XPKatana_eng

COLLABORATORS			
	TITLE : XPKatana_eng		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		February 12, 2023	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

XPKatana_eng

1.1 XPKatana 1.0 © 1996 by Eric Sauvageau.

```
                                                XPKatana V1.0

                          Chop yer file sizes to bits! :-)

                       Copyright ©1996 by Eric Sauvageau (Merlin)

                                     SHAREWARE

   First Part:
                  ~About~~~~~~~~
                       What is XPKatana?
                  ~Features~~~~~

                  ~Requirements~
                     Second Part:
                  ~Installation~
                       How to install it?
                  ~Tooltypes~~~~

                  ~Shell~Usage~~
                     Third Part:
                  ~Main~Window~~~~~
                       How to use it?
                  ~Progress~Window~

                  ~Set~Packer~Wnd~~

                  ~Menus~~~~~~~~~~~~

                  ~ARexx~Commands~~
                     Fourth Part:
                  ~Legal~Stuff~~~~
                       Final words,
                  ~Author~~~~~~~~~
                       Misc Stuff and
                  ~Thanks~~~~~~~~~
                       cie.
                  ~History~~~~~~~~
```

~Future~~~~~~~~~~

~Other~Programs~

~Note~

1.2 About XPKatana

One nice thing about the Amiga is those Amiga-standards established by various programmers. XPK is one of them: it allows the end-user to use an external packer which suits his system ressources and needs, and it allows the application programmer to easily add file compression capacities to his file loading/saving routines.

But one thing XPK was missing was a COMPLETE interface/server. Oh for sure, there are things like XDrop for the Workbench user, or PackX for the CLI fanatic. But how about those in-between who wants the power of the Shell interface along with the ease of use of the XDrop AppIcon? Or how about those who wants to be able to pack/unpack files from other applications but doesn't want to use a system patch? Here comes XPKatana.

XPKatana is a program with a complete GUI which will allow you to easily pack and unpack files using the XPK system. It also has a complete ARexx port, allowing you to interface it with other applications.

1.3 Features

XPKatana features:

\textdegree{} Supports about every XPK packer and features such as encryption, multi-mode packers, etc...

\textdegree{} Very complete and easy to use GUI, which doesn't need any external library.

\textdegree{} Helps you to decide which packer to use by analyzing the source's file type (thanks to
FileID.library
!)

\textdegree{} Complete ARexx port, allowing you to interface XPK with any other application which has an ARexx port, providing XPK support to them.

\textdegree{} Allows for automated batch packing/unpacking via ARexx scripts.

\textdegree{} Script Record mode, allowing you to record your actions to a ↩ script
for easy script generation.

\textdegree{} Special "Test Mode": allow you to try a selected list of packers ↩ on

```
    a given file to determine which one is the best.  Also able to
    generate a report in a text file about the packing figures.

  \textdegree{} ...And still more!
```

1.4 Requirements

```
                To work, XPKatana requires:

    \textdegree{} Any Amiga model with at least 512 Kb memory (depending on the ↩
       used
      packers).
    \textdegree{} Kickstart 2.04 or better (with asl.library and diskfont. ↩
       library).
    \textdegree{} xpkmaster.library (any version should work), and at least
      one packer installed.  I haven't included it in the archive
      because including the whole XPK archive would make the package too
      large.  Look for xpk25usr.lha (currently the latest release of
      XPK).  It should be available on most local Amiga BBSes, and on
      Aminet (for those with Internet access).


  Also STRONGLY recommended:

    \textdegree{} ARexx (shouldn't be a problem, it's part of Kickstart 2.04 and ↩
       up.)
    \textdegree{}
              FileID.library
               by Oliver Lange (for file type identification,
     included in the archive, but still optional for those with a
     floppy-based system, who can't afford the disk space).
    \textdegree{} A hard disk.
    \textdegree{} At least 1 Meg of memory.
    \textdegree{} At least a 68020.  Not for XPKatana itself, but for the packers ↩
       .
      File compression will benefit a LOT from a simple 68020.
```

1.5 Installation

```
   In the Install/ drawer you'll find various installation scripts for
XPKatana.  Note that these requires the Commodore Installer, which is NOT
included in the archive.  The Installer should be found on your Workbench
disks if you have Workbench 2.1 or better.

   If you don't have Installer, you can easily install XPKatana manually:

1) Copy the main executable (XPKatana) in the directory of your choice
   (example: Utilities/ on your Workbench).

2) Copy the FileID.library in the Libs: directory of your Workbench if you
   want XPKatana to be able to identify the submitted files, giving you a
   better idea of the packer to use to pack the given file.  This library
```

```
   is optional, so you can skip that step if you are short in space on
   your Workbench.

3) Copy the ARexx/ scripts in the directory where you usually keep the
   other scripts (usually where the REXX: assign points, or the s/
   directory on your Workbench.  Included scripts are:

   Examine_kat.rexx
   Pack_kat.rexx
   Unpack_kat.rexx

   These one allows you to examine, pack or unpack a given file from the
Shell with XPKatana running in background.  Read them to know how to use
them.

   FW-Pack_kat.rexx
   FW-Unpack_kat.rexx

   These two scripts allows the Final Writer 4 users (from Softwood) to add
XPK support to files created by this excellent word processor.  Note that
these scripts should be easily adaptable for other products from Softwood.

   DO5-Pack_kat.rexx
   DO5-Unpack_kat.rexx

   These two scripts adds XPK support to Directory Opus 5 (from
GP Software).


   These ARexx scripts are optional, so you can skip that step if you don't
need them.
```

1.6 Workbench Usage

```
   XPKatana can be launched from Workbench.  Here's a list of supported
tooltypes:


   DEFAULTDIR: This is the default directory that will be used in the
               file requester when selecting source.  Defaults to the
               current directory.

   FONT      : This is the name of the font you wish to use for the GUI, in
               case XPKatana is having problems with your system font and
               the fallback to topaz.font doesn't suit you.  Don't forget
               the ".font" suffix.

   FONTSIZE  : This is the font size you wish to use for the GUI.  Note
               that you don't have to specify both FONT and FONTSIZE: if
               only one is specified, the other one will use the settings
               in your "Font" Workbench preferences.

   ICONIFY   : Tells XPKatana to start in an iconified state.

   PUBSCREEN : Tells XPKatana on which public screen it should open the
```

GUI. Defaults to Workbench.

TEMPDIR : Tells XPKatana the path where the temporary files should be created while processing. Defaults to "T:" but could be relocated elsewhere on your HD if you don't have enough free memory.

1.7 Shell Usage

When launched from the Shell, XPKatana supports the following arguments:

DD=
DEFAULTDIR
FN=
FONT
FS=
FONTSIZE
I=
ICONIFY
P=
PUBSCREEN
TD=
TEMPDIR
You can either use the shortcut or the complete argument name when specifying arguments.

These works exactly like their tooltypes counterparts.

Note that when launched from the Shell, XPKatana will still use the tooltype settings. BUT, any specified Shell argument will OVERRIDE the corresponding tooltype. So, if you have in your tooltypes "DIRWORK" as the pubscreen, specifying "P Workbench" from the Shell will override it, and open XPkatana on the Workbench.

1.8 Main Window

The main window is separated into three areas.

1) The Information Area. Located in the upper left corner of the window, this large area holds a button labeled "Source" next to a string gadget. You can select the source file by either entering it directly into the string gadget, or by clicking on the "Source" button, which will open a file requester. You can also drag icons and drop them in the window to have them selected as the source.

Below that line are various informations concerning the current source file:

Filetype: This area tells you what type of file is currently

selected as the source, according to FileID.library.

Size: This is the size in bytes of the source file.

Packer: If the file is packed, displays the packer used if it's a standard XPK packer, or "alien" if it's a non-XPK packer. If the file is also encrypted, an "*" will be added to the packer name.

Ratio: Displays the compression ratio of the file (if it's an XPK-based packer).

Unpack: Displays the size of the source file when unpacked (only for XPK-based packers).

Status: Displays the current status of XPKatana.

2) The Packer Area. Located to the upper right, this is where you can select which packer you wish to use. Just below the list of available packers is a button labeled

Set...

. Click on this gadget to set some settings for the current packer (such as the chunk size, password if you're en/decrypting an encrypted file, and also select the packing mode to use.

3) The Actions Area. This row of gadgets at the bottom of the window holds three gadgets:

Process: Click on this gadget to process the current source file using the Operation selected to the right. The source file will be overwritten with the new file with the same filename, unless you have "

Handle~Suffix

" selected, in which case the .xpk suffix will be added or removed, depending if you're packing or unpacking the file.

Process As...: This is the same as "Process", except that a file requester will be opened, asking you to select the destination directory and filename. The source file won't be affected, unless you have "

Delete~Source

"

selected in the preferences.

Operation: This cycle gadget lets you select which type of operation you wish to perform on the source file. Can be:

\textdegree{} Pack - Will compress the file using the current ↩ packer.

\textdegree{} Unpack - Will unpack the file with the required ↩ packer,

no need for you to select which packer to use.

```
\textdegree{}
 Test
  - Test a given selection of packers on the current
        source file.
```

1.9 Test Packers

When processing in "Test Mode", a new window will be opened, with two listviews. The left list holds all the available packers on your system, while the right list holds the selected packers. Select a packer on the left list by double-clicking on it, or remove it by double-clicking on it in the right list.

On the right side, a column of buttons are available:

About: Display informations about the last packer you clicked on in any of the listviews (that packer's name is shown just below the "About" button).

All/Clear: Select all packers for testing, or clear the selected list.

Test: Launch the test procedure.

Abort: Abort the whole thing (clicking on the window's close gadget has the same effect).

At the bottom of the window there's a "Output..." button. If you want XPKatana to generate a complete report of the results in a text file, click there to select the filename for the output file where the generated report will be written.

When you click on "Test", XPKatana will try each of the selected packers on the source file. When finished, it will tell you which packer gave the best compression ratio, with the ratio attained. Then, you can select the "Pack" operation, select that packer in the packer list, and process the file as usual.

Note that while in Test mode, the source file WON'T be modified. This mode will only let you "test" a given selection of packers on a file, without affecting that file.

1.10 Set Packer Window

This window is separated into three main parts:

1) The packer informations. In this area will be displayed a short description of the current packer (the name is displayed as the window title), with informations about the packer's supported options:

The maximum supported chunk size.

The default chunk size.

Encryption: Does this packer allows encryption (and the use of a password, by the same time)?

Lossy: If this packer does lossy compression, which would make it useless for files which needs to preserve their EXACT content, like a text file or a program. Could be acceptable for files such as a RAW sound sample, when losing some quality is okay to you.

2) The packer settings.

Chunk Size: The size (in bytes) of each blocks of data while packing. It also determines how often the progress window will be updated while packing a file. Adjusting this might be usefull if you're short in memory, or if you want to fine tune the packing efficiency.

IMPORTANT: The Chunk size will be resetted to the default each time a new packer is selected! This is done to avoid problems since each packers have different limits for this value.

Password: If you want to encrypt or decrypt a file, you must supply a password. Note that you will also have to enable the "Use Password" flag in the Prefs~menu .

3) The packer mode. Many packers support more than one packing mode. This is where you will adjust the desired packing mode, depending on the ratio vs speed issue desired. The slider allows you to select the desired packing mode, between 0 and 100, 100 giving the best compression but the slowest (un)packing.

Note that the packmodes are usualy implemented as "steps". Example:

A given packer can give you a fast but loose compression from 0-25, a cheap compression from 26-50, average compression from 51-75, and the best compression from 76-100. So any value between each boundaries of a given mode will give you the same result. Check the mode name just above the slider to determine the current mode.

Just below there is some average specs about the current mode, in term of speed and memory usage for both packing and unpacking.

Clicking on the close gadget will close the window, and reopen the main window.

1.11 The Progress Window

While processing, a progress window will be opened, showing you various
informations about the current processing.

Note that if the
No~Progress
flag is enabled, the Progress Window won't be
opened. If the main window is opened, minimal informations will be shown in
the status gadget about the processing's progress.

In the progress window, you'll see:

\textdegree{} The filename being processed
\textdegree{} The processing type (and the destination filename)

\textdegree{} The packer's name

\textdegree{} The packer's efficiency - 0% = No compression, 100% = whole file
packed into a single byte :)
\textdegree{} How much of the processing has been completed yet.

\textdegree{} The source's length in bytes
\textdegree{} How many bytes have been read from the source yet
\textdegree{} How many bytes have been written to the destination yet
\textdegree{} The Bytes Per Second (bps) speed ratio

This progress window will get updated between each data chunks, as setted
in the
Set~Packer
window.

During processing, you can click on the "Abort" gadget to stop processing
(note that due to a limitation in XPK itself, processing can only been
aborted between each chunks. So, it may take a few seconds before your
Abort click gets acknowledged.)

When processing is completed, click on the window's close gadget to close
it.

1.12 Menus

Attached to the main window of XPKatana, you'll find three menus:

~Project~
- These are general functions like Iconify, Delete File, etc...

~Prefs~
- These are various configurable options.

~ARexx~
- Allow to execute a script or to record a new one. Will be disabled if you don't have ARexx installed.

1.13 Menu - Project

About

Displays the "About" requester.

Iconify

Will close the main window, and put an AppIcon on the Workbench. Files dropped on it will get processed accordingly to the settings setted before you iconified (or as changed via ARexx while iconified). Double-clicking on the icon will bring back the main window.

Delete File...

Will open a file requester, asking you to select a file which you wish to delete from the disk. The icon will ALSO get deleted, if there's one.

Quit

Close down XPKatana and exits.

1.14 Menu - Preferences

Task Priority

Allows you to select between three task priority for the (un)packing subprocess:

-1 for background processing (won't slow down other tasks)
0 for regular processing
1 for foreground processing, giving more CPU time to this task

These 8 items are various flags which can be turned on or off:

Overwrite

If you want XPKatana to quietly overwrite files without asking you first.

Stepdown

If you don't have enough memory for the selected packing mode,

XPK will step down to a lower packmode (when allowed by the packer).

Copy Icon

If you want XPKatana to copy the source's icon when saving a processed file with a different filename (Process As...)

Allow Lossy

If you want to allow the use of a lossy packer (shouldn't be activated unless you KNOW what you're doing!)

Use Password

If you want the packer to protect the file against unauthorized unpacking by using the specified password (the selected packer must support encryption).

Handle Suffix

If you want XPKatana to add an .xpk to the filename of files it packs, and remove it from files he unpacked, if there's one.

Delete Source

If you want XPKatana to delete the source file when saving under a different filename (Process As...)

No Progress

If you don't want XPKatana to open the progress window, in which case the process's status will be shown in the status gadget of the main window (if it is opened). Recommended for script operation, as an example.

Load Prefs

Will load the preferences from disk.

Save Prefs

Will save the current preferences to disk: current packer & packmode, password, and the state of the above flags.

1.15 Menu - ARexx

Record Script...

Will record an ARexx script to disk. These scripts allows you to automate some repetitive process, or create a batch file to

process several files in one shot without the need of having the user to watch the whole process.

XPkatana will first ask you under which name you wish to save it. By default, the script will be created in the T: directory. This is where you would create temporary scripts which are going to be generated just for a single batch processing, while scripts you wish to keep should be created somewhere on disk instead (like in REXX:). It will then ask you if you want to save the current settings in your script, like the current packer, the state of the current options, etc...

Then, everything you will do will be logged to the script for later playback. That is, rather than pack a file on "Process", it will write the equivalent ARexx command to the script (that is, "

PACK

"). Changed flags will be noted to the script to. And so on.

Note that while in Script Record mode, you can select more than one source file. The selected files will be put in some file queue, waiting for you to press "Process", where the "

SETSOURCE

/ (UN)PACK" commands will be written to the script according to the queued filenames, and the queue will get emptied. More files may be added to the file queue by clicking on the "Source..." button more than once (if you wish to select files from more than one directory, as an example.) Usefull if you want to do a batch processing.

Stop Recording

Will stop the recording of an ARexx script, closing it and returning to normal operation.

Execute Script...

Allow you to launch an ARexx script. Note than only ONE ARexx script can be launched at a time from XPKatana! Also, you can't record a script while in Execute mode, and vice-versa.

1.16 ARexx interface

XPKatana has an ARexx port named "KATANA", and has about a ↩ dozen of
implemented commands, allowing you to use XPKatana from other applications, or automate some of its functions.

To access XPKatana from an ARexx script, you must tell your script to address the 'KATANA' port. This is done like this:

```
Address 'KATANA'
```

You must also tell your script to use "RESULT" to obtain informations from

```
the function:

  Options Results


    Commands:


                ~EXAMINE~~~~~~

                ~PACK~~~~~~~~~~

                ~SETSOURCE~~

                ~GETFLAG~~~~~~

                ~QUIT~~~~~~~~~~

                ~SETTASKPRI~
                ~

                ~GETLIBINFO~~~

                ~SETCHUNKSIZE~~

                ~SETPACKER~~

                ~GETMODEINFO~~

                ~SETFLAGS~~~~~~

                ~UNICONIFY~~

                ~ICONIFY~~~~~~

                ~SETPASSWORD~~~

                ~UNPACK~~~~~
                ~~

See the supplied example scripts in the Rexx/ drawer, and your ARexx manual
for more informations on how to write ARexx scripts.
```

1.17 ARexx - EXAMINE

```
Description:

   Examine the current source file.

Template:

   EXAMINE

Input:
```

```
   None.

Result:

   Returns a string with the following format:

      "FILE PACKER ENCRYPTED? RATIO DESCRIPTION"

   FILE         is the source filename.
   PACKER       is a four-letters packer, or #NONE if it isn't packed.
   ENCRYPTED?   will be YES or NO, depending if the source file needs
                a password to be unpacked.
   RATIO        is the compression ratio (if applicable).
   DESCRIPTION  holds the file type as determined by FileID.library, if
                installed.

   If the source file couldn't be found, "NOFILE" will be returned.
   If another error occured, "ERROR" will be returned.
```

1.18 ARexx - QUIT

```
Description:

   Shutdown XPKatana.

Template:

   QUIT

Input:

   None.

Result:

   Around 75 Kb of free memory :)
```

1.19 ARexx - ICONIFY

```
                Description:

   XPKatana will close its window and put an
                AppIcon
                 on the Workbench.

Template:

   ICONIFY

Input:

   None.
```

Result:

None.

1.20 ARexx - UNICONIFY

Description:

If XPKatana was in an iconify state, will remove the appicon and re-open the main window.

Template:

UNICONIFY

Result:

None.

1.21 ARexx - SETSOURCE

Description:

Select the source file.

Template:

SETSOURCE [source]

Input:

SOURCE is the source file. If you use "?" as the source, a file requester will be opened, asking for a source file.

Result:

ERROR if no argument was supplied.
NOFILE if the selected file couldn't be found (note that XPKatana will still keep that filename as the source).

1.22 ARexx - SETTASKPRI

Description:

Select the task priority of the packing sub-process.

Template:

SETTASKPRI [priority]

```
Input:

   PRIORITY is one of the following keywords:

        LOW   - Task priority of -1 (in background)
        NORM  - Task priority of 0 (same as a normal process)
        HIGH  - Task priority of 1 (gives more priority to it)

   Usualy, leaving this to "NORM" should be enough for most needs.

Result:

   Returns ERROR if an invalid priority has been supplied, in which case the
priority will remain unchanged.
```

1.23 ARexx - SETFLAGS

```
Description:

   Set various options of XPkatana.

Template:

   SETFLAGS [option_1] [state] [option_2] [state] ... [option_x] [state]

Input:

   Any number of these options can be used:

   ALLOWLOSSY  - Allow lossy packers.
   COPYICON    - Will copy the source's icon with the destination.
   DELSOURCE   - Will delete the source file after processing.
   NOPROGRESS  - Won't open the progress window while processing.
   OVERWRITE   - Will quietly overwrite destination if it already exists.
   STEPDOWN    - Allow the packer to do a mode step-down when needed.
   SUFFIX      - Will take care of the .xpk suffix, adding/removing it
                 to the destination file as appropriate.
   USEPASSWORD - Enables password protection, if the packer supports it.

  Each options must be followed by a state flag:

     0 - Disabled.
     1 - Enabled.

Example:

     SETFLAGS USEPASSWORD 0 SUFFIX 1 DELSOURCE 1

Result:

  None.
```

1.24 ARexx - PACK

```
Description:

   Compress a file.

Template:

   PACK [destination]

Input:

   DESTINATION: An optional destination filename.  If omitted, the
                destination will be the same as the source, with a ".xpk"
                suffix added if the "Handle Suffix" option is enabled.

                If "?" is supplied as the filename, a filerequester will
                be opened, asking for the destination filename.

Result:

   Returns "ABORT" if the packing was aborted by the user or an error.
```

1.25 ARexx - UNPACK

```
Description:

   Uncompress a file.

Template:

   UNPACK [destination]

Input:

   DESTINATION: An optional destination filename.  If omitted, the
                destination will be the same as the source, with the
                ".xpk" suffix removed if there's one and the
                "Handle Suffix" option is enabled.

                If "?" is supplied as the destination, a filerequester
                will be opened, asking for a destination filename.

Result:

   Returns "ABORT" if the unpacking was aborted by the user or an error,
   or if the file wasn't packed with any known packer.
```

1.26 ARexx - GETFLAG

```
Description:
```

```
   Get the current state of the specified option.  Usefull if you are about
to change the state of some options, but want to be able to restore them to
their initial values afterward.

Template:

   GETFLAG [option]

Input:

   Option is one of these:

   ALLOWLOSSY  - Allow lossy packers.
   COPYICON    - Will copy the source's icon with the destination.
   DELSOURCE   - Will delete the source file after processing.
   NOPROGRESS  - Won't open the progress window while processing.
   OVERWRITE   - Will quietly overwrite destination if it already exists.
   STEPDOWN    - Allow the packer to do a mode step-down when needed.
   SUFFIX      - Will take care of the .xpk suffix, adding/removing it
                 to the destination file as appropriate.
   USEPASSWORD - Enables password protection, if the packer supports it.

Result:

   Will return '0' if the option is disabled, '1' if it's enabled, and
'ERROR' if the option is unknown.
```

1.27 ARexx - GETLIBINFO

```
Description:

   Return informations on the current packer.

Template:

   GETLIBINFO [info]

Input:

   INFO is one of the following values:

      1 - The packer's short name.
      2 - The packer's complete name.
      3 - A one line description of the packer.
      4 - Returns a formatted string, with some features about
          the current packer.  The string has the following format:

         "ENCRYPT <flag> NEEDPASSWORD <flag> MODES <flag> LOSSY <flag>"

            ENCRYPT:      Does this packer supports encryption?
            NEEDPASSWORD: Does this packer REQUIRES a password?
            MODES:        Does this packer supports more than one packing
                          mode?
            LOSSY:        Does this packer do lossy compression?
```

```
            <Flag> is either "YES" or "NO".

      5 - The largest chunk size for input (in bytes).
      6 - The default chunk size for input (in bytes).
      7 - The default packing mode (between 0 and 100).

Result:

   Returns a string according to the supplied argument, or "ERROR" if the
   argument was invalid.
```

1.28 ARexx - GETMODEINFO

```
Description:

   Return informations on the current packing mode.

Template:

   GETMODEINFO [info]

Input:

   INFO is one of the following values:

        1 - The current packmode (between 0 and 100)
        2 - The description of the current mode.
        3 - Memory needed for packing (usualy in bytes).
        4 - Memory needed for unpacking (usualy in bytes).
        5 - Average packing speed in Kb/s (usualy on an A3000/030)
        6 - Average unpacking speed in Kb/s (usualy on an A3000/030)
        7 - Average compression ratio (on an AmigaVision data file).
        8 - Desired chunk size (in Kb) for this mode.

Result:

   Returns a string according to the supplied argument, or "ERROR" if the
   argument was invalid.
```

1.29 ARexx - SETPASSWORD

```
Description:

   Select a password for (un)packing encrypted files.

Template:

   SETPASSWORD [password]

Input:

   The password to use for (un)packing encrypted files.
```

```
Result:

   ERROR if no password has been specified.
```

1.30 ARexx - SETCHUNKSIZE

```
Description:

   Set the file's packing chunk size.

   IMPORTANT: The chunk size will get resetted to the packer's default
              size each time you select a new packer!  So you must use
              this function AFTER selecting the packer (via SETPACKER or
              the GUI).

Template:

   SETCHUNKSIZE [chunksize]

Input:

   CHUNKSIZE is the size of each blocks of data in the packed file, given
             in bytes.  It will also determine how often the progress
             window will be updated (that is, each time a chunk has been
             (un)packed).  The chunk size must be between 10 and the
             packer's max chunk size.

Result:

   Will return "ERROR" if no value was supplied.  If the value doesn't fit
between 10 and "max chunk size", it will get adjusted to fit in these
limits.
```

1.31 ARexx - SETPACKER

```
Description:

   Set the packer and packing mode.

Template:

   SETPACKER [packer] [packmode]

Input:

  PACKER is the 4-letters XPK packer to use (like "NUKE").
  PACKMODE is the packing mode, between 0 and 100.

Result:

   Returns "OK", or "BADMODE" if the packing mode is outside of the 0-100
```

range, "BADPACKER" if the specified packer isn't present on the system.

1.32 Legal Stuff

XPKatana is released as Shareware. If you're using this product and like it, then I ask you to send me a fee of 15$ US (or 20$ CDN). I will also accept things like a registered version for another program or old Amiga games/applications you're no longer using as an alternative.

As you can see, the program isn't crippled in any way. You won't receive a keyfile in exchange for the payment. This is because I don't believe in crippleware. I HATE it when I download a program only to find it to be so crippled that it's of no use to me. So, why should you register? Simple. The Amiga's future relies heavily on Freeware and Shareware developpers, because many of the largest commercial developpers seem to ignore the Amiga. Many Shareware developpers ends up being hired by Amiga dedicated cies, giving you even better softwares. So, supporting us is an excellent way of concretely supporting YOUR Amiga. Also I usually re-invest shareware fees I receive by registering other sharewares or by buying Amiga hardware or commercial softwares. So, it makes it twice a good investment for the Amiga community :)

You cannot resell this program for profit, but inclusion on CD-ROMs, PD collections or coverdisks is allowed. The only thing I ask is that if you include it on some magazine coverdisk, that you send me a free issue of the magazine with the coverdisk. It's always a kick to see my name in a magazine <blush> :)

FileID.library
is written by Oliver Lange, and is Public Domain.

Xpkmaster.library and the associated packers are written by Urban Mueller and various other authors, and are Freeware.

I'm am not responsible for any data loss, or any kind of problem encountered while using this package. You're using it at your own risks, so if you find a bug that I didn't noticed before release, if your computer starts blowing smoke or your girlfriend slaps you and dumps you for being a morron, it's not my fault. If you do find a bug, please report it so I can fix it in a future release.

1.33 Author

I can be reached at one of these addresses:

Fidonet: Eric Sauvageau @ 1:242/907.0
(Freq for 'KATANA' for the latest version.)

Internet: dream@step.polymtl.ca (this is a friend's account. Specify in your message that it is addressed to me).

IRC nick: RMerlin.

Snail-Mail: Eric Sauvageau
5336 10th Avenue
Montreal, QC
CANADA
H1Y-2G6

1.34 Thanks

I want to thank the following persons who had some part in XPkatana:

Wouter Van Oortmessen : For the Amiga E package (which includes EasyGUI, used to generate the GUI.)

Urban Mueller : For the XPK standard, and Aminet.

Oliver Lange : For the FileID.library, used for file identification.

Phil Vedovatti : For the artwork (the About logo and the icons), the Installer script and beta-testing.

Georges Goncalves : For the portuguese translation, beta testing and the Directory Opus scripts. (NON y'aura pas de version MUI! ;)

Petter Nilsen : For the norsk translation of the install script.
Claudio Di Martino : For the italian translation of the install script.
Volker Schleifstein : For the german translation of the install script.

Rémi Létourneau: The Final Writer scripts and beta testing.

Misc. thanks goes to: Pepsi-Cola (for the liquid fuel)
Iron Maiden (for the audio fuel)
Sony (for the audio fuel feeder)
Amiga Technologies (for keeping our baby alive)
Commodore's ex-engineers (for my Amiga 1200!)

And the registered users:

<If you want to see your name here in the next version,
click~here
to

learn what you must do for it ;)>

XPKatana was written in E, and compiled with EC 3.2e.

1.35 History

1.0 (08-Jan-96)

- First public release.

"He's acting like a God - an ancient lunatic
Your mission - terminate with extreme prejudice."

-Iron Maiden (The Edge of Darkness)

1.36 The Future

This is a partial listing of what might be added in future versions,
depending on my time and the user's feedback:

\textdegree{} Support for alien packer decrunching (already half done, via xfdmaster)
\textdegree{} Localization (I'm still looking for good E example on localization :( )
\textdegree{} Ability to open its own public screen.
\textdegree{} Automatic mode selection while in iconify mode (select pack/unpack accordingly to the dropped file).
\textdegree{} Optimizations: I'd like to see the program shrink a bit in size...
\textdegree{} TEST, SETPUBSCREEN and SETTEMPDIR ARexx commands.
\textdegree{} XPKnife, a lite version for those with limited disk space (floppy systems, as an example). Same as XPKatana, but without goodies like ARexx port, Test mode, AppIcon, etc...
\textdegree{} ...Any suggestions? Any ARexx commands which might be usefull?

Tell~me!
Things which won't get implemented, for various reasons:

\textdegree{} GUI Hotkeys (at least, not before EasyGUI supports them :( )
\textdegree{} Support for non-XPK packers like powerpacker, crm, lh, etc... This is XPKatana, not PowerKatana ;)
\textdegree{} Complete Shell interface (Altough the supplied ARexx scripts allow Shell usage of XPKatana while it is running)

1.37 Other Programs I Wrote

```
  Other programs that I wrote:

        \textdegree{} DevsMan 1.4 - Devs: directory manager, allow easy handling
                        for your DOSDrivers, Datatypes, etc...

        \textdegree{} FileScroller 3.40 - File lister for TransAmiga BBS (3.50 and ↩
            up
                              for Excelsior! BBS.)

        \textdegree{} MFormat 1.8a - Replacement for CBM's "Format" command.  Has
                         a complete GUI, configurable device filter, can
                         install a bootable bootblock, etc...

        \textdegree{} TDPrefs 1.0 - Preferences editor for trackdisk.device, can
                        adjust the step rate, disable the drive click,
                        etc...
```

1.38 About FileID.library

FileID.library is a PD shared library for identifying files by checking it's contents. I decided to add support for some file identifier library because it can help you to select a packer if you know what kind of file you're going to pack. It also allow you to detect if a file has already been packed by an alien cruncher such as CrunchMania, in which case it's doubtfull that XPKatana will be able to further crunch it. The supplied version (7.0) is able to recognize more than 591 different filetypes.

Under Workbench 2.1 and up, this library supports localization. Although XPkatana itself doesn't, FileID.library will still use it when displaying the filetype description. In the archive I enclosed the German catalog, English being built in the library.

This library has been written by Oliver Lange (Bloodrock) of Syndicate. He can be reached at these addresses:

InterNet: Bloodrock@funboard.in-berlin.de

Snail-Mail:

```
   Oliver Lange
   Bartastr. 9
   D-12055 Berlin
   (Germany)
```

1.39 Note

```
              This document is AmigaGuide V39 enhanced ;)

  (Netscape ain't the only one who "benefits" from "enchancements" <grin>
```

```
There's a secret message hidden in XPKatana :)
Hint: 'nastybug' is it's source...
```