PowerData

COLLABORATORS

	TITLE : PowerData	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		January 5, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

PowerData

1.1 PowerData Documentation (26.7.1993)

Documentation for

PowerData
Version 38.200

A transparent, application independent
powerpacker file loader and saver for
Workbench 2.04 and above

Written by Michael Berg

1.2 Shareware Notice

A note on Shareware

PowerData 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^1 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^2, 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 will DO anything about it, if they do so. As a shareware programmer, you have very few means of physically enforcing the laws you are asking people to follow, so therefore you must rely 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.

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 do not release fully fledged versions their programs any more. In stead they release release "crippled" programs that have some features disabled, or only work on a limited set of data. They are usually called evaluation 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.

PowerData 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 almost to its full potential, since no features have been 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 started

To get the REAL, TRUE version of the program, you must send the amount of at least 10 US $, or the equivalent amount in another currency, to me. You are welcome to send more money (which will probably mean faster development), but US $10 is the minimum required fee.

Even though it entails a slight risk, I encurage you to simply send actual currency in plain old envelopes, along with a letter stating that you want to register. And please, no checks. I have to argue with the bank clerk every time I want to cash in a foreign check, because they are nervous it'll bounce (which is a tricky situation when the check is foreign). If you absolutely must use checks, add at least another 6 US $ to cover the exchange overhead, or (better still), make it out in Danish currecy (1 US $ ~= 6,50 DKK).

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

Anyway, for those 10 US$, I will register you as a user, and immediately mail you a branded 3,5" disk with the registered PowerData program. 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.

As a registered user, you also have the option to buy additional, registered copies of new versions of PowerCache at a rate of only US $5 a piece. Those $5 are ment only to cover materials and shipping, so if you have a modem, you can download a free update from me directly. Don't mail envelopes or stamps. It is cheaper for me to buy those locally, so mail cash in stead.

You may pre-order one version of PowerData, and mail US $15 in one go. This saves you the trouble of having to write me a second time, and it gets you the new version that much faster.

Registered users of another program of mine, PowerCache, may register PowerData for half the price - 5 US$. It really does pay to register shareware, people!

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

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

Notes

^{1}) Actually, there are many more.. PostcardWare, BeerWare, you name it..!
^{2}) Respecting the wishes (or restraints) of the author, of course.

1.3 Background

A bit of background info...

The Amiga is a very powerful machine, and is supported by a variety of sophisticated applications, utilities and games. One common property of most of this software is the fact that they use their own format for saving data in files.

The majority of software companies have agreed on a number of standard file formats, one of which is the IFF format (Integrated File Format). The IFF format can contain a variety of data types, ranging from pictures and sound to locale files for Workbench 2.1.

This is all very well, but there is one problem with this: One cannot compress these datafiles without rendering them useless to the application

that needs them (unless of course the application is capable of reading compressed data, which is probably unlikely).

One way of overcoming this would be to add compression and decompression code to all applications, so that they could automatically handle compressed datafiles. However, as you may have guessed, this is virtually impossible to accomplish, because no-one has the time, the money or even the skill to do it.

What is really needed is some way of enabling applications to read compressed datafiles without changing one single line of code in any of them. And this is exactly what PowerData does.

1.4 Introduction

A brief introduction to PowerData...

Briefly, this program enables all applications to read files compressed with Powerpacker, and to compress the files they create themselves. PowerData is implemented as a patch so it is completely transparent to both applications and to the operating system itself.

The implications of transparent compression and decompression are far reaching. For example, imagine being able to compress all your Workbench icons, and still have Workbench read and display them. Or how about saving some text from your text editor, and have it compressed on-the-fly, without having to turn to Powerpacker? Imagine your paint program being able to save its picture files in a compressed format, and being able to read compressed files as well.

You can do all this, and much more, using PowerData.

Those of you who are already familiar with the product "PowerPacker Patcher", will hopefully have noticed that PowerData is written by the author of PPPatcher.

1.5 Overview of features

Feature overview

Below is a run-down of some of the most important features of PowerData. All of these features will be described in detail later.

- Completely transparent to all applications
- Enables applications to read powerpacked files
- Compresses (powerpacks) files created by all applications
- Completely configurable
- Friendly 2.0-style GUI configuration window
- Hotkey controllable
- Installs as a standard AmigaDOS 2.04 commodity
- Uses powerpacker.library for fast, efficient decrunching
- Uses reqtools.library for requester handling

- Partially localized for use with Workbench 2.1 and later
- Works with all CPU's, including 68040 (68020 version included)
- Comes with plenty of documentation in AmigaGuide format
- Installation scripts for both IconX and The Installer
- Includes ARexx interface

As previously mentioned, I will elaborate on all of these points later on in this document.

1.6 System requirements

What is required to run PowerData?

As opposed to the previous version of this program, Powerpacker Patcher, this program will only work on machines running Kickstart 2.04 and above. There is no way to make it work on 1.2/1.3 machines.

I did a lot of thinking about this. A program that would work on both pre- and post-2.04 operating systems would be MUCH larger, and more prone to errors. Also, some features are only possible under 2.04 anyway.

Lastly, I find that continuing to write software for an OS that is quickly becoming obsolete, is unnecessary, and quite simply plain wrong. By doing so, you prolong the life of an OS that should be laid to rest on ALL Amigas. I will not do this.

Writing good software that works ONLY on 2.04 and above will encourage Amiga owners all over the world to upgrade their hardware, which is both in their own interest - and in the interest of software developers all over the world.

Apart from AmigaDOS 2.04, you will need powerpacker.library and reqtools.library, both by Nico François. You should find both included in this distribution.

1.7 Installing PowerData

How to install PowerData...

It is very easy to install PowerData. Basically, you need to make sure that you have both powerpacker.library and reqtools.library in LIBS:, and then just copy the PowerData program itself to whatever location you keep your commodities
SYS:Tools/Commodities, and still others keep commodities somewhere else entirely.

Note that if you already have powerpacker.library in LIBS:, you may still have to install the version supplied here. You need version 38 or later of reqtools.library, and version 35 or later of powerpacker.library. To find out which versions you have, type

```
1> Version reqtools.library
```

```
reqtools.library 38.1042
```

The resulting string, as illustrated above, should read 38.something. If it reads anything lower than that, you must copy the reqtools.library supplied here over the version you already have in LIBS:.

The same thing goes for powerpacker.library. The version string must show 35.something (the library provided in this distribution is version 35.344).

If this procedure seems too complicated, an IconX script has been provided that will do it all for you. All you need to do is click it, and everything should end up in the right place. It will make intelligent decisions on what libraries need to be installed, and only copy those which are newer than the ones you already have.

For those of you who have Workbench 2.1 or later, I have also included an Installer script, which is a good deal more pleasing to watch :^) Just double click the script to activate the Installer.

1.8 Using PowerData

```
                Using PowerData
```

PowerData is completely transparent in use. After running it, you should't really notice much of a difference from your old (non-PowerData) configuration, other than the fact that files take up less space on your disk. Obviously the compression engine requires some additional CPU time to be able to compress and decompress data, so you will notice a slow-down when loading powerpacked files, and when saving files. The powerpacker routines are quite fast compared to how efficient they are, so the extra delay should not be too intolerable. You can lower crunch efficiency to improve crunch speed if you find it to be too slow.

You can always have PowerData pop up its window by pressing the "show window" hotkey. This will work even if the Workbench screen has been closed or if another program has installed a new, default public screen. PowerData always appears on the default public screen.

For further info on how to configure PowerData, and how to fine-tune it to your particular needs, please refer to the section on
Configuring
PowerData.

1.9 Configuring PowerData

```
How to configure PowerData
```

It should be easy to configure the program. If you have not yet run PowerData, I suggest you do so now. It makes the following somewhat more understandable.

(If the window does not show up when you run the it, press Ctrl-Alt-S.

This should do the trick)

When you examine the configuration window, the first thing you'll see is the Temporary Path gadget. This is the path that PowerData will use for its background file decrunching, and it should be set to RAM:, T: (if T: is in RAM:) or at least to somewhere on your harddisk. PowerData will be doing a lot of reading and writing in this path, so things will run much smoother if the temporary path is set to a FAST device -- And RAM: is just about as fast as you can get.

By default, it is set to T: and if you leave it like this, you will of course have to make sure T: is properly assigned.

To select another temporary path, you can either enter it manually by selecting the string gadget and keying it in, or you can click the select path gadget right next to the string gadget. This will give you a file requester, making it easier to quickly traverse various paths. Select a directory and click "Ok". The filerequester will disappear and the path you selected will appear in the string gadget.

Note that any changes made to the temporary path takes place immediately. That is, if an application attempts to open a powerpacked file while the control window is showing, PowerData will still work, and the new temporary path you just entered will be used immediately. This is also true for any other changes you make while the window is on (including hotkeys)

Moving on, the next thing you will need to configure is the hotkeys. These are used to control certain features of PowerData, like popping up the configuration window and enabling or disabling PowerData itself.

You are free to reconfigure these keys if they do not suit your needs, or perhaps clash with the hotkeys of other commodities you happen to be using. When changing the hotkeys, you must respect the "commodities way" of describing keys. For example, to change the show hotkey from its default of Ctrl Alt S to the left command key, plus any "Alt" key, plus the right shift key, plus ESC, enter lcommand alt rshift esc. For a complete description of how commodities expects keys to be described, please refer to your AmigaDOS documentation.

Note that if you enter something which Commodities cannot understand, you will get a warning, followed by PowerData resetting the hotkey to what it was before you attempted to change it.

To the right of the hotkey control area of the window, you should find some crunch controls. These control the way PowerData crunches datafiles when applications create them. By default, they are set to "Best" efficiency, with a "Large" buffer. This will crunch files most efficiently, as fast as it can be done.

Unfortunately, using a large buffer requires plenty of memory. If you do not have very much to spare, you can click it a couple of times, selecting either "Medium" or "Small". Note that as you select a smaller buffer, crunching will be slower.

You may also want to use "Fast" or "Good" (or another value) for the crunch efficiency. What you have to notice here is that specifying "Fast" will

crunch files very fast, but not very efficiently, and that specifying "Best" will crunch files very efficiently, but more slowly.

If your system resources are limited, you'll have to find a compromise between fast and best crunching, and large and small buffers.

Note that while what you specify at configuration time may seem reasonable, PowerData may not be able to comply in low-memory situations. In these cases, PowerData will try to find another, less memory requiring combination, giving priority to efficiency over buffer size. This means that if you run low on memory, PowerData will lower the buffer size before worsening crunch efficiency.

If all fails (i.e. there simply is NO memory), then PowerData will not crunch the file at all, and simply transfer the file from its temporary path to the target file. This means that you will not loose data, even in low memory situations.

The Effect gadget controls what happens when the decruncher (or the cruncher, for that matter) is "active". It is set to "Pointer" by default, causing the mouse pointer to rapidly change color when a file is being decrunched or crunched. This serves as an indicator telling you that your machine is still alive and well - only it is working on decompressing or compressing some file.

Directly below the crunch parameters section, you'll find five flags, or "switches" if you prefer. These are controls that affect the way PowerData behaves.

If you set the Popup flag, then PowerData will pop up the control window every time you start the program. If you use it from your startup sequence, or from the WBStartup

Setting the Beep flag will cause PowerData to flash all screens whenever it is enabled or disabled (using the proper hotkey). PowerData only beeps when the action you request is actually carried out. For example, pressing Ctrl Alt D (disable) will blink the screen because PowerData has now been disabled, but pressing it again will not do anything (because PowerData already IS disabled). Pressing Ctrl Alt E (enable) will blink the screen because PowerData has now been activated, but pressing it again will have no effect.

PowerData does not blink the screen when you request the window to appear, by pressing Ctrl Alt S. The window is in itself an effect :-)

The Crunch switch is used to either enable or disable automatic crunching. This is useful for people who wish PowerData to only decrunch, and never attempt to crunch new (or updated) files. In this case, PowerData will behave exactly like its predecessor, Powerpacker Patcher. The default value of this flag is "on", meaning that PowerData will indeed attempt to crunch anything that matches the filters (see below).

The Decrunch switch has a similar effect. With it, you can either enable or disable transparent decrunching. You would typically disable decrunching if you are performing a backup of your harddisk (in which case PowerData would otherwise decrunch every single Powerpacker datafile that the backup program would back up). The default value of this switch is

"on", meaning that PowerData will indeed decrunch any file matching the filename filters (next).

The Busyptr switch controls whether or not the mouse pointer should turn into a "busy clock" image when when PowerData is working. If you enable it then the mouse pointer will change into a clock image as a file is being crunched or decrunched. Further, the window of the program you are working with will be "locked" from input, so that no gadgets or menus can be selected. Busy pointer is on by default, and you probably want to leave it that way.

Lastly, we come to the "filters". This is a bit complicated to explain, but simply put, these filters control which files PowerData will consider, and which it will not do anything to/about.

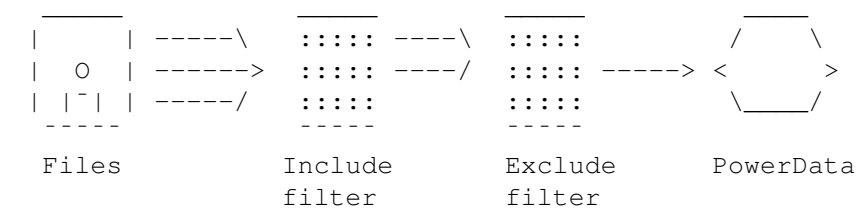
These filters offer a wide range of possibilities. For example, if you want PowerData to automatically decrunch all files with the postfix of ".pp", but NO OTHER files, set the include filter to "#?.pp". This means that no matter what, PowerData will only decrunch files that end in ".pp", and only attempt to crunch files that are opened with a name that matches this pattern.

The exclude filter can be used to fine-tune the include filter. For example, as it does not make any sense to try to decrunch files ending in .lha or .lzh (these are already packed with a different archiver), you should set the exclude filter to "#?.(lha|lzh)". This is done by default, and the file type dms is also excluded.

If all you want is to use PowerData to enable Workbench to read and write icon files, set the include filter to "#?.info" and the exclude filter to nothing (i.e. leave it blank).

Now, to make things interesting, you might have a few files that all begin with "packed_" and ending in ".info", and you do not want PowerData to consider these (they may already be packed using something else). All you need to do then is to set the exclude filter to "packed_#?" and PowerData will do what it is supposed to.

Here's my best shot at illustrating the filters graphically:



If you are still a bit confused about how to use the exclude and include filters, just leave both at their default values.

Note that all filters are case insensitive

By default pattens match filenames, NOT devices or paths. This means that if you enter DF0:#?.pp as a filter, it WILL NOT WORK. You can ONLY match file names.

As of version 38.200, patterns can now be made to optionally match both the filename and the path and volume on which it exists. For example, if a program attempts to open "datadir/file0", then PowerData can be made to expand the name to "Work:GlobalFiles/datadir/file0" before applying any filters.

This behaviour is controlled by the Match Path checkbox gadget located beneath the filter gadgets. If you set this gadget ON, then filenames will be expanded before they are matched by patterns. Otherwise, any path or devicename part will be stripped from the filename before matching, so that ONLY the name is matched. Note that expanding filenames takes extra time (which is why the filenames-only matching method was retained in the first place). Use it with some thought.

Note that you can only match volume names, not device names. You can match Work:, Empty: or DH0{80MB}:, but not DH0:, FF1: or DF2:. Use the volume label, not the device name. Similary, it is impossible to match assigned paths, so you cannot match SYS:, LIBS: or ENVARC: (simply match the assigned paths in stead).

Note that if you are able to precisely specify what files PowerData should operate on, then DO so. It will enhance the performance of DOS operations on those files not matching the filters.

All filters must comply to the syntax of standard AmigaDOS patterns (i.e. for selecting files).

Normally, filters apply for both crunching and decrunching. However, the Always Decrunch checkbox gadget can be used to instruct PowerData to only use filters when creating or modifying files, and to always decrunch files. This means that any crunched file will be decrunched by PowerData, even if it does not match the filters. This feature has been a popular request since v38.105, and now it’s finally here :-)

The Save gadget will save the current settings permanently (in the ENVARC: directory, to be more precise), and so PowerData will use these settings every time you start it, or every time you select "Last Saved" from PowerData’s menu.

Clicking the Use gadget will remove the window, but allow PowerData to continue operating, using the current settings. This is the normal way to get rid of the PowerData window.

The Cancel gadget will restore the settings that were in effect before the configuration window popped up.

Note that when you quit PowerData, it will save its current settings in ENV: (which is in RAM:, usually), so that it will use these settings if you start it again at some point. Of course, if you reboot the machine, everything in ENV: will probably go away, and PowerData will look in ENVARC: to find the last settings that were saved.

The gist of this is that it is not necessary to Save the settings if you only have to terminate PowerData for a few moments. Merely exiting the program is enough.

There are also a few menus attached to the PowerData control window. These correspond exactly to what you would expect a preferences program to have. Here's a brief description of the menus:

Project/Open... will allow you to load some preset defaults, which you have saved previously. You will be presented with a file requester, making it very easy to locate the file you are looking for. By default the requester will start off in the directiry "SYS:Prefs/Presets", which is the standard directory for keeping non-standard settings in.

Project/Save As... will allow you to save the current settings in a file, for later retrieval using the "Open..." function. Again, the file is selected using the file requester. It is suggested that you save the presets in "SYS:Prefs/Presets".

Project/About... will tell you a bit about PowerData.

Project/Quit will TERMINATE PowerData. It will not merely hide the window, but will kill PowerData entirely.

Edit/Reset to Defaults will completely replace the current settings by those settings that PowerData consider to be defaults. Select this menu item if you have somehow messed up the settings so that nothing works anymore :^)

Edit/Last Saved will load the settings that were last saved (using the Save gadget), or ultimately to the default settings, if no saved settings can be found.

Edit/Restore will undo any changes you have made to the settings since the window was opened.

Options/Create Icons? - Selecting this one will create an icon for every settings file you save (using Projects/Save As...). This way, whenever you want to start PowerData with a speciffic set of preferences settings, you can launch it by clicking on the "Prefs" workbench drawer, and on the "Presets" drawer inside it. The icon(s) you'll find inside the Presets drawer are the one(s) created by PowerData and other preferences programs.

Double click on a PowerData icon to start up PowerData with those particular preferences. For further info, please refer to your AmigaDOS 2.04 documentation.

1.10 Alternative ways of configuring PowerData

Alternative ways of configuring PowerData

First of all, a word on activating PowerData automatically from Workbench. If you put PowerData into the WBStartup drawer, you MUST add the string DONOTWAIT to the tooltypes in PowerData's icon. You do this by clicking once on the icon and selecting "Info" from the Workbench menu. Please refer to your AmigaDOS documentation for more information about the "Info" program and how to change tooltypes.

Besides using the gadgets in the window, you can also specify paramenters

directly on the command line, or as tooltypes parameters, using PowerData's icon (see above). Here are the parameters you may specify:

CX_PRIORITY=<priority>

This will set the priority of the PowerData commodity. This has nothing to do with task-priority, only it enables Commodities.library to give priority to one commodity over another, in case they both use the same hotkey. This value defaults to 0.

CX_POPKEY=<hotkey>

Here you specify the hotkey you want to use for making PowerData's window appear on command. It defaults to whatever settings were last saved by PowerData, or ultimately to "Ctrl Alt S".

ENABLEKEY=<hotkey>

Here you specify the hotkey you want to use for enabling PowerData. It defaults to whatever settings were last saved by PowerData, or ultimately to "Ctrl Alt E".

DISABLEKEY=<hotkey>

Here you specify the hotkey you want to use for disabling PowerData. It defaults to whatever settings were last saved by PowerData, or ultimately to "Ctrl Alt D".

INCLUDE=<pattern>

Here you can specify the include filter you wish PowerData to use.

EXCLUDE=<pattern>

Here you can specify the exclude filter you wish PowerData to use.

TEMPPATH=<path>

Here, you may specify the temporary path you wish PowerData to use.

CX_POPUP=[YES | NO]

Set this to YES if you want PowerData to show its window when it starts. Otherwise, set it to NO.

BEEPING=[YES | NO]

Set this to YES if you want PowerData to beep when it is enabled or disabled. Otherwise, set it to NO.

When entering tooltype values (and giving CLI parameters), you must use a somewhat strict syntax. Below is a couple of examples on how a valid tooltype entry looks:

```
INCLUDE=#?.(pp|pdat)  <- Recommended
INCLUDE=  #?    <- Spaces AFTER the '=' are allowed
```

That is, you may enter as many blanks after the '=' as you wish, but you may not enter blanks between the INCLUDE keyword and the '=', like this:

```
INCLUDE =#?   <- Spaces BEFORE the '=' are not allowed
```

If you want to specify the same thing as CLI parameters, remember to quote those parameters that contain spaces. For example, the following ways of starting PowerData are all valid:

```
1> PowerData INCLUDE=#?
1> PowerData "INCLUDE= #?"
1> PowerData "INCLUDE= #?" CX_POPUP=YES
```

While the following are examples of how you should NOT do:

```
1> PowerData INCLUDE = #?
1> PowerData "CX_POPUP"=YES
1> PowerData INCLUDE=#? CX_POPUP = NO
```

Now, don't get discuraged and confused about these tooltype- and CLI parameters. You are not supposed to use them (CX_PRIORITY being the exception). I have designed a very nice window that does away with both tooltypes and CLI parameters. They are provided for compatibility and completeness only.

1.11 Removing PowerData

Making PowerData go away...

As of yet, there is only one way to actually remove PowerData, and this is by making its window appear and then selecting Quit from the Project menu. I have not implemented a hotkey function to do this, because it would be too easy to unintentionally quit PowerData.

When you get used to it, it becomes very easy to exit PowerData. All you really have to do is press the hotkey to provoke PowerData to show itself, and then pres right-amiga Q, the menu shortcut for the "Quit" function.

If you use the Exchange program, you can always remove PowerData by selecting it in the exchange's list of commodities, and clicking "Remove". You can also have the PowerData window appear and disappear, by selecting "Show interface" and "Hide interface" respectively.

1.12 Sample usage

Sample usage

PowerData offers an unparalleled degree of freedom of file compression. With PowerData active, you create and load powerpacked datafiles, even when you are not aware of it.

For example, during the development of PowerData, I accidently left it on

when I did a couple of source code changes with my editor. I saved the modified C files, and exited the editor to go to the shell. I then issued a "make" command (recompile every changed file, and re-link the objects), and everything went okay.

Then I suddenly remembered that PowerData was actually running in the background, so I popped up its window, and terminated it. Looking in my source code directory, I found that about half of the source code files had automatically been compressed. Thinking about it, I realised that it had happened while my editor was saving these files.

Realising this, it became obvious that the object files created by my compiler would also have been be compressed, and examining them, I found that this was indeed the case.

Using my editor, I tried to reload one of the packed source codes, and of course this did not work. The files were compressed, and looked like greek on my monitor. Back to Shell and start PowerData. Now, I could load the source code and have PowerData automatically decrunch it.

I do not really like having my source code powerpacked, so I re-loaded every one of the packed source files into my editor, and then saved them again with PowerData's Crunch flag disabled. Naturally, this caused all the packed source files to become "normal" (i.e. not powerpacked) once again.

So if you enable PowerData consistently, then most of the files that you use regulary will become powerpacked sooner or later, without you even knowing about it. Here is a small list of sample usages for PowerData:

- Icons. You are free to crunch all your icons, without them losing any functionality. Workbench will continue to be able to read the icons, and you can even "snapshot" an icon without needing to use PowerPacker to recrunch it. PowerData does it all for you.

- AmigaGuide documents can be crunched, and AmigaGuide will still read them.

- If you are a Fidonet point with limited disk space available before an "import", you can use PowerData to transparently crunch all the messages that are extracted from the PKT (packet) and imported and written to disk. When your mail scanner (Foozle, PointManager etc) reads the messages, they will be transparently decrunched by PowerData. Any new messages you write will also be transparently crunched. The effect is that you can do all the things you did before, only you will use half the disk space doing so!

- You can crunch all your include files for your C compiler

- You can keep the source code for your projects powerpacked, and still have your compiler read it, and even write compressed object files.

- Pictures can be compressed, and will still load into your paint program, be it DPaint (© EA) or Real 3D (©~RealSoft).

- If you play games like Sid Meier's Civilization, or Fighter Duel Pro (made by ..?), you can crunch the datafiles that these games use, and still play the games. Saves lots of space on your HD.

As you can see the possibilities are endless. It is completely up to you to define the limits of PowerData.

1.13 Some Important Notes

A couple of important notes...

Copying files

Since PowerData causes powerpacked files to decrunch, and non-packed files to crunch, the "Copy" command will probably appear to be very slow indeed, when copying powerpacked files. This is because Copy first opens the source file, causing PowerData to decrunch this, and next opens the target file, causing PowerData to crunch the file it has just decrunched. The net effect is that while the copy command DOES in fact work, it becomes very slow. So remember to disable PowerData when you are copying files from one place to another.

Note that the same is true when you use the Workbench to copy files (by using the "Copy" menu item, or by dragging icons from one volume to another), so this would be a good place to disable PowerData, too.

Crunching Executables

Another important note to developers and programmers: When you are linking your projects, you might think that PowerData would crunch the resulting executable like any other file (knowing that the linker opens the output file as a datafile) but PowerData is smart enough to not attempt to crunch executables (if it did, you wouldn't be able to run them, since PowerData doesn't hook into LoadSeg()).

This also ensures that if you DO use the Copy command without disabling PowerData, then any executables that are copied will NOT be recrunched.

You must use PowerPacker or Imploder to crunch executables, not PowerData.

The Exchange Program

A note on the Exchange program. As you know, the Commodities Exchange program allows you to enable and disable commodities. Now, you may think that this is the same thing as pressing the appropriate PowerData hotkeys, but this is not the case. While the hotkeys enable and disable transparent crunching and decrunching, the Exchange program's enable and disable gadgets will render PowerData COMPLETELY inoperative. It will be totally deaf to hotkey presses (even the hotkey for "enable") and can only be revived using the "Enable" gadget in the Exchange program.

Using PowerData with CygnusEd

A note regarding CygnusEd is also in order. You will find that PowerData apparently crunches files saved from CEd, even when these do not match the

filters. This is because when CEd saves a file it does so by first saving the file under a different name in a temporary file, and then renaming that temporary file to the original name. This ensures that the original file is not lost if there is a read/write error on the destination media.

You can switch off this behaviour by selecting "file save method/simple saves" in CEd. Otherwise you will have to adapt your filters so that they match the filenames of CEd's temporary files.

BCPL Programs

PowerData works transparently with all "newer" programs. These use conventional, approved ways of calling the DOS, and PowerData has no problems intercepting these calls.

However, most of the old 1.2/1.3 "C:" files (the files in the C directory) are so-called "BCPL" programs that behave somewhat erratic. Be warned that these will bypass PowerData, and thus not work on Powerpacked files.

As a Workbench 2.04/2.1/3.0 user, you should not have any problems with this, since all C: programs were rewritten in C with the release of this new OS. But if you still happen to have a couple of the old C commands laying around, be careful with them.

PowerData and Virus Checkers

If you are using a virus checker, then PowerData is more than likely to trigger some of the checker's protection mechanisms when you start it. This is normal. PowerData needs to take over certain function vectors within the operating system in order to work, and this is what triggers the alerts.

PowerData currently hogs these vectors in the DOS library:

Open()
Close()
Examine()
Write()
OpenFromLock()

Future versions may patch additional vectors in order to make PowerData even more transparent to the system.

Compressing PowerData's Preferences file

One last thing that may surprise you. Enabling all programs to read Powerpacker datafiles, you'd expect PowerData to be able to at least read its own preferences files, if these are compressed. This is NOT the case, however. PowerData cannot read its own settings if these are compressed. You can easily compress all your other preferences settings (for other prefs programs), because PowerData will enable those programs to read compressed files, but do not this to PowerData's own preference files.

1.14 Known bugs

Known Bugs

As this is a major rewrite of the original PowerPacker Patcher program (which was relatively bug-free, by the way), there are bound to be at least a couple of bugs in there.

Sometimes you may run into a situation where you think PowerData is doing something wrong. In those cases, spend a few minutes and try figuring out exactly what may be wrong. Sometimes, what you think is a bug is really nothing more than PowerData doing its job in a way you had never even though of.

For example, had I not taken the time here to explain the different behaviour of the Copy command, you would probably have though that something was wrong with PowerData, since the Copy command suddenly did not operate as quickly as it did before. But as explained, this behaviour is normal, and not a bug.

I have not yet discovered any "strange" behaviour, so right now, there are no known bugs. Feel free to write to me and tell me about any bugs you find.

There is still a couple of functions that are still to be implemented. For example, even if you specify "Icon Creation" when saving preference files, no icons will be generated. That bit of code will be implemented in a future release.

1.15 The future

Things to come...

So, how can you expect this program to evolve? I have thought up a few items that I would like to implement at one point or another. Please note that this list is only a sketch. There is no guarantee that any one of the items will actually be implemented, but it IS what I am opting for:

- A hotkey to terminally quit PowerData? Maybe...
- XPK support
- Client/server operation
- I am hoping to improve the quality and contents of this manual a great deal. There are still MANY more tips and tricks that need to be explained. The glossary.guide should be expanded.
- As the GUI expands (it WILL), I may extract the GUI code itself into a genuine Preferences program. I've not done this yet, due to the relative discrete overhead of this particular bit of code.
- 100% localization in all the "standard" languages
- Multi-pattern support

- Full blown ARexx interface (current one is minimal, but serves its purpose just fine)

Also note that the order of these items do not reflect their respective priorities. More items may appear on this wish-list, as more users begin leaving me feedback. I urge those of you who register PowerData to also find a couple of likes and dislikes about the program, and put that into your letter as well.

1.16 Thanks to...

This software was developed using the following tools and utilities:

GadToolsBox, by Jan van den Baard
Aztec C, by Manx Software Systems
CygnusEd, by CygnusSoft Software
Chelp v1.30, by Robert Wahlström & Mathias Widman

And also a brief thank-you, to Olaf 'Olsen' Barthel for the FracBlank screen blanker. Besides being a great blanker, it serves as a good example on how to implement a commodity for Workbench 2.04 and above.

I must also thank the following persons for their excellent work in doing the various localizations needed for Workbench 2.1+ users.

Norbert Fährmann - German
Reza Elghazi - French
Mirco Zanca - Italian

I did the Danish ("Dansk") localization myself, and as PowerData speaks English by default (also done by myself :-), this comes to five languages. Plenty enough this early in the program's life. I am of course hoping to get PowerData fully localized at some point.

Note that this program would have very difficult, if not impossible to write, had it not been for Nico François' powerpacker- and reqtools libraries. Both are excellent and have outstanding documentation, making it very easy for me to write good code the first time around. Thank you Nico!

1.17 Error codes

Error Codes

Below is a list of the errors that may occur when you use PowerData. I have also listed a possible solution for each problem, so that you can easily rectify the problem.

Note that if you are using Workbench 2.1, the messages may appear in the language you have chosen as the system language (using the "locale" program).

> An error occurred while initializing with the commodities.library broker
> (code ###)

This is a serious problem. PowerData has been unable to communicate with the commodities library broker (which is a synonym for the set of routines that manage each commodity).

This message can occur in different situations, but most likely it will be because you are out of memory, or (more rarely) because your system software does not recognize the commodities version that PowerData is asking for.

The error code means:

1: System error (not enough memory, probably)
3: Did not understand PowerData's broker version string

If you get message #1, then try freeing up some ram by closing windows, terminating programs etc. This should do the trick.

If you get anything but 1 or 3, you are in serious trouble :^/

> Unable to interface with Commodities

Something went wrong when PowerData tried to set up the hotkeys for Commodities. Your configuration file may contain some strange hotkey specifications that the Commodities broker cannot understand.

Try deleting the file PowerData.prefs in the ENVARC: and ENV: directories.

> Could not create a message port for use with Commodities

A trivial error, caused by low memory. Free up some memory and try running PowerData again.

> The Commodities broker is acting strange. System may be unstable!

Wow.. This sounds dangerous, doesn't it? Well, it may just be so. You'll get this message during normal operation, if the Commodities broker suddenly starts sending strange messages to us.

You should never receive this message, but if you do, you are probably close to a system crash. Reboot the machine and start over.

> You need <some.library> <some.version> in LIBS:

PowerData has been unable to open some library that is required in order for it to function.

If you have installed PowerData with any of the installation scripts in this distribution, you should not an error message like this very often. All you have to do is to copy the required library from the distribution disk into the current LIBS: directory, and try again.

> Include-pattern is too complex
> Exclude-pattern is too complex

If you get any one of these messages, it is because the include (or exclude) filter is too complex for AmigaDOS. This should never happen. If it does anyway, free up some memory, and then try again.

Ultimately, you may have to delete PowerData.prefs from the ENV: and ENVARC: directories, because one of these may contain some very long pattern specifications.

Deleting the preferences file(s) will cause PowerData to start up using its own, internal preferences, which you can then edit to suit your needs. Don't forget to save the new preferences by clicking the Save gadget.

> Out of memory during exlude-pattern allocation
> Out of memory during include-pattern allocation

These messages relate very closely to the previous set of messages, only here PowerData is telling you that there is nothing wrong with the patterns themselves, only the system is VERY low on memory.

Free up some memory, or reboot the machine, and then try again.

> Unable to initialize ARexx interface

This message can occur only in the ARexx version of PowerData. It tells you that something went wrong as PowerData tried to set up the required structures necessary to get ARexx going.

The most likely cause of this error is that you are running low on memory, or because PowerData can't open rexxsyslib.library (which should never occur).

> 'filename' exists, but is not a valid PowerData preferences file

During the version change from 38.105 to 38.110, a new preferences file format was introduced, which, unfortunately, renders saved preferences settings created with PowerData 38.100 and 38.105 unreadable.

To fix this, simply click away the requester and (manually) set up the various settings once again. When done, simply select "SAVE" from PowerData's menu, making PowerData overwrite the old settings you have stored in ENVARC:.

> Unable to open Timer.device

This message can only occur in the evaluation version of PowerData. It indicates that something went wrong as PowerData attempted to start an internal 20-minute countdown timer.

This timer measures out the evaluation version timeout, which is why it could never happen in a registered version (since it does not have the time constraint).

The problem is most likely due to low memory, so freeing up some should do the trick.

Below is a list of the messages you may receive when working with the PowerData preferences window:

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

This should be pretty much self-explanatory. You have entered something that the commodities broker cannot make any sense of. Click the requester away and re-enter a valid hotkey. For a description on hotkeys, please read the relevant AmigaDOS system documentation.

> Resetting include-filter to <something>

You have entered a pattern that doesn't make any sense. Click away the requester and re-enter a valid pattern. For a description on AmigaDOS wildcard patterns, please refer to the relevant AmigaDOS system documentation.

> This is not a valid PowerData preferences file

You have tried to load a preset preferences file that did not belong to PowerData. You must select a file that you have previously saved with PowerData.

> Unable to save current preferences (<something>)

As you exit PowerData, it tries to save a copy of the current preferences in ENV:, so that those preferences will re-appear in the window if you start PowerData again later on. Since ENV: is in RAM:, the settings will only be remembered until you reboot the machine. Then PowerData will look in ENVARC: for its preferences (ENVARC: is on your system disk).

This error message means that an error occurred as PowerData was trying to save the current preferences in ENV: This can almost only mean that memory is running low. As PowerData exits immediately after displaying this message, you have no means of "retrying" to save the preferences.

> No changes made to current preferences

A little reminder from PowerData to show you that even if you did try to load another preferences file from disk, nothing has yet been changed. This is true if you cancel the load requester, or if something goes wrong reading the preferences file.

> Unable to read PowerData preferences (error reading file)

This may indicate that you have a physical problem with your disk drive (or hard disk), but more likely you are trying to use PowerData with an "old" preferences file.

Briefly, the preferences file format of PowerData was kind of unflexible up to (and including) version 38.105. At that point the file format was changed to allow for future enhancements and more preference settings than the old format would allow.

Unfortunately, this does mean that PowerData 38.110 and later will *not* be able to read any preferences file belonging to a prior version. The good side is that the preferences file format will never have to change again.

Sorry about this inconvenience.

If you suspect that the error message may be caused by an old PowerData prefs file, then simply click the error requester away, set up all the gadgets to what they were before you got your new version, and then just save the preferences over the old ones. You can also delete the PowerData preferences files in ENV: and ENVARC: by hand, if you like.

1.18 Authors Address

Authors Address

If you have comments or suggestions for improving PowerData, or if you wish to register PowerData, you can reach me at the following address:

Michael Berg
Sjællandsgade 56, 4 lejl. 3
8900 Randers
DENMARK

(You may substitute the "æ" with the more widely recognized "ä" if you like). If you have any urgent questions, you can also call me up directly on the phone, on

+45 864 22402

If you have access to a modem, and access to Fidonet or AmigaNet, you can reach me at the following addresses:

2:230/816.8@FidoNet
39:140/102.8@AmigaNet

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

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

More nodes will eventually support PowerData.

If you are writing to me about a problem you are having with PowerData, then please include a printout of your system's parameters if you can. This is a great help for identifying the problem.

If you do not have a program to print out system information, then try out Nic Wilson's SysInfo, which is capable of doing just that. If you do not have this program, you can get it off the Fred Fish PD series.

1.19 Copyright

COPYRIGHT

FINALLY...

A warning to would-be crackers: Each registered version of PowerData carries not only the owners name, but also a unique code that is very hard to find. If I pick up a registered version from a BBS, or from any other place, for that matter, I'll know right away who it is registered to, regardless of what the "About" requester claims. So there... You've been warned...

When you register, you agree to all terms involved in owning a registered version, which entails agreeing with this copyright message. And breaking a copyright IS a fellony in the eyes of the law.

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

DISKETTE LIMITED WARRANTY

The author of PowerData warrents to the original license that the diskette on which the registered PowerData distribution 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 by yourself (including, but not limited to, utilizing any remaining free space on the diskette for your own personal use, and including any repair or optimization operation), in which case you will void this warranty.