

xpkCybPrefs General documentation

COLLABORATORS

	<i>TITLE :</i> xpkCybPrefs General documentation		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		June 25, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	xpkCybPrefs General documentation	1
1.1	xpkCybPrefs General documentation	1
1.2	Table Of Contents	2
1.3	Some Blah-Blah about 'packing' :-)	3
1.4	Introduction	3
1.5	Features	4
1.6	Requirements/Installation	5
1.7	Requirements	5
1.8	Installation	5
1.9	Prefs File	6
1.10	Important notes	6
1.11	Crunchers definitions	7
1.12	Definition	7
1.13	Macros	12
1.14	Flags	13
1.15	Values	17
1.16	xpkmaster flags	18
1.17	Quiet Tasks	20
1.18	Usage	20
1.19	Configuring your tools	21
1.20	Behavior	21
1.21	Datamaster.library filetypes	22
1.22	Hints & Tips	23
1.23	Prefs file tips	23
1.24	Programs	25
1.25	Legal mumbo-jumbo	26
1.26	Disclaimer	26
1.27	Distribution	26
1.28	Thanks, Hellos, Infos...	27
1.29	History	29

1.30 The idea	30
1.31 XpkCYB1->XpkCybPrefs	30
1.32 Versions	31
1.33 ReleaseNotes	32
1.34 Future	32
1.35 Author	32

Chapter 1

xpkCybPrefs General documentation

1.1 xpkCybPrefs General documentation

xpkCybPrefs General documentation

XpkCybPrefs
Version 1.2, 12 september 1997, Giftware
** Release Version **
© Alexis 'Cyb' Nasr

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

IMPORTANT!!! -
ReleaseNotes

Table Of Contents

Some Blah-Blah about 'packing' :-)

Introduction

Features

Requirements/Installation

Prefs File

Usage

Hints & Tips

Legal mumbo-jumbo

Thanks, Hellos, Infos...

History

Author

1.2 Table Of Contents

Table Of Contents

MAIN

xpkCybPrefs General documentation	
0.	
Some Blah-Blah about 'packing' :-)	
1.	
Introduction	
2.	
Features	
3.	
Requirements/Installation	
3.1.	
Requirements	
3.2.	
Installation	
4.	
Prefs File	
4.0.	
Important notes	
4.1.	
Crunchers definitions	
4.1.1.	
Definition	
4.1.2.	
Macros	
4.2.	
Flags	
4.3.	
Values	
4.4.	
xpkmaster flags	
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.	
Legal mumbo-jumbo	
7.1.	
Disclaimer	
7.2.	
Distribution	

```

8.
Thanks, Hellos, Infos...
9.
History
  9.1.
The idea
  9.2.
XpkCYB1->XpkCybPrefs
  9.3.
Versions
  9.4.
ReleaseNotes
  9.5.
Future
10.
Author

```

1.3 Some Blah-Blah about 'packing' :-)

```
0. Some Blah-Blah 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 present 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: the internal "USER"
packer.When using this one instead of a "normal" (NUKE,RAKE etc) method,
information on the packed data is passed to a preferences protocol, allowing an
AUTOMATIC, TRANSPARENT compression with the best xpk-sublib according to its
filetype.(To say thing quickly, as there are MANY other features possible)
```

```
o All you have to do is configure your prefs once for all, then select
"USER" in your tools as xpk-packer. And off you go! This is a feature we've
been waiting for since a looong time!!!
```

o Thanks to a shared structure (the XpkPrefsSemaphore), xpkmaster is able to "communicate" with an external, INDEPENDANT prefs system. 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. You won't be disappointed I think. Just try both XpkPrefs & XpkCybPrefs systems and choose!

1.5 Features

2. Features

- 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 For those who use XFH, be happy: XpkCybPrefs FULLY works with it. That means all the packing on your HD can be 100% filetype-optimised!!
Note: it works with all the programs using xpkmaster.library "normally" :-)

Here is a quick summary of the features:

~~~~~

- \* Extremely user-friendly GUI (by David "Reez" Le Corfec)
- \* Data recognition/selection can be made on:
  - FileType (uses datamaster.library) [Filesize conditions too!]
  - TaskName pattern
  - Efficiency parameter of USER (USER.25 , USER.95 etc)
- \* Multipacking is possible (!). That means compression can be tried with multiple packing methods. 4 modes are possible:
  - Best : Best % compression factor
  - Fastest : Fastest depacking time (yes, autodepacking included!)
  - Average : A mix between previous ones.
  - Ask : Pops a requester, so you choose "by hand"
- \* Optional report window with progress bars, statistics and all the hell...
- \*
  - Quiet Tasks
    - list (useful for disabling output automatically when called from Backup programs, XFH...)
- \* 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 more...

\* Giftware :-) [SHAREWARE SUCKS!!]

## 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 xpkmaster.library V4+ [INCLUDED]  
(© Bryan Ford, Urban Dominik Müller, Christian von Roques, Dirk Stöcker)  
( Aminet: Util/pack/xpk\_user.lha)

Some changes were made recently in the XpkPrefsSemaphore protocol.  
This release of XpkCybPrefs needs xpkmaster V4.15 or higher.  
If you use an older v4 version, you may get some enforcer read-hits  
(nothing to worry about). Also, some new features like the "USERefficiency"  
criterion won't work.  
[Unfortunately, semaphore version can't be tested]

- o Reqtools.library (© Nico Francois) [everybody has this one eh...]
- o datamaster.library package (© Alexis Nasr ;- ) [INCLUDED]  
( 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 ENV: & ENVARC:

\* 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\_lib.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 :]

\* I also included current version of xpkmaster at the moment of the release, so you can have a quickstart (but getting the complete latest archive is of course recommended)

## 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 in Prefs file

~~~~~

(no need to read this if you use the GUI Prefs ;-)

* 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 ;)

Prefs loading/changing:

~~~~~

\* On startup, the prefs file is searched in ENV: & ENVARC:

\* When the ENV: prefs file is modified, 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.

## 1.11 Crunchers definitions

### 4.1. Crunchers definitions

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

Using the GUI Prefs program will surely make your life easier!!  
Anyway, read the explanations below ;-)

Definition

Macros

## 1.12 Definition

### 4.1.1. Definition

Each definition consists of:

---

```
-----
1)      A CRITERION OPERATOR
-----
```

Basically, it's what will make the choice of X or Y xpk packer.

There are 3 possible criterions: (\*\* In ascending priority \*\*)

- \* Filetype (size conditions are possible)
- \* Task Name
- \* USER.xxx efficiency

When data is analysed, the prefs file is scanned, searching for these criterions/keywords.

```
type=....
```

```
~~~~~
```

Just the filetype, as detected by the datamaster.library.

(You can know all the filetypes currently recognized by the datamaster.library by using one of the tools provided with it:use "DMcontrol LIST".)

You can define multiple type lines:the rest of the definition will be applicable for all the filetypes.It's nice when you want to define the same packing method for multiple filetypes.

(this is applicable for the other criterions, that can even be mixed)

```
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.

TaskName=...  
~~~~~

This operator has a higher priority than "Type=..." ones.

If the calling task matches the given pattern,condition if OK.  
(matching is case insensitive & patterns are accepted: \*, #? etc)

Example:       TaskName=\*Prog  
                  ....(rest of the definition)....

[the "MyBeautifulProg" task would match ;]

USERefficiency=...  
~~~~~

This operator has a higher priority than "Type=..." & "Taskname=..." ones.

When you select "xpkUSER" as packing method,you can specify the efficiency, as with any xpk method (USER.50 , USER.100 etc).  
Default is USER.100 of course.

But what is the advantage of using "USER.xxx" that will lead to use "xpkMASH" for instance, instead of directly setting "xpkMASH" in the calling program, you may ask?

Well, right you are: in that case, no advantage :-).  
The \*BIG\* advantage is when you want to do some "multipacking" (read further on).

NOTE: This needs xpkmaster.library v4.15+

Conclusion  
~~~~~

You can put multiple operator lines one after another, and even mix them. They are all INDEPENDENT. Doing so can only be useful if you want to use the same packing scheme with different operators,for instance if "type=XXXX" or "task=\*YYY" or "USERefficiency=80" then pack using "RAKE":

```

type=XXX
task=*YYY
USERefficiency=80
...
packer=RAKE
...

```

```

2) ONE OR MORE PACKER ENTRIES

```

```
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 criterion.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=ON" 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 "Mode=Best" 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

History  
/Future section.

```

-----
3)          MODE ENTRY
-----

```

When there is multipacking, you can choose between 4 modes:  
(if you don't define a "mode=..." entry, "mode=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.

Mode=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 *;-)

Mode=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!! :-)

Mode=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...

Mode=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 criterion.

```
-----
4)                END_DEF
-----
```

END\_DEF

~~~~~

Extremely important flag to know the end of the definition.

DON'T EVER FORGET IT or unpredictable things may happen ;-)
(very predictable in fact: ever tried 142 packers? ;-))

1.13 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
                          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.

## 1.14 Flags

### 4.2. Flags

```
-----
                        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).

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 frozen 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.)

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.)

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.)

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.)

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.)

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.)

UseTemp

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

When "multipacking" the data, an 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.)

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.)

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.)

1.15 Values

4.3. Values

Like for the flags, the definitions MUST be between:

```
START_CYBGENERAL (1st line)
END_CYBGENERAL   (last line)
```

```
-----
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 recommended 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 overridden by "ForbidPack" & "ForbidUnPack"

Defaults are PackPri=0 and UpackPri=15

```
-----
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.)

```
-----
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.)

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)

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

1.16 xpkmaster flags

4.4. xpkmaster flags

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 :-)

UseXFD

UseXFD=YES/NO or ON/OFF

Use xfdmaster.library to decrunch the input data.

(Default is OFF.)

UseExternals

UseExternals=YES/NO or ON/OFF

Use the new xex libraries.

(Default is ON.)

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.)

Timeout

Timeout= value in seconds.

Password timeout.

(Default is Timemout=120)

1.17 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 or XFH 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=ON" flag will be overridden (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
....
```

1.18 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 also.

Configuring your tools

Behavior

Datamaster.library filetypes

1.19 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
```

wowowooooow... 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" :-(

## 1.20 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. So data will be packed "for nothing" during the tests, then RE-packed with the "winning" packer by the calling task.

* Most of time, if you use one "packer=" per criterion, 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 "packer=" 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, Surveyemem, MCP version requester etc etc....).
 (no need to quit XpkCybPrefs, flush, restart etc :-)

1.21 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)

When using the GUI Prefs program, no need to read this, just have fun clicking on the info buttons :-)

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?

1.22 Hints & Tips

6. Hints & Tips

Prefs file tips

Programs

1.23 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 ;-)

Rough classifying of well-known xpk sublibs:

Fast packers

ACCA

BLZW

CBRO

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...)

1.24 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.

* DiskExpander:

This one has no excuse! It's rather 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!! :)

And you know what?? XFH, which is quite old, WORKS perfectly with
"xpkUSER" as it accesses to xpkmaster.library...

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).

1.25 Legal mumbo-jumbo

7. Legal mumbo-jumbo

Disclaimer

Distribution

1.26 Disclaimer

7.1. 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.

1.27 Distribution

7.2. 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, e-mail, 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.

1.28 Thanks,Hellos,Infos...

8. Thanks,Hellos,Infos...

My config:

- * A1200+ Blizzard A1230 IV Card (68030/50Mhz/16Mb/FPU/SCSI)
- * HD 540 Mb
- * CD300e x2, ZIP 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]

* Enormous thanks to my good'ol'friend Reez (David le Corfec) who made the fantaaaastic MUI-based GUI for XpkCybPrefs.
This will greatly "boost" the program I hope! I was aware that a script can be a pain to use (this one definitely WAS ;), but my C/MUI skills being close to zero, I was a bit stuck.
So thanks again Dav'!! (If there are more than 3 people using XpkCybPrefs, you'd have played quite a great part in the process ;)

* 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 cuuuuuute 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 ;)

* Warm thanks to the users of XpkCybPrefs who e-mailed me for suggestions, etc... (I suppose there are thousands of users but they are too shy, yes yes, it's surely that :-))

- Niels 'Frolic' Froehling (Seal@CHATLINE.MCNET.DE)

Sorry if I didn't implement the stuff like you requested it, but it was a bit too specific (and defining prefs would have become quite overcomplicated for "standard" users :) Hope the task & efficiency new options will be enough...

- Christopher S Handley (ELA95CSH@sheffield.ac.uk)

"Bugs? what bugs??" :-)
Hi, "SCSI-man" he he he...
Hope xpk correctly works on system by now ("gosh, what did he do for it to suck?" ;)

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 ;)

1.29 History

9. History

The idea

XpkCYB1->XpkCybPrefs

Versions

ReleaseNotes

Future

1.30 The idea

9.1. 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 (Surveyemem, 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 GRRRRRRrrrr :).

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.

1.31 XpkCYB1->XpkCybPrefs

9.2. 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 :-)

1.32 Versions

9.3. Versions

v1.2 (12 September 1997)

Additions:

~~~~~

- "TaskName=" & "USERefficiency" criterions.  
[ requested by Niels 'Frolic' Froehling (Seal@CHATLINE.MCNET.DE) ]
- GUI by David "Reez" Le Corfec included in the package.  
[ requested by me :-))) ]

#### Changes:

~~~~~

- Big rearranging/changes in the doc, to make it clearer (much less nodes ;-)

v1.0 (11 May 1997)

Initial release

1.33 ReleaseNotes

ReleaseNotes

Some changes were made recently in the XpkPrefsSemaphore protocol.
 **** This release of XpkCybPrefs needs xpkmaster V4.15 or higher. ****
 If you use an older v4 version, you may get some enforcer read-hits
 (nothing to worry about). Also, some new features like the "USERefficiency"
 criterion won't work.

[Unfortunately, semaphore version can't be tested]

1.34 Future

I don't have any more ideas. do you??

- * 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.

1.35 Author

I've invested some hard work on this program. I've done it just in hope that it
 might be useful to "someone out there" :-)

I absolutely REFUSE to FORCE users to pay something for this prog, as I am
 convinced that shareware SUCKS... at least HEAVILY crippled shareware, with
 the keyfiles & all the hell... Hey people, wake up! Stop asking \$10 for each
 CLI command you code!

Of course, any feedback is strongly encouraged :-) Feel free to send e-mails,
 postcards, HDs, girlfriends (over 18 please), chocolate bars, peanut packs, or
 toilet paper [hmmm..nah...forget this one!]. My dream would be to gain some
 money to be able to offer my poor Al200 a tower+zorroII+Gfx-card, but I know
 this will never happen unless I win some lottery (which I doubt, as I don't
 participate to such stuff ;)

Now ... adverts for some of my other progs:

~~~~~

- \* Surveymem v3.08: (util/cdity/Surveymem308.lha)
- \* TFMX-converter v1.0: (mus/misc/TFMX-Converter.lha)
- \* Chipsaver v1.83: (mus/misc/Chipsaver183.lha)

- \* datamaster.library: (util/libs/datamaster\_lib.lha)
- \* Amstrad CPC ripped tunes: (mods/misc/AYsongs\_CPC\_01.lha)

All for today...

Have fun, Amiga rulez!

Snail Mail:  
-----

The Cyb'

Alexis Nasr  
27, rue Formigé  
Résidence RENOIR  
33110 Le Bouscat  
(FRANCE)

E-mail: nasr@hol.fr  
-----

French Minitel network: 3615/RTEL1 - 3614/RTEL2  
----- BAL: "The Cyborg/NGC"

3614-3615 AMIGATEL  
BAL/ "The Cyborg"

---