PDPro

COLLABORATORS			
	TITLE : PDPro		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		March 1, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

PDPro

1.1 PowerData Professional

Documentation for

PowerData Professional 1.0b

An application independant, transparent
XPK file crunch and decrunch utility

Written by Michael Berg

This document fully describes how to use the PowerData Professional preferences editor. Please select an item from the table of contents listed below:

1.2 Background

Background

PowerData Professional is the result of literally years of development, and is the final descendant of a string of successful predecessors. It began as a relatively humble program called "PP" (Powerpacker Patcher) which performed the not so humble task of enabling applications to transparently load powerpacked datafiles as if these were normal files.

PP was the very first such utility for the Amiga and quickly became popular with users with limited disk capacity. Even though the program was fully functional and freely distributable, many people also registered their copy and wrote me asking for new features. This greatly encouraged me to continue developing the program and also gave me a reason to embark on other projects such as PowerCache.

As AmigaDOS 2.04 was released and accellerated machines became more wide spread, it became apparent that PP would need a major overhaul to keep up with these new high power systems. So I rewrote PP from scratch using more uniform and system friendly programming, and also adding some fundamental new features such as transparent file crunching, which was something beyond the reach of the old PP. The result of this efford was a program called PowerData, and this program is very much alive and well today.

However, with the introduction of XPK, I realized that PowerData would become a much more powerful utility if it could somehow read and write this format. It would also be nice if the program could use different XPK methods for different file types, and if it could control crunch/decrunch for individual applications in stead of just globally.

After yet another full rewrite, PowerData Professional, the software you’re using at this moment, was born. The program now supports all the above mentioned features, it uses an external preferences editor and generally supports the system in a much better way. In daily use it works better than dedicated packages such as XFH, yet it is signifficantly easier to set up. Transparent decompression works without any configuration at all. You simply run the daemon and away you go. Compression is also very easy to set up, since it requires only a few minor ajustments to be made using the preferences editor.

Like all its predecessors PowerData Professional is shareware. Unregistered it has a few limitations which makes it unfeasible for long term use, but

still allows you to try out all the features. Please see the section on

registering
for more information about this.

Note: All registered users of the old PowerData program will be able to use their existing key file with PowerData Professional. No silly upgrade fee, no need to register again. How about that! :-)

1.3 Introduction

Introduction

PowerData Professional is a utility which lets any application transparently load and save data in the XPK format. XPK is a powerful new data encoding standard which provides a uniform interface to a large number of different crunching and encryption algorithms.

In case you are wondering, XPK is Powerpacker compatible, so if you are upgrading from an old version of PowerData, you won't have to recompress all your crunched datafiles. You may eventually want to do this, however, because there are XPK methods available which are both better and faster than the Powerpacker algorithm.

So how does it work? Well, it's not that easy to explain, but the essential part of it is that PowerData takes over some internal DOS vectors so that whenever a program asks the operating system to open a file, PowerData intercepts the call, decrunches the file (if it was crunched, of course) and returns a handle on the decrunched file. The requesting application will then be reading from the temporary file, and so be fooled into thinking the original file wasn't crunched.

Transparent crunching works just like this, only in "reverse". When an application wants to save a file, PowerData steps in and redirects the application file to a temporary location where the application then saves its data. After the application closes the file PowerData crunches and copies the resulting XPK file over the file which the application was originally trying to save to.

It may sound complicated (and belive me, it is!), but the mechanics are well hidden and the overall effect is that applications suddenly seem able to load and save XPK compressed data files in any XPK format you like.

Here is a general list of features:

· Enables applications to load XPK compressed data
· Enables applications to save data in any XPK format
· Selectable crunch/decrunch mode per application
· Selectable XPK method per file type
· Localized
· Easy installation, Installer and IconX scripts supplied
· External, font sensitive, localized preferences editor
· Totally system friendly, works perfectly with SetMan-like utilities
· AmigaGuide documentation
· Much more! :-)

See the section on
Usage
for a complete description of all features.

1.4 Installation

Installation

Scripts for both IconX and Commodore's Installer utility (not included) are provided to make installation as convenient as possible. Double click either to install PowerData Professional on your system.

If you prefer to install the program manually, copy PDPro and its icon into an appropriate drawer. If you want it to be run automatically when the system is started, put it in SYS:WBStartup.

Next, copy the full contents of the included libs directory to your own LIBS: directory. This directory includes several important libraries such as guifront.library and xpkmaster.library, without which neither PDPro or PDProPrefs will run. It also contains a number of XPK compression methods for use with transparent crunching. The range of compression capability and efficiency of the included methods should allow you to set up a configuration which matches your system's capabilities. You don't actually NEED all the libraries included (in the compressors directory), so after you've configured PDPro, you can delete those which you never intend to use anyway.

Note that a preferences editor for guifront.library is also included although you need not install this to your hard drive. This editor allows you to control the visual appearance of GUIFront applications (in this case the PDPro preferences editor) in case the defaults do not suit your needs.

Also note that the encryption feature of PowerData Professional requires Nico François' reqtools.library. A copy is included in the libs directory, in case you don't have it already.

If you want PDPro and PDProPrefs to run localized (AmigaDOS 2.1 or later), you must also copy the necessary catalog files to your LOCALE: directory.

PDPro and PDProPrefs both require at least AmigaDOS 2.04 (V37) to work.

1.5 Usage

Usage

PowerData Professional may be invoked from either the Workbench or from a shell. Please select one of the following entries for a more detailed description of how this is done from either environment:

Workbench invocation

CLI invocation
Once PowerData Professional is running, you will get a new ↩
AppIcon on your
Workbench screen (unless you explicitly told PowerData not to do this, using the NOAPPICON tooltype/cli argument). After that everything will happen in a pretty transparent way. Clients which are defined to crunch in PDProPrefs will save data using the selected XPK method, and clients which are defined to decrunch will be able to load XPK crunched files.

To quit PowerData Professional, you can either double click its AppIcon, remove it with the Exchange program or simply run it again. Note that PowerData Professional might not be able to exit right away. If someone else has patched the same functions in AmigaDOS that PowerData has, or if an application still has temporary files open, you will get an error message prompting you to abort the exit operation, try again later or keep trying until it becomes possible to exit.

You may also enable and disable PowerData Professional using the Exchange program. However, the same restrictions concearning quitting the program (as explained above) also apply for disabling it. It may be impossible to do this if an application still has open files that PowerData is monitoring or if some other utility patched one of PowerData's AmigaDOS vectors.

1.6 Workbench invocation

Workbench Invocation

To start PDPro from Workbench either doubleclick its icon or move it into WBStartup and let Workbench do it for you next time your system is rebooted. PDPro currently accepts the following icon tool types:

NOAPPICON

Normally PowerData Professional puts up a small AppIcon on your workbench when it is running. If your Workbench is already cluttered with icons, or if you simply don't want it there, you may use this tool type to suppress the AppIcon.

ICONNAME=<name of AppIcon>

This tooltype allows you to specify the name of the AppIcon on your Workbench. Default is "PDPro Dock".

ICONPOS_X=<x position of AppIcon>

This tooltype enables you to put the AppIcon on a specific X (horizontal) position on your Workbench. Default is "no position", which means Workbench will just stick it where there is room for it.

ICONPOS_Y=<y position of AppIcon>

Like ICONPOS_X, this tooltype enables you to put the AppIcon on a specific Y (vertical) position on your Workbench. Default is "no

position", which means Workbench will put it where there is room for it.

If everything was properly initialized, PDPro installs a small AppIcon on the Workbench screen. Normally PDPro uses an image identical to its own tool icon for the AppIcon, but you can change this if you wish. Simply place your favourite icon under the name "PDPro_dock.info" in either ENV:Sys or ENVARC:Sys, and PDPro automatically uses it next time you run it.

1.7 Shell invocation

Shell Invocation

PDPro can be run from a CLI using the following command line template:

NOAPPICON/S

Here is a brief description of each argument:

NOAPPICON/S

This switch tells PDPro not to add an AppIcon to the Workbench screen. This is useful if your Workbench screen is a bit crowded or if you never use the AppIcon to exit the program anyway.

1.8 Author

Author

If you have any suggestions, comments or catalog translations for PDPro, you can reach me at the following addresses:

E-mail: mberg@scala.ping.dk (preferred)

Fido: 2:238/ 24.28
AmigaNet: 39:140/101.28

S-mail: Michael Berg
Sjællandsgade 56, 4
8900 Randers
Denmark

1.9 Program history

Program history

16-Dec-94 - Release 1.0b, internal revision 38.2 (public release)
- German translation provided by Henning Tietgens
(henning@Informatik.Uni-Bremen.DE)

```
    - Swedish translation provided by Magnus Holmgren (2:203/602.51)
    - Now works properly with xLoadSeg (Renze De Ruiter)
    - File notification bug fixed (several)
    - Fixed problem with 0-byte files not being loadable when PDPro
      was active (Lars Eilebrecht)

3-Dec-94 - Release 1.0, internal revision 38.1 (public release)

7-Sep-94 - Release 0.5, internal revision 38.1 (internal beta)
```

1.10 Credits

Thanks to...

· SAS Institute for their excellent C compiler, version 6.51 of which was used to compile both PDPro and PDProPrefs.

· All registered PowerData users. Your continued support has been invaluable and PDPro would not have become a reality without it.

· The authors of XPK in general, and Urban D. Müller in particular, for XPK and for letting me include xpkmaster.library in the PDPro distribution archive.

· The various authors of the included XPK encoding methods. I apologise for not crediting you individually, and for not including documentation for a few of the libraries, but I lost track of your names and original file distribution archives long ago. I'm confident your software can stand alone, however. You know who you are, people. Thanks! :)

· All betatesters for their invaluable help in getting the bugs out of PDPro and its editor.

1.11 Registering

Registering

As explained elsewhere in this document, PowerData Professional is released under the concept of shareware. What this means is that you are allowed to review the product as crasy as you like without any restrictions at all, but if after a period (a month or so) you start using it regulary, you must register your copy with me.

PowerData Professional has a few limitations when running unregistered. Firstly, it will display the 'About' requester from the menu whenever it is run. Secondly, it only allows you to configure the default client and exactly one custom client. You may freely configure more clients with PDProPrefs, but PDPro will ignore anything but the default and the first client.

Registering is very simple. Basically you mail me a small sum of money (currently US $20), and I send you back a disk with the latest version of

the program and a "key file". This file acts just like a regular key and unlocks the program so you can use it without the above mentioned restrictions. The program on the disk will probably be the same as the one you're already using, so to complete your registration you will only need to copy the key file to your own system. The rest happens automagically.

Once you have a key file, you will be "set for life" where updates are concearned. A key file for one version of PDPro is a key file for ALL versions of PDPro, both existing and future versions. Whenever I release a new version of the program, you simply download it from a BBS or any other PD outlet, copy the new binaries over your existing ones, and hey presto! you're fully updated and STILL registered.

To register your copy of PowerData Professional, fill out the included registration form, then print and mail it to me along with the registration fee. If you haven't got a printer, it's perfectly okay if you simply write the information on a piece of paper.

So why register? Well, if getting rid of the restrictions isn't enough, consider the fact that you are supporting an Amiga shareware programmer, which in turn means you are supporting the Amiga platform itself. The more registrations I receive, the more encouraged I become to produce updates and write new software for your favourite computer. No money, no code. It's that simple.

It might also be worth noting that I spend a lot of the money generated by PDPro for registering shareware myself. In turn this means your money is put to good use not just once, but two, perhaps even three times. This keeps the wheels moving and the Amiga rocking :-)

In all modesty, considering what PowerData Professional does, you're actually getting remarkable value for money. Look up the price of commercial Amiga software in any popular Amiga magazine, and you'll see what I mean. So let me rephrase my original question and ask you - Why NOT register?

1.12 Caveats

Caveats

This section contains a few pointers on how to use PDPro, and also how NOT to use it. Several things are covered here, and I strongly suggest you read all of it before using PDPro. Yes, ALL of it.

Asynchronous I/O

PowerData Professional is, like its predecessors, basically a hack. It works by taking over some internal AmigaDOS vectors which MOST applications go through in order to read from and write to files. However, there are applications which bypass these common functions to achieve faster throughput, and PDPro does not work reliably with these products. It can cause anything from program failure, a system crash or just make programs turn out erroneous output.

Whenever you receive some new software there are certain key words which

should catch your attention. The most important one is "Asyncronous i/o". If a program states that it uses async i/o, you MUST disable BOTH crunching and decrunching for that program. Not doing so may result in data loss or other serious misbehaviour.

Perhaps the most prominent example of a program which uses asynchronous i/o is Stefan Boberg's LhA. If you use this program often, you will have to configure it to "no crunch" and "no decrunch" using PDProPrefs, or the resulting .lha files will end up corrupted every now and then.

Another popular piece of software which uses async i/o is Nico François' mail reader "Spot". However, Spot offers internal message base compression, so the loss is not terribly important here. As long as you remember to disable PDPro crunching and decrunching for Spot, everything will be alright.

ExAll

Programs using the AmigaDOS function "ExAll" to gain information about file names and sizes in a directory may fail to operate properly. This is because PDPro doesn't patch the ExAll function to return the decrunched size of XPK packed files (like the Examine patch does) so file size information will be incorrect (too small) for XPK encoded files. If a program uses information obtained via ExAll to allocate memory for a file, it will allocate too little memory and the load procedure will either fail or overwrite innocent memory if the program is poorly written.

The reason PDPro does not patch ExAll is purely for speed considerations. The function would simply be excruciatingly slow if every file had to be opened and examined for XPK header information. Further, only a fraction of currently available software uses file size information returned by ExAll for other purposes than informational output (directory listings), so the problem is insignificant compared to that caused by asynchronous i/o, as explained above.

XPK Encryption Considerations

Although XPK encryption is fully supported in PDPro (as long as you have reqtools.library), it should be used only selectively, and with great care. Firstly, if you forget a password not I nor anyone will be able to recover data from your file. Secondly, some programs have a tendency to open and close files several times in a row, and with these programs encryption is not a particulary useful feature. In the case of MultiView, for example, every file to be viewed is opened THREE times, and if your file is password protected this means you will have to enter the password three times. Keep this in mind when planning which file types should use encryption.

Files NOT to Compress

There are certain files which should not be compressed under any circumstances. These include your startup-sequence, user-startup and files stored in ENV: and ENVARC:. It is okay to crunch Workbench icons, but don't crunch icons in WBStartup unless you are running PDPro out of your startup sequence (i.e. NOT from WBStartup itself, and before the LoadWB command in the startup-sequence). The explanation for this is related to the way Workbench detects and runs programs from WBStartup. It does this by looking for icons in WBStartup, and only programs which have icons get activated.

If an icon is crunched, Workbench does not recognize it as an icon (because PDPro itself hasn't been started) and so that particular program simply isn't started. If you crunch icons in WBStartup, either deliberately or by accident, your system may not boot as expected.

Keep in mind that doing a snapshot or modifying an icons tool types causes the icon to be saved and, if you have configured crunching for Workbench in PDProPrefs, crunched.

To summarize the above, your PDPro preferences configuration should include the following:

File types:

#?WBStartup/#?.info : No XPK crunching! (I.e. select "« None »" in the XPK configuration GUI for this file pattern)

Applications:

LhA : No crunching, No decrunching.
Spot : No crunching, No decrunching.