xpkCybPrefs General documentation

COLLABORATORS

	TITLE : xpkCybPrefs General documentation	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		July 20, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1 xpkCybPrefs General documentation **1**

1.1 xpkCybPrefs General documentation 1

1.2 Table Of Contents 1

1.3 Foreword about 'packing' :) 3

1.4 Introduction 4

1.5 What can I do with XpkCybPrefs?? 4

1.6 Requirements/Installation 5

1.7 Requirements 6

1.8 Installation 6

1.9 Prefs File 6

1.10 Important notes 6

1.11 Syntax conventions 7

1.12 Miscellaneous 7

1.13 Crunchers definitions 8

1.14 Basic definition 8

1.15 type 8

1.16 packer 9

1.17 mode 10

1.18 Best 10

1.19 Fastest 11

1.20 Average 11

1.21 Ask 11

1.22 END_DEF 11

1.23 Macros 12

1.24 Flags 13

1.25 Window 13

1.26 ForbidPack 13

1.27 ForbidUnPack 14

1.28 LoadLibs 14

1.29 Flushlibs 14

1.30 Frontscreen . . . 14
1.31 ScreenToFront . . . 15
1.32 UseTemp . . . 15
1.33 LoMem . . . 15
1.34 UserInfo . . . 15
1.35 Values . . . 16
1.36 PackPri/UPackPri . . . 16
1.37 Pubname . . . 16
1.38 WinX/WinY . . . 16
1.39 TestSize . . . 17
1.40 Delay . . . 17
1.41 xpkmaster flags . . . 17
1.42 What are these? . . . 18
1.43 UseXFD . . . 18
1.44 UseExternals . . . 18
1.45 AutoPassword . . . 18
1.46 Timeout . . . 18
1.47 Quiet Tasks . . . 19
1.48 Usage . . . 19
1.49 Configuring your tools . . . 20
1.50 Behavior . . . 20
1.51 Datamaster.library filetypes . . . 21
1.52 Hints & Tips . . . 21
1.53 Prefs file tips . . . 22
1.54 Programs . . . 24
1.55 Disclaimer . . . 24
1.56 Distribution . . . 25
1.57 Thanks,Hellos,Infos... . . . 25
1.58 History . . . 27
1.59 The idea . . . 27
1.60 XpkCYB1->XpkCybPrefs . . . 28
1.61 Future . . . 29
1.62 Author . . . 29

Chapter 1

xpkCybPrefs General documentation

1.1 xpkCybPrefs General documentation

xpkCybPrefs General documentation

This tool is GIFTWARE :-)

XpkCybPrefs

Version 1.0, 11 May 1997

RELEASE VERSION

(-: When you feel lost in the guide, use the INDEX button :-)

Table Of Contents

Foreword about 'packing' :)

Introduction

What can I do with XpkCybPrefs??

Requirements/Installation

Prefs File

Usage

Hints & Tips

Disclaimer

Distribution

Thanks,Hellos,Infos...

History

Author

1.2 Table Of Contents

Table Of Contents

MAINxpkCybPrefs General documentation

0.Foreword about 'packing' :)

1.Introduction

2.What can I do with XpkCybPrefs??

3.Requirements/Installation

3.1.Requirements

3.2.Installation

4.Prefs File

4.0.Important notes

4.0.1.Syntax conventions

4.0.2.Miscellaneous

4.1.Crunchers definitions

4.1.1.Basic definition

4.1.1.1.type

4.1.1.2.packer

4.1.1.3.mode

4.1.1.3.1.Best

4.1.1.3.2.Fastest

4.1.1.3.3.Average

4.1.1.3.4.Ask

4.1.1.4.END_DEF

4.1.2.Macros

4.2.Flags

4.2.1.Window

4.2.2.ForbidPack

4.2.3.ForbidUnPack

4.2.4.LoadLibs

4.2.5.Flushlibs

4.2.6.Frontscreen

4.2.7.ScreenToFront

4.2.8.UseTemp

4.2.9.LoMem

4.2.10.UserInfo

4.3.Values

4.3.1.PackPri/UPackPri

4.3.2.Pubname

4.3.3.WinX/WinY

4.3.4.TestSize

4.3.5.Delay

4.4.xpkmaster flags

4.4.0.What are these?

4.4.1.UseXFD

4.4.2.UseExternals

4.4.3.AutoPassword

4.4.4.Timeout

4.5.Quiet Tasks

5.Usage

5.1.Configuring your tools

5.2.Behavior

5.3.Datamaster.library filetypes

6.Hints & Tips

6.1.Prefs file tips

6.2.Programs

7.Disclaimer

8.Distribution

9.Thanks,Hellos,Infos...

10.History

10.1.The idea

10.2.XpkCYB1->XpkCybPrefs

10.3.Future

11.Author

1.3 Foreword about 'packing' :)

0. Foreword about 'packing' :)

"Packing"....

....Packing is really a great thing! On the Amiga, we had (roughly) 3

periods:

- The "stone age":the now massively outdated, outrageously unefficient,

hack-coded, absolute-addr packers...remember the Black&Decker, ByteKiller and

other junk... (loong list... :-)

- Then came the "middle age":PowerPacker & TurboImploder:two rather good

packers, CLEANLY coded (almost ;-).In those days, we were quite a number who

imploded the executables & PPacked the datas, hmm?

- And finally...the actual situation, with the excellent XPK concept.I

won't talk much of it here, as the xpk package has a good documentation you

surely read.

1.4 Introduction

1. Introduction

o Since its version 4, xpkmaster.library has a new feature: a preferences

protocol, to pack data with the best xpk-sublib according to its filetype.

o Thanks to a shared structure (the XpkPrefsSemaphore), xpkmaster

is able to "communicate" with an external, INDEPENDANT prefs system.

o Dirk Stoecker, the new xpk maintainer, has made his own "XpkPrefs"

program. My "XpkCybPrefs" is an alternative, which claims (of course :) to

be more powerful and flexible.

What does it mean for you?:

~~~~~~~~~~~~~~~~~~~~~~~~~~~

-> Well, you just select the "USER" packing method and then the

prefs system handles everything, choosing the best xpkXXXX.library

adapted to the type of data being crunched.

o In fact, I had began a similar project on my own, some 2 years ago, but

as a PATCH, because then, xpkmaster.library seemed to be "stuck" on V2, and

no more development seemed planned. So a patch to enhance this ageing

library seemed Ok.

When finishing my patch, I realized xpkmaster was under new development, as

Dirk continued the project. So I agreed to transform my patch system into

this "standard" xpkprefs protocol.

Read the History section if you knew/used xpkCYB1 before, or

simply if you're curious :-)

1.5 What can I do with XpkCybPrefs??

2. What can I do with XpkCybPrefs??

Hey what do you expect? Of course it will do the "minimal job", that's

to say associate xpk sublibs to filetypes. But there are LOTS of much

powerful options. It's a bit hard to summarize, so even if you're not

convinced by the quick description that follows or if you're ready to say

once again "Oh no! It's complicated! Xpk is complicated! Computers are

complicated! etc...", well.. PLEASE!!...

...STOP A FEW MINUTES AND READ THE MANUAL.

It's EASY to use, and will make your life EASIER, in a totally TRANSPARENT

way, once you've configured it.

But please do read the WHOLE doc, lots of details are spread everywhere :)

o Once installed and configured, you will be able to use the "fake" USER
~~~~~~~~~~~~~~~~~~~~~~~~~~~

packing method. There is NO Compressors/xpkUSER.library, it's just an internal xpk method signalling xpkmaster to pack the data using the prefs system.

o XpkCybPrefs does not "only" associate "one filetype->one packer".

Of course, this can be enough for most filetypes, but you have access to more flexible features:

- Size conditions: It "finetunes" the filetype conditions.

For example, a 300Kb Protracker module can be
packed with SMPL whereas a 50Kb one would be a
SQSHed, and a PTK-chip of 3Kb 'd be SHRI'd :-)

- You can define more than just ONE xpk-method for a filetype!!

That's what I call the "multipacking". YES! Multiple packers may be tried on the same data, with the following options:

* ask option:

Pops up a requester asking you which packer to choose.

* Best, Fastest, Average options:

You surely guessed: YES! With such options, the data will be packed (and possibly unpacked) in turn with all the selected packers and the final "winning" packer will be given back to xpkmaster.

(Respectively the best %, the fastest to decrunch, or a "mix" between both).

Other misc features:

~~~~~~~~~~~~~~~~~~~~~~

- Switchable Report Window with progress bars, statistics and all the hell :-)

- Quiet Tasks list (useful for disabling output automatically when called from Backup programs for example).

- Of course, you can define "NO packing" for a filetype.

- Memory optimization: The packing tests are done in a memory buffer or on a temp file for those who are short on memory.

(if depacking is needed, it will NOT require any extra buffer!! :=)

- and lots of options but you'll see...

Now read the rest of this doc :-))

1.6 Requirements/Installation

3. Requirements/Installation

Requirements

Installation
~~~~~~~~~~~~~~~~~~~~~~

1.7 Requirements

3.1. Requirements

This package needs:

o AmigaOS 2.04 / Kickstart V37 or higher.

o Reqtools.library (© Nico Francois)

o xpkmaster.library V4+ (© Bryan Ford, Urban Dominik Müller, Christian von Roques, Dirk Stöcker :)

(and some xpk sublibs I guess)

(Aminet: Util/pack/xpk_usr.lha)

o datamaster.library package (© Alexis Nasr ;-)

(Aminet: Util/Libs/Datamaster_lib.lha)

1.8 Installation

3.2. Installation

* XpkCybPrefs would usually be put in your SYS:WBStartup directory.

* The XpkCyb.prefs config file goes in S:

* datamaster.library, its "recognizers", and DMcontrol are provided.

copy contents of: "libs" directory to LIBS:

"c" directory to C:

Complete archive of datamaster on aminet: Util/Libs/Datamaster_vXX.lha

I strongly advise you to get the full archive, that provides complete docs

(user & programmer), and associated utilities such as DMcontrol and

DMlauncher. [end of the advert :]

1.9 Prefs File

4. Prefs File

Important notes

Crunchers definitions

Flags

Values

xpkmaster flags

Quiet Tasks

1.10 Important notes

4.0. Important notes

Syntax conventions

Miscellaneous

1.11 Syntax conventions

4.0.1. Syntax conventions

Some words about the syntax used in the prefs file:

~~~~~~

* The flags and values definitions MUST be between:
START_CYBGENERAL (1st line)
END_CYBGENERAL (last line)

* Everything is CASE-INSENSITIVE (the exceptions will be signalled).You may use REMs ";" and blank lines.

* "packer=RAKE" is good, "packer = RAKE" IS NOT! So please always stick everything together.

* NO "" or ” chars must be used for string parameters. Just type the strings normally.

* For the "switch" parameters, you can use indifferently YES/NO or ON/OFF (requested by Solo, I hope it’s clear enough now ;)

1.12 Miscellaneous

4.0.2. Miscellaneous

Sorry but for the moment there is no nice GUI to make the config, but it’s easy to do it by hand (it’s even more powerful in some cases, you’ll see). (There WILL be a GUI one day!!)

The prefs file is located in S:,not in ENVARC:/ENV:. This is because there is no GUI, so no "Use/Test/Save" features.... making usage of ENV: and ENVARC: a waste of space an memory.

A GUI (MUI) is planned, but it’s still a project. So if you want to make an interface or anything related to XpkCybPrefs, please DO CONTACT ME, so we don’t make twice the same work for nothing ;-)

Note: When you modify the prefsfile, new preferences are automatically loaded and taken into account. (DOS Notification is used).

If a program is currently using XpkCybPrefs or if the prefs file is incorrect in some way (bad definitions, etc...) a requester will inform you of the problem.
~~~~~~

1.13 Crunchers definitions

4.1. Crunchers definitions

That's the big one!

Here, you'll see how it's possible to ASSOCIATE packer(s) to filetypes.

You can know all the filetypes currently recognized by the datamaster.library

by using one of the tools provided with it:use "DMcontrol LIST".

(Refer to the documentation of the datamaster package for complete info)

Basic definition

Macros

1.14 Basic definition

4.1.1. Basic definition

Each definition consists of:

type

packer

mode

END_DEF

1.15 type

4.1.1.1. type

type=....

Just the filetype.

You can define multiple type lines:the rest of the definition will be

appliable for all the filetypes.It's nice when you want to define the same

packing method for multiple filetypes.

example: type=Ascii-Amigaguide

type=Ascii-Text

type=Generic

....(rest of the definition)....

For each filetype, you can add 2 size-conditions (in kilobytes):

example: type=Ascii-Amigaguide >50 <100

(amigaguide files between 50kb & 100kb)

type=Music-PtkClones <10

WARNING:if you define such conditions, respect this rule:

define packers for _ALL_ the size ranges:

ex: type=generic <10

....

type=generic >10 <120

...

type=generic >120

The "filetype" string can be either a major-filetype (like MUSIC-Synth for example), or a sub-filetype (ex: Hippel or PlaySid, etc, in this particular case). See Datamaster.library filetypes section for information.

Note:if a filetype has no associated packer(s) then the program will fallback to the 'generic' filetype prefs.

There MUST be a "generic" filetype definition or the prefsfile will be considered invalid.

1.16 packer

4.1.1.2. packer

packer=...

----

o The packer name is the classic 4 characters xpk sublib name:

example:NUKE, RAKE etc... (UPPERCASE!!!)

o You can add the usual efficiency parameter:

example:NUKE.100

SHRI.25

o There is an extra parameter:the chunksize:

example:RAKE.100.64 (use RAKE, efficiency mode=100%, chunksize=64 Kb)

Note:if you use this, you MUST define the efficiency TOO.

Note:Don't bother to know is this chunksize is supported by this xpk sublib, as it will be automatically modified if there's a problem.

Note:Remember that increasing the chunksize, will also increase the chunk-loading time (but surely lead to a better efficiency), but if you set 256 Kb chunks, keep in mind that you'll lose one of xpk's benefits:partial-loading (if you have a 200 Kb file, the WHOLE file would be loaded in that case=>memory usage while packing gets bigger). Set this depending of your config (memory, HD, etc..)

The classic chunksizes are 32-64Kb usually.

o One of the VERY powerful features of this, is that you can define MULTIPLE "packer=..." lines for a filetype.The mode option will affect what happens next in that case.

Multipacking can take quite a long time, & depends of the selected

packers of course. The Window option will open a window showing you all
the necessary informations (except if the calling task is in the
Quiet Tasks list).
It's self-evident that you must 'compare' packers that have the same
kind of action (using FAST & SHRI with a Best mode for instance
would be ridiculous)
See Hints & Tips.
o There is a very useful 'fake' packer name:"DONT" :-)
Setting "packer=DONT" means this type of file must not be packed.Quite good
no to try packing JPEGs don't you think?
o Crypted/Password packers are not supported.(Yet :)
See the Future section.

1.17 mode

4.1.1.3. mode
mode=...

When there is multipacking, you can choose between 4 modes:
(if you don't define a "mode=..." entry, best will be used).
Best,Fastest and Average test-modes will make packing & possibly
unpacking tests to find out which xpk library gives the best result for
the current data.
Best
Fastest
Average
Ask

1.18 Best

4.1.1.3.1. Best
"Classic"...the packer with the best "%CF" (compression factor) will be
chosen.
It would be ridiculous to make a comparison between SHRI/NUKE/BLZW:it's
clear that SHRI's going to win anyway.This reasonment is appliable to
the other modes too (use your brain *;-)

1.19 Fastest

4.1.1.3.2. Fastest

This will perform a depacking operation after each packing, in order to

calculate the depacking-speed.

The fastest-decrunching method will be used.

The depacking is made to NIL: (!), so this will NOT cost you any

extra memory buffer!! :-)

1.20 Average

4.1.1.3.3. Average

Performs a depacking like Fastest, but this time, it tries to find out

which packer has the best balance between the compression ratio & the

depacking speed.

example:SHRI:54% CF & 2 hours to depack (just joking!)

MASH:49% CF & 2 seconds........

Guess who wins the match, he he...I think even a FAST:25% CF could.

(against SHRI & in average mode :)

Like Fastest, the depacking is done in NIL:, no extra-mem needed...

1.21 Ask

4.1.1.3.4. Ask

Well, in this case, there won't be a multipacking. You'll be popped a

requester asking you to choose between all the packers defined for this

specific filetype.

1.22 END_DEF

4.1.1.4. END_DEF

END_DEF

Extremely important flag to know the end of the definition.

DON'T EVER FORGET IT or impredictable things may happen ;-)

(very predictable in fact: ever tried 142 packers? ;-))

1.23 Macros

4.1.2. Macros

Macros are a way to group packers under a same name:if you use for example the NUKE/MASH/RAKE multipacking often in your definitions, just create a macro instead of typing each time:

```
...
packer=MASH
packer=NUKE
packer=RAKE
...
```

The macros looks like:

```
START_MACRO
macro=MYPACKERS ;name
packer=MASH use like the normal "packer=..." in a Basic definition
packer=NUKE.100.20
packer=RAKE
END_MACRO
```

Then you can use it in the definitions, like "packer=..." :use "macro=..." instead!

```
type=generic
macro=MYPACKERS ;it's "macro=", NOT "packer", be careful!!
mode=best
END_DEF
```

This is fully RECURSIVE, that means you can use MACROS IN MACROS, or a mix of PACKERS & MACROS !!!!

examples:

```
~~~~~~~~~
START_MACRO
macro=BIG_ONE!
macro=FAST packers ;defined elsewhere
macro=SLOW packers ;...
packer=IMPL
macro=DUMMY packers ;...
END_MACRO
type=generic
packer=IMPL
macro=NICEMACRO
packer=PWPK.100.64
mode=average
END_DEF
```

You can define the macros anywhere in the prefsfile, in any order.

1.24 Flags

4.2. Flags

Window

ForbidPack

ForbidUnPack

LoadLibs

Flushlibs

Frontscreen

ScreenToFront

UseTemp

LoMem

UserInfo

1.25 Window

4.2.1. Window

Window=YES/NO or ON/OFF [Used when Multipacking only]

The report-window will open only when there is a multipacking (with mode set to best, fastest or average)...and if you've set this flag ON or YES.

(Default is ON.)

Information about the report-window:

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

* You can pause/abort packing by clicking on the window close-gadget, then:

click on: "Continue" =>resume packing

"Abort current" =>abort current packer (skips to next one)

"Abort All" =>well.....

(when aborting, the best packer so far will be used).

* The window uses the screen font if not proportional.Else, it falls back to the system default font (guaranteed not be proportional).

1.26 ForbidPack

4.2.2. ForbidPack

ForbidPack=YES/NO or ON/OFF [Used when Multipacking only]

If set, there will be a Forbid() during the packing, which means the system will be freezed and the packing will use 100% of the CPU.

Anyway we're not in the days of the A500/Powerpacker anymore, so I don't think you'll use it ;-)

If ON it overrides the PackPri/UPackPri value.

(Default is OFF.)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

1.27 ForbidUnPack

4.2.3. ForbidUnPack

ForbidUnPack=YES/NO or ON/OFF [Used when Multipacking only]

More interesting than ForbidPack:it's very useful when you're using an
Average or Fastest mode:as there will be a depacking of data, Forbid()ing
can be very useful for the ACCURACY of the depack-timing.I personally use
it, as depacking is usually quick anyway.

If ON it overrides the PackPri/UPackPri value.

(Default is OFF.)

1.28 LoadLibs

4.2.4. LoadLibs

LoadLibs=YES/NO or ON/OFF

If set, then all the the xpk-sublibs which names are found in the prefsfile
will all be loaded to memory at library start! Useful if you don't want to
reload these libraries all the time.

NOTE:the libraries are loaded, but not Openlib()'d, that means that if nobody
else uses them, they may be flushed from the system's memory (by the workbench
flush, CXflush, Surveymem ;-)

(Default is OFF.)

1.29 Flushlibs

4.2.5. Flushlibs

FlushLibs=YES/NO or ON/OFF [Used when Multipacking only]

If set, then each time a subpacker is used, it will be flushed from memory,
immediately after its use.This will save memory, but will force to reload
the xpk sublib each time.No problem if you have a harddrive.Avoid it if you
only have a floppy drive (whaaat? ;-).

Of course, it's a bit ridiculous to use LoadLibs AND FlushLibs , as you
surely guessed.

(Default is OFF.)

1.30 Frontscreen

4.2.6. Frontscreen

FrontScreen=YES/NO or ON/OFF [Used when Multipacking only]

If the frontmost screen is public, then assume it's the packing-application's
screen, & open the report-on it.

(Default is ON.)

1.31 ScreenToFront

4.2.7. ScreenToFront

ScreenToFront=YES/NO or ON/OFF [Used when Multipacking only]

Will bring the report window's screen to front on window opening.

Note that this can get really annoying sometimes :-)

(Default is ON.)

1.32 UseTemp

4.2.8. UseTemp

UseTemp=YES/NO or ON/OFF [Used when Multipacking only]

When "multipacking" the data, a output buffer is needed.

By default, memory is used. If UseTemp=ON, then a temporary file will be used instead.(Typically on the hard drive)

You need a CYBTMP: assign somewhere on your HardDisk. Of course, do not make the assign on RAM: ! ;) Or on T: if it's in RAM: etc

(well, as usual, use your brain!!).

(Default is OFF.)

1.33 LoMem

4.2.9. LoMem

LoMem=YES/NO or ON/OFF [Used when Multipacking only]

If UseTemp=OFF but the allocation of the output buffer fails (not enough memory), then packing fails. Best defaults are then used (See Usage).

But if LoMem=ON, then XpkCybPrefs will then try to use the temporary file option as second chance (like if UseTemp=ON).

(Default is ON.)

1.34 UserInfo

4.2.10. UserInfo

UserInfo=YES/NO or ON/OFF

This option will pop up a requester showing some informations about the crunched data and the packer used etc.

It's the only way yo have some information when there is a unique packer entry for the filetype (no multipack, no window output).

The requester will NOT appear in the following cases:

- if the calling task is in the Quiet Tasks list.
- if there was a multipacking and Window=ON (you got some info already).
- if you were asked to choose the packer.

(Default=ON.)

1.35 Values

4.3. Values

Like for the flags, the definitions MUST be between:

START_CYBGENERAL (1st line)

END_CYBGENERAL (last line)

PackPri/UPackPri

Pubname

WinX/WinY

TestSize

Delay

1.36 PackPri/UPackPri

4.3.1. PackPri/UPackPri

PackPri/UPackPri=numeric value (-127 -> +128)

The packing/unpacking priorities when multipacking.

This will in NO WAY affect the final "normal" packing.

It's highly recommanded NOT to set the unpack priority to low values, as

decrunch-timings would become really unaccurate, leading to bad packer

choices...You can set the UpackPri to high values, it can be enough, or use

the ForbidUnpack flag.

These values are overriden by ForbidPack & ForbidUnPack

Defaults are PackPri=0 and UpackPri=15

1.37 Pubname

4.3.2. Pubname

Pubname=FooScreen (no "" or ”, remember)

Define a public screen where to open report windows.

If the screen is not found, fall back to DefaultPubScreen (Workbench usually)

Note:The PubScreenName is case SENSITIVE.

(Default is Workbench screen.)

1.38 WinX/WinY

4.3.3. WinX/WinY

WinX/WinY=CoordValue

Defines the default X/Y coordinates of the report window.

’rem’ these (use ’;’) if you want the window to be autocentered.Note that

the autocentering is ’smart’:even if you have a "virtual" screen (example:

a 1280x256 WB with a PAL:640x256 resolution like me :), the window will be

centered on your "physical" screen.

(Default is autocenter.)

1.39 TestSize

4.3.4. TestSize

TestSize=value in bytes.

This is the size of the buffer that will be requested from

xpkmaster.library in order to analyze and recognize the filetype of the

data to pack.

Normally you should not have to modify this value. (setting too small values

will lead to mis-recognitions)

(Default=5000)

1.40 Delay

4.3.5. Delay

Delay=number of seconds between info messages of Report Window.

Quite useless.Nice when packing big files, you can see the progression, but

with small files, the delay can be longer than the packing itself! This may

slow down drastically a packing by a backup program for example.

Mainly used for early debugging :)

That’s why by default, Delay=0

1.41 xpkmaster flags

4.4. xpkmaster flags

What are these?

UseXFD

UseExternals

AutoPassword

Timeout

1.42 What are these?

4.4.0. What are these?
As already said, xpkmaster accepts an independant prefs system, but
there is a common shared structure (quite normal).
This structure provides some options that must be supported by ALL the
prefs systems.
In clear: you will find the following flags in Dirk Stoecker's XpkPrefs
system, in XpkCybPrefs, and in any other xpk-prefs system (I think 2 is
enough :-)
When writing these docs, this is still under development, so check out
your current xpkmaster archive to see if the options have been implemented.
(That's also why the explanations on the flags are not very accurate :-)

1.43 UseXFD

4.4.1. UseXFD
UseXFD=YES/NO or ON/OFF
Use xfdmaster.library to decrunch the input data.
(Default is OFF.)

1.44 UseExternals

4.4.2. UseExternals
UseExternals=YES/NO or ON/OFF
Use the new xex libraries.
(Default is ON.)

1.45 AutoPassword

4.4.3. AutoPassword
AutoPassword=YES/NO or ON/OFF
xpkmaster will open automatically a password requester if a password is
required but not passed.
(Default is OFF.)

1.46 Timeout

4.4.4. Timeout
Timeout= value in seconds.
Password timeout.
(Default is Timemout=120)

1.47 Quiet Tasks

4.5. Quiet Tasks

The definitions MUST be between:

START_QUIETTASKS (1st line)

END_QUIETTASKS (last line)

It's nice to have a window report during a CLI-xpack or on the Prowizard screen (Hi Nico! ;)...But it's quite annoying to see Abackup packing a huge number of small files, with windows popping in/out and...the window operation being too fast to be seen, and in the mean time, also slowing the packing... ;-[

That's why you can define a list of tasks, where the Window flag will be overriden (set to OFF).

Conventions:

----

* syntax:"task=taskname"

* The taskname is INSENSITIVE:this is unusual but I think you'll find this as handy as I do (I hate having to type a string byte for byte ;-)

* You can use aliases (ONE alias per line:no '|') [*, #?, ?, etc...]

Examples:

task=*backup

....

1.48 Usage

5. Usage

Once XpkCybPrefs has been started, you're able to use the "USER" packing method... Just like any "normal" xpk sublibrary.

Note that this will work with 99% of programs but NOT with some particular ones, because they don't respect the xpk conventions and access directly the xpk subpackers without making usage of the xpkmaster.library.

Have a look at the Programs section.

You should read the Prefs File section first. All the features are explained in it. Then read this usage part.

The Behavior section is important. be sure to read it.

Configuring your tools

Behavior

Datamaster.library filetypes

1.49 Configuring your tools

5.1. Configuring your tools

I personally think it's obvious, but hell... maybe it isn't!?

"How do I use this extremely complicated tool?":

~~~~~~

with xpack, for instance:

> xpack ram:Mybloodyfile method=USER

wowowoooooow... complicated eh??

I will *NOT* tell you how to configure your DirOpus buttons, nor Dirwork,

nor ANY other tool. Simply because there is *NO NEED FOR THIS*!.

You just set USER as method instead of your usual MASH, RAKE or whatever you

used before...

I you did not know how to use xpk before, well you're surely not reading

this doc at the moment, you must have deleted the archive saying "wheww

this is too powerful, it must be complicated, I will stick to powerpacker

- I'm sure some guys around still use it :-((-let's go back to some game

playing, damn where did I put my joystick? ha, if only I had a PC I could

play Doom, Quake and enjoy the great Windows 95" :-((((

1.50 Behavior

5.2. Behavior

Simply , that's what you can expect when using XpkCybPrefs:

~~~~~~

* When there is a multipacking, even if the data may be packed/unpacked

lots of times, you must understand that the Prefs program does NOT do the

"real" final packing.It will return the best packer to xpkmaster that WILL

then do the "normal" packing.

* Most of time, if you use one packer per filetype, everything should be

very "quiet and transparent". (Unless you use UserInfo=ON).

* If a problem occurs, the best possible definition will be used:

-> In case of multiple packers defined, the FIRST one in list will be

used, so choose it well :-)

-> Even in the extreme low-memory situations, a packer definition WILL

be returned: it's the "Generic" filetype one.

-> If a filetype has no definition, Generic will be used.

* When it appears that multipacking must be used, then XpkCybPrefs will

request from xpkmaster to LOAD THE WHOLE SOURCE DATA INTO MEMORY.

Yes, this is a bit heavy, but that's life :-)

Anyway, memory is cheaper everyday now :-)))

If the file can't be loaded entirely, then the 1st packer in list will be

used.

Note 1: to remove XpkCybPrefs, just launch it again. (if not possible, a

requester will inform you).

Note 2: xpkmaster.library is used only when needed, so when you update new

versions of xpkmaster.library, and if XpkCybPrefs is not currently packing

some data, you can just flushlibs (with, "c:avail flush", Workbench debug

menu, Surveymem, MCP version requester etc etc....).

(no need to quit XpkCybPrefs, flush, restart etc :-)

1.51 Datamaster.library filetypes

5.3. Datamaster.library filetypes

I advise you to read the full docs of datamaster.library.

(Aminet: Util/Libs/Datamaster_lib.lha)

Anyway, here are some hints, for a quick startup.

What you want is to know the filetypes datamaster recognizes, so you can type

them in the xpkcyb.prefs file. Right? Well, there is an easy solution:

The "MAJOR-filetypes" can be listed using "DMcontrol LIST"

Some of these "MAJOR-Filetypes" offer more precise additional "sub-filetypes".

example: the "SMPL-Others" recognizer "contains" the ADPCM, AIFF, SUN, VOC

and Wave sub-filetypes.

To have full info, including sub-filetypes, use: "DMControl ti=all".

This will output all the information contained in all the currently loaded

recognizers.

Well..."everything printed in BOLD WHITE can be used as filetype"

EASY no? :-)

So it's up to you to use the global filetype or the sub-filetype

separately if you need them. Nice no?

1.52 Hints & Tips

6. Hints & Tips

Prefs file tips

Programs

1.53 Prefs file tips

6.1. Prefs file tips

When making the config file, you have to choose well the packers & packing

modes, but...just imagine this is done ONCE FOR ALL.Then you will be able to

pack everything transparently, with a simple click, with a UNIQUE packer, in

ANY of your tools supporting xpk!!!

The default configfile is the one I use.Have a look at it, I hope it'll suit

most of your needs.

Some hints now:

* DO NOT USE the CRMS packer. DELETE IT, IT'S CRAP!

(It just trashes data!! :-(

Read the following xBench stuff ....

* I can't give you 'the' super benchmark...YOU will have to do some of it

yourself.Just try, change your config prefs , retry etc.. have fun ;-)

To do an optimal multipacking, you'd better have a rough idea of what packers

can be compared.

A good thing is using the new xBench program, that can be found in the xpk

package:

USAGE:

~~~~~~

> xBench <file> TEST

This will try ALL the xpk compressors (with all modes), and give you

complete stats.

> xBench <file> TEST PASSWORD <Password>

Same thing, but also tries password modes.

RESULTS:

~~~~~~~~

xBench is useful for benchmarks of course, but also helps to detect which

xpk sublibs are BADLY CODED and TRASH DATA. (!).

If you get a message like "Decrunched buffer different to source!", well,

DELETE (YES!!!) the sublib that caused this message, of your

LIBS:Compressors/ directory.

This will improve your system stability.

Some buggy sublibs are more "lethal" and may crash your machine during the

test. If this happens, the faulty library that must be deleted is THE ONE

AFTER THE CURRENT OUTPUT. (compressors are alpha-sorted)

So for example if you have something like this as last output... :

RAKE: 100 1.7 24052 0.26 196969 51212 0.03 1707066 53.1

... then the library causing the crash is the one AFTER "RAKE" in your

Libs:Compressors/ directory.

In my case, it would be "RDCN" (no offense meant to its author ;-)

* Sorry if I didn't provide a GUI for showing all the informations on the

xpk sublibs.It was hard enough to make the job & all the hell of this! :)

To get info on the sublibs, use the xpk related cli commands, or use the GUIs

of tools like PDPro, Abackup, Diavolo...Anyway, here is the "classification"

that comes out, grouping packers that have comparable abilities:

Fast packers

ACCA

BLZW

CBR0

FAST

RDCN ;<--- CAUTION. TEST IT WITH XBENCH

RLEN ;<--- CAUTION. TEST IT WITH XBENCH

Average Packers

NUKE

IMPL

PWPK

Good Packers

GZIP ;WOW :)

ILZR

LZCB

MASH

RAKE

LHLB ;quite impressive sometimes!

NUKE ;sometimes, beats the others on small files

FRHT

Specific packers

SQSH ;for sample data.SMPL can be better for huge files with

SMPL ;digitized voices & so on...

Megaslow packers

SHRI (that one can't be compared to any other ;-))

* All filetypes that are already packed should have a "packer=DONT"

(JPEG, GIF, LZX, LHA...)

1.54 Programs

6.2. Programs
Some programs don't respect xpk conventions and access DIRECTLY the xpk
sublibs instead of passing through xpkmaster.library :(
So, of course, they can't access the "USER" method.
~~~~~~
* XFH: One of the 1st programs using xpk. Really old, but efficient :)
Maybe xpkmaster wasn't completeley finished (!!) when this program
was made so it has an excuse!! :=)
* DiskExpander:
This one has no excuse! It's more recent but seems xpkmaster was
too easy for the programmers. Maybe they wanted to do it "the hard
way"?! Well...sure... :-(
(And you know what? I bought it!! :)
Special cases:
~~~~~~~~~~~~~~
* Diavolo:
From my experience when developing this prog, it appears that
Diavolo does NOT pack whole files. It cuts everything to small
16Kb-or-so parts, that are passed to xpkmaster INDEPENDENTLY.
So recognition might be Ok for 1st part, but sadly not for the rest
of the data. So there's no interest in setting "USER" packing in
Diavolo :-(
[it would work, but with random results, probably 80% of the data
would be recognized as "Generic" :[
* In general:
All progs that do not pack a file in "one pass" won't benefit of a
good filetype recognition : the 1st part of data will be recognized
correctly, but the following parts surely won't (as information for
recognition of most filetypes are in the beginning of the files).

1.55 Disclaimer

7. Disclaimer
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 quality and performance of this program is
with you. In no event can I be liable to you for damages of any kind
arising out of the use of this program, or the inability to use it.

1.56 Distribution

8. Distribution

o This package is Copyright ©1997 by Alexis Nasr.

All rights are reserved.

o This package is "Giftware" so no donation is required.

If you use this package, I would enjoy receiving a postcard or anything

you'd think worth from you. See the Author paragraph.

(It is *not* in the public domain)

o It may be freely distributed provided all the files of the

original package remain unaltered and are included in the distribution.

They may, however, be archived to conserve space.

o No profit is to be made by selling this software. You may only

charge enough to cover reasonable production and distribution

costs.

o This software may not be included in a commercial package, without

the author's written permission.

o Magazines can include this on a coverdisk, provided the author gets a

copy of the magazine.

o This software may not be uploaded onto any BBS that claims

copyright on uploaded material.

1.57 Thanks,Hellos,Infos...

9. Thanks,Hellos,Infos...

My config:

----

* A1200+ Blizzard A1230 IV Card (68030/50Mhz/16Mb/FPU)

* HD 540 Mb

* Squirrel + CD300e drive.

* 33.6 Sony Modem.

* 1085S monitor (thanks to Blackbird.."Wow le luxe, c'est dingue!" :)

Development:

Entirely in assembler, using the hmm... <NO COMMENT> Asm-One v1.29 ;-)

It has been successfully tested with Enforcer & Mungwall.(the program, of

course, NOT Asm-one....just joking...really?).If it works fine on my own

machine, with the 1000% patched system I have, it should be fine hehehe.

Thanks:

----

* Big thanks to Dirk (Stoecker) for his help finding solutions to
keep as much options of the "xpkCYB1 patch" in the XpkCybPrefs ;-)
(and for the notification and semaphore setup sources)
Wow... The daily E-mail exchange surely helped!
(Gosh! How could I do without Internet some months ago??? ;=))
[e-mail:stoecker@rcs.urz.tu-dresden.de]
* Hellos fly to 'Solo' for taking time reading the docs, and testing
this package in order to "have an idea"...
(Thanx also for delirious daily e-mail, pal :-))
[e-mail:johant@club-internet.fr]
* The cuuuuuuute icons were designed by Eric Gerard (marchiii :-)
[e-mail:egerard@club-internet.fr]
You can also get his MagicCountry package (aminet: pix/mwb), if you want
a good bunch of icons for your tools! I'm sure you'll find anything you
need in this ;)
* Thanx in avance to good'ol' Reez (David le Corfec) who plans (?) to
make the MUI GUI for XpkCybPrefs. Nothing sure for the moment, it's just
"in study" but I have good hope.
(if he doesn't do it, he'll be flamed, anyway ;)
(bon maintenant ke t'as eu ton heure de gloire -warf- dans ma doc, tu te
mets au boulot quand? :)
Adverts:

Just some tools I really use & appreciate...
No particular order, no logic, no classification ;)
* Eagleplayer (EP v2.00 rulez!!!). Gosh how would I spend endless hours
in front my amiga without a good musical support? With EP it's a real
pleasure... Well if you use Delitracker or Hippoplayer what can I say?
Better 3 good progs than just one :) But if you don't know EP or still
have an old version watch out for the new v2.0 series...
[No I'm not paid by the EP dudes :]
* Mods Anthology 4CD-pack by Gryzor. ("ben ki c'est lui?" :-)
Maybe I'll finish listening to all the mods within 4 or 5 years he he.
Thanx Nico for the masterpiece and hope it will get distributed as it
deserves in the PC-world... teach these dudes what a MOD is once for
all! ((-: "It began on Amigaaaaaaa" :-))
[this should remind you a well know tune ha ha :]
* Xit , by Laurent Kempé.

A DAMN GOOD replacement for the outrageously old xData... MUI

rulezzz, GUI rulezzz....

Xit+XpkCybPrefs, and here you go!!!!!

* Ami-CPC , by Ludovic Deplanque.

The best Amstrad CPC emulator around! Keep up the development!!

* ZXAYEMUL player by Patrik Rak.

Spectrum, Amstrad, AY3-8912... the good old sound of 8bits... (enough of

all these ugly PlaySID versions!!! Now it's time for AY3 tunes!!! ;-)))

* MCP by AlienDesign.

Needless to present this one I guess... Well, anyway if you don't know

about it and still use 34 commodities & patches, losing memory and

cpu-time for nothing... have a look at this piece!!...

* Filemaster 2

Still beats all the Dopus and memory hungry "fat" dirbrowsers... When

you need to do a quick operation on the fly, it's still a MUST...

* CED

Nearly same remark as for FM2 :)

* FastBoot.

Fantastic tool! ("hack"? did I say "hack"?? ;)

It saved me HOURS (DAYS? ;) of rebooting, during the development!!! ;-D

Booting in FIVE SECONDS with a full system has become something natural

now, he he...

Well, I'll stop now, or I'll begin coping my startup-sequence file ;)

1.58 History

10. History

The idea

XpkCYB1->XpkCybPrefs

Future

1.59 The idea

10.1. The idea

(this part was written in the original xpkCYB1 package).

(Obsolete now of course eh ;)

Two years ago or so, I thought of a way of using the xpk library to its

maximum.I also saw some tentatives like "XpkKnight".Not very convincing, and

you had to pass all your files to ONE program...Then the idea came:"hey why

not do an xpk-sublib that would use the other ones?? It'd detect the
filetype & crunch to the best".Simple to think of it...harder to begin some
paper job...and even harder to do it for real :-)
I really had very few time, and anyway, this time was already used, as I had to
work on other programs of mine (Surveymem, new versions of ChipSaver, other
stuff...).
Anyway, last year, I found some time & began the work:the xpkCYB1.library
itself was rather quickly made (huhu), being an "empty shell".Then I began
the xpkcybhandle.library.Painful job (very hard to debug a library), and I
had to do everything from scratch:I still lacked the filetype-recognition
device, & the patch itself :-).So all I could do was simulate a single
filetype & work 'blind'.Hard job, it GURUed very often, and I finally gave up
because there wasn't any programming pleasure in it, no visible results, just
an endless game of assembling/GURUing/rebooting.
Nevertheless, the sources I left were not so hideous:it still bugged but the
patch job was 50% done, the xpkcybhandle something like 60%.Crunching worked
a bit, the windows opened (followed by flashing red screens GRRRRRRrrr :).
So a few months ago I decided to take back the sources & try to make
something of them.After some "cleaning" thing were becoming better, but I
still had to make a BIG part:the datamaster.library ENTIRELY FROM SCRATCH.
This took me quite a long time, but it was worth it.

1.60 XpkCYB1->XpkCybPrefs

10.2. XpkCYB1->XpkCybPrefs
DO NOT USE xpkCYB1, xpkCYB2 etc. THEY'RE BUGGED "TO THE BONE".... :-(
Unfortunately, xpkCYB1 bugged as hell :-)
I had not tested it thoroughly enough, and my beta-tester had not noticed
anything (well.. without using Enforcer, it's harder! :).
Anyway, in the beginning of 1997 I decided to take back the whole project,
debug it, and make a real "good" release.
...and I've made it. After some long and hard work, xpkCYB1 finally worked!
It worked perfectly, no bugs etc...BUT...
...It's at this particular moment that I saw that xpkmaster had a new
maintainer, and he was willing to do improvements. I contacted him to
inform him that xpkCYB1 existed etc...
Well... it was a shock to learn that he was developing his own prefs
system!!! :)
So, instead of keeping the patch, we agreed I'd adapt xpkCYB1 and transform

it into the standard xpkprefs system. So I've lost a few unimportant features,
but now my job has become MUCH simpler (I don't have to analyze and patch
the whole xpack() input/output streams anymore !!!).
With daily e-mail, we managed to setup the "special" features needed by
XpkCybPrefs, while keeping the "standard" prefs compatibility etc...
So I've "lost" some options, but the whole thing is much nicer this way,
because it's cleaner and will keep ascending compatibility in the future.
that's all folks :-)

1.61 Future

10.3. Future
I don't have any more ideas. do you??
* GUI! GUI! GUIIIIII! We need a GUI with hundreds of buttons. You'll have
it!!... :-)
* Password passing if requested by users. (YOU!!)
The feature is nearly implemented both in xpkmaster & xpkcybprefs but the
idea of a plain-ascii password in a prefsfile is a bit unsafe, so it's
your choice.
* Implementing the default Progress-Report feature.

1.62 Author

11. Author
This package is the result of some painful MONTHS of work (if you don't
believe me you're free to do it yourself :-)).
As usual I release this as GIFTWARE, E-MAILWARE etc... as:
* I know nobody pays sharewares, coz nobody cares of the authors.
* There are some crackers that spend their days hacking every keyfile:
I don't want to spend more time protecting my work with a heavily
protected code that would annoy ME during the development, and would
annoy NOBODY after the program was cracked.
(Hé Nico si tu veux j'ai CrackWizard.exe ;-)
* Freeware/Giftware is a nice thing: I'm really fed up because now every
single program is a $30 F...ING shareware!! You have to pay for
everything, this is a shame...
Anyway.....
I would really appreciate any sign of gratitude, ya know :-)
I know you're not bad guys but you "forget", you "don't have time" to send a

letter, a postcard, a 2 Gigas SCSI-II HD, to the people whose programs you're using EVERY DAY...HMmmmmmm? Am I right? Well, if you took the time to copy the stuff, install it, and then use it all the time, maybe you could move your [BIIP.... US-Censorship passed] & let the poor programmer know at LEAST that his job was useful to you?! Is this TOO MUCH??

Or maybe I should put lots of bugs in the code so people will then SURELY write & say "oh it's so nice but it bugs". Saddening....

Now some adverts for some of my other progs:

~~~~~~

* Surveymem v3.08: (util/cdity)

* TFMX-converter v1.0: (mus/misc)

* Chipsaver v1.83: (mus/misc)

* datamaster.library (util/libs)

Well, that's all folks! Have fun, Amiga rulez!

Snail Mail: The Cyb'

Alexis Nasr E-mail: nasr@hol.fr
27, rue Formigé ------
Résidence RENOIR French Minitel network: 3615/RTEL1 - 3614/RTEL2
33110 Le Bouscat ----------------------- BAL: "The Cyborg/NGC"
(FRANCE)
~~~~~~