

# **Default**

Jens T. Thielemann

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i> Default		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Jens T. Thielemann	July 22, 2024	

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>Default</b>	<b>1</b>
1.1	SignArch v1.1 . . . . .	1
1.2	Introduction . . . . .	1
1.3	Requirements . . . . .	3
1.4	Disclaimer . . . . .	3
1.5	Distribution . . . . .	3
1.6	Installation . . . . .	4
1.7	Usage of shell commands . . . . .	4
1.8	The commandfile . . . . .	9
1.9	Creating the commandfile . . . . .	9
1.10	SUMFILE . . . . .	11
1.11	ARCHIVE . . . . .	11
1.12	ADDARMOR . . . . .	11
1.13	ADDCOMM . . . . .	11
1.14	ARCLIVE . . . . .	12
1.15	ASCCHAR . . . . .	12
1.16	AUTOCHK . . . . .	12
1.17	BANNERS . . . . .	13
1.18	BINCHAR . . . . .	13
1.19	DEEPCHK . . . . .	13
1.20	ENCRYPT . . . . .	14
1.21	ENDWAIT . . . . .	14
1.22	EXECCMD . . . . .	14
1.23	INCTEXT . . . . .	14
1.24	LASTOPT . . . . .	15
1.25	NOKEYADD . . . . .	15
1.26	NOWILDS . . . . .	15
1.27	ROOTDIR . . . . .	15
1.28	SIGNAME . . . . .	16
1.29	SUMICON . . . . .	16

1.30 SUMONLY . . . . .	16
1.31 WAITCOM . . . . .	17
1.32 WAITRET . . . . .	17
1.33 Notes . . . . .	17
1.34 Bugs . . . . .	20
1.35 Copyrights . . . . .	20
1.36 Thanks . . . . .	21
1.37 Address . . . . .	21
1.38 E-mail address . . . . .	22
1.39 Usage of GUI . . . . .	22
1.40 File Control Part . . . . .	23
1.41 Root directory . . . . .	24
1.42 Archive . . . . .	24
1.43 Checksum-file . . . . .	24
1.44 Include textfile . . . . .	24
1.45 Filelist . . . . .	25
1.46 Add files . . . . .	25
1.47 Rem. Files . . . . .	25
1.48 Sort (files) . . . . .	25
1.49 Encryption Control Part . . . . .	25
1.50 Encrypt archive . . . . .	26
1.51 Wipe nonencrypted . . . . .	26
1.52 Recipient list . . . . .	26
1.53 New key . . . . .	26
1.54 Remove . . . . .	26
1.55 Sort . . . . .	27
1.56 Prefs Part . . . . .	27
1.57 Checksum .info . . . . .	27
1.58 Use ASCII armor . . . . .	27
1.59 ASCII deep check . . . . .	28
1.60 Add PGP public key . . . . .	28
1.61 Add script commands . . . . .	28
1.62 Only wait at end . . . . .	29
1.63 Wait command . . . . .	29
1.64 Signature ID . . . . .	29
1.65 Overview of menu items . . . . .	30
1.66 Project->Open... . . . .	30
1.67 Project->Save... . . . .	31
1.68 Project->Save as... . . . .	31

---

1.69	Project->Global Settings . . . . .	31
1.70	Project->Create archive . . . . .	32
1.71	Project->About . . . . .	32
1.72	Project->Quit . . . . .	32
1.73	Settings->Prefs->Load... . . . .	33
1.74	Settings->Prefs->Save... . . . .	33
1.75	Settings->Prefs->Clear... . . . .	33
1.76	Settings->Encryption->Load... . . . .	33
1.77	Settings->Encryption->Merge... . . . .	34
1.78	Settings->Encryption->Save... . . . .	34
1.79	Settings->Encryption->Clear... . . . .	34
1.80	Settings->Files->Load... . . . .	34
1.81	Settings->Files->Merge... . . . .	35
1.82	Settings->Files->Save... . . . .	35
1.83	Settings->Files->Clear... . . . .	35
1.84	PGP key-select window . . . . .	35
1.85	Use key . . . . .	36
1.86	Search . . . . .	36
1.87	Search next . . . . .	36
1.88	Update . . . . .	36
1.89	PGP . . . . .	36
1.90	Index . . . . .	37

# Chapter 1

## Default

### 1.1 SignArch v1.1

SignArch v1.1

Message digests a list of files, and archives everything

or

A Graphical User Interface for LhA

or

YAPS (Yet Another PGP Simplifier)

Distributed under the GNU General Public License

Introduction	-	What is this?
Requirements	-	What do I need to use this?
Disclaimer	-	IMPORTANT to read!
Distribution	-	How may I distribute this?
Installation	-	How to install everything
Usage of GUI	-	How to use the graphical user interface
Usage of Shell commands	-	How to use the Shell programs
Creating the commandfile	-	How to create a commandfile by hand
Notes	-	Misc. info worth reading - kind of FAQ
Bugs	-	No...there can't be...:)
Thanks	-	People I like
Copyrights	-	Legal mush
Address	-	How to contact the author
Index	-	Table of contents

### 1.2 Introduction

When releasing a software package, nothing is more irritating than when you discover that some idiot has modified your code, added some banners to be printed, or even worse, added a virus. The solution is of course to create message digests and signatures with MD5SUM and PGP, thus making it impossible to pass a modified file unnoticed, but this is usually a quite big and boring task. Not any more.

All you have to do is to select the files to be included in the

archive using a filerequester, point'n'click according to whether you wish encryption, etc. Then, just lean back and watch the show.

Still, if you hate GUI's, all you have to do is to add T W O lines to the file containing the files that are to be added, which you normally would pass to LhA, and pass it to this program instead. A file with MD5SUM signatures, signed with PGP, will then be created, and archived with the remaining files.

Features include:

- Full featured graphical user interface. Just point'n'click, and off you go! EASIER than using LhA itself!
- Full support for wildcards, even when signing!
- Optional including of script commands, making the file 'self-checking'.
- Automatic adding of user specified script commands, banners, etc.
- Optional encryption of final archive, also for multiple receivers.
- Automagically detects whether a file is ASCII or binary, and signs it accordingly. Can be overridden by user flags.
- Will by default automagically NOT sign icon files, as just changing their window position is enough to make the check fail. Can be overridden by user flags.

By the way, this package does of course include a sum-file, be suspicious if that one is missing!

Changes from 1.0

ADDED:\* The '-----BEGIN' file will now be created automagically when the AUTOCHK option is specified. It will not do nothing, like the previous release, but try to check the Sumfile to detect a bug in PGP.

\* Shell-command called much the same way as LhA (thus named ShA :), so you don't have to create command files first.

\* Graphical User Interface

\* You may now force the script to NOT add the PGP key.

FIXED:\* Would add a ".lha" postfix to the archive name, no matter whether it already had a such extension

\* Several small problems when some filenames were specified absolute, instead of relative to the ROOTDIR.

\* AUTOCHK did only function correctly when one or more ADDCOMM's where specified

\* Included a workaround for a bug in LhA, which doesn't like filenames beginning with an '-'.  
\* Lots of other, small bugs...:)

OTHER:\* Guide'd the documentation.

---

\* Tried to comment/improve the readability of the scripts.

## 1.3 Requirements

- An Amiga with Kick 2.0+.
- ARexx and PGP 2.3a.3 or higher properly installed, in the path.
- MD5SUM (included with PGP) and LhA in the path.
- reqtools.library (NOT INCLUDED - get this from somewhere else)
- rexxreqtools.library (optional, not included)

## 1.4 Disclaimer

Distribution and (non-existent) warranties for functionality as specified in the GNU General Public License, which was included in this archive (the COPYING file). The essential points have been extracted:

BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "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 THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS IN THE LICENSE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## 1.5 Distribution

As specified in the GNU General Public License version 2 or higher. NOTE WELL: You are NOT allowed to charge more than the cost of media plus a nominal copying fee. The total cost may not exceed the cost of obtaining a disk from Fred Fish.

If you use this program, I would highly appreciate if you mentioned my name in your documentation, and sent me a letter telling me that you are doing so.

---



## 1.6 Installation

Just double-click the 'InstallSignArch' icon, and follow the prompts.

If you don't have or like the C= 'Installer' utility, it's still simple. Just copy 'SignArch.rexx' to your REXX: assign/directory, and 'SignArch' to c: or somewhere else in your path. Please mind that both files have the 's'-flag set, use the Protect command to do this. In addition, you'll have to copy the 'ChkASCII' to somewhere in your path, and make sure that as well 'PGP' as 'MD5SUM' also are there.

## 1.7 Usage of shell commands

```
--+<>+==  
Rexx script invocators  
(these will archive, sign, etc.)  
--+<>+==
```

### NAME

SignArch -- signs & archives according to commandfile given

### FUNCTION

This will startup the main REXX script, and parse the commandfile given, performing the necessary signing, archiving and (eventual) encryption.

### TEMPLATE

SignArch COMMANDFILE

### COMMANDFILE

The filename of a commandfile created using the Graphical User Interface (GUI), or handcrafted using the guidelines found here.

If no COMMANDFILE is specified, it will be requested using a file-requester, if both 'reqtools.library' and 'rexxreqtools.library' is installed. If not, a shell prompt will be issued.

### RESULT

A sumfile and an archive will be created, according to the user's desires & wishes.

### NOTES

This is exactly the same as selecting the Create archive item in the Project menu, found in SignArchGUI.

Much of the same effect may be obtained by using Sha.

### BUGS

No more than those found in the main REXX script, in other words:  
None known :)

SEE ALSO

ShA, Usage of Graphical User Interface, Commandfiles and the creation of them.

--+<>+==

NAME

ShA -- LhA 'substitute', signs & archives according to template

FUNCTION

Will process the commandline given, and create a CommandFile, which will be passed to SignArch. May be considered as LhA with message digesting possibilities.

TEMPLATE

ARCHIVE, SUMFILE/A, ADDARMOR/S, ADDCOMM/K, ARCLIVE/S, AUTOCHK/S, BANNERS/K, DEEPCHK/S, ENCRYPT/K, ENDWAIT/S, INCTEXT/K, NOKEYADD/S, ROOTDIR/K, SIGNAME/K, SUMICON/S, WAITCOM/K, FILES/A/M

Although the template may seem enormous, only three parameters are needed:

ARCHIVE

Name of the archive to put files within. Must be specified, unless SUMONLY is specified.

SUMFILE/A

Name of the file to put message digests within

FILES/A/M

Files to sign/archive. Multiple files are of course allowed.

You may also specify the following keywords/switches, which are equivalents to CommandFile options:

ADDARMOR/S

Make the archive output as 7-bit ASCII, suitable for e-mail.  
(Boolean)

ADDCOMM/K

Add a user-specified command to the SUMFILE, making it a script

ARCLIVE/S

Don't delete the unencrypted archive when ENCRYPT is specified.  
(Boolean)

AUTOCHK/S

Add commands to the sumfile, making it self-checking (Boolean)

BANNERS/K

Add a user-specified line of info, copyright, etc.

DEEPCHK/S

Check whole file when determining whether it is text (Boolean)

---

**ENCRYPT/K**

Encrypt the archive to the person specified

**ENDWAIT/S**

Only issue the WAITCOM command at the end of the checking script (Boolean)

**INCTEXT/K**

Include the user-specified textfile in the SUMFILE

**NOKEYADD/S**

Don't include the PGP public key used for signing in the SUMFILE (Boolean)

**ROOTDIR/K**

The specified directory will be used as current directory when archiving

**SIGNAME/K**

The name of the PGP key to use when signing

**SUMICON/S**

Message digest .info files also (Boolean)

**SUMONLY/S**

Only create the sumfile, don't create any archive. The ARCHIVE should be omitted if this is specified.

**WAITCOM/K**

Command issued between the cryptographic commands when the user executes the SUMFILE - only relevant when AUTOCHK is specified.

Currently, the current directory when running ShA must be the same as the files to be archived are located within.

**NOTES**

Needs Kick 2.0 or higher.

"/K" after a parameter in the template, means that it is a keyword, "/S" that it is a switch (boolean), "/A" that it is required and "/M" that multiple files may be specified as args.

Pure, may be made resident.

**RESULT**

The files will be archived, message digested, signed and, if selected, the final archive will be encrypted.

**BUGS**

You may currently only specify one encryption recipient and one banner via this interface (the SignArch program supports unlimited).

**SEE ALSO**

Documentation for AmigaDOS templates

--+<>+==

#### NAME

AddSum -- add a signature file to an existing archive

#### FUNCTION

This one will unarchive an existing archive, message digest the files and sign them, and add this file to the archive. The name of the sumfile will be set to the base of the archive's name, e.g. 'ram:foobar.lha' will get a sumfile of 'foobar.sum'.

#### TEMPLATE

ARCHIVE, TMPDIR

##### ARCHIVE

Archive that is to be message digested, each file individually. If no archive is specified, a file-requester will pop up, if you use the shell version to invoke the script, and have installed RequestFile.

##### TMPDIR

By default, the archive will be unarchived to T:. If the archive is big, you may wish to unarchive them somewhere else. This may be specified here.

#### RESULT

A sumfile will be created and added.

#### NOTES

#### BUGS

You are currently set with a preset options.

#### SEE ALSO

--+<>+==

Other programs

(used by the script, may be nice for other purposes also)

--+<>+==

#### NAME

ChkASCII -- Checks whether a file is text or binary

#### TEMPLATE

FILENAME/A,WHOLEFILE/S,VERBOSE/S

##### FILENAME/A

Name of the file to determine whether is text or binary.

##### WHOLEFILE/S

Normally, if the first 1k can be classified as text, the entire file will be assumed to be text. Specifying this will force that the whole file to be checked before it is

---

considered as text.

#### VERBOSE/S

Normally, processing is 100% quiet. This gives you progress info, and results.

#### NOTES

Needs Kick 2.0 or higher.

Pure, may be made resident.

The check is performed, by looking for non-ASCII characters. Aborts at the first binary character found, most binary files will therefore be processed very quick.

Allowed 'text' bytes are the one defined in ANSI X3.64-1979, that is:

Graphic group 0:	' ' to '~' plus DEL	( 32 to 127)
Graphic group 1:	' ' (Non-breaking space) to 'ÿ'	(160 to 255)
Control group 0:	TAB, NL, FF, CR, ESC	
	0x9,0xa,0xc,0xd,0x1b	
Control group 1:	SS2, SS3, DCS, CSI, ST, OSC, PM, APC	
	0x8E,0x8F,0x90,0x9B,0x9C,0x9D,0x9E,0x9F	

Is used by the REXX script.

#### RESULT

OK	= file is ASCII
WARN	= file is binary
FAIL	= Unable to check

#### BUGS

Rather weak check.

--+<>+--

#### NAME

ChkDate -- checks that the date is set correct

#### TEMPLATE

#### NOTES

Works with all Kickstarts.

Pure, may be made resident (for whatever purpose that has).

Is used by the REXX script.

#### RESULTS

Returns WARN if system time is before 15. November 1994, OK if it is after. Thus, we can warn in most cases when the user doesn't have a realtime clock, and has forgotten to set the date correctly.

#### BUGS

---

The user should be able to push the 'checkdate' forward.

## 1.8 The commandfile

The command file is the file which partly control the script, partly give it the information it needs, such as filenames, names of PGP keys, etc.

By using the GUI, you will avoid the details of such files, it will take care of everything.

And example of a hand-crafted commandfile, can be found in 'SA.flr', which was included with this distribution.

If you wish to know the exact specification for this command file, [click here](#).

## 1.9 Creating the commandfile

Here's how to create the commandfile, step by step.

NOTE: You DON'T have to read/understand this in order to use the Graphical User Interface! This information is provided for those who wish to write their own utils utilizing the REXX script. 'AddSum.rexx' is an example of such.

First of all, please note that lines beginning with ';', are ignored, if you wish to comment your work. For an example, look at the SA.flr file, the file used to create this distribution.

1. The first line of the file MUST read '!!FILELIST!!' (excluding the quotes, of course).
2. The next lines have to begin with '\*\*\*', thus specifying them as options, with a possible parameter. Please do not use more than one space between option and parameter. There is no need to enclose the parameter with double quotes, that is '"', unless the parameter contains leading or trailing blanks. If not, the rest of the line will be taken as parameter. Neither the option nor an eventual parameter is case-sensitive.

The following options are valid:

Name	Required	Needs parameters
SUMFILE	Yes	Yes
ARCHIVE	Yes	Yes
ADDARMOR	No	No
ADDCOMM	No	Yes
ARCLIVE	No	No
ASCCHAR	No	Yes

AUTOCHK	No	No
BANNERS	No	Optional
BINCHAR	No	Yes
DEEPCHECK	No	No
ENCRYPT	No	Yes
ENDWAIT	No	No
EXECCMD	No	Yes
INTEXT	No	Yes
LASTOPT	No	No
NOKEYADD	No	No
NOWILDS	No	No
ROOTDIR	No	Yes
SIGNAME	No	Yes
SUMICON	No	No
SUMONLY	No	No
WAITCOM	No	Optional
WAITRET	No	No

3. Now the list of files should follow, all relative to the ROOTDIR parameter. Blank lines are ignored. Remember that disk information should NOT be present, e.g. skip volume names like 'dh0:'. Also, leading and trailing blanks and double quotes ('') will be stripped off. Of course, use wildcards as much as possible - you won't have to type that much then. The sumfile should NOT be included in this list, it will be added automagically.

When you are about to create the commandfile (by hand, that is), things get simpler if you type the following commandline, being in the rootdir of your distribution:

```
List >RAM:Fls LFORMAT "%f%s" ALL FILES
```

The file RAM:Fls will now contain a list over all the files, which you can edit.

4. That's all! In a nutshell:  
First line: !!FILELIST!!

The next line should contain:

```
***ROOTDIR <dir-used-as-current-dir-while-archiving>
```

Next lines MUST contain:

```
***ARCHIVE <name-of-archive>
```

```
***SUMFILE <name-of-sumfile-relative-to-rootdir>
```

If you wish to add more options, do it here.

The remaining lines should contain a list of files & wildcards, separated by a linefeed.

If you still are confused, please take a look at the included 'SA.flis' file, which is an example of practical use.

## 1.10 SUMFILE

Name : SUMFILE  
Required : Yes  
Needs parameters: Yes  
Description:

The following parameter specifies the name of the file where the message digests and sums will be placed, relative to the ROOTDIR parameter.

## 1.11 ARCHIVE

Name : ARCHIVE  
Required : Yes  
Needs parameters: Yes  
Description:

This is the name of the LhA archive the files will be put within, relative to ROOTDIR, as all filenames are.

## 1.12 ADDARMOR

Name : ADDARMOR  
Required : No  
Needs parameters: No  
Description:

This command will make sure that the final archive output ends up as 7-bit ASCII, suitable for posting via e-mail. Currently, only PGP is supported as encoder (yes, you may ASCII-fy with PGP without necessarily encrypting), UU-encoding is currently not supported.

## 1.13 ADDCOMM

Name : ADDCOMM  
Required : No  
Needs parameters: Yes  
Description:

The parameter will be added to the front of the signature of the script, enabling specifying a kind of script. If the AUTOCHK option is specified, the command will be issued before the cryptographic part. If you do not specify the AUTOCHK option, please let the final ADDCOMM parameter be 'Quit'. More than one ADDCOMM option may be specified, and will be added in the same order as they were specified.

---



## 1.14 ARCLIVE

Name : ARCLIVE  
Required : No  
Needs parameters: No  
Description:

If you specify the ENCRYPT option, the noncrypted archive will by default be deleted/wiped. Specifying ARCLIVE will leave the archive on disk.

## 1.15 ASCCHAR

Name : ASCCHAR  
Required : No  
Needs parameters: Yes  
Description:

Exactly the same as BINCHAR, although an ASCII message digesting is forced. Please read the description of BINCHAR.

## 1.16 AUTOCHK

Name : AUTOCHK  
Required : No  
Needs parameters: No  
Description:

By default, only the signatures are put in the SUMFILE file, plus a quick description of how to check the files. In some cases you may wish to make this process automatic. To make the script work properly, a '-----BEGIN' executable is included automagically with your distribution (which btw. is included with this one).

NOTE: THIS IS A SECURITY HOLE. THE '-----BEGIN' FILE MAY BE INFECTED, OR REPLACED. COMMANDS INFECTING THE SYSTEM MAY BE ADDED IN FRONT OF THE PGP SIGNED PART. IT IS INDEED DISCOURAGED THAT YOU USE THIS OPTION.

To make sure you are aware of this problem, you will be asked each time for confirmation. To avoid this, add this line to your S:User-Startup file:

```
SetEnv "SignArchOpts" "do_not_ask_for_confirmation_of_autochk"
```

The -----BEGIN file is added because doing so will:

- 1) avoid AmigaDOS message like 'Can't find command '-----BEGIN' (the ENTIRE file is interpreted as a normal AmigaDOS script!)
  - 2) check that the SUMFILE begins with '-----BEGIN PGP SIGNED MESSAGE-----' and TWO linefeeds. This is necessary to avoid a bug in PGP. Both the script and the executable should work with all kickstarts.
-

As the file has to be recreated for each run and it differs from archive to archive, the only equality is that there is one code chunk of 300 bytes, and then a data chunk which is more or less the length of the SUMFILE's name.

## 1.17 BANNERS

Name : BANNERS

Required : No

Needs parameters: Opt

Description:

The parameter will, if the AUTOCHK is not specified, just be added at the top of the SUMFILE file (of course behind eventual ADDCOMM commands). If AUTOCHK is specified, the parameter will be Echo'ed upon execution of the SUMFILE file. More than one BANNERS option may be specified, and will be added in the same order as they were specified.

## 1.18 BINCHAR

Name : BINCHAR

Required : No

Needs parameters: Yes

Description:

Usually, the detection between an ASCII text and a binary file is done automatically. In some cases, however, it might be useful to override this setting on a particular file. If you put the character you use as parameter here, in front of a file, it will be message digested in binary mode. If the filename contains wildcards, ALL files matching that pattern will be forced to binary.

Some characters are not allowed used: `;`, `/`, `:`, `\*`, `"` and ` ` (blank space).

## 1.19 DEEPCHK

Name : DEEPCHK

Required : No

Needs parameters: No

Description:

Specifying this flag makes the ASCII check run through the entire file (still aborting at first 'binary' value), to determine whether the file is ASCII or not. In some cases this may be preferable, the default is to only check the first kilobyte.

## 1.20 ENCRYPT

Name : ENCRYPT  
Required : No  
Needs parameters: Yes  
Description:

In some cases, you may wish to encrypt the final archive. Specifying this option will do so, and the parameter is the name of the recipient. If you wish to encrypt the archive to multiple recipients, specify multiple ENCRYPT options, one for each name.

Please note that the extension of the archive will be changed from ".lha" to ".lzh", to fit with the internal recognition in PGP of LhArc archives. The nonencrypted archive will be wiped off the disk, unless you decide to let the archive live by specifying the ARCLIVE option.

## 1.21 ENDWAIT

Name : ENDWAIT  
Required : No  
Needs parameters: No  
Description:

This option has only effect if you specify it together with the WAITCOM and AUTOCHK wait command will ONLY be executed at the end of the script, not between each check, as default.

## 1.22 EXECCMD

Name : EXECCMD  
Required : No  
Needs parameters: Yes  
Description:

Will execute the parameter as a shell command. Might be useful in some cases, if you consider the commandfile as a 'makefile' for the archive. Please read the notes for ROOTDIR option.

## 1.23 INCTEXT

Name : INCTEXT  
Required : No  
Needs parameters: Yes  
Description:

Will include a text file with the sumfile, making the sumfile also a kind of readme file or whatever. The text will be put after eventual

---

commands added by AUTOCHK or/and ADDCOMM. The parameter is the name of the text file, and must be relative to the ROOTDIR parameter. Currently, multiple INCTEXT's are not allowed. Please read the notes for ROOTDIR opt.

## 1.24 LASTOPT

Name : LASTOPT  
Required : No  
Needs parameters: No  
Description:

This option simply tells the script that this was the last option, and that the list of files begin now. Not really needed, but just in case all your files begin with '\*\*\*'...:)

Warning: This MUST, naturally, be the very last \*\*\* option!

## 1.25 NOKEYADD

Name : NOKEYADD  
Required : No  
Needs parameters: No  
Description:

Specifying this option will DISABLE the automatic adding of your PGP public key required to check the signature. May be useful if the archive only will be distributed to people that already possess your key.

## 1.26 NOWILDS

Name : NOWILDS  
Required : No  
Needs parameters: No  
Description:

This option will turn off wildcard matching and file name expansion. Mind the latter, you MUST give the filenames in the same format as 'List~LFORMAT~%f%s' would give them! If not, strange things may happen.

## 1.27 ROOTDIR

Name : ROOTDIR  
Required : No  
Needs parameters: Yes  
Description:

---

The parameter will be used as current directory when creating the archive. If not specified, the current directory when starting the script will be used instead. You should put this as the very first line in your commandfile, to avoid trouble with for instance EXEC CMD & INCTEXT.

## 1.28 SIGNAME

Name : SIGNAME  
Required : No  
Needs parameters: Yes  
Description:

The parameter will be used as your name when talking to PGP, and this name will be used to find the key to sign everything with. It's up to you whether you choose the hex value or a part of your real name.

Please note if you don't specify this option, the MyName parameter found in pgp.config in your PGPPATH dir (if properly initialized, that is) will be used as your name when talking to PGP. If this can't be found, the first key on your secret keyring will be used.

If this information is not available, or initialized, the first key on your secret keyring will be used for signing, and its matching key on the pubring will be extracted.

## 1.29 SUMICON

Name : SUMICON  
Required : No  
Needs parameters: No  
Description:

By default, icon files are not summed, because just changing their position on the WB is enough to make the check fail. To override this, specify this option. If you wish to only override this setting on one file, specify the BINCHAR option, and prepend the filename with the BINCHAR character.

## 1.30 SUMONLY

Name : SUMONLY  
Required : No  
Needs parameters: No  
Description:

Specifying this option will just create the Sumfile, and not do any archiving, encryption, etc.

---

## 1.31 WAITCOM

Name : WAITCOM  
Required : No  
Needs parameters: Opt  
Description:

When AUTOCHK is enabled, you may wish to execute a command between each of the cryptographic commands, so that the user can evaluate the output. However, this is making the security hole even bigger. For instance, if you include the wait command with your distribution, a command will be run BEFORE it has been checked with MD5SUM. This command may have been rewritten by someone, to patch PGP and MD5SUM to not work correctly.

## 1.32 WAITRET

Name : WAITRET  
Required : No  
Needs parameters: No  
Description:

This option will, if anything goes wrong, or the script is broken, ask the user to press return. Useful the console will be closed immediately when the script finished, as the user may get a chance to see what actually went wrong.

## 1.33 Notes

When can you trust such a file of message digests? (READ THIS)

Such a file of message digests can ONLY be trusted, if the PGP public key can be verified. The entire security of the package depends on that the receiver can get your key confirmed, e.g. verified that it really is yours. PGP keys with only a few signatures, are difficult to verify the ownership of, because it isn't probable that the receiver has the keys of the people who signed your key and trusts them. A PGP key without signatures is worth close to nothing.

Therefore, you should try to collect as many signatures as possible. Also, you should include a notice in your documentation, saying that the message digests only can be trusted if your public key can be verified.

But, what exactly is a message digest?

Message digests are more or less similar to checksums, except that they can't be reverse engineered. As the designer says:

"It is conjectured that the difficulty of coming up with two

---

messages having the same message digest is on the order of  $2^{64}$  operations, and that the difficulty of coming up with any message having a given message digest is on the order of  $2^{128}$  operations. The MD5 algorithm has been carefully scrutinized for weaknesses. It is, however, a relatively new algorithm and further security analysis is of course justified, as is the case with any new proposal of this sort. The level of security provided by MD5 should be sufficient for implementing very high security hybrid digital signature schemes based on MD5 and the RSA public-key cryptosystem."

As you may understand, there's no other way known than the brute force attack to create a file with a given message digest (checksum). Thus, if one can with PGP verify that the message digests are unmodified, and the MD5SUM doesn't report any errors when checking, one can trust that the files have passed untouched. Another word for message digests is signatures.

I can't get my PGP keys displayed, the program refuses to archive everything and so on...what can I do?

If this only happens once in a while, the problem is probably only that not enough memory is available to start PGP to get the keys. Remember, PGP is a big program, and needs about 150kb free memory to start.

However, if the program never shows any keys on your keyring (assuming that there are some there :), something is probably configured wrong. Check that the following is correctly initialized:

1. PGPPATH (look in the PGP documentation)
2. Global settings

I'm fed up with the requester when I use the AUTOCHK option! I AM aware of the dangers, and do NOT wish to be disturbed. What do I do?

Read one of these sections: AUTOCHK or Add script commands, and add the line specified to your S:User-Startup.

What exactly does the '-----BEGIN' file?

To make the autochecking script work properly, a '-----BEGIN' executable is included automagically with your distribution (which btw. is included with this one).

This file will:

- 1) avoid AmigaDOS message like 'Can't find command '-----BEGIN' (the ENTIRE file is interpreted as a normal AmigaDOS script!)
- 2) check that the SUMFILE begins with '-----BEGIN PGP SIGNED MESSAGE-----' and TWO linefeeds. This is necessary to avoid a bug in PGP. Both the script and the executable should work with all kickstarts.

However, many programs do not like that program names starts with

---

one or multiple "-"'s (for instance, MD5SUM, LhA). I've tried to create workarounds for these, but future releases of MD5SUM and LhA may have corrected these problems, so the workarounds may then just cause trouble....:)

The workarounds work OK with the Sept 1993 version of MD5SUM, and v1.50 of LhA.

Can I use previously created files with all the filenames to archive?

Yes, you can, as long as each filename is separated with a return. In the GUI, if you already have a list of filenames separated with a <return>, use Settings->Files->Load... to load the files into the display. Just ignore the warning that shows up, click that you wish to continue.

If you still prefer to create the file by hand, you may of course simply include the file at the end of the file.

Why ARexx, instead of some (quicker) compilable language, like C?

First of all because it is simple, and needs no compilation. Thus, it is easy for even a novice to examine that the program does not contain any trap doors, something which may occur also if a C-source is included, as most people don't mind recompiling the programs they get.

Secondly, because it is an easy language to develop within, and is indeed supported by C=, being the only programming language directly runnable uncompiled from Shell, with no command in front.

Of course, the GUI is written in C....:)

The REXX script is far too slow! Can something be done?

It might be a good idea to run a REXX optimizer/compiler on the script - especially the option parsing will be speeded up then. This was not done with the distributed script to maintain readability.

A REXX optimizer I recommend, is REXXOpt by Proximity Softworks (Ulrich Sibiller/sibilluh@trick.informatik.uni-stuttgart.de), found in the archive REXXOPT\_1.5.LHA somewhere on AmiNet. This will strip all unneeded comments, blanks, blank lines - thus making the program about 20% shorter. This will make the REXX parser run faster.

Why isn't the source for the executables included?

As you may have noticed, the source for the GUI and the shell commands are not included. It is not because I wish to hide something, but rather because it's big (The GUI itself is 120 kilobytes! - LOTS of tabs, though...), and, for most people, not very interesting. Thus, most people are more likely to appreciate the reduced size of the archive, than the possibility to study the source.

The only source included, is the one for the '-----BEGIN' file, because it is important that people trust and understand the function

---



this file.

However, if you wish to have the source for the GUI and the remaining executables I've made, just send a letter to one of the addresses you may contact me at, and I'll happily return it.

MIND YOU: If you wish to receive the source via snail-mail, make sure to include:

- 1) A disk
- 2) A self-addressed envelope
- 3) \$2 USD if you live within Europe, \$3 if you live outside, to cover postage.

I will also NOT upload it to boards outside Oslo.

## 1.34 Bugs

Haven't found any...hopefully I've cleaned it up...:) There has been some problems with crashes upon startup, but this has apparently been caused by other programs.

There is a bug in most versions of PGP, making its clearsigning not secure at all, as one may add lines at the top without PGP noticing it. The AmigaDOS script avoids this problem partly, by not using the file itself, but the output PGP produces as source for message digests. You should also encourage the user to use the PGP output as source when checking the message digests.

I also believe that there also is a bug in amigaguide.library v39.11, because when I specify that the link should go to line 158 in a node, it jumps to line 168, and so on. It seems like this only occurs when the linenumber is above 100. When this bug is fixed, strange links may occur.

Anyway, if you have found an undocumented feature, reproducible ones are preferred. That is, you ought to enclose a detailed description of what you did, so the bug can be reproduced. If possible, please include the files used as arguments when the error occurred.

## 1.35 Copyrights

\* SignArch.rexx, SignArchGUI, ChkASCII and ShA are all copyright (c) 1994/95 Jens T. Berger Thielemann, and are distributed under the GNU General Public License. Parts of the SignArchGUI code (the opening of the windows) were created using GadToolsBox (copyright (c) 1992-93 Jaba Development (Jan van den Baard)). However, the GadToolsBox code output is (c) 1994 Jens~T.~Berger~Thielemann - still following Baard's regulations on distribution.

\*~The icons for the executables are (c) 1994/95 Sigbjørn Skjæret.

---

Some of the icons' origin is unknown, please inform if any copyright has been violated.

- \* PGP is copyright (c) 1990-1993 Philip Zimmermann, Amiga port Peter Simons. MD5SUM is (c) 1993 Branko~Lankester, Colin~Plumb and RSA Data Security, Amiga port Peter Simons.
- \* LhA is copyright Stefan Boberg
- \* reqtools.library is copyright (c) 1991-94 Nico François
- \* rexxreqtools.library is copyright (c) 1993-94 Rafael D'Halleweyn
- \* Arexx is copyright (c) 1987 William S. Hawes/Wishful Thinking Development
- \*~If any trademarks have been used, I'm aware of that they belong to someone

## 1.36 Thanks

Greetings are sent to the following people:

- ~Sigbjørn Skjæret for encouraging comments, cool phone-calls, key-signing, bug reports, some icons & suggestions! This program would certainly be delayed months if it wasn't for his support!
- Marius Mortensen for helping me out with paths when started from WB.
- Phil Zimmermann & Peter Simons for PGP.
- Jan van den Baard for GadToolsBox
- D. L. McPaul for making AmigaGuideWriter, which the initial version of this document was created with.
- Miner, Scheppner, Sassenrath, Haynie, etc. for the Amiga - what would computing been without it?
- All kewl friends I have - always encouraging me!

## 1.37 Address

If you have any comments/suggestions/bug reports/questions, have signed my PGP key, or simply wish to send a smiley to someone, send your response to:

<jensthi@ifi.uio.no>

or

Jens Berger  
Spektrumveien 4  
N-0666 Oslo  
Norway

or

send a message in POST to J. BERGER on Trashcan (A)BBS [(+47)~22~25~74~78  
or (+47)~22~25~88 22].

All signs of intelligent life are welcomed; that should exclude piracy.

Have fun.

Signed,

-JeT-

## 1.38 E-mail address

Wondering where in the world my e-mail address is located? It's isn't anything more exciting than the Department of Computer Science at the University of Oslo in Norway, where I'm currently studying maths and informatics.

## 1.39 Usage of GUI

Usage of the Graphical User Interface (SignArchGUI) should be quite easy, especially if you take a look at the explanation of the commandfile below and is a little familiar with the Amiga.

### STARTUP

To start the GUI, just double-click its icon. You may also start it by shift-clicking, that is holding <shift> down, click once on the SignArchGUI icon, and double-click on a CommandFile. The commandfile selected will then automagically be loaded into the GUI. The same will happen if you simply double-click the CommandFile's icon.

If you just wish to create the archive the CommandFile specifies, you may click once on the SignArch (the dummy script's) icon, and double click on the CommandFile.

It may also be started from a Shell, template:

SignArchGUI COMMANDFILE

The commandfile argument is optional, it will be loaded.

### MINIMUM EFFORT

The minimum effort to create an archive, is to:

- 1) select the files you wish to archive
  - 2) select the name of the sumfile (here called Checksumfile)
  - 3) the name of the archive
-

4)~select Project->Create Archive to create the archive (execute the REXX script).

#### ONLINE HELP

To get help on an item while the program is running, make sure the program window is active, position the mouse pointer over the gadget (or hold it over a menuitem), and press the <HELP>-key. A node in this AmigaGuide will then pop up accordingly, if it is installed in "HELP:" or the same directory as the program. The Installer script will take care of this.

#### BREAKING AN OPERATION

Some operations, like adding a zillion files to the the filelist, may take some time. To interrupt this, press CTRL-C in the window, or use the Shell "Break" command.

#### GADGETS AND MENUS

The user interface can roughly be separated into four parts:

Files	- Where you control file settings
Encryption	- Where you set encryption recipients, etc.
Prefs	- Where you can set misc. flags
Menus	- Loading/saving/etc.

Needless to say, clicking the key underlined in a gadget text, will change/activate that gadget.

NOTE: If there is a 'fileselect' gadget to the right of the gadget, pressing <shift> while clicking the key underlined in the gadget text, will pop up a requester accordingly.

For instance, if you hold <shift> and press 'r', a directory requester pops up. If you just push 'r', the string gadget is activated.

## 1.40 File Control Part

The file control part has the following gadgets:

Root directory	- The current directory while archiving
* Archive	- The file to archive the files within
* Checksum-file	- The file to put the message digests within
Include textfile	- Text file to include into the checksum-file
* Filelist	- List of the files to be archived
Add files	- Add files to the list above
Rem. files	- Remove files from the list above
Sort	- Sort the list above

The items marked with an asterix (\*), must be initialized to something. If not, the final archiving script won't do anything, and you will be warned if you try to save the datas to a file.

## 1.41 Root directory

The contents of this string gadget, will be treated as the ROOTDIR parameter when creating the commandfile, and will thus be used as current directory when archiving begins.

By clicking the box to the right of the string gadget, you may use a directory requester to select the root directory, or even easier: Just let the program try to autofind it.

## 1.42 Archive

The contents of this string gadget, will be treated as the ARCHIVE parameter when creating the commandfile, must be initialized. Thus, the final archiving script won't do anything unless this gadget is initialized, and you will be warned if you try to save the datas to a file.

The filename specified here, is the name of the LhA archive everything will be put within.

By clicking the box to the right of the string gadget, you may use a file-requester to select the archive.

## 1.43 Checksum-file

The contents of this string gadget, will be treated as the SUMFILE parameter when creating the commandfile. Thus, the final archiving script won't do anything unless this gadget is initialized, and you will be warned if you try to save the datas to a file.

The filename put here, is the file that the message digests are put within, and which, finally, is signed with PGP.

By clicking the box to the right of the string gadget, you may use a file-requester to select the checksumfile.

## 1.44 Include textfile

The contents of this string gadget, will be treated as the INCTEXT parameter when creating the commandfile. This file will be included in the sumfile, after the script, if such is specified (see: Add script commands).

The filename put here, should specify the name of a textfile to be included into the sumfile.

By clicking the box to the right of the string gadget, you may use a file-requester to select the text to be included.

---

## 1.45 Filelist

The listview named FileList contains a list of the files which are to be archived. You may use the arrows and scrollbar to the right to move up and down in the list view, select items by clicking on them, and use the gadgets below to manipulate it; namely Add files, Rem. files and Sort.

## 1.46 Add files

By clicking this gadget, a file-requester will pop up, where you can select the files which are to be archived. You may of course select multiple files.

Sometimes, after pressing "Ok" in the file-requester, this requestor may show up:

```
WARNING: Some files exist above the
rootdir. May cause problems when
archiving.
```

This is just a reminder that the Root directory gadget is not correctly initialized, and that you should correct it.

## 1.47 Rem. Files

By clicking this gadget, the currently selected file (if any), will be removed from the FileList gadget, and will thus NOT be archived.

## 1.48 Sort (files)

When you click this gadget, the contents of the FileList gadget will be sorted alphabetically, making it easier to read.

## 1.49 Encryption Control Part

The encryption part has the following parts:

Encrypt archive	- Encrypt the final archive with PGP?
Wipe noncrypted	- Keep the unencrypted version of the archive?
Recipient list	- List of people able to decrypt the archive
New key	- Add a key to the list above
Remove	- Remove a key from the list above
Sort	- Sort the list above

These gadgets will be ghosted unless the Encrypt archive prefs item is checked.

---

## 1.50 Encrypt archive

Checking this box, indicates that you wish to encrypt the final archive. The following gadgets will then become active: Wipe nonencrypted, Recipient list, New key, Remove and Sort.

If this box is checked, and the Recipient list is not empty, one or more ENCRYPT options will be put in the commandfile.

## 1.51 Wipe nonencrypted

When this item is checked, the encrypted archive will be deleted - this is the default behaviour. Deselecting this item equals specifying the ARCLIVE option in the commandfile.

This gadget will be ghosted unless the Encrypt archive checkbox is checked.

## 1.52 Recipient list

The listview within the Encryption box, contains a list over the people who may decrypt the final archive. The keys are shown with their hex ID first, and the name of the person is listed to the right. You may use the arrows and scrollbar to the right to move up and down in the list view, select items by clicking on them, and use these gadgets to manipulate it: New key, Remove and Sort.

Double-clicking on an item will show more info about that key, if PGP is properly installed (PGPPATH, etc.).

This gadget will be ghosted unless the Encrypt archive checkbox is checked.

## 1.53 New key

By clicking this gadget, a list of keys will pop up, where you may select the keys of new people who may decrypt the archive. For docs about how to use this window, [click here](#).

This gadget will be ghosted unless the Encrypt archive checkbox is checked.

## 1.54 Remove

By clicking this gadget, the currently selected key (if any), will be removed from the Recipient list gadget, and that person will thus NOT be able to decrypt the archive.

This gadget will be ghosted unless the Encrypt archive checkbox is checked.

## 1.55 Sort

When you click this gadget, the contents of the Recipient list gadget will be sorted alphabetically, making it easier to read.

This gadget will be ghosted unless the Encrypt archive checkbox is checked.

## 1.56 Prefs Part

The prefs part has the following parts:

Checksum .info	- Message digest icon files?
Use ASCII armor	- Make final archive suitable for e-mail?
ASCII deep check	- Make ASCII test check the whole file?
Add PGP public key	- Include your pubkey in the Checksumfile
Add script commands	- Add script, which checks signatures?
Only wait at end	- Don't issue the Wait command between each cmd?
Wait command	- Command issued between PGP & MD5SUM in script.
Signature ID	- The user name to sign the checksum file with.

## 1.57 Checksum .info

By default, icons are not summed, as only changing their position before redistributing is enough to make the check fail. If you check this box, they will be summed too, as the SUMICON option will be added to the commandfile.

## 1.58 Use ASCII armor

When this option is checked, the final archive output will end up as 7-bit ASCII, suitable for posting via e-mail. Currently, only PGP is supported as encoder (yes, you may ASCII-fy with PGP without necessarily encrypting), UU-encoding is currently not supported.

This is equivalent to specifying the ADDARMOR option in the CommandFile.

---



## 1.59 ASCII deep check

Checking this box, is the same as specifying the DEEPCHK option. This will make the ASCII check run through the entire file (still aborting at first 'binary' value), to determine whether the file is ASCII or not. In some cases this may be preferable, the default is to only check the first kilobyte.

## 1.60 Add PGP public key

When this option is turned on, your PGP public key will be added to the final checksum file. This is the default behaviour, but the addition of the key may be unnecessary for internal distributions, especially if you have a lot of signatures, making the key big. In other cases, leave it on, cause if the recipient haven't got your key, he/she can't check neither the signature nor the message digests.

## 1.61 Add script commands

If this box is checked, a script will be put in front of the checksum file. To make the script work properly, a '-----BEGIN' executable with your distribution (which btw. is included with this one). The script added can be controlled somewhat with these gadgets: Only wait at end and Wait command.

By default, only the signatures are put in the SUMFILE file, plus a quick description of how to check the files. In some cases you may wish to make this process automatic. To make the script work properly, a '-----BEGIN' executable is included automagically with your distribution (which btw. is included with this one).

NOTE: THIS IS A SECURITY HOLE. THE '-----BEGIN' FILE MAY BE INFECTED, OR REPLACED. COMMANDS INFECTING THE SYSTEM MAY BE ADDED IN FRONT OF THE PGP SIGNED PART. IT IS INDEED DISCOURAGED THAT YOU USE THIS OPTION.

To make sure you are aware of this problem, you will be asked each time for confirmation. To avoid this, add this line to your S:User-Startup file:

```
SetEnv "SignArchOpts" "do_not_ask_for_confirmation_of_autochk"
```

The -----BEGIN file is added because doing so will:

- 1) avoid AmigaDOS message like 'Can't find command '-----BEGIN' (the ENTIRE file is interpreted as a normal AmigaDOS script!)
- 2) check that the SUMFILE begins with '-----BEGIN PGP SIGNED MESSAGE-----' and TWO linefeeds. This is necessary to avoid a bug in PGP. Both the script and the executable should work with all kickstarts.

As the file has to be recreated for each run and is differs from archive to archive, the only equality is that there is one code chunk

---

of 300 bytes (which is equal from file to file), and then a data chunk which contains the name of the SUMFILE's name.

This is btw. the same as specifying the AUTOCHK option.

## 1.62 Only wait at end

When this box is checked, the script will only issue the Wait command at the end of the script, when the Add script commands box is checked.

Checking this box equals specifying the ENDWAIT option.

This gadget will be ghosted unless the Add script commands prefs item is checked.

## 1.63 Wait command

In this string gadget, enter the command issued between each of the cryptographic commands, or only at the end if the Only wait at end Prefs item is checked. Preferably, it will wait some seconds, ask for user input or whatever, so that the user may read the output from PGP and MD5SUM.

Ideas for wait commands:

```
Ask "Press <return> to continue..."    ; will wait until the user
                                         ; presses return
```

```
Wait 5                                  ; waits 5 seconds
```

etc.

Home-made waiting commands should be avoided, as they can be infected, and may infect PGP, which will set off a tragic chain of events.

This gadget will be ghosted unless the Add script commands prefs item is checked.

## 1.64 Signature ID

Enter your name within PGP in this string gadget, the key this name corresponds to will be used for signing the checksumfile. By clicking the gadget to the right of the gadget, you may select the key with the PGP key-select requester.

Please note if you leave this gadget uninitialized, the MyName parameter found in pgp.config in your PGPPATH dir (if properly initialized, that is) will be used as your name when talking to PGP. If this can't be found, the first key on your secret keyring will be used.

---

The key ID returned from this requester is the hexadecimal ID number, not your name. This is done to avoid confusion (if you for instance, have more than one secret key).

## 1.65 Overview of menu items

The following menu items are available:

### Project

- Open... - Open previously created commandfile
- Save... - Save settings to commandfile
- Save as... - The same, but requesting file first
- Global settings - Set paths for needed programs
- Create archive - Create the archive (and crypt it?)!
- About - Shows some info...
- Quit - Only included for testing purposes...:)

### Settings

#### Prefs

- Load... - Loads the prefs settings ONLY
- Save... - Saves the prefs settings ONLY
- Clear... - Clears all prefs settings

#### Encryption

- Load... - Loads the names of keys ONLY
- Merge... - Merge a namelist into the current
- Save... - Saves the names of keys ONLY
- Clear... - Clears all encryption settings

#### Files

- Load... - Loads filenames into the filelist ONLY
- Merge... - Merge a filelist into the current
- Save... - Saves filenames in the filelist ONLY
- Clear... - Clears filelist and/or rootdir, etc.

## 1.66 Project->Open...

Selecting this menu item will open a previously created commandfile. The options will be parsed, and the gadgets will be initialized accordingly. The following options are ignored:

- ASCCHAR - strips off eventual leading ASCCHAR's in filenames
- BINCHAR - strips off eventual leading BINCHAR's in filenames
- ADDCOMM - ignored completely
- BANNERS - ignored completely

You may also only load parts of the files, for this, see:  
Settings->Prefs->Load... to load the prefs settings ONLY,  
Settings->Encryption->Load... to load the pubkeys ONLY and  
Settings->Files->Load... to load the files in the filelist ONLY.

## 1.67 Project->Save...

Selecting this menu item will save the settings of the gadgets into the previously selected option file. If no option file is previously selected, the program will request one with a filerequester, as in Project->Save as...

If you wish to only save a part of the gadget settings, see: Settings->Prefs->Save... to save the prefs settings, Settings->Encryption->Save... to save the pubkeys and Settings->Files->Save... to save the files in the filelist.

## 1.68 Project->Save as...

Selecting this menu item will save the settings of the gadgets into the option file entered in the file requester. If no option file is selected, nothing will happen.

If you wish to only save a particular part of the gadget settings, see Settings->Prefs->Save... to save the prefs settings, Settings->Encryption->Save... to save the pubkeys and Settings->Files->Save... to save the files in the filelist.

## 1.69 Project->Global Settings

This menupoint will make a window pop up, where you may configure the paths for PGP, RX and the REXX script. You should not do anything else than specifying the program name, WITH PATH. For PGP you may also specify options, assuming you have PGP in "C:" :

```
C:PGP +PKCS_COMPAT=0
```

Just remember to NOT add any commands to the line, like "-kvv", etc.

Quick explanation of the string gadgets should contain:

PGP

- The name of your PGP binary program file, NOT the directory you've set up as PGPPATH.

RX

- The command used for starting REXX programs. Usually found in the Sys:Rexxc/ directory.

REXX

- Full path & name for the SignArch.rexx script.

AmigaGuide

- ~The path (with filename) for this document. If it isn't installed, set the filename to "::::N/A::::", and the program won't search for it.

LhA

---

- Full path & name (plus possible options) for LhA

MD5Sum

- The path & name for MD5SUM, the tool which was enclosed with PGP, and which will be used for creating signatures.

The other gadgets will do the following:

Auto-include icons?

- When checked, this gadget will cause that when pushing the Add~files button, automagically include associated .info files for the files you select.

Generate sumfile icon?

- Will generate an icon for the sumfile, if it doesn't exist.

Save - Save the settings for future use

Use - Use the settings only this session (until reboot)

Cancel - Do not use settings at all

Please note that all these parameters are initialized by the Installer script.

## 1.70 Project->Create archive

By selecting this menupoint, signing and archiving will start! The settings will be saved to a temporary file, and the REXX script, which does the job, will be invoked. The input/output from this script will appear in the console window.

## 1.71 Project->About

Shows some info about me and the program. Please remember these addresses when it's Christmas, in case some unwanted A4000/040's, Indy's, HD's, etc. should show up...:)

Nah, but I WOULD LOVE a postcard, anyway - at anytime of the year!

## 1.72 Project->Quit

This menuitem was included for testing purposes, and will probably have no practical use. :)

Anyway, it will check whether you've changed the settings since last save, and give you the choice of saving them first.

---

### 1.73 Settings->Prefs->Load...

This menupoint will load the prefs settings from disk, ignoring eventual crypt and file information. Thus, the only gadgets affected by this operation, will be those within the Prefs box. Please note that these files begin with '!!PREFS!!' instead of '!!FILELIST!!', however, you may choose to ignore this when loading the datas.

Thus, you may load only the prefs settings from another, previously created, commandfile.

These files are normally saved with Settings->Prefs->Save...

### 1.74 Settings->Prefs->Save...

This menupoint will save the prefs settings from disk, not including crypt and file information. Thus, the only gadget settings saved by this operation, will be those within the Prefs box. Please note that these files begin with '!!PREFS!!' instead of '!!FILELIST!!', however, you may choose to ignore this when loading the datas with Project->Open...

These files are normally loaded with Settings->Prefs->Load...

### 1.75 Settings->Prefs->Clear...

By selecting this menupoint, everything in the prefs rectangle will be reset to default, and string requesters will be cleared. Confirmation will be asked for. Please note that the Wipe noncrypted gadget also will be affected by this.

### 1.76 Settings->Encryption->Load...

This menupoint will load the files contained in the file selected into the Recipient list, ignoring eventual prefs and file information. Please note that these files begin with '!!CRYPT!!' instead of '!!FILELIST!!', however, you may choose to ignore this when loading the datas.

NOTE: This will replace the contents of the Recipient list gadget. If you wish to simply add to the contents of the gadgets, use the Settings->Encryption->Merge... menupoint instead.

These files are normally saved with Settings->Encryption->Save...

---

## 1.77 Settings->Encryption->Merge...

This menupoint will load the files contained in the file selected into the Recipient list, ignoring eventual prefs and file information. Please note that these files begin with '!!CRYPT!!' instead of '!!FILELIST!!', however, you may choose to ignore this when loading the datas.

NOTE: In contrast to Settings->Encryption->Load... menupoint, this one is non-destructive. If you wish to replace the contents of the recipient list gadget, use that one instead.

These files are normally saved with Settings->Encryption->Save...

## 1.78 Settings->Encryption->Save...

This menupoint will save the files in the Recipient list gadget to the selected file, not including eventual files and prefs information. Please note that these files begin with '!!CRYPT!!' instead of '!!FILELIST!!', however, you may choose to ignore this when loading the datas with Project->Open...

These files are normally loaded with Settings->Encryption->Load...

## 1.79 Settings->Encryption->Clear...

This menupoint will, after the selection has been confirmed, flush all recipients from the recipient list, and turn off encryption of final archive.

## 1.80 Settings->Files->Load...

This menupoint will load the files contained in the file selected into the Filelist, ignoring eventual crypt and prefs information. Please note that these files begin with '!!FILES!!' instead of '!!FILELIST!!', however, you may choose to ignore this when loading the datas.

Thus, you may load files you normally sent to LhA with this menupoint.

NOTE: This will replace the contents of the FileList gadget. If you wish to simply add to the contents of the gadgets, use the Settings->Files->Merge... menupoint instead.

These files are normally saved with Settings->Files->Save...

---

## 1.81 Settings->Files->Merge...

This menupoint will load the files contained in the file selected into the Filelist, ignoring eventual crypt and prefs information. Please note that these files begin with '!!FILES!!' instead of '!!FILELIST!!', however, you may choose to ignore this when loading the datas.

Thus, you may load files you normally sent to LhA with this menupoint.

NOTE: In contrast to Settings->Files->Load menupoint, this one is non-destructive. If you wish to replace the contents of the FileList gadget, use that one instead.

These files are normally saved with Settings->Files->Save...

## 1.82 Settings->Files->Save...

This menupoint will save the files in the Filelist gadget to the selected file, not including eventual crypt and prefs information. Please note that these files begin with '!!FILES!!' instead of '!!FILELIST!!', however, you may choose to ignore this when loading the datas with Project->Open...

These files are normally loaded with Settings->Files->Load...

## 1.83 Settings->Files->Clear...

When you wish to reset the filelist and/or the Checksum-file, Root directory, Archive or Include textfile, select this menupoint. Beware that clearing the rootdir without clearing the filelist may cause all filenames to expand to full path - making it difficult to get an overview.

## 1.84 PGP key-select window

There are four gadgets, by clicking them, the following happens:

Use key	- Return the currently selected key
Search	- This will search for a specified string.
Search next	- Searches for the next occurrence
Update	- Re-reads the keyring from disk

If you change your mind, and don't wish to select a key anyway, click the close button in the upper left part of the window.

PLEASE NOTE: Sometimes, nothing is displayed, although PGP is properly installed and configured in Global Settings. This is probably because you're running low on memory - try free some and click the Update button.

---



## 1.85 Use key

Click this button, and the key selected will be returned. If this window was invoked by pressing New key, the key selected will be added to the recipient list, with the hex ID first and then the name. If it was started by clicking the box beside Signature ID, only the hex ID will be returned, to avoid later confusion.

If you have problems with locating a specific key (although they're sorted alphabetically by name, not hex ID), try the Search and Search next buttons.

## 1.86 Search

When you click this button, a requester will pop up, asking for string to search for. The program will then search downwards, case-insensitive, for any match within a key's hex ID or name, starting at the currently selected item, or at the top if nothing is selected. If nothing is found, you will be informed.

To continue searching for the same string, press Search next.

## 1.87 Search next

Does the same as Search, except that it won't re-ask for the search string, unless the search string is empty. For more info on the search procedure, look at the docs for Search.

## 1.88 Update

Sometimes the keyring is changed while the program is running, a key may be added. Thus, the list of keys in the PGP key-select window may be incorrect. In such case, click this button to re-read the keyring from the disk.

This will also correct problems when you're running low on memory - if there's no mem available, nothing can be displayed. If you have freed some memory, click this gadget to retry loading.

## 1.89 PGP

Pretty Good(tm) Privacy (PGP), from Phil's Pretty Good Software, is a high security cryptographic software application for MSDOS, Unix, VAX/VMS, and other computers. PGP allows people to exchange files or messages with privacy, authentication, and convenience. Privacy means that only those intended to receive a message can read it. Authentication means that messages that appear to be from a particular

---

person can only have originated from that person. Convenience means that privacy and authentication are provided without the hassles of managing keys associated with conventional cryptographic software. No secure channels are needed to exchange keys between users, which makes PGP much easier to use. This is because PGP is based on a powerful new technology called "public key" cryptography.

PGP combines the convenience of the Rivest-Shamir-Adleman (RSA) public key cryptosystem with the speed of conventional cryptography, message digests for digital signatures, data compression before encryption, good ergonomic design, and sophisticated key management. And PGP performs the public-key functions faster than most other software implementations. PGP is public key cryptography for the masses.

(Taken from "Volume I: Essential Topics" of the PGP documentation).

## 1.90 Index

Overview over the nodes in this hypertext, sorted alphabetically. (GUI) means that this is something found in the Graphical User Interface.

	--> A <--
ADDARMOR	- Makes output suitable for E-mail
Add files	- Add files for archiving (GUI)
Add script commands	- Make sumfile self-checking (GUI)
AddSum	- Add sumfile to existing archive
Add PGP public key	- Add key used for signing the SUMFILE
ADDCOMM	- Add command to script put in SUMFILE
Address	- How to contact me
ARCHIVE	- Name of archive to put it all into
Archive	- Name of archive to put it all into (GUI)
ARCLIVE	- Don't delete non-crypted archive
ASCCHAR	- Force file to be summed as ASCII
ASCII deep check	- Thorough text/binary check (GUI)
AUTOCHK	- Make sumfile self-checking
	--> B <--
BANNERS	- Add a banner/comment to the sumfile
BINCHAR	- Force file to be summed as binary
Bugs	- Are there any?
	--> C <--
Checksum .info	- Message digest icon files? (GUI)
Checksum-file	- File to put message digests into (GUI)
ChkASCII	- Is a file text or binary
ChkDate	- Is date valid
Copyrights	- Legal mush
Creating the commandfile	- What commands may be put within?
	--> D <--
DEEPCHECK	- Thorough text/binary check
Disclaimer	- READ THIS! NO WARRANTY!
Distribution	- Conditions on distribution

```

--> E <==
E-mail address      - Where in the world is it?
Encrypt archive     - Encrypt final archive with PGP (GUI)
ENCRYPT              - Encrypt final archive with PGP
Encryption Control Part - PGP key handling within the GUI
ENDWAIT             - Only run WAITCOM at end of script
EXECCMD             - Run command BEFORE archiving

--> F <==
FAQ                 - Frequently Asked Questions
File Control Part   - File handling within the GUI
Filelist            - List of the files to archive

--> I <==
Include textfile    - Include textfile in SUMFILE (GUI)
INCTEXT             - Include textfile in SUMFILE
Installation         - How to install the program
Introduction         - What the program is and does

--> N <==
New key             - Add recipient able to decrypt (GUI)
Notes               - Misc. info worth reading - kind of FAQ
NOKEYADD            - DON'T add key used for signing
NOWILDS             - Don't do any wildcard parsing

--> O <==
Only wait at end    - Only run WAITCOM at end of script (GUI)
Overview of menu items - Menus available in the GUI

--> P <==
PGP                 - Quick introduction
PGP key-select window - Point'n'click PGP keys
Prefs Part          - Prefs settings within GUI
Project->About        - About the GUI program (GUI)
Project->Create archive - Archive everything (GUI)
Project->Open...       - Load commandfile (GUI)
Project->Quit          - Terminate GUI
Project->Save as...     - Save commandfile to specified name (GUI)
Project->Save...        - Save commandfile (GUI)

--> R <==
Recipient list      - People able to decrypt the archive
Rem. Files          - Remove files from the filelist (GUI)
Remove              - Remove crypt-recipient from list (GUI)
Requirements        - What you need to run it
Root directory      - Name of current dir when archiving (GUI)
ROOTDIR             - Name of current dir when archiving

--> S <==
Search              - Search for PGP key
Search next         - Search for next PGP key
Settings->Encryption->Clear... - Clears all encryption settings (GUI)
Settings->Encryption->Load...  - Load PGP-keys only (GUI)
Settings->Encryption->Merge... - Merge PGP-key list into recipient gadget (GUI)
Settings->Encryption->Save...  - Save PGP-keys only (GUI)
Settings->Files->Clear...      - Clears filelist and/or rootdir, etc. (GUI)
Settings->Files->Load...       - Loads commandfile into the filelist ONLY (GUI)

```

---

Settings->Files->Merge...	- Merge filenames into FileList gadget (GUI)
Settings->Files->Save...	- Saves the names in the filelist ONLY (GUI)
Settings->Prefs->Clear...	- Clears all prefs settings (GUI)
Settings->Prefs->Load...	- Load prefs only (GUI)
Settings->Prefs->Save...	- Save prefs only (GUI)
ShA	- LhA replacement
SIGNAME	- Commandfile option
SignArch v1.1	- Main index/screen
Signature ID	- Secret key used for signing (GUI)
Sort	- Sort filelist (GUI)
Sort	- Sort recipient list (GUI)
SUMFILE	- File to put message digests into
SUMICON	- Message digest icon files?
SUMONLY	- Only generate the sum file?

--> T <==

Thanks	- People I like
The commandfile	- What is this mysterious file?

--> U <==

Update	- Reload list of PGP keys
Usage of GUI	- How to use the Graphical User Interface
Usage of shell commands	- How to use the Shell programs
Use ASCII armor	- Makes output suitable for E-mail (GUI)
Use key	- Use currently selected key

--> W <==

Wait command	- Command issued in SUMFILE script (GUI)
WAITCOM	- Command issued in SUMFILE script
WAITRET	- Wait for return in case of failure?
Wipe noncrypted	- Delete non-crypted archive? (GUI)

---