Features
 - Presenting PowerCache!
- Requirements
 - Hardware requirements
- Installation
 - How to install the program
- Starting PowerCache
 - How to start the program
- General Information
 - Some general info on cache software
- Configuring your Caches
 - How to configure PowerCache
- Using Gadget Shortcuts
 - Keyboard navigation
- PowerCache in use
 - Using PowerCache's hotkeys
- Editing active caches
 - How to edit a cache
- Removing a cache
 - Returning a cached unit back to normal
- Using cache statistics
 - How to interpret statistics info
- Menus
 - Overview and explanation
- Stopping PowerCache
 - How to end the program
- Troubleshooting
 - In case of problems
- Error messages
 - Understanding PowerCache's error messages
- Important notes
 - Some VERY important stuff
- Known bugs
 - Things to be aware of
- Benchmarks
 - How fast is PowerCache?
- The Future
 - What's in store for PowerCache?

Your key file

The file you receive upon registering is yours exclusively. As expressed above, you may NOT give this key file to anyone. If you do so, for whatever reason, it is regarded as piracy and will be dealt with as such. Key files contain both the name and the registration number of the owner, so there will be no difficulties in identifying the origin of a key file.

Cracking

If you find a copy of PowerCache somewhere on a BBS, which is labelled "Cracked" or something like that, and download that program and use it on your machine, you are taking a gigantic risk. You are putting all your faith in the skill of some teenage cracker who may or may not have found and disabled the various protection mechanisms within PowerCache. Naturally, that is not always a such good idea. The slightest mistake on the part of that cracker, and PowerCache becomes the most expensive disk formatting utility ever made.

The choice is yours, of course. However, before you start the program, I urge you to pause and think for just two more seconds. Remember, a key file for PowerCache costs just US \$20 - a small price to pay for the safety of your disks.

Registering

When you register, you agree to all terms involved in owning a key file, which in turn entails agreeing with this copyright message. And breaking a copyright IS a felony in the eyes of the law.

ANY VIOLATION OF THIS COPYRIGHT NOTICE WILL LEAD TO PROSECUTION
AND CONVICTION IN A FEDERAL COURT, RESULTING IN UP TO SIX YEARS
OF IMPRISONMENT.

1.3 Program Disclaimer & Liability

DISCLAIMER & LIABILITY

This program is provided "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the results and performance of this program is assumed by you.

Should the program prove defective, you alone assume the entire cost of all necessary servicing, repair, or correction. Further, the author of PowerCache neither warrants, guarantees, or makes any representations regarding the use of, or the results of the use of, the program in terms of corrections, accuracy, reliability, currentness, or otherwise; and you rely on the program and results solely at your own risk.

The author of PowerCache can in no event be held responsible for any data or information which may be lost or rendered inaccurate by PowerCache, even if the author has been advised of the possibility of such damages.

DISKETTE LIMITED WARRANTY

The author of PowerCache warrants to the original licensee that the diskette on which your key file for PowerCache is recorded shall be free from defects in material and workmanship only for a period of 90 days from the date of original purchase.

If a defect covered by this warranty occurs during this 90 day warranty period, and it is returned to the author not later than 5 days of the end of such 90 day period, the author shall, at his own option, either repair or replace the diskette, and return it, free of charge.

This warranty shall remain effective for the duration of the 90 day warranty period, or until an attempt is made to change any information on the diskette (including, but not limited to, utilizing any remaining free space on the diskette for personal use, and including any repair or optimization operation), in which case you will void this warranty.

1.4 Background Information

Some Background Information on Disk Caching Systems

Amigas are becoming more powerful every day, it seems. Faster CPU's, lots of colors on super high-resolution screens and increasingly more complex software to control all of it. Evidently, you need massive harddisks to accommodate it all.

Most Amiga harddisk controllers are of the SCSI type, with the exemption of the new Amiga 4000 and Amiga 1200, which are equipped with PC-style "IDE" controllers. In a one-on-one comparison, an IDE interface is roughly as fast as an SCSI interface (sometimes even faster, actually), but if you put more devices on an single bus, the SCSI system wins every time. With the new SCSC-II (or SCSI/Wide) standard on the way, SCSI must undoubtedly be the interface of the future.

The common factor for all harddrives is their immense capacity and speed of data retrieval. It is not uncommon to see an Amiga read 500 kb off a hard disk in less than a second. Impressive, considering that a standard Amiga floppy drive will deliver some 10-12k per second on a good day.

However, even with such speeds, harddrives still seem a bit on the slow side when you are doing things like running a BBS or a Fidonet mail system, or when you tend to keep lots of files in the same directory, as any programmer will confirm. Many programmers prefer copying their "include files" and compiler executables to a RAM device during development, rather than to keep these dormant on the harddisk.

While keeping files in RAM does indeed give an impressive speed boost over the harddisk, it is still a cumbersome task to maintain these files. Did you remember to copy the modified files back onto the harddisk before cutting power? No? Too bad. Also, directory manipulation and maintenance are things which cannot be sped up this way.

The best solution to all of this is what is commonly known as a cache. A cache is a kind of buffer which logically slides in between the harddisk

and whatever software controls the drive. By keeping frequently used disk sectors in memory, such a cache can sometimes make a harddisk much faster than even the fastest of RAM disks.

The real value and usefulness of a cache program depends on how much memory you are willing to set aside for it, and how able the cache is to choose which sectors to keep in memory, and which ones to leave on the drive as the cache buffer fills.

As such, you control what kind of performance you want from a cache. It is a simple matter of memory versus speed, meaning that generally, a smaller cache buffer will be less efficient than a larger one. In any event, the program will need an efficient cache algorithm to be able to utilize the available buffer memory to the fullest.

Unfortunately, there aren't really any good cache programs available for the Amiga, today. One program, SmartDisk, is in the public domain, and actually performs quite well, considering the software is free. However, its cache parameters are fixed and cannot be configured in any way. You'll have to settle for its preconfigured defaults, which may not always suite your needs. Mostly, however, this isn't much of a problem. SmartDisk offers good value for money, and can be found on most BBS systems.

Another cache program currently available for the Amiga is HyperCache. HyperCache is quite configurable, very fast, and generally performs extremely well. When configured correctly, it makes everyday harddisk use a breeze. But before you all rush out to get HyperCache, you'd better check your cash deposit, because it will set you back some 50 US \$.

Another drawback of this otherwise very powerful program, is the complete lack of any form of user interface. If you don't know your CLI like the back of your hand, you probably won't be able to utilize the program to its full potential either.

Last, but certainly not least, both SmartDisk and HyperCache suffer from one common, very significant drawback: Both are read-only caches. That is, sectors that are read off the drive will be cached in memory, but sectors written to the drive will not. In fact, the presense of such a read-only cache will actually degrade your drives performance during write operations.

This is where PowerCache steps in. In response to the growing need for a fast, flexible, but most importantly, reliable, cache program, and out of frustration of how much existing cache software costs, I decided to write one myself. The overwhelming number of features, combined with very high performance and ease of use must make PowerCache the absolute Cadillac of cache programs. I'm sure you'll agree with me on that, once you reach the bottom of the features list, later on in this document :-)

1.5 Introduction to PowerCache

Introduction to PowerCache

PowerCache is a new style, user friendly, completely configurable cache program that both delivers top performance and has a neat Intuition user

interface. And since it can be used as both a read/write cache and as a traditional read-only cache, it will not handicap your drive during write operations, as certain other currently available disk caching utilities do. It will in fact speed up write operations just as it does read operations.

For maximum performance, PowerCache uses an N-way, set associative cache buffer layout. This method is traditionally not quite as memory efficient as using fully associative cache designs, but requires much less program overhead. PowerCache breaks this tradition by implementing a design where memory is allocated on demand rather than all in one piece. Therefore it matches the memory usage efficiency of any fully associative cache, while still retaining the simplicity of the set associative design. Further, since PowerCache has a low-memory handler built in, it will also pay attention to system memory panics and release most of the buffer memory whenever the system calls for it (Kickstart 3.0 only).

With PowerCache, you get an an extremely versatile and flexible disk caching system. Not only does it offer more features than any other cache program currently available, it also outruns most of its competition. Check out the

benchmark section
for a look at some hefty numbers :-)

1.6 Overview of Features

Below is a brief overview of the most important features of PowerCache. A (*) denotes that no other cache program for the Amiga currently offers that particular feature.

- (*) · Completely Intuition driven
- (*) · Friendly and familiar gadtools look
- (*) · Keyboard shortcuts for all gadgets (automatically mapped)
- (*) · Online help available for every single gadget and menu function
 - Extensive syntax and consistancy checking of input parameters
- (*) · Uses the superb requesters of Nico François' reqtools.library
- (*) · Uses NewLook menus (and gadgets) under Kickstart 3.0 and beyond
- (*) · Adapts to whatever font you happen to be using
 - Easy installation - runs as a standard Commodity
- (*) · Can be called up using any hotkey you like
- (*) · Slides conveniently into your WBStartup drawer
- (*) · ~Can be instantly enabled and disabled using hotkeys
 - Unmatched range of cache features
 - Allows any configuration of 'lines', 'sets' and 'prefetch'
- (*) · Will optionally suggest the best combination of the above
 - Handles devices running with non-standard sector size
- (*) · Can cache units on the same device seperately
- (*) · Offers unique "cache-in-a-file" caching facility
- (*) · Knows most device drivers, and how to patch these correctly
- (*) · Can give off detailed device information
- (*) · Caches both READ and WRITE operations

- (*) · Any cache can be designated as being read/write or read-only
 - (*) · Memory friendly (includes v39-style low-memory-handler)
 - (*) · Memory type for main cache buffers can be specified
 - (*) · Selectable caching algorithm (though only LRU as of yet)
 - (*) · Memory usage can be efficiently controlled and limited
 - (*) · Offers statistics graphs for each cached unit, for fine tuning
- Easy, transparent use
- (*) · Has reached speeds of 8+ megs/second on primed reads and writes
 - (*) · Any cached unit can be returned to normal at any time
 - (*) · Any cached unit can be edited (enlarged, shrunk) at any time
 - Found to work reliably with GVP and Advantage controllers
- Loads of other convenient features
- (*) · Includes an extensive ARexx interface
 - (*) · Localized - runs in two languages (CatComp source included)
 - ~68000/68020+ versions included
- Plenty of documentation
- (*) · 100k+ AmigaGuide documentation
 - (*) · Context sensitive online help everywhere (AmigaGuide hyperhelp)

One quick note about the speed rating. It should be clear to everyone that the somewhat radical transfer rate of 8 megs per second is only possible for accelerated machines (68040), and typically only when using disk benchmarking programs (such as DiskSpeed), which aren't particularly close to real world application use anyway. I personally achieved 8 megs/sec during a disk benchmark test on my '2000, which is equipped with the RCS Fusion Forty 28 mhz 68040, and a 105 MB Quantum LPS drive.

In practical use, you can't really expect performance like that. And if you do not have an accelerator, you must be prepared for a great deal less than 8 megs per second. I'm not a magician, you know! :-)

1.7 Installing PowerCache

Installing PowerCache

PowerCache installs quite easily. For the sake of convenience and future compatibility, I strongly suggest that you use the script which is provided for use with Commodore's Installer utility. However, since not all people have this utility (yet), a script for IconX has also been provided. In either case, just click the appropriate icon to get going.

For those of you who prefer doing things manually, here's how:

- Copy libs/reqtools.library from the distribution disk to your own LIBS: directory
- Make a directory called "PowerCache" on your system disk (hard disk)
- Assuming the distribution disk is in DF0: and your system disk is DH0:,

issue the following CLI command:

```
1> Copy DF0:PowerCache DH0:PowerCache ; Or Copy DF0:PowerCache.020...
```

Next, make a "Documentation" directory, in the PowerCache directory you made in the previous step. Then issue this command:

```
1> Copy DF0:PowerCache/Documentation/#?.guide#? DH0:PowerCache/Documentation
```

Note that it is not strictly necessary to install the guide files. PowerCache will run without them, but doing so means giving up all forms of online help.

•~Put the following few lines into your S:User-Startup file:

```
;BEGIN PowerCache
Assign PowerCache: DH0:PowerCache DEFER
;END PowerCache
```

PowerCache can be run without this assign, but only if you organize it on your harddisk exactly as it is on the distribution disk. That is, to use PowerCache without the Assign, it must be able to open "Documentation/PowerCache.guide" from its home directory.

For future compatibility it is recommended that you do indeed include the assign statement in your user-startup file, since later versions of the program may *require* this assign in order to work at all.

Both installation scripts automatically fix your user-startup file in this way.

1.8 Starting PowerCache

Starting PowerCache

PowerCache can be started by simply clicking its icon, but may also be run from the CLI.

Workbench users can simply drop PowerCache into the WBStartup drawer, remembering that doing so will require PowerCache: to be properly assigned. See the installation notes on for a more detailed explanation of this).

You may provide a couple of startup parameters for PowerCache. These are:

CX_POPUP=<YES or NO> If YES, then PowerCache will display its main window upon startup. If NO, then it will not do this.
This flag overrides the setting of the Popup checkbox gadget on PowerCache's main window.

CX_PRIORITY=<number> PowerCache's priority in the chain of installed commodities. Used to give priority to one of two (or more) running commodities that uses the same hotkeys. Normally zero.

PREFSFILE=<filename> This will cause PowerCache to initialize everything from the preferences file you specify, in stead of using the default preferences filename(s).

DONOTWAIT Well, this isn't really a parameter that PowerCache understands, but Workbench needs this as a tool type parameter if you run PowerCache from your WBStartup drawer, or it will wait forever for it to quit.

Parameters may be specified as tool types in PowerCache's icon, or as command line parameters (in any order) when using the CLI.

Running PowerCache EARLY in the Startup-sequence

Naturally, the sooner in your startup-sequence you start PowerCache, the sooner it starts doing its job, and the faster your startup procedure will be. But just how soon can you run it? The answer is that you may run PowerCache just as soon LIBS: is valid. If your startup-sequence doesn't make a LIBS: assignment, then you could run it as the very first thing (after SetPatch).

One thing must be taken into consideration, though. Unless told otherwise, PowerCache looks in ENV: and ENVARC: for its default settings, so you will need to have these assigned properly if you want PowerCache to start up using the last saved preferences (which you probably want). If you want to start PowerCache before assigning ENV: and ENVARC:, use the PREFSFILE parameter to specify the preferences file directly. I do this myself, like this:

Excerpt from my startup sequence

```
C:SetPatch >NIL:
C:Run DH0:bin/PowerCache.020 PREFSFILE=SYS:Prefs/Env/PowerCache.prefs
...script continues...
```

So as you see it is possible to start PowerCache very early in the startup sequence. The use of a cache in your startup sequence will typically halve the time it takes before your Workbench screen comes up, and sometimes it will do even better (depending on the size of the cache and the complexity of the startup script, of course)

PowerCache can effectively replace any validate-and-wait type tools you may be using, in that it will wait for a validating drive to settle down before adding a cache to it. Of course, it will not wait for drives that it is not supposed to cache.

1.9 Some General Information on Disk Caches

Some General Information on Disk Caches

Before continuing, there are a few things about cache organization that you must know. In order to make cache access as fast and efficient as

possible, the main cache 'buffer' must be organized in a special way. We talk about "sets" and "lines" (and "prefetch", as we will see later on).

These values all relate very closely to sector numbers. All devices that are used to store files on, have sectors. One sector is a tiny bit of information - typically something like 512 bytes. A floppy drive has approximately 1700 sectors, and a medium sized harddisk (100 MB) 170,000 or more.

All devices have a controlling task (a running program) called a device driver, whose primary task it is to read and write these sectors. If a file system (like FFS) wishes to read sectors off a drive, it must politely ask the device driver to please fetch those sectors. There is (almost) no other legal way to do it.

The file system is a kind of protocol that describes how data is organized on the physical device. Older Amigas (pre-1.3) use mainly OFS (Old File System), while newer Amigas use FFS (Fast File System) which employs a more efficient way of storing file information. The latest Amiga models offer sophisticated filing systems such as DCFS (Dir Cache File System) and International modes. Since PowerCache is layered directly on top of the device driver, and plays no part in file block organization, it will work with any file system currently in existence. There shouldn't be any problems with future file systems either.

The term "prefetch" is an indication of how many sectors the cache will read or write for each physical device access. If prefetch is set to 4, then this means that even if the file system only asks for two sectors, four will be read. Since each drive access is a lengthy and time consuming procedure, much time can be saved by "guessing" what sectors will be needed next. Since the best guess is often that "the next sector will be needed next", prefetching sectors in clusters is indeed a Good Thing. Keep in mind that it only takes slightly longer to read four sectors than to read one. It would take much longer to read four sectors individually than to read them all in one go.

Prefetch, or "read-ahead" if you want, can boost performance a great deal, providing the device is not too fragmented. Common to all known Amiga file systems is the fact that one single file can be spread out all over the disk. It is not required to be all in one piece (that is, stored in consecutive sectors). This helps the file system utilize the available space in the most efficient manner.

Unfortunately, this flexibility comes at a price. If a file is scattered across a large partition, it will also take much longer to load the file into memory. In this case one says that the file has been scattered, or that the device on which it is stored, is fragmented. Fragmentation can be corrected with what is known as a disk optimizer.

If you use too much prefetch on a fragmented disk, it will not help you one bit. In fact, it will probably work against you, slowing everything down. This is because the next sector in sequence probably is NOT the next sector that will be needed, and so the cache will be reading four times more than it should, on each access.

The morale is to use prefetch with some thought. Tune it to your disk

fragmentation level. Use a defragmentator (a tool that will reorganize data on disks) when necessary.

Moving on to the term "lines". Lines is a means of grouping together sectors in "blocks" of a certain length. If a cache buffer uses 32 lines, this means that the cache can remember up to 32 different entries, with each entry consisting of 'prefetch' number of sectors. To illustrate this, consider the following sketch of a 32 line cache, with a prefetch of 4:

Line number	Line contents
0	[sector][sector][sector][sector] (4*512=2k)
1	[sector][sector][sector][sector] (4*512=2k)
2	[sector][sector][sector][sector] (4*512=2k)
3	[sector][sector][sector][sector] (4*512=2k)
4	[sector][sector][sector][sector] (4*512=2k)
....etc....	
31	[sector][sector][sector][sector] (4*512=2k)
	32*2k=64k

Each line can hold any four consecutive disk sectors with this prefetch configuration. If the file system asks for sector number 8, the cache will read sectors 8, 9, 10 and 11 into one of the lines. The cache will choose the actual line number to use based on the sector number, but the exact procedure is irrelevant here.

If the cache program selects a line that is "dirty" - that is, one that already contains data that has been updated by the file system in a previous operation, then that line must be flushed out onto the physical device before it can be reused. This happens automatically, but it is important to know.

By now it must be obvious that more lines also mean more space available for sector data, and therefore a higher chance that a read or write operation will be able to complete with no or only little physical disk access. Unfortunately, it also means you have to set aside more memory for the buffers. A compromise you often encounter with computers.

The last term you need to know about in this context is "Sets", and this is easy. A set is exactly one cache buffer, like the one illustrated above. By keeping several of these buffers (sets) around for the same device, you increase the cache's hit-rate.

A typical cache configuration is an 8-set, 32-line cache, with a prefetch of 4. The memory required by such a cache buffer will be (fully loaded)

$$\begin{aligned}
 & 8 \text{ sets} \times (32 \times (4 \times 512 \text{ bytes})) \\
 &= 8 \text{ sets} \times (64\text{k bytes}) \\
 &= 512\text{k bytes (half a megabyte)}
 \end{aligned}$$

For a medium sized hard disk (~100 megs), you would want more lines and

probably also more sets. For example, 64 lines and 12 sets. This works out to (fully loaded)

$$\begin{aligned} & 12 \text{ sets} \times (64 \times (4 \times 512 \text{ bytes})) \\ &= 12 \text{ sets} \times (128\text{k}) \\ &= 1536\text{k bytes (one and a half megabyte)} \end{aligned}$$

All this probably leaves you as confused as you were when you started reading all this. I apologize. I'm a programmer, not a writer :-)

Anyway, I'll give you a few tips on how to tune your caches:

- If a cache is generally meant to be used for manipulating relatively small amounts of data, use fewer lines.
- Otherwise, the larger the drive, the more lines should be reserved.
- The more generally useful you want the cache to be, the more sets (and lines, for that matter) should be used.
- Prefetch can work against you if a partition is fragmented. If your drive is, or becomes, fragmented, then lower the prefetch value for better cache performance. On the other hand, if you have just optimized your drive, increase prefetch.

It's simply not possible for me to present you with a recipe on how to make a perfect cache. The range of possible combinations of drive size, number of lines, sets and value of prefetch is simply too vast. You will have to experiment to find the combination that best suit your needs. The

statistics
window will probably be of some use to you here.

The cache preferences window contains a gadget that will suggest some good values (well..) to use for any small, medium or large cache buffer and for any set of memory and disk capacity parameters (which PowerCache works out on its own -- no need to learn more new terms~-). This function can be used as a guiding line to finding a usable configuration. The procedure PowerCache employs can be studied in the technical reference section in this manual.

1.10 Configuring your Caches

This entire section describes how you use PowerCaches gadgets and menus. Much of this will be easier to understand if you have PowerCache running in the background, so if you haven't already done this, then go ahead and start the program now.

The first thing that should happen is that PowerCache opens its main preferences window. It is from here you control everything.

On the left hand side is a list of those devices (and units) that PowerCache believes it is able to add a cache to. Note that the unit number may not always seem to be accurate. For example, the AdSCSI controller (Advantage) uses the icddisk.device to control the drives, and this device driver lies about unit numbers (i.e. it claims that all units are unit 0). Gvpscsi.device (GVP's device driver) will play a similar trick on you. This is no problem for PowerCache, though. It is quite capable of distinguishing between individual units on a device.

On the right hand side is a similar list that shows you which units are currently being cached by PowerCache. If this is the first time you run PowerCache, then this list will probably be empty (if it isn't, your machine must be haunted :-)

Moving back to the left hand side of the window, you'll find three checkbox gadgets that control various aspects on PowerCache's behaviour.

The first one, `Popup window?`, tells PowerCache if you want it to show its window every time it is started. On long-term use you will probably want to disable this gadget, and just have PowerCache start up silently from the startup-sequence or from the WBStartup drawer.

The second gadget, `Beep` tells PowerCache if you want it to flash all screens and sound a beep every time it is either disabled or enabled. Leave this on.

The third one, `Use Help`, enables or disables online help. If you plan to use PowerCache regularly (like in your startup-sequence or WBStartup), then leave this gadget off. PowerCache starts up faster without online help.

To the right of the checkbox gadgets are three string gadgets. These allow you to change the hotkeys that control PowerCache.

The first one, `Show Window`, is the hotkey that you use to make PowerCache show its main preferences window. Note that PowerCache always opens its window on the default public screen, falling back to Workbench if this is somehow not possible.

The next two string gadgets define the hotkeys that are used to enable and disable caches. These hotkeys affect ALL caches in operation. That is, if you press the "disable" hotkey then ALL caches will be disabled.

Below the list of devices, you will find three regular gadgets. The first of these, `Info`, will show you an impressive arsenal of detailed device information. If you are a novice user, this information may appear to be completely mumbo-jumbo, but it comes in handy when you impress your friends with the enormous amount of knowledge you possess about your harddisk :-) (and for expert users, this information is often crucial in selecting an appropriate cache size)

You may open any number of Info windows (even several windows for the same device), and close the main window without closing any of the Info windows. (PowerCache will close all Info windows on exit, so there's no need for you to chase around looking for Info windows when you want to stop the program).

The `Rescan` gadget should be used if the list of devices seems to be

incomplete, or if you mount new devices after opening PowerCache's window, and want these to appear on PowerCache's device list.

In the bottom of the window you will find three gadgets, labelled Save, Use and Cancel. To save your current cache configuration permanently (that is, the entire program and cache configuration as it is now), select Save. PowerCache will then save the preferences file in ENVARC:, which is usually somewhere on your hard disk. This means that all caches will be reinstalled automatically when you start PowerCache next time.

Note that when you exit PowerCache, it will also save the current preferences, but this time in ENV: (which is in RAM:). This is done so that if you exit the program, and then start it again, you won't have to configure everything all over, yet you won't be stuck with the configuration forever (since ENV: is in RAM: which is volatile and dies on a reset)

To hide PowerCache's window, either click the Use gadget, the close-window gadget or select Hide from PowerCache's menu.

The Cancel gadget is a bit special. If you click it, all changes you have made to the three checkbox gadgets and to the three string gadgets will be reset to the values that were in place when PowerCache opened its window, but any changes made to installed caches will not be returned to the original state. That is, if you install additional caches, remove installed caches or edit any of the active caches, then these changes will NOT be cancelled.

By now you must be dying to find out just exactly how you add a cache to a device. Well, start off by finding the unit you want to cache, in the list of cacheable devices. Now click it. Under 2.04, it will show up below the listview itself, and under 3.0+ it will be highlighted in blue.

Notice that as you select a device and a unit to cache, the Install gadget below the listview becomes active, indicating that PowerCache will allow you to install a cache on this unit. Click Install to configure a cache for whatever device you selected.

Let's try installing a cache. Find a device in the list (DF0: for example), and then click Install. With any luck, a new window with a bunch of gadgets will pop up, and the main window will become inactive.

There are three main blocks of settings in this new window. The first one controls the number of sets and lines, plus the amount of prefetch to use. After the long monologue I delivered in the previous section, these numbers should have some meaning to you. Change these values as see fit.

Note that "lines" and "prefetch" are both required to be a power of 2. You cannot have 20 lines, for example. You must select either 32 or 16 (which is exactly the main disadvantage of set associative caches). If you enter a value that does not fit this rule, then PowerCache will correct the value to the closest power-of-two value it knows (so PowerCache will change an entry of 20 to 16, which is the nearest power-of-two value). Sets may be any value.

Please note that none of these three values may be zero. You cannot have

zero lines or sets, and prefetch must be at least one (namely the sector the filesystem is requesting). PowerCache will not allow you to enter zeroes in any of the three gadgets.

Those of you who are a bit unsure about how large or small to make a cache may simply click the "Auto Configure" gadget. This will make PowerCache suggest some values which it believes to be appropriate for any "large", "medium" or "small"~cache, based on disk capacity and the amount of free memory you have. The "technical details" section in this manual describes in detail how this is done.

To the right of the "Organize as..." box you will also find three lines of text. This is a calculation of how much memory the cache buffer you request will use, when it is fully loaded (a cache is not allocated all at once, but as it fills). If necessary, trim your cache parameters to match the amount of free memory you have.

Moving downwards and to the left, you will find a group of gadgets that all have something to do with the way the cache operates. The first gadget you encounter is the Mode gadget. By default it is set to R/W (read/write) meaning that both read and write operations will be cached. Clicking it once will turn it into R/O (read-only), meaning that only read operations will be cached and that write operations will be passed through to the device controller unchanged.

So when is the right time to use a read/write cache versus a read/only cache? Well, the short answer is that using a R/W cache is generally the best thing. The long answer is that read-only caches may protect you better from data loss, in the event of a sudden system crash during a write operation. A R/W cache will leave you with much less data having actually been stored on the drive than a R/O cache would have done. However, a R/O cache is not necessarily safer than a R/W cache as far as crash protection is concerned. Either one will trash your harddisk when the machine crashes at just the right (wrong) time. In fact, ordinary hard disk use can do this as well.

My advice is: Stick to read/write caches for maximum overall performance. Read-only caches have their place, but only under very special circumstances. For example, suppose a program spends 10 minutes of constant drive reading and writing (it is creating a very complex file), only to crash the machine just before it is done. With a read-only cache, the complex file may well have survived the crash and be sitting on your harddisk (ready for use), but with a read/write cache, this might not be the case. The file might have been partially (or completely) stored in the cache when the crash occurred. One important note, though: Read/only caches are some 40% faster than read/write caches when reading sector data.

Moving on to the next gadget, you'll find a cycle gadget that controls what algorithm the cache routines should use for this particular cache. It is set to LRU by default, and you probably want to leave it there. Least-Recently-Used is by far the best, generally applicable algorithm available.

As of yet, LRU is also the only algorithm that has actually been implemented. The other algorithm currently selectable, SPC, is in effect a do-nothing algorithm that mirrors LRU completely. In time, a number of

different algorithms will be implemented, each tailored to deliver top performance for certain well defined tasks.

Right below the Algorithm gadget is the Purge timeout integer gadget. This gadget defines for how long a cache should be idle, before it begins purging its dirty buffers. The number is in seconds, and is set to 2 by default.

For a read/write cache, sectors which are cached during a write operations will remain in the cache only for a specified amount of time, after which the cache will switch into purge mode and start flushing out "dirty" lines back to the drive. The Purge timeout gadget specifies this value.

This next gadget is the Free after Purge gadget. Setting this gadget will cause read/write caches to write out dirty buffers and release the cache buffer memory when the drive has been idle for the amount of time specified as "purge timeout". It has no effect on read-only caches. Earlier versions of PowerCache allowed read-only caches to behave similarly, but this is no longer the case, due to speed considerations. Read-only caches can be made to work that much faster if they don't have to keep track of idle time.

This gadget is not set by default for one very good reason. Caches work best when they are primed. A cache is primed when all read/write requests can be satisfied entirely by using only cached sectors, and no physical disk access is necessary. As you can imagine, freeing the cache buffers will also free up any sectors which may have been cached, and the cache will then have to prime itself once again.

There is no real rule to use when configuring this gadget. However, I do recommend that if a device is to be used for intense read/write operations, followed by long periods of drive inactivity, then setting the Free gadget to true is probably a good idea.

On the other hand, if the drive remains inactive only for relatively short periods of time (but longer than the Purge Timeout value), then you should probably leave this gadget unset. Obviously, if you have lots of memory to spare, also leave this flag unset.

Note for Kickstart 3.0 users: If a low-memory situation occurs when PowerCache is running, then all caches will be forced to release whatever they can (which means all non-dirty sectors), regardless of the setting of the Free-after-purge checkbox gadget. Actually, I recommended you to leave the gadget un-checked when running 3.0 or later. This allows buffers to remain in memory for as long as possible, usually resulting in better cache performance (yet allowing the system to reclaim the memory). This feature is only available to 3.0+ users, since AmigaDOS 2.04 is not smart enough to ask running programs to free up memory in low-mem situations.

Moving to the right you find a new group of gadgets. These all control what kind of buffer should be used for this cache.

PowerCache knows two different methods of caching: Memory caching, and file caching. A memory cache is a "normal" cache, where lines and sets are allocated in memory as needed.

A file cache is used to cache one device on top of another, so to speak. For example, it allows you to cache your floppy drive in a file on your harddisk. This means that your floppy will become just about as fast as your harddisk (when primed), but not as fast as a regular memory cache. On the other hand, you don't have to sacrifice precious RAM to accommodate the cache buffer, since it will be located in a file on your harddisk.

File caches are typically used to create small or medium sized caches for floppy drives, with the cache buffer files being located in a temporary directory somewhere on your harddisk. There are no restrictions on the use of file caches, however. PowerCache will allow you to cache your harddisk on a floppy drive, even though this doesn't make much sense.

One very, very important note about file caches, though: DO NOT USE prefetch on file caches. Always set prefetch to 1. For some reason or other, Prefetch works unreliably with file caches. I am working on the problem, and I expect to have it fixed before long.

The Type gadget selects what form of cache you want. It is set to Memory by default, indicating that a normal memory buffer will be used. You may select the type of memory to be used by clicking the Memory type gadget (below the Type gadget) until the type you want is shown.

If you chose file caching, then you must also enter the name of the file to be used. You can use the select file gadget right next to the filename gadget to easily pick a directory for your file. PowerCache will suggest an appropriate filename to use, so that all you need to do is find the right directory to put it in. However, you may change the filename if you like.

The cache file will be created when you leave the cache preferences window, and it will be set to the size indicated in the Main Cache: text field shown at the top of the window. Expect this, and be sure you have enough space available on the target drive.

What remains is the bottom three gadgets, labelled Okay, Purge and Cancel. The purge gadget will be ghosted, since only active caches may be purged. We'll come back to this later on.

If you believe everything is correctly configured, then click Okay to actually install the cache. If not, then just click Cancel, and nothing will happen.

If everything goes well, the device should now appear in the right-most listview gadget, in a format looking like this:

```
Device:  A x B x C
```

A is the number of sets, B is the number of lines, and C is the prefetch value. For example:

```
DH0: 10 x 32 x 4
```

If something goes wrong during the cache setup procedure, then PowerCache will issue an error message (meaningful, hopefully) and leave you back in

the main preferences window.

1.11 PowerCache in Use

PowerCache in Use

When in use, PowerCache should be pretty transparent. The first thing you should notice is that access to cached devices will be tremendously sped up. Install a cache on your harddisk, and watch those icons whizz along at light speed. Or try clicking the icon of a really large program. On the first invocation, the program will be read off the drive and into the cache (with little or no speed increase), but if you start it again, it will start immediately, probably with no disk access at all.

If for some reason you feel you must enlarge or shrink an existing cache, you can always bring up PowerCache's main window by pressing the appropriate hotkey (Ctrl-Alt-C by default). Please refer to the section on

cache editing
for more details on how to do this.

1.12 Editing Active Caches

Editing Active Caches

PowerCache allows caches to be edited without limitations. To do this, select the cache you wish to edit in the list of cached devices, and click Edit. The familiar cache preferences window will appear, and will contain the current values for the cache being edited. Change any value you like. You may even change the cache from being a memory cache to being a file cache, or vice versa.

When done, select Okay to force the changes to take effect, and Cancel otherwise.

Note that the Purge gadget (bottom part of the window) will be selectable when editing caches. Clicking this gadget will flush any dirty sector data currently present in the cache buffers out to disk and then free the cache buffer itself. The cache will be completely empty, as if had you just installed it. This can be helpful if you need some more memory and you don't want to have to remove the cache completely.

Also note that purging a buffer may involve considerable disk activity, depending on how much dirty sector data there is, as well as the prefetch value. Larger prefetch probably means more time is used to write out cache lines.

1.13 Returning a Cached Unit to Normal

Returning a Cached Unit to Normal

Any cached device unit can be restored to normal (non-cached) operation at any time. This is done by selecting it in the list of cached devices, and then clicking the Remove gadget.

Note that if you try to remove a read/write cache that contains dirty sector data, then this data must first be written back to disk. This may take some time, especially for floppy drives. During that time the drive will be locked from any other disk activity that may be going on (meaning that all pending disk operations will be halted until the drive has become completely updated by PowerCache).

Read/only caches can be removed instantaneously, since they never contain dirty data anyway.

When you exit PowerCache, it will effectively remove every installed cache, which - as just explained, may take some time. Expect this.

1.14 Using Cache Statistics

PowerCache offers a sophisticated graphical display of the performance and memory usage of each cached unit. This allows you to fine-tune a cache so that it uses just the right amount of memory to suit your needs.

Select any active cache, and then click the Stats gadget, located below the list of active caches. When you do this, a small window with a few gadgets will appear.

The large rectangle in the left half of the window is used as a monitor and continuously displays two graphs.

The first graph is the Memory Usage graph, and is (usually) drawn in black. Note that this graph is meaningless in the context of a file cache, since the entire cache file is created when the cache is installed.

The second is a rough indication of how well the cache is performing, and is drawn in blue.

Both these graphs are displayed as percentages of full capacity, and can be individually turned on and off by using the two checkbox gadgets on the right hand side of the graph display.

The Update Frequency gadget controls how fast you want the graphs to update, and is measured in tenths of a second. Possible values range from one tenth of a second to once every five seconds (50/10 sec). Statistics windows are somewhat CPU hungry, and the more often you want updates, the more CPU time will be used.

The time delay is automatically adjusted to match the system load once every second. This means that even during the most heavily multitasking system, the statistics windows will always update at correct time intervals.

The Memory usage graph can be used to find out how accurately the cache

size fits drive activity. If the graph never reaches maximum, then your cache is probably too big. Otherwise, if it hits the ceiling very quickly, with a subsequent drop in performance, then the cache is probably too small.

If you never seem to get the Efficiency graph to go above 50%-70%, even though the cache is still not full, then it is probably because your drive is too fragmented for the prefetch value you are using. Try lowering the prefetch value. It could also mean that you are not using enough cache sets, so try increasing the number of sets. If you want the cache buffer to remain the same size, lower the number of lines.

There are a couple of other things which can be derived from observing the graphs, but I'll leave you to experiment with this on your own.

One note, though. The Efficiency graph is ROUGH. It is not a one hundred percent accurate read-out of cache performance, but more of an indication of how well the it is doing. The usefulness of 1/10 sec updates is questionable (but amusing to watch none the less). If you want a more realistic image of your cache's performance, then use update intervals above one second and watch how the graphs behave with real world applications.

Click Done or the close-window gadget to close the statistics window. Note that PowerCache also closes statistics windows automatically when you quit it.

The Clear gadget will erase the graph window. This is useful if you don't want differently timed graphs to appear in the graph window at the same time (i.e. you mainly use it when changing the update frequency rate)

If you temporarily want a statistics window out of the way, click the window's Zoom gadget (top right, next to the Depth gadget). This will turn the window into a tiny menu bar, which can easily be dragged out of the way. Also, 'zoomed' windows will not render graphics and will not use nearly as much CPU time as 'unzoomed' windows.

1.15 The Menu System

PowerCache has a few menus attached to its main window. The structure of the menus resembles the kind of structure you'd expect any preferences type program to have. In fact, most of the menu functions are exactly the same as those of any other commodity.

Briefly, the menu functions do the following:

Project_Menu

Open...

This function will open a set of preferences that you have previously saved with PowerCache. The current (active) set of preferences will be overwritten.

Save As...

This function will save the current set of preferences in a file chosen by you. A file requester will be opened, allowing you to select an appropriate directory and file to use.

About

Gives you some version information

Hide

Closes the main PowerCache window, but allows PowerCache to remain active, running in the background. This is the same as clicking Use or the close-window gadget in the main window.

Quit

Select this menu item to permanently quit PowerCache. The current set of preferences will be saved in a file in ENV: so that PowerCache may be restarted without having to reconfigure everything all over.

Edit Menu

Reset to Defaults

This menu item will remove any active caches that are installed and then reset the hotkeys and checkbox gadgets to their default values.

Last Saved

Select this item to load the set of preferences that were last saved using the Save gadget. Logically equivalent to using the Open... menu item with ENVARC:PowerCache.prefs as the file to load.

Restore

Restores all settings to the values that were active before the main preferences window was opened. This does NOT apply to any changes made to running caches, nor does it remove new caches that were added in the mean time.

Options menu

Create icons?

Select this item to make PowerCache save icons with every set of preferences it saves. This is handy if you keep a number of different preference settings around, and you want to easily start PowerCache with any one of these. Simply click the icon of the preferences file to run PowerCache with those preferences.

The drawback of icon creation is (of course) that icon files take up valuable hard disk space (not much, but some), so don't save icons if you never use them anyway.

Under Kickstart 3.0+ (V39+), the menus will appear in the attractive new black-on-white menu style. Kickstart 2.04 (V37) users will have to settle for the traditional grey-on-black menus.

1.16 Stopping PowerCache

PowerCache can be stopped in two ways. The easiest way is to select Quit from the menu, but you may also use the Commodities Exchange program. This program is part of the Amiga operating system distribution, and can usually be found in the SYS:Tools/Commodities drawer. For further information on how this program works, please refer to your AmigaDOS manual.

Regardless of how you chose to exit PowerCache, it will first close all windows (info-windows and statistics windows), then remove each installed cache (with whatever subsequent disk activity this may cause) and finally free the miscellaneous system resources it uses.

Note that clicking the close-window gadget in PowerCache's main preferences window doesn't make PowerCache quit. The close-window gadget is just another way of selecting Hide from the menu, and only makes PowerCache close the main preferences window.

When you exit PowerCache, it will save the current cache configuration in a preferences file in ENV:. This means that the next time you start PowerCache, all caches will be re-installed automatically. However, since ENV: is usually in RAM:, this preferences file will not survive a reset. To permanently save a set of preferences, click the Save gadget in the main preferences window, or select "Save as..." from the menu.

1.17 Troubleshooting

If for some reason you can't get PowerCache to work, always feel free to contact me. I will try to help as much as I can. Also, if you find a bug in the program, then PLEASE send a report to me as soon as possible. I want this program to be as bug free as I can possibly get it.

Here's a small table of a couple of typical problems that may occur:

Situation	Solution
-----------	----------

PowerCache does not run	reqtools.library installed correctly? Low on memory?
-------------------------	---

PowerCache is not localized	locale.library present in LIBS:? PowerCache catalog installed in LOCALE:? Correct language selected with Locale prg?
-----------------------------	--

System guru on the first	Driver contains a strange jump table.
--------------------------	---------------------------------------

access to a cached drive Select "Info" on the device and mail author the result.

Using file caches result in This is a very strange problem that occurs strange checksum errors on the on some systems. Using a prefetch of 1 device being cached usually cures the problem.

Cache apparently does not work PowerCache still has problems with overlaid device drivers, such as mfm.device (PC0:) which runs on top of DF0:. No work-around yet.

No online help PowerCache: assigned?
 Help files installed?
 AmigaGuide.library installed in LIBS:?
 'Use Help' checkbox (in main window) set?

PowerCache refuses to let you You're running low on signal bits. I'm sorry, add more than X caches. there is no way I can fix this - the problem is in the operating system. It reserves 16

Error messages in windows signal bits for each task, and that's cutting seem to fail - as does it a bit short for a program like PowerCache. statistics and info windows PowerCache uses at least 6 bits for pure administrative purposes, plus 1-2 bits for window handling, 2 bits per read/write cache and 1 bit per read-only cache.
 Fix: Remove one of the installed caches,~or close one or more statistics or info-windows.

I'll try keeping this table as up to date as I possibly can, but that does mean that you'll have to write me and ask some questions :-)

1.18 PowerCache Error Messages

PowerCache is a very paranoid program. It checkes everything before using it, and so it is capable of giving off a huge amount of different error messages. I've tried to categorize these messages, so that you know how to respond to most of them. In any event, if PowerCache aborts with an error of some kind, and the following table doesn't help you, feel free to write me. Here goes:

"Low memory" type error messages include the following:

- Unable to create a file cache for <device name>
- Unable to allocate primary cache memory for <device name>
- Unable to open '<some.device>', unit <some number>
- Could not create IO Request block for <some device>
- Could not create a message port for <some device>
- Can't get a commodities.library broker
- Unable to create a message port for use with commodities.library
- Not enough memory
- Cannot open window
- Cannot create message port
- Unable to create a message port for cache buffer communication
- Unable to create a message port for use with Intuition windows
- Unable to spawn stat window bumper task

Could not add main window to list of active windows
Could not open main preferences window
Could not create menus for main preferences window
Could not create gadgets for main preferences window
Failed to create intuition BOOPSI "GetFile" gadget class
No visual info available for default public screen
Out of memory reading cache preferences
Unable to add stat window to list of active windows
Unable to initialize ARexx host interface
Unable to set ARexx variable
Unable to allocate cache IO request
Unable to initialize cache timer
Out of memory adding device to cache list
Could not access system default font

Any one of these messages indicate that you are running low on memory. You should be able to correct the situation simply by freeing up some. This includes closing windows, deleting files from RAM:, RemRAD'ing, stopping programs or even rebooting to reduce memory fragmentation.

"DOS/file problem" type messages include the following:

Unable to create a file cache for <somedevice:>
Unable to set up AmigaGuide online help system [amigaguide error]
Cannot open database [amigaguide error]
Can't read preferences! (Read error)
Can't read preferences! (File is mangled)
Preferences file is mangled. Will use last saved/defaults
Unable to read preferences file. Using last saved/defaults
Can't read preferences. (Could not open file)
Unable to save project icon. (Write error)
Unable to open project icon file. (Open failure)
Can't write preferences! (Write error)
Can't write preferences! (Could not open file)

These messages all indicate some problem with AmigaDOS. Typically, you'll get an error message if some file is being created and the device fills up, or if the device has a read/write error.

It is almost impossible to provide a general method for correcting these errors. If the problem is related to a write operation (saving prefs files, writing file caches etc), then try a different disk. Otherwise, if the problem is related to reading a file (preferences, AmigaGuide database, icon files, etc), then try manually copying the file(s) to some other disk. This will reveal any hidden AmigaDOS read/write errors that may be present on the source disk.

A reboot may also sometimes help - especially if you've been fooling around with some new software that has played tricks on AmigaDOS.

"Commodities problem" error messages include the following:

An error occurred while initializing the commodities broker system
Can't get a commodities.library broker

- Try disabling or quitting whatever commodities you are using, one by one. The error is most likely an indication of some conflict with the software you're using.

'<some hotkey>' doesn't seem to be a valid hotkey. Resetting to '<some other hotkey>'

- Self-explanatory. You have specified a hotkey of some kind, which the Commodities broker does not understand.

Commodities hotkeys follow certain guidelines. You may use any number of "qualifiers" and a key to specify a hotkey. Qualifiers are "dead" keys that do nothing on their own, such as "shift", "alt" or "ctrl", but may also be events, such as disk insertions, mouse clicks or even timer events. Here's a very brief list of qualifiers:

```
rshift    - Right shift key
lshift    - Left shift key
shift     - Any shift key
ralt      - Right alt key
lalt      - Left alt key
alt       - Any alt key
control   - The "ctrl" key (all kickstart versions)
ctrl      - The "ctrl" key (2.1 and upwards ONLY)
lcommand  - Left Amiga key
rcommand  - Right Amiga key
command   - Any Amiga key
```

There are more of them, and they can all be found in your AmigaDOS documentation.

The "key" part of a hotkey may be any key whatsoever. You may use meaningful labels for special keys, such as "help" for the HELP key, or "uparrow" for the up-arrow key. Here are a couple of legal hotkeys:

```
control alt s
lcommand shift y
control lshift rshift rcommand help
```

"Library problem" error messages include the following:

```
You need icon.library v37+ in LIBS:
You need commodities.library v37+ in LIBS:
You need gadtools.library v37+ in LIBS:
You need utility.library v37+ in LIBS:
You need reqtools.library v38+ in LIBS:
You need rexsyslib.library V35+ in LIBS:
```

These are all pretty obvious. You're going to have to find the library in question and place it in your LIBS: directory. This shouldn't be a problem, since reqtools.library is included in the PowerCache distribution, and the rest of the libraries are part of the standard Workbench 2.04 distribution.

In rare cases, you may also get a message like this when memory is running low. Lastly, remember to check that your LIBS: assignment points to the right place.

There are a couple of unique error messages that deserve special attention:

'somedevice:' is not a proper filing device

- You should never get this message. It indicates that a device appears to be a filing device, but really isn't. In any event, there really isn't anything you can do about it. Reset the machine, and if that does not cure the problem, select Info on the device in question and mail me the result. I'll have a look at it, then.

'somedevice:' has no controlling task

- Devices usually configure themselves and start up the device driver as soon as it is Mounted. However, if something goes wrong during the Mount operations, the device may linger on in a state of "limbo" - not really active, and not really inactive.

For example, if you only have the internal DF0: drive, and try Mounting PC0: and PC1:, PC0: will start up as normal, but PC1: will fail in opening the trackdisk.device unit 1 (which is the first external drive (A2000 notwithstanding)). Now, if you try adding a cache to PC0:, this will work just fine, but if you try adding a cache to PC1:, you'll get this error message.

There is no cure for this problem - other than to get one more external drive, that is :-)

Loop detected. You are caching two devices on top of eachother!

- Even though PowerCache is extremely liberal regarding the use of file caches, there are some situations that PowerCache will not let you do. For example, if you add a file cache for DF0: on DH0:, and then try adding a file cache for DH0: on DF0:, you'll get this message.

The solution is simply to make one of the caches a memory cache in stead of a file cache.

WARNING! The volume in drive <somedrive:> has been removed. Dirty buffers DO exist for this drive. You MUST re-insert the volume, or face the possibility of such massive data loss and structural corruption that only a complete re-formatting will again make it usable to AmigaDOS for data storage.

- This is a very large error message, and for a good reason, too. PowerCache goes to great length trying to explain this one, and it also gives you the cure: Re-insert the disk!
-

PowerCache will give you this warning if you eject a disk too soon, and the (read/write) cache still contains dirty (unwritten) sector data.

Can't find a public screen

- You'r machine is floating in never-never land. You probably won't even see this message before the machine crashes. Funny, isn't it? :-)

Of course there are other error messages, but these are all so obvious that they are pretty much self-explanatory. If you feel that some message needs an elaboration in this document, feel free to write me.

1.19 A few Important Notes

Rule number one: Never, NEVER, EVER, EVER, eject a floppy disk with a read/write cache until the drive has been silent for the amount of time specified as the Purge timeout value for that cache. PowerCache does have exception code to handle dangerous situations like this, but there are no guarantees that the disk will survive. This warning also applies for doing a DiskChange on a fixed disk (a harddisk).

Also, BE CAREFUL about using an optimizer on a unit that has a read/write cache. Make sure it is properly purged before you do anything like that. In fact, you should probably quit PowerCache while you optimize the drive. Just to be on the safe side.

1.20 Known Bugs

PowerCache is reasonably free from bugs. The program is still a bit on the shady side when it comes to certain internal structure arbitration, but this will be fixed very soon. In any event, it should not cause any problems in all but the most extreme conditions.

PowerCache still has problems with file caches when you use a prefetch of more than one. I haven't got a clue why this happens, but I am working on it.

I'm sure there are a few bugs still in there. You should probably test PowerCache to the fullest before allowing it to access your most precious data. Try running a disk speed program (such as DiskSpeed) on each cached device. This usually reveals any problems right away.

1.21 Benchmarks

The following charts are the result of a series of benchmarks I did, using MKSoft's DiskSpeed version 4.0. It is a relatively crude speed comparison of PowerCache against the commercial product HyperCache.

The test was carried out on an Amiga 2000, equipped with RCS Management's awesome Fusion Forty card (68040/28 mhz), with all caches on (but without Copyback), running Kickstart 3.1 (v40.9, shadowed in 32 bit memory) on the second (20 meg) partition of a Quantum LPS 105 meg drive. Please take all results with a grain (or a truckload) of salt. Disk benchmarking programs are notorious for giving out results that don't compare particularly well with actual (daily) drive usage, and DiskSpeed is no exception.

1. NORMAL HARDDISK CAPABILITY

Testing directory manipulation speed

```
File Create      212 files/s
File Open       302 files/s
Directory Scan  651 files/s
File Delete     434 files/s
```

```
Seek/Read:      38 seeks/sec
```

Testing with a 512 byte buffer (LONG-aligned, MEMF_FAST)

```
Create file     163088 bytes/s
Write to file   214080 bytes/s
Read from file  350208 bytes/s
```

Testing with a 4096 byte buffer (LONG-aligned, MEMF_FAST)

```
Create file     172544 bytes/s
Write to file   185856 bytes/s
Read from file  540672 bytes/s
```

Testing with a 32768 byte buffer (LONG-aligned, MEMF_FAST)

```
Create file     330125 bytes/s
Write to file   437179 bytes/s
Read from file  615418 bytes/s
```

Testing with a 262144 byte buffer (LONG-aligned, MEMF_FAST)

```
Create file     520385 bytes/s
Write to file   627138 bytes/s
Read from file  696517 bytes/s
```

2. CACHED HARDDISK PERFORMANCE

(Cache Configuration: 6 sets, 64 lines, prefetch 8)

	PowerCache.000	PowerCache.020	HyperCache.030
--	----------------	----------------	----------------

Testing directory manipulation speed

File Create:	232 files/s	243 files/s	161 files/s
File Open:	448 files/s	455 files/s	561 files/s
Directory Scan:	1162 files/s	1171 files/s	1727 files/s

File Delete:	714 files/s	714 files/s	909 files/s
Seek/Read:	967 seeks/s	980 seeks/s	1259 seeks/s

Testing with a 512 byte buffer (LONG-aligned, MEMF_FAST)

Create file:	615232 bytes/s	629120 bytes/s	167232 bytes/s
Write to file:	690496 bytes/s	704960 bytes/s	203200 bytes/s
Read from file:	697536 bytes/s	709824 bytes/s	1074560 bytes/s

Testing with a 4096 byte buffer (LONG-aligned, MEMF_FAST)

Create file:	2932224 bytes/s	2963456 bytes/s	173568 bytes/s
Write to file:	3411456 bytes/s	3438592 bytes/s	185856 bytes/s
Read from file:	3418112 bytes/s	3446784 bytes/s	4282880 bytes/s

Testing with a 32768 byte buffer (LONG-aligned, MEMF_FAST)

Create file:	5419008 bytes/s	5439488 bytes/s	350503 bytes/s
Write to file:	6565888 bytes/s	6582272 bytes/s	385551 bytes/s
Read from file:	6610944 bytes/s	6631424 bytes/s	6840320 bytes/s

Testing with a 262144 byte buffer (LONG-aligned, MEMF_FAST)

Create file:	6651904 bytes/s	6619136 bytes/s	491520 bytes/s
Write to file:	8159232 bytes/s	8159232 bytes/s	588353 bytes/s
Read from file:	8159232 bytes/s	8159232 bytes/s	7536640 bytes/s

The test reveals two differences. First of all, I'm sure you've noticed HyperCache's absolutely appalling speed when it comes to write operations. This is not really because of bad programming on HyperCache's part, but more because of the way a read-only cache is implemented. A test with PowerCache in read-only mode would probably turn out the same way.

Secondly, you will notice that HyperCache is somewhat faster than PowerCache, for small reads. This is because of a number of things:

- PowerCache was tested in read/write mode. Since PowerCache's read-only mode is significantly faster than the read/write mode, the above test isn't entirely fair (HyperCache uses the read-only caching mode). See below for a comparison of PowerCache in read/write mode, and PowerCache in read-only mode.
- HyperCache is optimized for devices with 512 byte sectors. PowerCache's routines are completely generic.
- HyperCache's cache routines are written in assembly language. PowerCache is written almost entirely in C.
- HyperCache.030 is optimized for 68030 processors - PowerCache only for 68020s. No big deal, but it does count.

Eventually, PowerCache will have special case code for devices with 512 byte sectors, because these are so common. Right now, I want to keep PowerCache as generic as possible, so that I can easily implement new features.

Similarly, I am not going to hand-write the cache routines in assembly just yet. There are still some algorithmic problems to address (and features to add), and this is best accomplished using C.

Lastly you will notice how PowerCache eats in on HyperCaches lead in transfer speed, as the buffer size increases. This is (probably) because of the very highly optimized memory copy function that PowerCache has.

Let's take a look at how well PowerCache performs in read-only mode. As previously mentioned, read-only caches are quite a bit faster than read/write caches, so it actually makes more sense to compare PowerCache with HyperCache if PowerCache is running in read-only mode. Well, I did that, and this is the result:

3. READ-ONLY CACHE PERFORMANCE (Cache Configuration:~6 sets, 64 lines, prefetch 8)

	PowerCache, R/W	PowerCache, R/O	HyperCache
Testing directory manipulation speed			
File Create:	238 files/s	166 files/s	161 files/s
File Open:	446 files/s	499 files/s	561 files/s
Directory Scan:	1171 files/s	1514 files/s	1727 files/s
File Delete:	714 files/s	833 files/s	909 files/s
Seek/Read:	973 seeks/s	1170 seeks/s	1259 seeks/s
Testing with a 512 byte buffer (LONG-aligned, MEMF_FAST)			
Create file:	620864 bytes/s	187626 bytes/s	167232 bytes/s
Write to file:	696832 bytes/s	213440 bytes/s	203200 bytes/s
Read from file:	702976 bytes/s	905536 bytes/s	1074560 bytes/s
Testing with a 4096 byte buffer (LONG-aligned, MEMF_FAST)			
Create file:	2936320 bytes/s	184320 bytes/s	173568 bytes/s
Write to file:	3421184 bytes/s	191488 bytes/s	185856 bytes/s
Read from file:	3408384 bytes/s	4076544 bytes/s	4282880 bytes/s
Testing with a 32768 byte buffer (LONG-aligned, MEMF_FAST)			
Create file:	5410816 bytes/s	392235 bytes/s	350503 bytes/s
Write to file:	6557696 bytes/s	466944 bytes/s	385551 bytes/s
Read from file:	6602752 bytes/s	7114752 bytes/s	6840320 bytes/s
Testing with a 262144 byte buffer (LONG-aligned, MEMF_FAST)			
Create file:	6619136 bytes/s	671344 bytes/s	491520 bytes/s
Write to file:	8093696 bytes/s	726423 bytes/s	588353 bytes/s
Read from file:	8126464 bytes/s	8192000 bytes/s	7536640 bytes/s

I have repeated the HyperCache results from table 2, as well as a table of PowerCache running in read/write mode. As you can see, the difference

between HyperCache and PowerCache has narrowed considerably, and indeed PowerCache is faster than HyperCache for most write operations. I expect PowerCache to completely overtake HyperCache before long. So if it is speed you're after, you can't go wrong with PowerCache.

All in all, PowerCache compares quite well with HyperCache. And considering the amount of documentation and additional features that you get only with PowerCache, I'm sure you'll agree that PowerCache is a very viable alternative. (The fact that HyperCache costs US\$ 50 doesn't help HyperCache, either)

1.22 Using Online Help

Using the Online Help Features

PowerCache has complete online help for every single menu and gadget. What this means is that if you are using the program, and can't remember what a specific gadget or menu does, you can make PowerCache look up the place in this manual where it is explained. Of course, you need to know exactly how to make PowerCache do that.

There are two distinct methods of activating help. One for menus, and another for gadgets. Menu help is activated by pressing the right mouse button, moving the pointer to an item in the menu and then pressing the HELP key (don't let go of the right mouse button until the AmigaGuide page is showing). For example, to get help on the menu item Open..., you press the right mouse button, move the pointer to the Open... item in the project menu, and then (still holding down the right mouse pointer), press HELP on your keyboard. It may sound complicated, but once you get a hold of it, it's easy.

Getting help on a gadget function is even easier. You just move the mouse pointer over the gadget you want help on, and then press the HELP key. That's all it takes.

While the AmigaGuide help window is showing, it is possible to simply resize or depth arrange it and then continue to use PowerCache. Pressing HELP several times without closing the help window is also perfectly legal, and will make AmigaGuide look up the new location in the manual. You won't get another window, the existing one will simply shift to the new location.

In case you're wondering, it is okay to quit PowerCache while the help window is still open. It will be closed by PowerCache, as a part of the cleanup procedure.

I have tried to make this manual as comprehensive as I possibly could. Still, there may be functions that you feel should be explained, or sections that are explained, only not good enough. For whatever the reason, feel free to make suggestions on how to improve the manual.

1.23 The Future

PowerCache is far from finished. There are still many ways to improve it. However, I do not want to turn PowerCache into a huge "do-all" thing. I do not plan to build in virus checking, screen blanking, mouse accelerating, PopCli shell hotkeys, new file systems, disk repair routines, disk optimization routines, device stacking (crunching) or anything like that. Too many great programs have become cluttered with out-of-context features, and I don't want this to happen to PowerCache.

You are most welcome to make suggestions regarding the future of PowerCache. Don't be shy, or hold back because your suggestion sounds strange, unreasonable or just plain dumb. Let me be the judge of that.

Here are a couple of things that I might incorporate into the program some time in the near future:

- Improved ARexx Interface

Even though PowerCache has an extensive ARexx interface, there are still a few commands left to implement. Also, some of the existing ARexx routines could do with a bit of optimization.

- More statistics

Currently, PowerCache keeps track of two statistical variables: amount of buffer cache memory used, and relative cache efficiency. I have a range of other interesting things planned that will help you choose the right cache configuration parameters.

One thing I am planning is an indicator on exactly how well a prefetch value is working for a drive. It is very difficult to hit a good prefetch value, and when (if) you finally succeed, it may not remain good as your drive becomes fragmented over time. Some form of indicator on prefetch efficiency is absolutely required.

- ~More cache algorithms

I have a whole textbook of different caching algorithms lying around. I'll try to have a few of the most interesting incorporated into PowerCache sometime soon. The internal algorithm routines are fairly modular, so this should not be a problem. I may even implement cache algorithms as external library protocols that can be developed and maintained by third party developers.

- ~Transparent sector encryption and protection system [TRANSEPS?]

This is a though I have been playing with for some time. Being hooked into such a central piece of code as a device driver really is also means that PowerCache has the potential to become the ultimate encryption utility. Without PowerCache and the correct password, the cached device would be completely useless. Let me know what you want, people.

- Direct support of "overlaid" devices

Currently, you are not able to cache devices like PC0: (mfms.device), which are layered on top of another device (DF0: in this case). The next release of PowerCache will most certainly support these devices.

- Faster cache routines

The current cache routines run pretty fast as it is, but can be made to run even faster. My goal is to achieve 8 megs/second on a standard Amiga 4000 (primed). As previously mentioned, PowerCache has already reached 8 megs per second on a 28 mhz '040, so 8 megs on a 25 mhz '040 should not be impossible (hard, but not impossible).

- Verify option

It would not be too difficult to incorporate a Verify mode on cached device units. I may include this feature as soon as in the next release, even though I don't think too many people will use it (as it will obviously hurt performance)

- Smart caches

PowerCache already includes just about everything needed to make it the worlds first truly intelligent cache. With "Smart mode" on, a cache will be able to utilize drive idle time to fill the cache buffer with data that hasn't been requested yet, but will be later. This will have a tremendous, positive impact on cache performance in general, and even more so for specialized tasks such as directory manipulation.

- ~AppIcon/AppWindow for configuring caches

This way you could add a cache to a device simply by dragging the Workbench icon for that volume over PowerCache's AppIcon/Window.

- Whatever you want

I'm open for suggestions.

1.24 Technical Information

Autoconfiguring cache sizes

The procedure of automatically determining an appropriate cache size is relatively easy - and also relatively relaxed. It may not always derive values which are 100% perfect, but it gets pretty close.

We simply find a percentage of the available memory, and convert this number into sets and lines (using the current amount of prefetch). The percentage is 10% for a small buffer, 30% for a medium sized buffer, and 50% for a large buffer. The resulting value will be rounded down to match the volume capacity if it turns out to be larger than that (a 3 MB cache on DF0: is a bit of a waste).

The procedure has been made to work for file caches as well. In stead of using the amount of available memory as the basis for the calculation, the amount of free space on the file caching device will be used. For example, caching DF0: on DH0: will use the amount of free space on DH0: as the basis for the percentage calculation.

Turning the result into sets and lines is pretty easy, but keep in mind that it is done using the current amount of prefetch. In other words, the auto configuration procedure will determine sets and lines based on

- The amount of free memory
- The disk capacity
- The current prefetch setting

This allows you to use auto configuration on fragmented devices as well as on optimized devices. You simply set prefetch first and then make the appropriate auto configuration selection afterwards.

1.25 Author's Address

How to Reach the Author

As mentioned several times in this manual, I am always open to suggestions on how to improve the PowerCache program, and how to improve this manual. You are most welcome to write to me regarding this, and my address is:

Michael Berg
Sjællandsgade 56, 4
8900 Randers
DENMARK

You may substitute the "æ" character with the more widely recognized "ä" (a umlaut) if you wish. If you need to get in touch with me quickly, you can always call me up on the phone. The number is:

+45 864 22402

+45 is the country code for Denmark. You'll find information on how to dial another country in your phonebook, or from your local telecom service.

I do not have any access to Internet or Usenet, but I do have an account on Fidonet on this address:

2:230/816.8

If you want to send me a matrix, please consider crashing it directly to my boss node. Overseas matrix netmail routing is not what it used to be (was it ever? :-)

1.26 Credits

PowerCache was developed using the following software:

Program compiler: Aztec C v5.2a - © Manx Software Systems, Inc.
User interface: GadToolsBox v2.0c - © Jan van den Baard
ARexx interface: ARexxBox v1.0 - © Michael Balzer
Requesters: Reqtools.library - © Nico François

The author of PowerCache gratefully acknowledges the excellent work by these authors, and also by the ones that were forgotten. Thank you for making life so much easier :-)

The following tools proved invaluable in the development of PowerCache:

Chelp v1.30e - © Robert Wahlström & Mathias Widman
DiskSpeed v4.0 - © MKSoft Development
CygnusEd - © CygnusSoft Software

Further greetings & thanks go to:

Kenneth Perto - For excellent Fidonet support and for risking his hardware for testing PowerCache :-)

David Le Blanc - For showing the Amiga community how a cache works, by
Andreas Arens releasing the SmartDisk source code (Dave authored
SmartDisk and Andreas beefed it up)

Reza Elghazi - For the french (français) translation
Gerhard Kupka - For the german (deutsch) translation
Mario Gonzales - For the norwegian (norsk) translation

1.27 Shareware notice

A note on Shareware

PowerCache is shareware. Let me just take a few moments of your time to explain this popular concept for those of you who are still unfamiliar with it.

There are at least two forms of "free software". One is freeware, and as you may have guessed, this means that the software is completely free of any charges. You may use it, copy it and do whatever you want with it, without having to pay anything to the author.

Shareware is almost the same thing. You are free to use and copy the program as crazy as you want, but continued use of shareware software requires registration. That is, if you use the program frequently, you must register it with the author. This usually entails that you have to mail a couple of bucks to the author, who will then be extatic with joy and probably send you something in return (his latest creation, for example).

However, most people still do not really do this. They continue to use shareware programs without registering, because they know that nobody has the authority to actually DO anything about it. As a shareware programmer, you have very few means of physically enforcing the laws you are asking people to follow, so therefore you rely entirely on the conscienceness of the end user to actually register.

Unfortunately, consciencious users are an endangered species. It has become very popular to simply take free software for granted, and this abusement of the shareware concept is growing to be a real problem for those of us writing shareware programs. We are only receiving a fraction of what we should be receiving, and there's nothing we can really do about

it. Other than to stop programming, of course.

To make things worse, there are currently quite a few shareware programmers giving the rest of us a hard time. They ask you to register, but when you do, you never hear a word from them again. This is unfortunate, since any first-time registree will be strongly discouraged from ever trying something like that again.

Due to these facts, many programmers no longer release fully featured versions their programs any more. In stead they release release "crippled" programs that have some features disabled, or only work on a limited set of data. They are usually called evaluation or demo versions. This is not only a sad trend, but also disappointing for the end user who would of course like to try the Real Thing before registering. However, I feel this tendency is here to stay. Programmers are simply fed up with the overly relaxed attitude people have towards shareware.

PowerCache is being released into the public domain in a crippled, but freely distributable form. This version is useless over long periods of time, but still allow you to try it out to its full potential, since no features are disabled. The evaluation version differs from the registered version in these counts:

- It will exit quietly after approximately 30 minutes (you may start it again as may times as you like, however)
- It insists on showing you the About requester when you start it

These constraints are relatively easy to live with, and will certainly allow you to try out the program before
registering
.

1.28 How to Register

How to Register

To register you must fill out the included registration form and mail it to me along with at least US \$20. You are welcome to send more money (which will probably mean faster development and more updates), but US \$20 is the minimum required fee.

I will then register you as a user, and immediately mail you a key file on a branded 3,5" disk. Not wanting to waste perfectly good disk space, I'll probably throw in a couple of other PD utilities as well. Please allow a few weeks for delivery (typically much less).

The keyfile you receive will unlock the evaluation version you already have, as well as future update releases of the program.

Even though it does entail a slight risk, I encourage you to simply send actual currency in plain old envelopes. Any currency is accepted, but do not send checks or money orders made out in anything but DKK. I have a hard time arguing with the bank clerk every time I want to cash in a

foreign check, since they are always so terribly nervous it will bounce (tricky situation with a foreign check).

Naturally, postal money orders (those reddish cardboard forms :) are also welcome. Ask at your local post office.

Registered users of another program of mine, PowerData, may register PowerCache for half the price - 10 US\$. It really does pay to register shareware, people!

Those of you with modems, and Fidonet access, can always FReq the latest evaluation version of PowerCache at one of the following (Danish) sites (magic name POWERCACHE):

```
2:230/815, +45 86-720273, V32B, HST 16800, V42B
2:230/816, +45 86-720274, HST 14400, V42B
2:230/817, +45 86-293910, V32, V42B
2:230/1815, +45 87-370010, UISDNB,ISDNC
```

1.29 Keyboard Navigation

Using PowerCache Gadget Shortcuts

As you have probably already discovered, most gadgets in PowerCache's windows have keyboard shortcuts. The key to use to activate any particular gadget will be underscored in that gadget's text field. For example, the Save gadget in PowerCache's main preferences window, looks something like this:

```
+-----+
| Save |
+-----+
```

This indicates that 'S' is the shortcut to use for the Save function. Actually, this keyboard shortcut causes PowerCache to perform the exact same operation as if had you physically clicked the Save gadget using the mouse.

Most gadget shortcut operations are pretty obvious, but others require a small explanation. First of all, there's the matter of lowercase versus uppercase (shifted) shortcut keys.

Simple Button style gadgets have only one mode of operation. You press it, and all the magic happens. However, gadget types like Cycle, Listview and Slider actually have two shortcut keys associated with them. Pressing the normal, lowercase shortcut key will make the next selectable item the current one, and pressing the shifted, uppercase shortcut key will make the previous item the current one.

Easy, huh? Go ahead and try it out on the Cacheable Devices listview gadget in the main preferences window. Start by pressing the normal shortcut key. The first item on the list will become selected. Press the shortcut key once more, and the next item becomes the selected one. Now

try using the uppercase shortcut key. The first item will become selected once again.

I'm sure this system will grow on you quite fast. Using a mouse is convenient, but not always as fast as you'd like it. After having knocked over a couple of beers reaching for the mouse, I'm sure you'll agree with me. Use the shortcut keys, and save your rug a trip to the cleaner's.

Please note that listviews, cycle and slider gadgets only have keyboard equivalents when running Kickstart 3.0 or later. This is mainly because of deficiencies in the 2.04 implementation of the gadtools.library, and may be fixed in some future release of the program.

One more thing: PowerCache is localized, and since different shortcut keys are applied for different locales, be sure to select the appropriate language before getting too accustomed with using a particular set of shortcut keys.

(PowerCache automatically maps shortcut keys to gadgets, so if you want to mail me a locale translation, you don't have to worry about which keys to use with which gadgets. Couldn't be much easier, could it? :-)

1.30 Hardware Requirements

Hardware Requirements

Where memory is concerned, PowerCache is a fairly modest program. It will run on a 512k machine without problems, only you probably won't be able to use cache buffers of any significant size with this amount of memory. You really need 1 MB or more if you want any real performance. As always, but more so in the case of PowerCache, more memory means better performance.

One tip on memory: Online help is quite expensive, in terms of memory usage. For example, running PowerCache without help (by unselecting the Use Help checkbox gadget in the main window) requires approximately 105k, while running it with online help (which is the default) requires an overwhelming 325k. Those are the results I got on my 9 meg machine, anyway. It could be that AmigaGuide is allocating all that memory because it can see that I have enough available. On a machine with less memory, it might not allocate that much. I advise you to experiment and find out exactly. No need to waste memory unnecessarily.

Other than that, PowerCache needs at least Kickstart 2.04 (V37.175) and reqtools.library v38+. If you want online help, you'll need the AmigaGuide distribution, and in particular, the amigaguide.library. This wonderful tool has just been made freely distributable by Commodore, so you'll be able to pick it up from popular PD distribution sites before long.

Certain features of PowerCache require Kickstart 3.0, but these are mostly related to the internal workings of PowerCache. An example of such a 3.0-only feature would be the low-memory handler. Under Kickstart 2.04 there is no such thing as a low-memory handler, so obviously PowerCache can't install one.

Besides having a bit more features under 3.0, PowerCache also does certain

things differently, depending on which OS it is running under. Under 3.0 it utilizes extra system features and functions whenever possible.

PowerCache is currently being distributed in two versions. One is the 68000 version, and the other is a 68020 optimized version. Naturally, you should use the version that matches the CPU you have. The 68020 version can be significantly faster than the 68000 version at certain tasks, so use it if you have a 68020, 030 or 040. You can use SysInfo (© Nic Wilson) to find out what CPU you have.

Future distributions may include special 68030 and 68040 optimized versions.
