

PowerCache

COLLABORATORS

	<i>TITLE :</i> PowerCache		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		October 27, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	PowerCache	1
1.1	PowerCache Documentation (5-Apr-1994)	1
1.2	Copyrights	2
1.3	Program Disclaimer & Liability	4
1.4	Background Information	4
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	11
1.10	Configuring your Caches	14
1.11	Close Window Gadget	15
1.12	Install Cache	15
1.13	Close Window Gadget	16
1.14	Title of Cache Config Window	16
1.15	Main Cache Size	16
1.16	Sets	17
1.17	Overhead	17
1.18	Power-of-two?	17
1.19	Lines	18
1.20	Total	18
1.21	Prefetch	19
1.22	Auto Configure	19
1.23	Caching Mode	19
1.24	Cache Type	20
1.25	Cache Algorithm	21
1.26	Cache Memory Type	21
1.27	Purge Timeout	22
1.28	Cache Filename	23
1.29	Get Cache Filename	23

1.30 Free After Purge	23
1.31 Show Statistics Window	24
1.32 Accepting the Configuration	24
1.33 Cancelling the Configuration	24
1.34 Rescan AmigaDOS DeviceList	25
1.35 Device Information	25
1.36 Remove an installed cache	25
1.37 (De)activate an installed cache	26
1.38 Purge an installed cache	26
1.39 Edit an installed cache	27
1.40 Obtaining cache statistics	27
1.41 Closing the Statistics Window	28
1.42 Title of the Statistics Window	29
1.43 Toggling Used %	29
1.44 Toggling Efficiency %	29
1.45 Changing the Update Frequency	29
1.46 Disposing of the Statistics Window	30
1.47 Clearing the Graphics Area	30
1.48 Show Window Hotkey	30
1.49 Enable Caches Hotkey	31
1.50 Disable Caches Hotkey	31
1.51 Popup Window	31
1.52 Beep	31
1.53 Save Settings	32
1.54 Use Settings	32
1.55 Cancel Changes	32
1.56 PowerCache in Use	33
1.57 The Menu System	33
1.58 Stopping PowerCache	35
1.59 Troubleshooting	35
1.60 PowerCache Error Messages	36
1.61 A few Important Notes	40
1.62 Known Bugs	41
1.63 Benchmarks	42
1.64 Using Online Help	45
1.65 The Future	46
1.66 Technical Information	48
1.67 Author's Address	49
1.68 Credits	50
1.69 Shareware notice	50
1.70 How to Register	52
1.71 Keyboard Navigation	53
1.72 Hardware Requirements	54

Chapter 1

PowerCache

1.1 PowerCache Documentation (5-Apr-1994)

Documentation for

PowerCache v37.115

A Flexible and Powerful Disk
Caching System for the Amiga

Written by Michael Berg
Copyright (C) 1993, 1994 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:

General Stuff

Shareware Notice	About shareware
Registering	How to register
Author's Address	How to reach me

Legal Issues

Copyright	You MUST read this
Disclaimer	Important disclaimer note

Documentation for PowerCache

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 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?
Technical information	Info on PowerCache's inner workings
Credits	Who did what?
Other	
Online help	How to use online help
AREXX	The ARexx Interface

- Aim for the impossible and you will achieve the improbable

1.2 Copyrights

Copyright Notice for PowerCache

Copyright

This program and all related documentation is copyright. You may not use, copy, modify or transfer the programs or documentation or any copy hereof, except as expressly provided in this document.

This program and the related documentation may be redistributed to and by public networks or individuals only in whole, and only for a basic fee covering the distribution overhead. PD services take note!

The PowerCache program and/or related documentation must not be distributed as part of any commercial product without a written permission from the author.

The PowerCache distribution may be archived and transferred onto and between BBS systems, as long as no files are added to, or removed from the original distribution. BBS sysops may add the usual "displayme" and BBS banner files to the distribution archive, but no more than that.

You may not electronically transfer your PowerCache key file or any copy of it, over a network, or otherwise distribute it to others in any form. You may make any number of backups of it, but all backups must always remain in your immediate possession.

It is strictly prohibited to reverse-engineer the PowerCache executable, in order to find out anything about the internal workings of the program, or indeed for any other reason. It is also strictly forbidden to modify the PowerCache executable in any way, except perhaps for compressing it for long-term storage. You may ABSOLUTELY NOT run any "patch" type program that removes any of the restrictions present in the evaluation version of PowerCache, or in fact does anything at all to change the way PowerCache works, either dynamically (as it is running) or statically (on-disk). Be warned that any such attempts may trigger certain built-in protection mechanisms in PowerCache, resulting in severe repercussions on the part of PowerCache.

Where clarification is required on any of the above paragraphs, the author of PowerCache shall decide the clarification.

Your key file

The file you receive upon registering is yours exclusively. As expressed above, you must 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 determining the origin of a runaway key file is not particularly difficult :-)

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.

- Sooner or later, the worst possible set of circumstances
is bound to occur

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 license 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.

- No one can feel as helpless as the owner of a sick goldfish

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. Consequently, 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 capacity to store large amounts of data, and to retrieve it quickly. It is not uncommon to see a stock 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. Some 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.

To futher the problem a bit, it is also impossible to speed up directory listings and do general harddisk maintenance (copying files from one place to another) simply by using a RAM disk.

The best solution to all of this is what is commonly known as a cache. A cache is a kind of buffer which 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, it is very old and uses fixed cache parameters that cannot be altered 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, and reliable cache program, and out of sheer 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, revealed later on in this document :-)

- Any sufficiently advanced technology is indistinguishable from magic

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. It can be used as both a read/write cache and as a traditional read-only cache, so 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. This scheme results in memory usage comparable with any fully associative cache, while still retaining the simplicity of a set associative cache. Further, since PowerCache has a low-memory handler built in, it will also pay attention to system memory panics and release unused buffer memory whenever the system calls for it (Kickstart 3.0 only).

For easy configuration, PowerCache offers a user friendly "gadtools" user interface. Every setting has a gadget, and extensive online help is available at the flick of the HELP button.

The advanced user will appreciate PowerCache's built in ARexx interface, which offers more than 30 commands. Using ARexx, you can in fact control every aspect of it from within other ARexx compatible applications.

With PowerCache, you get 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 the competition. Check out the benchmark section for a look at some hefty numbers :-)

- Horse sense is the thing a horse has that keeps it from betting on people

1.6 Overview of Features

Below is a brief overview of the most important features of PowerCache. Each item will be further explained later on in this document.

- 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
 - 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 (or ARexx)
 - Includes localized installation script for C's Installer utility
 - 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
 - Will intercept premature resets via a reset handler
 - 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 disk READ and WRITE operations
 - Any cache can be designated as being read/write or read-only
 - Memory friendly (includes v39-style low-memory-handler)
 - Selectable memory type for cache buffers
 - Selectable caching algorithm (though only LRU and RND 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 (more than 30 commands!)
 - Localized - runs in eight languages (CatComp source included)
 - ~68000-68040 versions included
- Plenty of documentation
 - 180k+ 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.

- If a listener nods his head when you're explaining your program, wake him up

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/regtools.library from the distribution disk to your own LIBS: directory
- Make a directory called "PowerCache" on your system disk (hard disk)
I recommend either SYS:Tools/Commodities/ or SYS:WBStartup/
To make the following a bit clearer, let's just say you picked 'TARGET:'.
- Assuming the distribution disk is in DF0:, issue the following CLI command:

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

Next, make a "Documentation" directory, in the PowerCache directory you made in the previous step:

```
1> Makedir TARGET:Documentation
```

Now copy the documentation files using this command:

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

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

•~Put the following few lines into your S:User-Startup file:
(assuming that TARGET: was DH0:)

```
;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.

- Life starts at '030... fun at '040... impotence at '86!

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, See the installation notes on for a more detailed explanation of how to do 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).

LANGUAGE=<language> PowerCache allows you to directly specify which language

(locale) it should use for strings. Normally, IPrefs takes care of language preferences (though the Locale prefs program), but if you are running PowerCache PRIOR to IPrefs in your startup-sequence, the system always defaults to english. Since PowerCache doesn't respond to a subsequent change of language (or startup of IPrefs), I've provided this parameter so that you can specify the language directly.
Example: LANGUAGE=français

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: and ENV: are valid. If your startup-sequence doesn't assign these, 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 it is not supposed to cache.

Disk: A piece of magnetic material used to wipe dust off the disk head.

Disk Head: The point upon which the collection of dust is maximised.

Disk Drive: Equipment for collecting dust.

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 of 100 MB has some 170,000.

All devices have a controlling task (a running program) called a device driver, whos 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 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 FileSystem) 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 in existance. There shouldn't be any problems with future file systems either.

Prefetch

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 every time the the file system asks for a sector, PowerCache reads FOUR sectors into memory. 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 in sequence will be the next one the system will need", prefetching sectors in clusters is indeed a Good Thing. Keep in mind that it only takes slightly longer to read four sectors than it does to read one. It takes much longer to read four sectors individually than it does 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, such as Holger Kruse's ReOrg utility.

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. This means slow reads/writes, plus the fact that the cachebuffer will contain only one quarter of usable sector data. The rest is simply prefetched data which will never be used anyway.

That having been said, I feel I have to say that a too low prefetch value can reduce performance, too. The less prefetch, the more PowerCache has to work shuffling sector data around. This means that a prefetch of 4 on a fragmented disk may still be desirable over a value of, say, 2 or even 1. Finding just the right prefetch level can be a tricky situation.

The lesson here is that prefetch should be used only with some thought. Tune it to your disk fragmentation level. Use an optimizer whenever necessary, try lowering prefetch to 2 or even 1 (which means NO prefetch at all), but don't forget to test higher values, too. In short, experiment until you find just the right value.

Lines

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 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 already contains data that has been updated by the file system in a previous operation (known as a "dirty"

sector), then that line must be "flushed" back 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. More sector data in the buffer means a higher chance that a read or write operation will be able to complete with no or only little physical disk access. Naturally, it also means you have to set aside more memory for the buffers. Speed for memory (or vice-versa) is a common tradeoff with most computers.

Sets

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 reserving several of these buffers (sets) for the same device, you increase the cache's hit-ratio. Double the number of sets, and you double the ratio. It's that simple.

A typical cache configuration would be 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:

- The larger the drive, the more lines should be defined.
 - 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.
 - When run under V39 (Kickstart 3.0) and beyond, PowerCache uses a low-memory handler to automatically purge all cache buffers when the
-

system runs low on memory. This means that you can set up a cache buffer of a~comfortable 30 or 40 megabytes, and still have all of your memory available for normal applications. See the explanation of the 'Main Cache Size' display field in the Cache Configuration Window. Take me there!

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 detail in the technical reference section in this manual.

- If your only tool is a hammer, every problem looks like a nail (Anon)

1.10 Configuring your Caches

Configuring your Caches

This section describes how you use PowerCaches gadgets and menus to set up the cache configuration. 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.

When run, the first thing that should happen is that PowerCache opens its main preferences window. It is from here you control most of PowerCache:

```
+-----+
| [X] PowerCache Preferences | [^] [V] |
+-----+-----+
| | | |
| | | |
| | | |
| | | |
+-----+-----+
| Install Rescan Info | Remove Active: [/] |
+-----+-----+ Purge Edit Stats |
| Show Window [ ] +-----+
| Enable Caches [ ] | [/] Popup Window |
| Disable Caches [ ] | [/] Beep |
+-----+-----+
| Save Use Cancel |
+-----+
```

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 device driver to control the drives, and this device driver lies about unit numbers (i.e. it shows LUN addresses rather than AmigaDOS unit numbers, so a SCSI unit appearing at LUN #0 will show up as unit #0, a unit at LUN #6 will appear as unit #6, etc. etc). Gvp SCSI.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 :-)

The rest of the gadgets are devoted to installing, editing and removing cache buffers. Initially, some of them will be ghosted, which is because no appropriate selections have been made in one or both listviews.

Please select an entry from the absolutely brilliant and incredibly well drawn illustration shown above, for a more detailed explanation of each gadget.

- It is impossible to make anything foolproof
because fools are so ingenious

1.11 Close Window Gadget

Close Window Gadget

As I am sure you are already aware of how window closing works, I am not about to plunge into an in-depth explain of this very complicated operation. I merely wanted to point out to you that closing the window is functionally the same as pressing the Use gadget. PowerCache will remain dormant and active, and can be recalled using the Show Window hotkey.

- A continuing flow of paper is sufficient to continue the flow of paper

1.12 Install Cache

Installing a Cache

This gadget is used to install caches to selected devices. To use it, you must first select a device and unit in the list of cacheable devices.

Upon pressing Install, a new window with a bunch of gadgets will (hopefully) pop up, inactivating the main window. This window is the Cache configuration window, and contains gadgetry to specify every detail of a cache buffer:

```
+-----+
[X] Cache Configuration For [...]          [^][V] |
```

```

+-----+
|   Organize As...   |
|                   |
|                   | Main Cache: [#####] |
| [###] Sets        | Overhead: [#####] |
| [###] Lines       | Total: [#####] |
| [###] Prefetch    | Auto Configure: [S/M/L] |
+-----+-----+
| Mode: [RW/RO] | Type: [Memory/File] |
| Algorithm: [LRU/RND] | Memory Type: [A/C/F/24] |
| Purge Timeout: [###] | Filename: [_____] |
[?] |
| Free After Purge: [/] |
+-----+-----+
| Show Statistics Window: [_____] |
+-----+-----+
|   Okay           | Cancel   |
+-----+-----+

```

For a more detailed explanation of individual controls, please select any of the above links.

- Real knowledge is to know the extent of ones ignorance

1.13 Close Window Gadget

Close Window Gadget

Closing the Configuration window is functionally the same as clicking the Cancel gadget (i.e. to cancel the changes/installation)

- A fool must now and then be right by chance

1.14 Title of Cache Config Window

Title of Cache Config Window

Note - The title of the Cache Configuration window will include the name of the device which you are working on ('DF0:' for example).

- The seminar on Advanced Time Travel will be held a week ago

1.15 Main Cache Size

Main Cache Size

This read-only field displays the amount of memory required by the actual cache buffer. This is the buffer that holds the actual sector data during

caching, and is computed as sector size x prefetch x lines x sets.

If you are running kickstart 3.0 or later, PowerCache automatically installs a so-called low-memory handler. This handler responds to low memory system panics by releasing all unused or non-dirty sector buffer memory, thus allowing the system to stay on its feet a while longer.

Because of this low memory handler, you can in fact install caches which by far exceed the total amount of memory in your system. Say, 30 or 40 megabytes for example. When you do this, obviously PowerCache will eventually end up having allocated every single byte of memory you have. However, as soon as some other program (or the system itself) asks for memory, the low memory handler steps in and releases most or all of the 30-40 meg buffer, leaving more than enough free to satisfy the request. It can be argued that in a situation like this you will in fact be utilizing your memory better, all else being even. Memory which is not allocated is not really doing any good, you know.

- An elephant is a mouse with an operating system

1.16 Sets

Sets

This gadget is used to specify the number of sets you wish to reserve for this cache. Valid entries are any integer larger than 0.

For a more detailed explanation of the meaning of a "set", please refer to the section on general cache information. Take me there!

- Truth is stranger than fiction, because fiction has to make sense

1.17 Overhead

Overhead

This read-only field displays the total amount of memory overhead required by PowerCache to run the cache. You will need at least this amount of memory to install the cache at all.

- Sharks don't eat lawyers. Professional courtesies

1.18 Power-of-two?

What does "Power of Two" mean?

An integer (I) is said to be a power of two, if an integer N exists for

which the following equation is true:

$$2^N = I$$

That is, two lifted to the power of N must equal the integer. If you still feel a bit in the dark about this, have a look at the following numbers, which are all powers of two:

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, ... etc.

For each of these numbers, N equals:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, ...

In PowerCache, it is okay to enter values which aren't exactly powers of two, even though such a value may be expected by the program. In this case PowerCache automatically rounds the number you entered up or down to match the nearest power of two value. For example, an entry of 60 will be corrected to 64, and an entry of 40 will be truncated to 32.

- Jimi Hendrix's modem was a Purple Hayes..

1.19 Lines

Lines

This gadget is used to specify the number of lines you wish to reserve for this cache. Valid entries are power-of-two integers larger than 0.

For a more detailed explanation of the meaning of a "line", please refer to the section on general cache information. Take me there!

- If a cup is lowered into water, it gets extremely wet

1.20 Total

Total

This read-only field displays the total amount of memory required by PowerCache to set up and run a cache of the size you've specified as sets, lines and prefetch. This number is a sum of the size of the raw cache buffer itself, plus any administrative overhead which PowerCache needs to run the cache.

This reading is valid only for memory caches. File caches only require the amount of memory indicated by the Overhead display field, as the buffer itself will be residing somewhere else (on your HD, typically).

- If you need n items of anything, you will have $n - 1$ in stock

1.21 Prefetch

Prefetch

This gadget is used to specify the amount of prefetch you wish to define for this cache. Valid entries are power-of-two integers larger than 0.

For a more detailed explanation of the meaning of "prefetch", please refer to the section on general cache information. [Take me there!](#)

- Washing your windows to make it rain doesn't work

1.22 Auto Configure

Auto Configure

Those of you who are a bit unsure about how large or small to make a cache may simply click this 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.

Note that the autoconfiguration function obeys the current prefetch setting. If you set prefetch to 4, and then autoconfigure a "medium" cache buffer, the function will try to find a number of sets and lines that go well with a prefetch of 4. PowerCache does not attempt to find a suitable prefetch value, since it does not know the degree of fragmentation your harddisk has.

The technical details section in this manual describes in detail how the function actually selects lines and sets.

- All general statements are false

1.23 Caching Mode

Caching Mode

By default this gadget is set to R/W (read/write), which means that you want both read and write operations to 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 would you choose read-only over read/write mode? -It's hard to give a clear answer to this question. I've heard many arguments both for

and against either, and really can't clearly recommend one over the other. Generally speaking, a read/write cache will serve you very well, and may actually protect your drive from damage if a crash occurs. Corrupted data which would have been written out to your HD (thereby invalidating it) would end up in PowerCache's buffers, where they would do no harm.

On the other hand, if the crash occurs while other programs are also using the harddisk, things get complicated :)

A read-only cache will be a better choice in some situations. For example, if you create files on the drive, but hardly ever read them back, a read/write cache may hurt more than help. This is because sectors which are written out to disk will claim space in the cache buffer - space which would be more useful for caching those blocks which are read all the time.

One important note: Read/only caches are some 25% faster than read/write caches. The reason for this is simply that read-only caches require much less overhead to run. So if REAL SPEED is the name of the game, go for a read-only buffer.

Note: PowerCache is the only caching software currently in existence that allows you to select cache mode on a per-unit basis :-)

- All I want is a warm bed and a kind word and unlimited power

1.24 Cache Type

Cache Type

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

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 in a file on a floppy drive, even though this doesn't make much sense.

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 until the type you want is shown.

If you select 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 and filename 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 itself 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.

If you delete the cache file when PowerCache is not running, it will be recreated by PowerCache as soon as it is started.

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.

Note: PowerCache is the only caching software currently in existence which allows you to cache one device in a file on another device :-)

- An object will fall so as to do the most damage

1.25 Cache Algorithm

Cache Algorithm

This gadget allows you to select the 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, only one other cache algorithm has been implemented - RND. RND is a random sector selection scheme, and is used mostly in situations where data is read and written in more or less random patterns, and where LRU would not make much of a difference anyway.

My advice to you is to use LRU for maximum performance. RND is there as an option, so by all means feel free to experiment and find the setting which suits your level of drive activity the best.

Note: PowerCache is the only caching software in existence which offers a selection of caching algorithm :-)

- Manly's Maxim:

Logic is a systematic method of coming
to the wrong conclusion with confidence

1.26 Cache Memory Type

Cache Memory Type

This gadget selects the type of memory you wish to use for a memory cache. Currently, four types are available:

- CHIP - Chip memory
- FAST - Fast memory (including 32 bit memory, if available)
- 24BitDMA - Memory in the 24 bit range
- Any - Any memory type

For maximum performance, you'd better leave this gadget at "Any".

Note: If PowerCache can't allocate memory of the requested type, it won't allocate anything at all. For example, if you are running a FAST-memory-type cache in a 1 MB fast ram, 2 MB CHIP ram environment, and you suddenly run out of FAST memory, then PowerCache will NOT begin stealing CHIP memory in stead (even if there is plenty left).

Future versions of PowerCache will allow a fall-back memory type selection.

- Only adults have difficulty with childproof caps

1.27 Purge Timeout

Purge Timeout

This gadget is used for defining for how long a cache should be idle, before beginning to purge (write back) any updated ("dirty") buffers. The number is entered 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.

Note: This value is also the MINIMUM amount of time you MUST wait before powering down or rebooting. This is the only way to ensure that no data is left unwritten in PowerCache's buffers. (And if you don't have the patience for that, simply quit PowerCache before powering down, and wait for the drive(s) to stop).

As of version 37.115, PowerCache has a built-in reset handler, which tries to intercept and postpone premature resets long enough to allow remaining data in read/write caches to be written to disk. However, there are a number of restrictions to this feature, so don't count on it to work. See notes for a more detailed explanation of this.

- Faith is the quality that enables you to eat blackberry jam on a picnic without looking to see whether the seeds move

1.28 Cache Filename

Cache Filename

This gadget is used to enter the name of the file in which to place a file cache. It will be disabled if the current cache is defined as a memory cache.

Valid entries are all valid filenames with or without paths. The Get-File gadget right next to this string gadget allows you to pick a file using a file requester.

- Intel put the "backward" in "backward compatibility."

1.29 Get Cache Filename

Get Cache Filename

This gadget is used to pick the name of the file to use for file caching. When pressed, a file requester will prompt you for a selection. The result of the selection is placed into the cache filename string gadget, for later editing if necessary.

- If at first you don't succeed, redefine success

1.30 Free After Purge

Free After Purge

Setting this gadget will cause read/write caches to actually release (free) the cache buffer memory once any dirty sector data has been written back to the drive. It has no effect on read-only caches. (Earlier versions of PowerCache did in fact allow read-only caches to behave almost similarly, but this has been dropped 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 occurs. As you can imagine, freeing the cache buffers means the cache will have to prime itself once again.

There is no real rule to use when configuring this gadget. I would probably use it in a situation where a device is to be used for intense read/write operations, followed by long periods of drive inactivity. Experiment for best results.

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 this checkbox gadget. Actually, you probably want 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).

- A system tends to grow in terms of complexity rather than of simplification, until the resulting unreliability is intolerable

1.31 Show Statistics Window

Show Statistics Window

You may optionally enter a hotkey description here. This hotkey works exactly like the Statistics gadget and allows you to show statistics for a cache without having to actually open the main window.

You may leave this gadget blank if you wish (no hotkey will be set up).

- Nothing happens to you that hasn't happened to someone else

1.32 Accepting the Configuration

Accepting the Configuration

If you believe everything is correctly configured for this cache, then click this gadget to actually install the cache. If not, then just click Cancel. If everything goes well, the device should now appear in the right-most listview gadget of the main window (the list of cached devices), in a format looking like this:

Device: A x B x C, <type>, <mode>

A is the number of sets, B is the number of lines, C is the prefetch value, <type> is the cache type (memory or file) and <mode> is the cache mode (read-only or read/write). For example:

DH0: 10 x 32 x 4, MEM, R/O

If something goes wrong during the cache setup procedure, PowerCache issues an error message (meaningful, hopefully) and simply sends you back to the main preferences window without doing anything further.

- The greatest masterpiece of literature is only a dictionary out of order

1.33 Cancelling the Configuration

Cancelling the Configuration

Pressing this gadget will simply cancel any changes you've made to the cache you are editing. If you are working on a new cache, it just means that no cache will be installed.

To accept changes you press the Okay gadget.

- The best way to destroy an enemy is to make him your friend

1.34 Rescan AmigaDOS Devicelist

Rescan AmigaDOS Devicelist

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. This gadget is rarely used, since most devices are mounted when your machine is booted.

- People don't plan to fail, they fail to plan

1.35 Device Information

Device Information

The Info gadget is used to obtain detailed information about the currently selected device. If you are a novice user, this information may appear to be completely mumbo-jumbo, but it'll come in handy when you want to impress your friends with the enormous knowledge you possess about your harddisk :-)

Advanced users will need this device information for finding an appropriate cache size.

You may open any number of Info windows (even several windows for the same device, although you probably don't want to do this), and you may close the main window without closing any of the Info windows. If you exit PowerCache while one or more Info windows are open, PowerCache will close them for you as a part of its cleanup procedure.

- Paper is always strongest at the perforations

1.36 Remove an installed cache

Remove an installed cache

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 removing a read/write cache which contains updated sector data requires that this data first be written back to disk by PowerCache. 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). Also, PowerCache will not respond to gadget and menu selections (or Rexx commands, for that matter).

Read/only caches can be removed instantaneously, since they never contain updated ("dirty") data.

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

- It's better to be a lion for a day, than a sheep all your life

1.37 (De)activate an installed cache

(De)activate an installed cache

This gadget allows you to selectively enable or disable any installed cache. In effect, it does exactly the same as if you pressed the hotkey to disable (or enable) all caches, only it operates on single caches rather than all of them.

Note: If you press the hotkey to enable all caches, then all caches will indeed be enabled - even those which had manually disabled with this gadget.

- Any system that depends on human reliability is unreliable

1.38 Purge an installed cache

Purge an installed cache

Clicking this gadget will flush any updated sector data currently present in the (write)cache buffers out to disk and then free the cache buffer itself. The cache will then 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 with Remove.

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.

- An unbreakable toy is useful for breaking other toys

1.39 Edit an installed cache

Edit an installed cache

PowerCache allows caches to be edited without limitations. To do this, you must first select the cache you wish to edit in the list of cached devices.

Once you click the gadget, The familiar cache preferences window will appear, containing the current values for the cache you intend to edit. 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 accept the changes (or Cancel to do the opposite). For a more detailed description of the gadgets in the Cache Preferences window, please refer to the cache installation documentation.

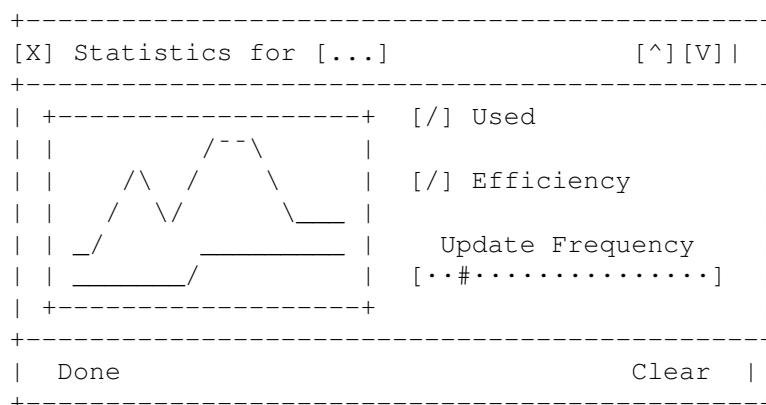
- One way to stop a runaway horse is to bet on him

1.40 Obtaining cache statistics

Obtaining cache statistics

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

Select any active cache, and then click this gadget. A small window with a few gadgets and a graphical area will now 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. For this reason, the Memory Usage graph is simply disabled for file caches.

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 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 (which is not really a problem, considering the dynamic nature of PowerCache). 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 accordingly.

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.

Please select any of the above gadgets, for further details on the functions in the statistics window.

- Help stamp out, remove and abolish redundancy

1.41 Closing the Statistics Window

Closing the Statistics Window

Clicking the CloseWindow gadget will make the statistics window go away safely. You can also use the Okay gadget.

- I can resist anything but temptation

1.42 Title of the Statistics Window

Title of the Statistics Window

The title of all statistics windows contain the name of the device they belong to ('DF0:' for example). In fact, they contain the entire contents of the entry in the listview for that device (caching mode, size, etc.)

- If God had wanted us to go naked we would have been born that way

1.43 Toggling Used %

Toggling Used %

This gadget can be used to turn the Memory-Used graph on and off (to make the graph view more easily readable).

Note: This gadget will be disabled for file caches. Files are created in full when the cache is installed, and there would be very little point in drawing a graph which is always at 100% anyway.

- If I cannot bend Heaven, I shall move Hell

1.44 Toggling Efficiency %

Toggling Efficiency %

This gadget can be used to turn the Efficiency graph on and off (to make the graph view more easily readable).

Note: This gadget will always be enabled file caches, since efficiency is the only graphical indicator available for these cache types (no point in rendering an empty graph).

- If it jams, force it. If it breaks, it needed replacement anyway

1.45 Changing the Update Frequency

Changing the Update Frequency

This 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. Statistics displays also has a tendency to degrade cache performance in general (due to internal structure arbitration), so during

drive intensive operations, either close the statistics window for that drive, or at least set the update frequency to no more than once every half second.

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

- If it looks easy, it's tough. If it looks tough, it's damn well impossible.

1.46 Disposing of the Statistics Window

Disposing of the Statistics Window

Click this gadget to make the statistics window close safely. You may use the CloseWindow gadget, too.

Note: PowerCache automatically closes all statistics windows when you quit the program.

- If the facts do not conform to your theory, they must be disposed of

1.47 Clearing the Graphics Area

Clearing the Graphics Area

Clicking this gadget simply clears the graphics area. 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 fool around with something long enough, it will eventually break

1.48 Show Window Hotkey

Show Window Hotkey

This string gadget is used to specify the hotkey which makes PowerCache open 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.

- It is difficult to soar with eagles when you work with turkeys
-

1.49 Enable Caches Hotkey

Enable Caches Hotkey

This string gadget is used to specify the hotkey to enable all installed caches.

Note that pressing this hotkey will enable ALL installed caches - even those which have been manually disabled using the Activate gadget.

- Never, ever use repetitive redundancies

1.50 Disable Caches Hotkey

Disable Caches Hotkey

This string gadget is used to specify the hotkey to disable all currently installed caches.

Note that pressing this hotkey will disable ALL installed caches - even those which have been manually enabled using the Activate gadget.

- A celebrity is a person who is known for his well-knownness

1.51 Popup Window

Popup Window

This checkbox gadget 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.

- A man wrapped up in himself makes a very small package

1.52 Beep

Beep

This checkbox gadget tells PowerCache if you want it to flash all screens and sound a beep every time it is either disabled or enabled. You probably want to leave this on all the time.

The only reason for PowerCache to have such an option in the first place is to accommodate users who have their DisplayBeep function redirected to show a picture or play an 8SVX sample (you can do this with the Sound

preferences program). Having pictures shown and samples played each time the caches are disabled and enabled would probably be too much to bear :)

- Air is water with holes in it

1.53 Save Settings

Save Settings

Pressing this button will make PowerCache save the current cache configuration permanently (in ENVARC:). This means that next time you start the program (even after a reboot), the cache configuration will be restored automatically when you start the program.

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)

- All I ask is a chance to prove that money can't make me happy

1.54 Use Settings

Use Settings

To use the currently configured caches, either click this or the CloseWindow button (or select Hide from PowerCache's menu). Either will make PowerCache hide its window (but remain dormant, naturally).

- All things are possible, except skiing thru a revolving door

1.55 Cancel Changes

Cancel Changes

Pressing this gadget will simply cancel all changes which have been made to the current cache configuration since PowerCache opened its main window.

Note: This includes configuration changes made to any of the installed caches during this time.

- Any small object that is accidentally dropped
will hide under a larger object.

1.56 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. 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 when you start it again, it will load 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.

- Every program has (at least) two purposes: the one for which it was written and another for which it wasn't

1.57 The Menu System

Using the Menus

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.

- Captain Hook died of jock itch

1.58 Stopping PowerCache

How to Quit 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 Workbench 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 choose 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 allocated during the startup procedure

Note that clicking the CloseWindow gadget in PowerCache's main preferences window doesn't make PowerCache quit. The CloseWindow 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.

- By doing just a little every day, you can gradually let the task completely overwhelm you.

1.59 Troubleshooting

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?
	Running Workbench 2.04+ ?
PowerCache is not localized	locale.library present in LIBS:?
	PowerCache catalog installed in LOCALE:?
	Correct language selected with Locale prg?
Note: If you run PowerCache before IPrefs,	

PowerCache will use its built-in (english) strings, not those selected with the Locale prefs program.

Note: Changing language using the Locale prefs program while PowerCache is running will not make PowerCache switch languages too. You must quit and restart PowerCache to do this.

System guru on the first Driver contains a strange jump table, or you access to a cached drive are using an unsupported driver.

Select "Info" on the device and mail me the result.

Note: Alf and Oktagon controllers are NOT supported as of this writing.

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.

No online help PowerCache: assigned?
 (and if not, do you have the docs installed
 in a 'Documentation' directory in the same
 directory as PowerCache?)
 Help files (docs) correctly installed?
 AmigaGuide.library installed in LIBS:?

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
 Cached unit freezes on window handling and 1 bit per read/write cache.
 first access Read-only caches are "free" since they require
 no extra signal bits to be allocated.
 Fix: Remove one of the installed R/W caches,~or
 close one or more statistics or info-windows.

PowerCache disappears after This is intentional. You have to register to
 about half an hour fix this! :-)

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 :-)

- A large number of installed systems work by fiat. That is,
 they work by being declared to work.

1.60 PowerCache Error Messages

Common Error Messages

PowerCache is a very paranoid program. It confirms every allocated resource prior to using it, and is capable of giving off a huge amount of 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 and ask. 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 <some.library> [someversion+] 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 regtools.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

- Your 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, then by all means tell me :-)

- All that glitters has a high refractive index

1.61 A few Important Notes

Some 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 cached unit. If it is a read/write cache, make sure it is properly purged before you do anything like that. In fact, you should probably disable all caches, or simply quit PowerCache while you optimize the drive. Just to be on the safe side.

The reset handler

The internal reset handler (present in PowerCache as of version 37.115) will try to intercept and postpone resets (Ctrl-Amiga-Amiga), hopefully

buying PowerCache enough time to flush out any remaining, dirty sector data. The delay can be no longer than ten seconds, after which a hardware reset is enforced by the system.

Not all drives, nor machines, can handle this. The Amiga autodocs states the following to this effect:

"Few of the Amiga keyboard models generate the communication codes used to implement this reset processing. Specifically, only the Euro a1000 (rare), and the B2000 keyboard generate them."

Apart from the lack of hardware support, it appears the reset feature does not work with certain drives either. The problem appears to be that as soon as one of these drives detect a reset, they refuse to write any further data to the disk in the drive (presumably in an attempt to avoid corruption), and so effectively lock out PowerCache from writing to the disk as well.

This appears to be the case with the Amiga floppy drives, so if you reset the machine while a R/W floppycache still contains unwritten data, most likely you'll end up with an incomplete or corrupted disk.

Furtunately the majority of harddisks remain in operation until the actual reset occurs, and will continue to obey commands from PowerCache.

Note: The reset handler does NOT handle gurus or system crashes. It only handles the well known three finger keyboard salute :-)

- Angular momentum makes the world go round

1.62 Known Bugs

Known Bugs

PowerCache is reasonably free from bugs. The program has been tested in many different environments and found to work totally reliably in all cases. Further, there are no known Enforcer or Mungwall hits.

However, there are still a few known bugs/quirks that you should be aware of:

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.

Ever since PowerCache was first released, zipping (zooming) of statistics windows has been very unreliable, and has often caused spurious crashes. I'm sorry to say that I have been unable to nail this one, and so for the time being I have simply removed the Zip gadget from the statistics windows. I hope to have this bug taken care of in time for the next release. Sorry, everyone.

PowerCache does not work with ALF and Oktagon controllers. I'm sorry I haven't been able to fix this, but I do not have access to such a

controller. Anyone wanna lend me one? :-)

I have been getting a few reports from people with CD-ROM drives, and it appears PowerCache has problems with certain drive types. I am working on a way to fix this, and will hopefully have it ready in time for the next release.

I'm sure there are a few bugs still in there. You should probably test PowerCache to the fullest on a floppy disk 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.

- There are no bugs, only unrecognized features

1.63 Benchmarks

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 and Copyback on, running Kickstart 3.1 (v40.9, shadowed in 32 bit memory) on the second (20 meg) partition of a Quantum LPS 105 meg drive. The controller used was the Advantage AdSCSI cache controller.

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	232 files/s
File Open	268 files/s
Directory Scan	813 files/s
File Delete	499 files/s

Seek/Read: 45 seeks/s

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

Create file:	221312 bytes/s
Write to file:	240384 bytes/s
Read from file:	466624 bytes/s

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

Create file: 179200 bytes/s
 Write to file: 191488 bytes/s
 Read from file: 604672 bytes/s

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

Create file: 397312 bytes/s
 Write to file: 465779 bytes/s
 Read from file: 692224 bytes/s

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

Create file: 699899 bytes/s
 Write to file: 761677 bytes/s
 Read from file: 799219 bytes/s

2. CACHED HARDDISK PERFORMANCE

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

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

Testing directory manipulation speed

File Create:	294 files/s	301 files/s	161 files/s
File Open:	510 files/s	511 files/s	561 files/s
Directory Scan:	1510 files/s	1531 files/s	1727 files/s
File Delete:	909 files/s	975 files/s	909 files/s
Seek/Read:	1254 seeks/s	1278 seeks/s	1259 seeks/s

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

Create file:	770816 bytes/s	800515 bytes/s	167232 bytes/s
Write to file:	864192 bytes/s	898388 bytes/s	203200 bytes/s
Read from file:	922624 bytes/s	963782 bytes/s	1074560 bytes/s

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

Create file:	3294208 bytes/s	3313281 bytes/s	173568 bytes/s
Write to file:	3825152 bytes/s	3844124 bytes/s	185856 bytes/s
Read from file:	3912192 bytes/s	3933709 bytes/s	4282880 bytes/s

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

Create file:	5836800 bytes/s	5882529 bytes/s	350503 bytes/s
Write to file:	7057408 bytes/s	7098511 bytes/s	385551 bytes/s
Read from file:	7069696 bytes/s	7102521 bytes/s	6840320 bytes/s

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

Create file:	7012352 bytes/s	7045585 bytes/s	491520 bytes/s
Write to file:	8585216 bytes/s	8618109 bytes/s	588353 bytes/s
Read from file:	8486912 bytes/s	8552192 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 (as shown later) reveals similar results.

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. (Also notice that PowerCache is in fact so blindingly fast for large i/o operations that it actually beats HyperCache's read-only implementation by about a megabyte per second 8^{\wedge})
- ~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.

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.

Similary, 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. Besides, SAS' code generator is one of the best around. It generates code which is very close to what a human programmer could do by hand.

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

3. READ-ONLY CACHE PERFORMANCE

(Cache Configuration:~Read-only, 6 sets, 64 lines, prefetch 8)

	PowerCache, R/W	PowerCache, R/O	HyperCache
Testing directory manipulation speed			
File Create:	294 files/s	192 files/s	161 files/s
File Open:	510 files/s	564 files/s	561 files/s
Directory Scan:	1510 files/s	1993 files/s	1727 files/s
File Delete:	909 files/s	999 files/s	909 files/s
Seek/Read:	1254 seeks/s	1539 seeks/s	1259 seeks/s

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

Create file:	770816 bytes/s	202112 bytes/s	167232 bytes/s
Write to file:	864192 bytes/s	239936 bytes/s	203200 bytes/s
Read from file:	922624 bytes/s	1236800 bytes/s	1074560 bytes/s

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

Create file:	3294208 bytes/s	181760 bytes/s	173568 bytes/s
Write to file:	3825152 bytes/s	191488 bytes/s	185856 bytes/s
Read from file:	3912192 bytes/s	4805120 bytes/s	4282880 bytes/s

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

Create file:	5836800 bytes/s	394354 bytes/s	350503 bytes/s
Write to file:	7057408 bytes/s	465779 bytes/s	385551 bytes/s
Read from file:	7069696 bytes/s	7729152 bytes/s	6840320 bytes/s

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

Create file:	7012352 bytes/s	671344 bytes/s	491520 bytes/s
Write to file:	8585216 bytes/s	726423 bytes/s	588353 bytes/s
Read from file:	8486912 bytes/s	8814592 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. If you take a look at the results, you will notice that PowerCache in read-only mode exceeds HyperCache in speed for ALL i/o operations, small or large. Hoo-ray! :-)

All in all, PowerCache compares extremely favorable with HyperCache. 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 attractive alternative.

- Never replicate a successful experiment
- There are lies, damned lies, and statistics
- Did you know the Cray 2 is so fast it can do an infinite loop in two and a half seconds?

1.64 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.

- If you can't learn to do it well, learn to enjoy doing it badly

1.65 The Future

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:

- Prefs/daemon implementation

One of the most important things I want to do with PowerCache, is to turn it into two separate programs: One prefs program (for your SYS:Prefs/drawer), and one cache program. Right now, you are forced to have ALL of PowerCache's code (including GUI code) lying in memory, even though you're only using the cache functions.

Some early estimations I've done indicate that the in-memory resident part would take up only some 20-30k, or in other words be more than cut in half, compared to the current ~80k executable.

Unfortunately, such a split-up almost certainly means a total rewrite of PowerCache, so I'm a bit reluctant to get on with it :)

(If/when I do start, it is quite possible that I will do a special MUI edition of the Prefs program, for all you devoted MUI freaks. We'll see...)

- Improved ARexx Interface

Even though PowerCache as 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 chose 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: (mfm.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.

- Never do today what you can put off until tomorrow
- Adding manpower to a late software project makes it later
- The most useful program will be continually improved until it is useless

1.66 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.

- Any given program, when running, is obsolete

1.67 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.

E-mail

I can be reached at any of the following sites:

FidoNet: 2:230/166.28
AmigaNet: 39:140/101.28
Internet: mberg@scala.ping.dk

Feel free to send comments, suggestions and - naturally, bug reports to any of these addresses.

- Any given program will expand to fill all available memory
-

1.68 Credits

Credits

PowerCache was developed using the following software:

Program compiler: SAS/C C 6.51 - © SAS Institute, 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 Internet and Fidonet support
Stig Poulsen - For extensive Fidonet support and betatesting

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 original german (deutsch) translation
Steffen Salzmann- For an improved german (deutsch) translation :-)
Mario Gonzales - For the norwegian (norsk) translation
Paul van der Heu- For the netherlands (dutch) translation
Magnus Holmgren - For the svenska (swedish) translation
Roberto Tosco - For the italiano (italian) translation
Gábor Nagy - For the magyar (hungarian) translation

Also thanks to all of my betatesters, for willingly allowing early beta versions to raid their precious harddisks. Thanks guys! :)

- Integer math - it's all so pointless

1.69 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 obey, 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. 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.

- Mistakes are often the stepping stones to utter failure

1.70 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 [postal] 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 (again, only issued in Danish Kroner (DKK)). 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!

Note: I have been getting requests from persons who ask for BOTH programs (PowerData and PowerCache), for \$20. Their reasoning is, that if they buy PowerData for \$10, then they can get PowerCache for half price, or \$10, a total of \$20. THIS IS NOT THE INTENDED USE OF THIS OFFER! I am offering this cut-rate deal as a special favour to persons who are ALREADY registered users of PowerData, and have been so for some time. Any future requests like this will be turned down, plain and simple.

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 87-370012, UISDNB, ISDNC

I have also arranged for PowerCache to be uploaded to AmiNet, so Internet users can FTP it.

- Spend sufficient time confirming the need and the need will disappear

1.71 Keyboard Navigation

Using PowerCache Gadget Shortcuts

As you have probably already discovered, all 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.

Note: All button-kind gadgets can be "cancelled", using the shift key. Example: Press and hold down the 'C' key in PowerCache's main window. The Cancel gadget will now be highlighted. Now press Shift. The Cancel gadget will be returned to normal, and the selection thus be cancelled.

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.

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? :-)

- Enough research will tend to support your theory

1.72 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 kind of performance. As always, but perhaps 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 activating help requires approximately 100k, but activating help immediately gobbles up an additional 220k (a whooping total of 320k!). Those are the results I got on my 9 meg machine, anyway. It could be that AmigaGuide is allocating all this memory because it can see that I have enough available. On a machine with less memory, it might not allocate that much. Also, neither amigaguide.library or any of the resources it needed were loaded prior to my pressing the HELP key, so the loading of these obviously factor in as well. I advise you to experiment and find out exactly what costs are involved, memory-wise. 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 creature 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.

Since version 37.104, PowerCache is being distributed in 68000, 68020, 68030 and 68040 optimized versions. Naturally, you should install and use the version which matches your CPU. The most common machines and their CPU types are as follows:

A500/A600 - 68000
A500 + A530 - 68030
A1000 - 68000
A2000 - 68000
A2500 (A2630) - 68030

A3000 - 68030
A1200 - 68020
A4000 - 68040
A4000/030 - 68030

Further, there are a wide range of accellerator boards available for all Amiga models (A1200 in particular). Listing them all here is simply not possible. Try the CPU command (in your C: directory), and if you don't have this command, look for the program 'SysInfo', by Nic Wilson. SysInfo is publically available on BBS systems and via Fred Fish.

- You never know what is enough until you know what is more than enough