

PackX

COLLABORATORS

	<i>TITLE :</i> PackX		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		February 12, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	PackX	1
1.1	PackX Guide Main Menu	1
1.2	copyright	1
1.3	Info about PackX	2
1.4	Requirements	2
1.5	usage	3
1.6	history	5
1.7	Bugs Report	5
1.8	Hall of Fame	6
1.9	About XPK	6

Chapter 1

PackX

1.1 PackX Guide Main Menu

```
                PackX v1.5 (30-Oct-95)
Written by Adam Ciarcinski
Copyright ©1995 Imagic Software
```

```
Contact via InterNet
E-Mail:  ciarcins@beta.seatech.tuniv.szczecin.pl
        adam@felix.univ.szczecin.pl
```

```
Copyright
  Read it first
```

```
Features
  PackX?  What is that?
```

```
Requirements
  What you need to run PackX
```

```
Usage
  How to use PackX
```

```
History
  What was changed in PackX
```

```
Bugs Report
  What you should do if PackX does not work correctly
```

```
Thanks
  Hall of Fame
```

1.2 copyright

PackX is KEYWARE. It means that you pay nothing to the author and you MUST spread it for FREE (with InfoX, I suppose), except the following case: if you make software that need a KEYFILE - you SHOULD send it to the author.

Registered version of your program will also be accepted. :-)

You are NOT ALLOWED to modify the program in any way, or any part of it for any other purpose than mentioned in this documentation.

The author is not responsible for any damage (file lost, disk corrupt, mouse dizziness, fire on motherboard :-)) caused by PackX.

IMPORTANT: PackX was not tested on animals!

1.3 Info about PackX

- * PackX is CLI/Shell utility which allows to pack/unpack or ←
encrypt/decrypt
files using
XPK
. Never run it from Workbench!
- * PackX was written in assembler, that's why it is small and fast.
- * PackX can pack files selected by given pattern and scan directories.
- * PackX can move or copy packed files to the given directory.
- * PackX can be interrupted by Ctrl-C.
- * PackX never writes files that are larger when packed.
- * PackX can work with the given task priority.
- * PackX doesn't change the protection and comment of packing file. When running on V39+, it also saves owner and group ID.
- * PackX can add suffix to the packed file and remove suffix from unpacked file. When adding or removing suffix, the source file may (or not) be deleted.
- * PackX can open progress window on the default or specified public screen. Progress window is font-sensitive, but if the screen's font is too large, PackX will use topaz/8. Packing can be aborted by pressing Abort gadget or by pressing either Esc or A. Note that progress update rate depends of the packer's chunks size, so sometimes you will see only 0% and 100%, but the file will be packed.
- * PackX returns ERROR if some IO error occurs or not enough memory; returns INVALID RESIDENT LIBRARY if a library cannot be opened; returns WARN if something went wrong with file or xpk returned error, but file processing was not stopped (except user break).

1.4 Requirements

PackX needs the following things:

- * Kickstart v2.04 (37.175) or newer
- * xpkmaster.library v2 or newer (in LIBS: directory)
- * some xpk sublibs (in LIBS:Compressors directory)

Read

XPK
documentation how to install XPK package.

1.5 usage

PackX is CLI/Shell utility. Never run it from Workbench!

```
PackX <file/path> [TO|DEST <path>] [ALL] [LIB|METHOD <lib name>]
[PWD|PASSWORD <password>] [LOSSY] [SD|STEPDOWN] [NOOW]
[PRI <priority>] [CS|CHUNKSIZE <size>} [EXT|SUFFIX <suffix>]
[KEEP] [QUIET] [FORCE] [GUI] [PUBSCR <pubscr>] [DELAY <ticks>]
```

Template: FILE/M/A, TO=DEST/K, ALL/S, LIB=METHOD/K, PWD=PASSWORD/K, LOSSY/S,
SD=STEPDOWN/S, NOOW/S, PRI/N, CS=CHUNKSIZE/N, EXT=SUFFIX/N, KEEP/S,
QUIET/S, FORCE/S, GUI/S, PUBSCR/K, DELAY/N

Where:

FILE - file(s) to pack/unpack/encrypt/decrypt, for example:

```
PackX Buster    unpacks file Buster
PackX DH1:Docs/#?.txt LIB NUKE
    packs all files with .txt suffix in DH1:Docs directory
```

TO=DEST - a destination path, where to move (or copy if you set KEEP option) packed files; if you also set ALL option, all processed files from source subdirectories will be moved/copied, but their directories will not be created

ALL - enable subdirectory scan, for example

```
PackX ALL DH1:Docs ALL LIB NUKE
    packs all files and subdirectories in DH1:Docs directory
```

LIB or METHOD - specifies method to pack file(s) with, for example:

```
PackX DF0:More LIB SHRI.75
    packs file More using SHRI method, mode 75
```

PWD or PASSWORD - specifies password to encrypt/decrypt with, for example:

```
PackX Secret LIB ENCO PWD RosesAreRed
    encrypts file Secret using ENCO method
    and password RosesAreRed
PackX Secret PWD VioletsAreBlue
    deencrypts file Secret
    with password VioletsAreBlue
```

LOSSY - enables lossy packing (lossy means that file packed with lossy packer will not be the same after unpacking as the original. For example JPEG is lossy)

SD or STEPDOWN - enable reducing packing efficiency to save mem if necessary

NOOW - no overwrite; do not delete file that already exists while saving output, for example:

```
PackX Demo LIB RLEN NOOW EXT .xpk
  packs file Demo and saves it as Demo.xpk,
  but if Demo.xpk exists, Demo won't be packed
```

PRI - set PackX's task priority (-128 to 127), for example:

```
PackX WC.doc PRI 3  unpacks file WC.doc with priority 3
```

CS or CHUNKSIZE - specifies the chunk size; small chunks speeds up packing, but reducing pack efficiency. WARNING: very small chunk size is dangerous and may cause packed files larger than their originals.

EXT or SUFFIX - specifies a suffix to append to the name of the packed/encrypted file, and remove from the name of the unpacked/decrypted file, for example:

```
PackX S:Start LIB BLZW SUFFIX .xpk
  packs file S:Start and saves it as S:Start.xpk
  file S:Start will be removed
PackX S:Start.xpk EXT .xpk
  unpacks file S:Start.xpk and saves it as S:Start
  file S:Start.xpk will be removed
```

NOTE: Extensions are case sensitive.

KEEP - do not delete original file (works with TO=DEST and/or EXT=SUFFIX), for example:

```
PackX S:Stop LIB BLZW EXT .xpk KEEP
  packs file S:Stop and saves it as S:Stop.xpk
  file S:Stop will not be removed
PackX S:Stop.xpk SUFFIX .xpk KEEP
  unpacks file S:Stop.xpk and saves it as S:Stop
  file S:Stop.xpk will not be removed
```

QUIET - enable quiet mode; no progress display, no error messages

FORCE - enable processing files already packed/encrypted with XPK

GUI - open graphical progress window instead of printing progress text in the output; all messages are printed in progress window

PUBSCR - specifies the public screen you want the graphical progress window be opened; if the given pubscreen doesn't exist, PackX will use the default public screen; this option works only in GUI mode; example:

```
PackX Demo.text GUI PUBSCR Workbench
  unpacks file Demo.text and displays progress window
  on the Workbench screen
```

DELAY - gives a delay (in ticks; 1/50 sec) between messages displayed on progress window; works only in GUI mode; valid values are 0-255 (from 0 to about 5 seconds)

1.6 history

Version	Date	What was changed
1.0	25-May-95	first release
1.1	11-Jun-95	removed Enforcer Hit new name of temporary file added CHUNKSIZE option added SD as alias of STEPDOWN added SUFFIX option
1.2	2-Jul-95	fixed problem with locked files added NOOW and KEEP options splitted 'file (un)packed' message changed version string now file attributes are not changed fixed error with too long suffix few optimizations to reduce the code
1.3	20-Aug-95	little changes in text formatting added GUI and PUBSCR now QUIET shuts up PackX for good added skipping files if no gain skip messages changed to 'Error with'
1.4	06-Oct-95	added DELAY and DEST more legal window calculation return WARN in some cases
1.5	30-Oct-95	fixed suffix problems and added "couldn't append suffix" message now protection and comment are kept in DEST mode now files will be packed if KEEP and Delete protection flag not set

1.7 Bugs Report

PackX may not work correctly. If you found any bugs in PackX or in this guide please contact the author. The E-Mail address is on the Main page.

KNOWN PROBELMS:

Packing files with pattern in RAM: causes several file scanning. That's because PackX uses dos.library's MatchFirst(), MatchNext(), MatchEnd() functions, which doesn't work correct with files written to RAM: during scanning.

For example: PackX RAM:#? LIB RLEN FORCE causes endless packing (until you break PackX with CTRL-C).

PackX may not work correctly with some file systems (like CrossDOS).

```
\/  
+oo+  
/||\  

```

1.8 Hall of Fame

I want to thank the following people:

```
Georges "Melkor" Goncalves (melkor@ramses.telesys-innov.fr)  
Bruno Rohee (rohee@univ-mlv.fr)  
Nils Goers (n.goers@art-line.art-line.de)  
Niels Van Willigenburg (niels@aobh.xs4all.nl)  
Leo "Nudel" Davidson (leo.davidson@keble.oxford.ac.uk)  
Rich "Lego-Master" Jesse (rjesse@qgraph.com)
```

1.9 About XPK

XPK is a package written by Urban Dominik Mueller
(umueeller@amiga.physik.unizh.ch)

XPK is available on AmiNet sites.
For more information read the documentation for XPK.