The KBackup User Manual

Version 1.2.11

by Karsten Ballder
(Ballueder@usa.net)

Maintained by David Frascone
(frascone@users.sourceforge.net)

October 13, 2000

Contents

Chapter 1

Copyright

Chapter 2

Copyright

Copyright (C) 1995-2000 Karsten Ballder
(Ballueder@usa.net)

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

The full text of the GNU General Public License is included in Appendix B.

You can contact the author by the e-mail address given above or write to: **Karsten Ballder, 11 (2F3) Yeaman Place, Edinburgh EH11 1BR, Scotland**.

2.1 If you use this software...

Well, writing KBackup was quite some work and maintaining it, answering people's questions, fixing bugs, ... is some more. To encourage me to continue work on this program,please, send me a postcard of your area. This should take just a minute or two, but *please* take the time to do so. It is always nice to see who uses one's work and getting postcards from all over the world is even better.

If you use KBackup on any other system but Linux — which should be absolutely no problem — please let me know.

2.2 If you spread this software...

I am happy for anyone who helps to make this software available to others. If you put KBackup on a CD-ROM, I would like you to send me a copy of it. This is not a legal condition, but only a wish that is easy for you to fulfill and would save me some money.

Please understand that I have no objections against KBackup appearing in any Linux distributions. If you need any support or help for this, let me know.

The current Maintainer of the software is David Frascone. He can be reached at (frascone@users.sorceforge.net).

Chapter 3

Introduction

3.1 Purpose of KBackup

KBackup is intended for handling of backups no matter whether they reside on disk or tape or even in files. As most other backup programs available today are either confusing the user with lots of long command line options or user friendly but not powerful at all, the aim behind writing KBackup was to provide a user friendly yet powerful backup program. Further on I wanted to keep it compatible to existing and well proven archive formats so you can restore your archives even if you should not have KBackup around anymore.

To achieve this goal, KBackup uses established and well proven archive tools like `tar(1)` or `afio(1)` to do the archive handling and improves their functionality by adding quite a lot of additional features. By using the `dialog(1)` program I could make archive handling as easy and user friendly as possible. This means that you will get the power and reliability of the best archive handlers combined with a customisable and easy to use interface.

3.2 Features

Although KBackup uses other common tools to handle its archives, it provides additional features to make your life easier. These include

- Each archive on a tape is preceded by a header containing option settings, parameters, compression format, date, backup type, *etc.* This maintains compatibility across different versions of this software and, as it is stored in readable format, helps you restore the data manually if you cannot use KBackup for what reason ever [1] .

[1] For restoring KBackup version 1.0 or 1.1 archives, please refer to the section 10.3.3 or the FAQ.

- When using tapes or mountable block devices (see section 9.2.3), a complete directory of all files in the archive gets stored at the beginning of the archive. You no longer need to read all the long archive to find out what files it contains.
- Archive format and compression method as well as other parameters get detected automatically during restore or verify operation. You do not need to know how you created the archive, KBACKUP will find it out for you.
- You can use `gzip(1)` or `compress(1)` to compress your archives.
- If you have some sensitive data, you can use `pgp(1)` (Pretty Good Privacy) to compress and encrypt your archives using public key encryption. (Ever wanted to send a tape or disk of sensitive data to someone else? Now you can!) [2]
- You can schedule backup or restore operations to be executed at a later time or even regularly. KBACKUP supports full `at(1)` functionality.
- KBACKUP can use double buffering to keep your tape drive streaming continuously even on slow or heavily loaded machines. No longer corrupt your tapes by starting and stopping the drive every few seconds.
- You can keep lists of files and directories that you want to exclude from backups. You can also include and/or exclude files by using standard shell patterns.
- You have the choice between full or incremental backups.
- KBACKUP supports tapes, floppy–drives and removable media like removable hard–drives, magneto-optical drives or others.
- You can keep different configuration files for your backup tasks. Sample configurations files for backups to tape or floppy disk are included. No manual configuration file editing, just go through the menus and change whatever you like.
- Reliably handle multi–volume archives. There is no limit for your archives' sizes. By using the MULTIBUF program, KBACKUP automatically detects the length of your tapes.
- The same MULTIBUF program is capable of performing any operation on the end of a tape, like e.g. telling an automatic loading mechanism to change tapes.
- support for accessing remote devices for easy support of LAN backups.

[2]For legal reasons, `pgp(1)` is not included in this package. You can obtain different version from file servers all over the world. I have been using KBACKUP with `pgp` version 2.6.2i. `Pgp` is written and copyright by Philip Zimmermann.

3.3 Compatibility

Although KBackup adds lots of functionality, its archives are still readable by `afio(1)` or `tar(1)`, depending on which one was used to create it. To do so, on tapes, KBackup writes several files per archive: an archive header containing information about the archive type, compression method, date, ..., the table of contents (`gzip(1)` –compressed) and the actual archive file itself (If using sequence information for the MultiBuf program, it creates another two (short one–block) files.). To restore multivolume archives, all you need is the archive handler (`afio(1)` or `tar(1)`) and MultiBuf, which is included in ANSI–C source code an should easily be compiled on any system.

If you do not want to use MultiBuf, you can switch it off and write plain `tar(1)` or `afio(1)` archives.

Unfortunately, the many new features, especially the secure multivolume support, required some changes to the way archives get written to the tape. However, KBackup is able to autodetect whether MultiBuf has been used in archive creation and therefore is still capable or reading archives that have been created by earlier versions of KBackup. It is currently *not* possible to restore multivolume archives that have been created with older versions of KBackup. As they were somehow unreliable anyway, better use a version newer than 1.2 to make new backups.

To restore old multivolume archives, created with version 1.0 or 1.1, you have to use `tar(1)` or `afio(1)` manually.

Restoring *any* archive manually (optionally using MultiBuf) is always possible, anyway.

3.3.1 supported devices

Tape drives

KBackup should works with all tape drives supported by Linux. You need to have a properly configured tape driver in your kernel. If other programs work with your tape drive, KBackup will as well. KBackup is able to autodetect your tape device and test it for compatibility.

KBackup has been reported to work with tape drives on a `QIC-02` controller, many different `SCSI` streamers and `DAT` drives as well as with the `ftape` driver version 2.03 (drives controlled by the floppy controller). It has reported *not* to work with `zftape` yet. It has also been reported to work with the IOmega ZIPdrive^{TM}, both the SCSI version as well as the alpha driver for the parallel-port version.

I am using and testing KBackup with a (rather old) Tandberg drive on a noname `QIC-02` controller. The `QIC-02` Linux driver implements the "`/dev/tape-reset`" device, so KBackup can use it to reset the driver at your request. I have not heard of any other driver supplying this pseudo–device.

For more details on using floppy streamers (the `ftape` and `zftape` drivers), see the Ftape-HOWTO[3].

Floppy disks

KBackup can support floppy disks in either of two ways:

1. By writing straight to the disk using `afio(1)` or `tar(1)` to write directly to the block device. This is not recommend, because error detection is not too good. However, this gives the highest capacity per disk.

2. By treating floppy disks as "general block devices" . This is highly recommended, see next section for details.

If you want to use higher than normal capacity floppy formats with KBackup , this is no problem. Just tell it about it in the options menu.

Block devices, ZIP drive

KBackup can use *any* block device which can be formatted and mounted, using the `BLOCKDEV` archive type. In this case, it will (optionally) format and mount the device and write to it. This is the best way to use floppy disks and the only way to use removable hard disks, like the IOmega ZIPdrive.

3.3.2 MultiBuf and Compatibility

All that MultiBuf does, is to split the archive in multiple parts at well defined block sizes. If a block does not fit onto the tape, it prompts you for the next tape and starts it with this block again. Without MultiBuf , restoring these archives might be tricky, especially if you are using the sequence information option. However, as MultiBuf is written in plain ANSI–`C` it should compile cleanly on *any* system. The source of it is included with KBackup , so is a `Makefile` for compilation.

3.4 Limitations, Bugs

Well, all limitations of the archive tools used by KBackup still apply. If e.g. `afio(1)` has a bug, KBackup will have it, too.

Another serious limitation occurs when using `pgp(1)` for encrypting archives. `Pgp(1)` creates temporary files during operation, so that, when using `tar(1)` the maximum size of the archive is somehow limited by your free disk space in the directory for storing temporary files. Better use `afio(1)` for writing encrypted archives, not `tar(1)` .

[3] ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/Ftape-HOWTO

3.5 Requirements

KBackup requires some other software to be installed on your system. However, most of them should be available on any properly installed Unix system. If you are using a Slackware–based Linux , all you need is to install `afio(1)` on your machine.
The programs required by KBackup are:

- `dialog` : the dialog program is being used for doing all the user interaction. If `dialog(1)` is not available, KBackup will fall back to . Without `dialog(1)` , it is still fully functional (just a bit ugly)!
- `gtar` : the GNU tape archiver. *A normal `tar(1)` will not do!* Under Linux , `tar(1)` *is* GNU's `tar` , so do not worry.
- `afio` : the archive program. It is KBackup 's preferred archive handler, somehow more powerful than `tar` , but less well tested.
- `grep` , `fgrep` , `sed` : some text processing programs that should be installed on every proper Unix system.

In fact, you do not need both, `tar(1)` and `afio(1)` . KBackup can use either of them to handle its archives. `Afio` is strongly recommended. *For handling encrypted archives , you should get `afio(1)` version 2.4.2 or newer.*

In the file `paths` , you can set environment variables pointing to the various programs. Just in case the standard settings should not work for you.

KBackup should run with any Bourne Shell (`sh(1)`) compatible shell. However, I have mainly tested it using the Bourne-Again Shell (`bash(1)`), therefore I cannot guarantee that it will work with others, but I do not see any problems. Especially, it appears to be working well with the rather small `ash(1)` shell, which is of interest for putting it onto boot disks.

3.6 Operating Systems supported

KBackup is being developed and maintained on a standard Linux system.

As it is a shell script, it is highly portable and *should* run on any UNIX system around. However, there might still be some work required to get it to run on other systems but Linux . You no longer need the `dialog(1)` program as KBackup can now run without it.

There is currently work going on in making KBackup run under ULTRIX.

3.7 Reliability. . . a warning

Well, I believe KBackup is well tested by now.

Of course, I have tried to test it as much as I could before releasing it. **But**, I can only test KBackup in my working environment with my hardware and my software setup. Although I have fixed all reported problems from previous versions, there might still be one with your setup. So, before relying on KBackup , **test it for yourself**! Make a small backup and check whether you can successfully restore it (I recommend you using the "`/tmp`" partition, not your real data directories.).

These comments apply to backup software in general, not only to KBackup . So don't be frightened, I believe it is safe to use and trust it for my own backup needs.

So far I have received some hundred mails concerning KBackup , therefore most bugs should be fixed.

3.8 Further Information

Further information on KBackup and related topics can be found at:
my homepage[4].

In that place you will always find the latest up-to-date information, bug–fix release and an online version of the complete KBackup and MultiBuf documentation (this manual).

Where to get those programs?

In the online version of the manual, you will find hyperlinks to the programs mentioned throughout this manual. These are:

- The `pgp(1)` [5] public key encryption program.
- The `afio(1)` [6] archive handler.
- The `dialog(1)` program in its official release[7] and in the latest beta/developement release[8] .

  There are also other (non official) variants that might work for you. However, KBackup does no longer require `dialog(1)` for its operation.

3.9 Contacts, Help, Support

In case that you experience any problems with this software, you can do the following:

[4] http://www.uni-karlsruhe.de/~Karsten.Ballueder
[5] ftp://ftp.th-darmstadt.de/pub/security/pgp
[6] ftp://sunsite.unc.edu/pub/Linux/system/backup
[7] ftp://sunsite.unc.edu/pub/Linux/utils/shell
[8] ftp://foggy.systemy.it/pub/dialog

1. read the documentation
2. search the relevant newsgroups
3. contact the KBackup mailinglist, by sending mail to kbackup@sitewerks.com[9]. To subscribe to the mailing list, send a mail containing the line `subscribe kbackup` to majordomo@sitewerks.com[10].
4. send e-Mail to KBackup@usa.net[11], the current maintainer.

in emergencies, do not send e-Mail to the original author (Ballueder@usa.net), as he is no longer maintaining this software!

3.10 Changelog

3.10.1 Changes from 1.2.10 to 1.2.11

- fixed rpm archives
- fixed multibuf makefile
- bugfix in restore-src
- added #!/bin/sh to source files

3.10.2 Changes from 1.2.9 to 1.2.10

- moved to GPL
- fixed backup-size bug, using awk to sum sizes
- re-introduced "Archive size: " in report
- excluded "Waiting..." messages from report generation
- changed check-error routine to use /dev/tty for IO
- made error message for non-existing datestamp disappear

[9] mailto:kbackup@sitewerks.com
[10] mailto:majordomo@sitewerks.com
[11] mailto:KBackup@usa.net

3.10.3 Changes from 1.2.8 to 1.2.9

- added option to change waiting time
- added input padding to blocksize for multibuf, to avoid dropping of last few bytes (like e.g. in contents directory)
- added comments to bug report feature
- added option to manually type in pathnames
- create ZFTAPE configuration, blocking to 10240 bytes
- added waiting to write-data code to avoid QIC02 troubles
- fixed autodetect code
- added "
#" detection for filenames with "#" in lists
- applied syncto for write/autodetect
- added "syncto" parameter for setting blocksize
- changed a few mt into my-mt
- added command line parameters "–auto-*"
- some changes to scheduling code, no longer creates/deletes configurations, schedule-only-once obsolete
- updated configuration files, added new config "File"
- fixed problem with decompressing logfiles
- changed copyright to "Artistic License" (like perl)
- (re)created new archive type FILE
- changed BLOCKDEV to allow multiple archives by setting archive file name
- incorporated double-buffering multibuf and changes made by Thomas Moore (remote mode should work fine now)
- changed a few mt into my-mt

3.10.4 Changes from 1.2.7 to 1.2.8

- changed multibuf code to use scripts using ”mt” on tapes
- fixed multibuf Makefile for non-linux-a.out
- kbinstall asks before installing things
- no more annoying messages when starting kbackup without required programs
- better (earlier) error detection in device-autodetect
- fixed problem with include-file list
- fixed cleanup code to remove all temporary files
- fixed typo: afio-user-opt -¿ tar-user-opt
- added ”mt offline” to multibuf options for tapes
- fixed (?) option-setting for multibuf commands

3.10.5 Changes from 1.2.6 to 1.2.7

- fixed problems with incremental backups
- put all temporary files into one common directory
- added ttymode parameters for MultiBuf
- added user definable parameters for tar/afio
- fixed dirlist in ttymode
- fixed parameter handling for afio 2.4.2
- fixed redirection problem in restore-src
- added working(!) configuration file for IOmega ZIPdrive

3.10.6 Changes from 1.2.5 to 1.2.6

- manual still outdated
- added `<errno.h>` to multibuf.c
- improved logfile/contdir filtering in restore-src
- fixed bugs in logfile handling, cont. dir handling during restore

- mt should work on remote host as well now
- eventually (!!!) got pgp and afio to work fine together (???)
- pgp should work safe(!!!) now, by using input redirection
- added protect-newer-files option for AFIO format
- added support for afio 2.4.2 features (pgp)
- changed kbinstall to support ttymode
- created simple text mode for use without dialog (opt: `--ttymode`)
- fixed typo in auto-rewind()
- fixed tar specific problem with restore from contentsfile (Do NOT use -s for tar restore operation!)
- fixed problem with afio version detect
- fixed problems with restore from contentsfile
- created dialog-script to replace dialog program if not available
- changed multibuf/Makefile
- fixed Delete-Logfile() function
- changed order of this file (was getting too long) :-)
- fixed configuration of directory for temporary files
- fixed logfile message in report generation
- added report-msg for broken contents dir in verify

3.10.7 Changes from 1.2.4 to 1.2.5

- multibuf seems to work with all types of tape drives now
- made code more portable (if statements)
- better handling of `SIGINT`
- bug report can be saved to a file now
- detection of configuration change and prompt whether to save it
- included icons for HTML version of manual
- cleaned up detection whether device is tape or not

- changed Help-Menu text
- added more info to `MultiBuf` Menu
- added "rewind"-warning
- fixed "mkfs"-bug
- added report generation
- added automatic verify for tape archives
- changed KBackup directory hierarchy, made kbinstall move old files
- fixed some minor bugs
- clarified Shareware-Copyright notice
- fixed sequence number problem in MultiBuf
- some minor fixes on BLOCKDEV code
- some fixes to code handling change of volumes
- removed icons from HTML manual

3.10.8 Changes from 1.2.3 to 1.2.4

In 1.2.3, the autodetection code was missing and there was a problem with multibuf on SCSI tapes as well as minor problems with the installation program. I hope to have them fixed in 1.2.4 (released Jan, 18, 1996).

- - added "fmark-fix" to multibuf: retries reading if 0 bytes read (should?!? skip filemarks reliably on non QIC-02 tapes)
- - added warning about erasing DAT drives
- - fixed some bugs in kbinstall: writes proper pathnames to src/paths
- - added missing autodetect-src to archive

3.10.9 Changes from 1.2.0 to 1.2.3

Most of the changes are only minor improvements or bugfixes:

- fixed $findxpertflags= assignment
- added lots of doublequotes in assignments to improve safety
- minor changes to restore and verify routines to share common code

- fixed bugs introduced by new doublequotes in read-data/write-data
- made it able to autodetect and read non-multibuf archives (V1.0/V1.1 type)
- (re-)added menu to switch off MultiBuf use
- fixed a few bugs in read-data/write-data
- started work on improved afio
- directory requester is sorting entries now
- fixed a missing close() in multibuf.c
- added last-command variable to be used by check-error
- introduced common "¡" key to go back to previous menu
- made directory requester show symlinks
- fixed Goto-Operation to handle archives with any count of files on the tape — important for mixing multibuf/non-multibuf archives
- changed copyright to BSD like

New features that have been added, are:

- first try at remote device support via rsh(1), works fine on my machine
- added autodetection/check tape & restructured device menu

3.10.10 Changes from 1.1 to 1.2.0

Among them…

- fixed the directory–exclusion bug, completely rewrote routines for including/excluding files
- added pattern matching capabilities
- created MultiBuf , giving automatic end-of-tape detection
- started modularising the software for faster start up (this also enables you to save disk space by leaving out unnecessary options, important for rescue–disks)
- fixed code for verifying archives
- restructured menus
- KBackup now handles filenames containing whitespace properly

- support for mountable block devices
- improved installation program
- changed archive formats (sorry)

3.10.11 Changes from 1.2.8 to 1.2.9

- added command line parameters "–auto-*"
- some changes to scheduling code, no longer creates/deletes configurations, schedule-only-once obsolete
- updated configuration files, added new config "File"
- fixed problem with decompressing logfiles
- changed copyright to "Artistic License" (like perl)
- (re)created new archive type FILE
- changed BLOCKDEV to allow multiple archives by setting archive file name
- incorporated double-buffering multibuf and changes made by Thomas Moore (remote mode should work fine now)
- changed a few mt into my-mt

3.10.12 Changes from 1.2.7 to 1.2.8

- changed multibuf code to use scripts using "mt" on tapes
- fixed multibuf Makefile for non-linux-a.out
- kbinstall asks before installing things
- no more annoying messages when starting kbackup without required programs
- better (earlier) error detection in device-autodetect
- fixed problem with include-file list
- fixed cleanup code to remove all temporary files
- fixed typo: afio-user-opt -¿ tar-user-opt
- added "mt offline" to multibuf options for tapes
- fixed (?) option-setting for multibuf commands

3.10.13 Changes from 1.2.6 to 1.2.7

- fixed problems with incremental backups
- put all temporary files into one common directory
- added ttymode parameters for MultiBuf
- added user definable parameters for tar/afio
- fixed dirlist in ttymode
- fixed parameter handling for afio 2.4.2
- fixed redirection problem in restore-src
- added working(!) configuration file for IOmega ZIPdrive

3.10.14 Changes from 1.2.5 to 1.2.6

- manual still outdated
- added `<errno.h>` to multibuf.c
- improved logfile/contdir filtering in restore-src
- fixed bugs in logfile handling, cont. dir handling during restore
- mt should work on remote host as well now
- eventually (!!!) got pgp and afio to work fine together (???)
- pgp should work safe(!!!) now, by using input redirection
- added protect-newer-files option for AFIO format
- added support for afio 2.4.2 features (pgp)
- changed kbinstall to support ttymode
- created simple text mode for use without dialog (opt: `--ttymode`)
- fixed typo in auto-rewind()
- fixed tar specific problem with restore from contentsfile (Do NOT use -s for tar restore operation!)
- fixed problem with afio version detect
- fixed problems with restore from contentsfile
- created dialog-script to replace dialog program if not available

- changed multibuf/Makefile
- fixed Delete-Logfile() function
- changed order of this file (was getting too long) :-)
- fixed configuration of directory for temporary files
- fixed logfile message in report generation
- added report-msg for broken contents dir in verify

3.10.15 Changes from 1.2.4 to 1.2.5

- multibuf seems to work with all types of tape drives now
- made code more portable (if statements)
- better handling of `SIGINT`
- bug report can be saved to a file now
- detection of configuration change and prompt whether to save it
- included icons for HTML version of manual
- cleaned up detection whether device is tape or not
- changed Help-Menu text
- added more info to `MultiBuf` Menu
- added "rewind"-warning
- fixed "mkfs"-bug
- added report generation
- added automatic verify for tape archives
- changed KBackup directory hierarchy, made kbinstall move old files
- fixed some minor bugs
- clarified Shareware-Copyright notice
- fixed sequence number problem in MultiBuf
- some minor fixes on BLOCKDEV code
- some fixes to code handling change of volumes
- removed icons from HTML manual

3.10.16 Changes from 1.2.3 to 1.2.4

In 1.2.3, the autodetection code was missing and there was a problem with multibuf on SCSI tapes as well as minor problems with the installation program. I hope to have them fixed in 1.2.4 (released Jan, 18, 1996).

- - added "fmark-fix" to multibuf: retries reading if 0 bytes read (should?!? skip filemarks reliably on non QIC-02 tapes)
- - added warning about erasing DAT drives
- - fixed some bugs in kbinstall: writes proper pathnames to src/paths
- - added missing autodetect-src to archive

3.10.17 Changes from 1.2.0 to 1.2.3

Most of the changes are only minor improvements or bugfixes:

- fixed $findxpertflags= assignment
- added lots of doublequotes in assignments to improve safety
- minor changes to restore and verify routines to share common code
- fixed bugs introduced by new doublequotes in read-data/write-data
- made it able to autodetect and read non-multibuf archives (V1.0/V1.1 type)
- (re-)added menu to switch off MultiBuf use
- fixed a few bugs in read-data/write-data
- started work on improved afio
- directory requester is sorting entries now
- fixed a missing close() in multibuf.c
- added last-command variable to be used by check-error
- introduced common "¡" key to go back to previous menu
- made directory requester show symlinks
- fixed Goto-Operation to handle archives with any count of files on the tape — important for mixing multibuf/non-multibuf archives
- changed copyright to BSD like

New features that have been added, are:

- first try at remote device support via rsh(1), works fine on my machine
- added autodetection/check tape & restructured device menu

3.10.18 Changes from 1.1 to 1.2.0

Among them...

- fixed the directory–exclusion bug, completely rewrote routines for including/excluding files
- added pattern matching capabilities
- created MultiBuf , giving automatic end-of-tape detection
- started modularising the software for faster start up (this also enables you to save disk space by leaving out unnecessary options, important for rescue–disks)
- fixed code for verifying archives
- restructured menus
- KBackup now handles filenames containing whitespace properly
- support for mountable block devices
- improved installation program
- changed archive formats (sorry)

Chapter 4

Making Backups

4.1 Creating a backup

. . . is rather simple. Make sure that you have set the parent directory for your backup to the directory whose contents you want to back up (You can set this in the Configuration– menu. Then select Backup in the main menu.

If all the options are set to proper values, this will work. Easy, isn't it? :-)

4.2 Backup types

In principal, there are two different classes of backups. Full backups or incremental backups . During a full backup, all files specified get backed up, while in an incremental backup you will only find those files modified after a certain date or time. This usually is the date and time of the last backup. Doing incremental backups can save you lots of time and space on your backup medium, but for a full restore you will always need the first full backup as well as all incremental backups. So better do a full backup from time to time or you might run into trouble.

There is *no* difference between normal and multivolume backups. All backups are handled by the MultiBuf program.

4.3 Incremental backups

To save only the files that are newer than a given date and time, KBackup provides incremental backups. In the Configuration/File selection –menu you can set the date to compare the files to. Whenever you create a backup using the same configuration, this date gets set to the current time. If you make a backup using the configuration "Default", this does not affect the date for the configuration "Example" and so on.

Caution: Using incremental backups can lead to inconsistencies! After restoring them, all files get written back to the filesystem, but files that have been deleted intensionally will get rewritten as well. Therefore you will generally have more files than before which might cause unexpected behaviour from some applications, especially if you restore lock–files. If you can, better do full backups.

Chapter 5

Restoring Backups

5.1 Restoring Backups

For restoring backups it is important to understand that the files get restored relative to the current parent directory. For this reason you are prompted to change or confirm it before any files get restored. The original parent directory under which the contents of the archive were located at the time the backup was made, is used as the default; so you do not need to remember it by yourself.

5.2 Selecting files for Restore operation

Before restore operation starts you can choose between several options.

5.2.1 Restore all files from archive

This does just what you expect from it. It restores the complete archive. If any of the files do already exist, they get overwritten.

5.2.2 Initialise editable list of files from contents directory

This reads in the contents directory from the archive (if it is not a simple floppy archive) and lets you edit this list. You can then manually remove all files that you do not want to restore. After you save the file and exit your editor, those files remained in the list will get restored.

5.2.3 Initialise editable list of files from log file

This is like the operation described above, but instead of initialising the list from the contents directory, it gets initialised from a log file.

Chapter 6

Other Actions

6.1 Tape Operation–Menu

All functions from this menu operate on tape drives only. They do not make any sense on archive files or floppy drives.

Most of them are simply a front end to the UNIX `mt(1)` command.

Rewind Tape

Obviously, this will rewind your tape.

Try to Reset Tape Driver

This will try to read from the filename given for resetting tape drives. Such a device is only implemented for the `QIC-02` driver and it is usually called "`/dev/tape-reset`".

If your `QIC-02` drive does no longer seem to work, this might help.

Go to n-th Backup on Tape

This will wind the tape to the beginning of a specified backup.

Note: This does *not* go to the `n`–th *file*, but to the `n`–th *backup*! It can handle backups with and without sequence information mixed in any order on one medium. *This will not work properly with backups written by other programs!*

Go to End of Written Data

This will wind the tape to the end of written data, for appending data to a tape without overwriting its contents.

Erase Tape

This will erase the whole tape.

Retension Tape

This will wind the tape to its end and back to its beginning to retension it.

Set tape offline: rewind and unload it

This will issue the `mt offline` command to the tape device. On devices that support this, it will cause the tape cartridge to be rewound and unloaded.

6.2 Other Actions

This section describes the other options from the Actions –menu.

Verify Backup

This will try to verify a backup. It also shows valuable information about the backup type, date ... – everything it finds in the backup's header.

Get Backup Size

This estimates the amount of media space required for doing a backup with the current settings. It simply adds up the file sizes of all files specified for backup. The size given is the *uncompressed* size of the backup, if using compression you can roughly expect it to be half or one third as big.

View Archive Contents

This will show an archive's contents directory, or if nonexistent, will generate a list of files in the archive.

View a Log file

This shows you a list of all log files and you can view them.

Delete a Log file

This shows you a list of all log files and you can delete one of them.

Send a Bug Report

To mail a bug report to the author[1] of KBackup , you should use this function. It will create a list of the configuration settings and collect some information about your system's setup and prompt you to fill in a bug report which subsequently will get mailed to me. Of course, this can also be used for suggestions how to improve KBackup ! :-)

[1] mail://Karsten.Ballueder@stud.uni-karlsruhe.de

Chapter 7

Scheduled Operations

7.0.1 Introduction

KBackup offers you the possibility to schedule a backup or restore operation to be executed automatically (by using the `at(1)` command) at any later time.

For scheduled operation to work properly, you need to have the `crond(8)` – demon running. Also the `at(1)` program needs to be available and a suitable entry be in your `crontab(1)` file. This should be the case on any properly installed system, so do not worry about it unless you get error messages.

By setting the commands to execute by MultiBuf , you can even support automatically changing tapes if your hardware supports this. [1]

7.0.2 The Scheduling–Menu

This menu provides rudimentary support for scheduling for those who do not know anything about shell programming. A possibly easier way is to create a configuration of your choice and use the `--auto-backup` option described in section 9.1.1. Thes Scheduling–Menu is left over from earlier times of KBackup and not really needed any longer.

Set date and time

Here you can set the date and time at which to execute the operation in a format suitable for the `at(1)` command. See its man-page for details.

Schedule a backup

After choosing the Schedule Backup function, KBackup terminates and will perform the backup at the time specified.

[1] If you do so, I'd appreciate if you could tell me about it.

Schedule a restore

After choosing the Schedule Restore option, KBackup first needs to read some information from the archive, so it must already be available. Further on you get prompted the usual Restore options. Then KBackup terminates and the files specified will get restored when their time has come.

Toggle Once Only flag

Normally, after having performed the scheduled operations, KBackup will automatically remove the temporary configuration files that it had created for this purpose.

If you schedule KBackup to run more than once, e.g. every night, youu must activate this flag, so it will keep the configuration files required for scheduled operation.

Chapter 8

Configurations

8.1 The rôle of Configurations

When you want to create a backup, you are confronted with lots of options that affect which files get included, the way the archive is written, compressed or whatever. Instead of having you set these options manually whenever you write a backup, KBackup lets you create named configurations.

For example, you can set all the options and the parent directory to back up your `home`–directory in a compressed archive to tape. Now you save these settings under a name (e.g. "Home"). Whenever again you want to make a backup of your home directory, all you have to do is enter the Configuration –menu and load the configuration "Home". This will automatically restore the settings you had chosen.

When using incremental backups, the configuration gains some more importance. The date and time to compare the files to while creating the incremental backup, are dependent on the actual configuration. So making a backup with the configuration "A" does not affect the date and time for incremental backups using configuration "B".

8.2 How to create a Configuration

That is pretty simple. Just set all options and parameters in the Configuration –menu as you like them and then choose Write a Configuration File from it. You will then be prompted a name for it.

8.3 Load a Configuration

If you choose the Load a Configuration entry from the Configuration –menu, you will be shown a list of available configurations.

8.4 Where to start?

When you install KBackup for the first time, it will create some example configurations for you. Try modifying these to fit your situation.

8.4.1 Configuration `Default`

This is a configuration for using tapes. It assumes that you have a link called `/dev/tape` pointing to a non-rewinding tape device. It is a good starting point for creating your own configurations.

8.4.2 Configuration `Floppy-and-Blockdevices`

If you want to use floppy disks, this is the configuration to use. It will not write straight to the device but create a filesystem on it and write to that. This is safer than writing straight to the device. This configuration can easily be adopted for use with removable hard drives.

8.4.3 Configuration `ZIPdrive`

This configuration file has been sent to me by a user who uses an IOmega ZIPdrive^{TM}. Try whether it works for you. It has not been tested with multiple volumes yet.

Chapter 9

Options Overview

9.1 Command Line Options

9.1.1 –auto-backup Config

This will cause KBACKUP to run in "auto–mode", without user interaction. This mode is especially useful for making automated or script controlled backups. KBACKUP will load the configuration given with the `--auto-backup Config` option and automatically do a backup using that configuration.

9.1.2 –auto-restore Config

Like the `--auto-backup` Option, this will cause KBACKUP to run in "auto–mode", without user interaction. KBACKUP will load the configuration given with the `--auto-restore Config` option and automatically do a backup using that configuration.

9.1.3 –ttymode

`--ttymode`

This will cause KBACKUP to run in "tty–mode", not using the `dialog(1)` program, while still offering full functionality. Generally this will not be needed, but it might be helpful in debugging (e.g. using `script(1)`), for porting it to other systems but Linux and for creating a boot disk without the rather big `dialog(1)` binary.

If `dialog(1)` is not installed on your system, KBACKUP will detect this and automatically start up in "tty–mode".

Well, in fact, there are some other options, but they are not meant to be used by you. In order to recognise what to do when called in scheduled operation mode, `atrun(8)` calls it with some options. You might, however , use them to perform a scheduled operation if your `at(1)` `/crond(8)` does not work. Have a look at the source to see what they do.

(e.g. kbackup –scheduled-backup .–scheduled–1234 to perform the scheduled backup number 1234)

9.2 Options set from within KBackup

All configuration settings are done within KBackup . You reach them by selecting the Configuration –menu.

Set Parent Directory

This allows you to set the parent directory for all backup operations. If you do a backup, only files under this directory are seen. If you restore from an archive, the files get positioned relative to this path.

If you want to make a backup of your "`/home`" directory tree only, set this to be your parent directory. So KBackup will back up only these files.

Configuration–Menu

This gets you to the Configuration –menu, where you can configure the way KBackup behaves.

Set File Selection Parameters–Menu

All settings that can be made in this menu work together to decide which files will get backed up and which won't. See the next section for more details.

Configure KBackup –Menu

In this menu you can set options that influence the way KBackup works, like archive format, compression, ...

Write a Configuration File

Do I really need to explain this? Just give a name for this configuration.

Load a Configuration File

You will see a list of configurations. Choose one.

Delete a Configuration File

You will see a list of configurations. Choose one to delete it.

9.2.1 Set File Selection Parameters–Menu

Toggle Local Filesystem Mode

If set to YES, KBackup will backup only files within one filesystem. If you want to backup a whole directory tree with all volumes mounted within, you must set this to NO.

Toggle Incremental Mode

If set to YES, backup only files that have been changed after the date given. See section 4.3 for more information.

Set Date For Incremental Backup

If in incremental mode, KBackup will only backup files that are newer than this date. Here you can change it. This date gets updated every time you perform a backup with the same configuration.

Edit List of Files to Exclude

There are always some files you do not want to include in a backup. Examples might be: `proc`, `dev`, `home/fred/secrets`

Give their names relative to the parent directory and list them one per line. See the default entries for more examples. Lines beginning with a "`#`" are ignored and can be included as comments. *This list gets interpreted always, no matter whether you use pattern matching or not.*

An example:
You have set your parent directory to "`/home`", and the exclude list contains the entries:
`fred/secrets`
`guest`
This will backup all files of the "`/home`"–tree, except for the trees "`/home/guests`" and "`''/home/fred/secrets''`.

Set shell pattern for files to exclude

Instead of listing all files to exclude from the backup separately, you can exclude files by using the shell's pattern matching rules. By using expressions like e.g. "`*.log`", you can exclude all files matching this pattern.

In the list, patterns must be specified one at a line. Lines starting with "`#`" are ignored and can be used for comments. For the patterns to take effect, they must be explicitly activated.

Set shell pattern for files to include

Like using shell pattern matching to exclude files from the backup, you can also specify a pattern for the files to include. *The standard setting should be "`*`". Do not leave this empty when using pattern matching!* A useful(?) example of this would be to backup only your TEX–files, by specifying "`*.tex`".

Toggle use of exclude/include patterns

With this switch you decide whether KBACKUP uses the exclude and include patterns or not. Usually this is switched off so the patterns get ignored. This does not influence interpretation of the list of files to exclude.

Protect newer files during restore

This will cause KBACKUP to restore only files that have not been changed since the backup was made, thus protecting any files that are newer than the backup from being overwritten.

Set Advanced Options–Menu

This menu gives you deeper control about the way KBACKUP looks at your files. Normally you will not need to use it.

9.2.2 Configure KBackup –Menu

In this menu you find the settings that control KBACKUP 's operation, apart from file selection.

Create Report on Backup

This leads to an input box where you can specify a command for handling report files. If you leave this empty any reports will be discarded. Otherwise, KBACKUP will create a detailed report during backup operation and process it with this command. It can for example be used to print a report to store it with your backup medium, or to mail the report to the system administrator.

Set Name of Editor to use

If you do not have your `EDITOR` environment variable set and do not like the standard setting (`vim(1)` , the "vi–improved" [1], just tell KBACKUP about your favourite editor. If you like the feeling of `DOS` –editors, try using `joe(1)` .

[1] Apparently, there is a problem with `vi(1)` that occasionally makes it unusable from within KBACKUP . All other editors I use work fine.

Follow Backup/Restore–Process on Screen

This will show you file by file how KBackup processes the archives.

Toggle use of restore–log file

With this switch you can decide whether KBackup creates a log file during restore operation. Disabling log file creation can be useful when disk space is few, like e.g. when running from a floppy disk.

Set Compression/Encryption Mode

Here you can choose whether you want to compress or encrypt your archives. You can use either `compress(1)` , `gzip(1)` or `pgp(1)` .

In the menu for choosing the compression/encryption filter you can also pass any useful command line options to the filter. A rather useful one is the parameter "`-1`" for `gzip(1)` . A `gzip`-compressed file with option "`-1`" is about 10–20% larger than with the standard compression, but compressing it is about twice as fast.

Do not be surprised, compressing a long backup can take quite some time (during which your tape might not operate if you are using double-buffering).

A word about encryption: You can use `pgp(1)` public–key encryption with both, `tar(1)` and `afio(1)` . Using `afio(1)` is *highly* recommended, unless your archives will be small. Using `tar(1)` together with `pgp(1)` encryption limits your maximum size for archives to the free disk-space in your temporary directory.

When using encryption, you will be prompted for a default pass-phrase whenever you try to read the archive. This will temporarily get stored in a file unreadable for others. If you leave it empty, `pgp(1)` will prompt you for the pass-phrase again and again with every singe file it decrypts. *Please*, do read the `pgp(1)` manual page and documentation before using it!

There is also a problem with old versions of `afio(1)` and `pgp(1)` : If the encrypted file is bigger than the original one, current versions of `afio(1)` will store it unencrypted! So check your logfiles carefully when using encryption. *Please* use at least `afio(1)` version 2.4.2 when using encryption.

For legal reasons, I cannot include `pgp(1)` with KBackup . Its several versions are available from ftp–servers all over the world, e.g. from ftp.th-darmstadt.de[2].

Toggle Use of Double Buffering

If your machine is slow or heavily loaded or if your tape drive is too fast, it will start and stop the tape every now and then. This is unhealthy for both, the tape

[2] ftp://ftp.th-darmstadt.de/pub/security/pgp

and the drive. To avoid repeated starts and stops of the drive, you can use double-buffering. By collecting a large buffer of data and writing it at once, you reduce the number of starts and stops.

In fact, starting with release 1.2.9 of KBackup , the term "double buffering" is no longer correct. MultiBuf does handle this now and allows you to have an arbitrary number of buffers in memory that get filled and written one after another. This will keep your tape drive streaming most of the time.

Note: this is only intended for use with tape drives. For all other media this hardly makes sense, so better do not try it then.

This is extremely useful when using compression or encryption because these operations reduce data throughput. However, you will see how your tape drive performs without double-buffering and then you can decide whether to use it or not.

Set # of buffers for double buffering

MultiBuf allocates several buffers of shared memory for double buffering. The size of each buffer is the product of the blocksize and the number of blocks given in the Multibuf Options –Menu. An example: If you set the blocksize to 10240 and the number of blocks to 100, MultiBuf will have a buffersize of $10240 \times 100 = 1024000$bytes. If you choose 10 buffers in this menu, you will have approximately 10 Mbytes of buffers which should be a good setting for slow tape drives on mashines with enough physical memory. When not using compression, you might not need any double buffering.

Verify Archives After Writing

If this option is set, KBackup will try to verify the archive while or immediately after writing it. When using floppies , you should activate this. On tapes, this will not work (it is physically impossible), so you have to verify your tape archives manually, using the Actions/Verify –function.

Archive Type Menu

Here you choose which archive type to use. KBackup uses either `tar(1)` –format or `afio(1)` –format. See section 9.2.6 for more.

Change KBackup Directory

Here you tell KBackup in which directory to store its configuration files and log files. The default place is `/var/adm/KBackup`. *This does not create the necessary hierarchy of directories.* You should exit KBackup and copy the old directory tree to the new location. Actually, I am not sure why I implemented this option at all, but maybe you want to use it someday.

Select device for backup–Menu

In this menu you can select which device to use for your backups. It is also being used to tell KBackup how to treat the device, whether as a tape, a simple floppy disk or a mountable blockdevice.

Remote mode settings–Menu

This is for the (still experimental) support for accessing remote backup devices, that is, devices of another machine. The way it works is very similar to the `rmt(8)` demon. KBackup will call a remote shell and redirect the data to a MultiBuf process running on the remote machine.

On the remote machine, at least MultiBuf , `dd(1)` and the archive handler (`afio(1)` or `tar(1)`) have to be installed. It is best, if you install the complete KBackup package on both machines to make sure everything is available.

NOTE: Remote mode does currently not work with autodetection of devices (it will always detect local devices only) and will not work with multivolume archives. Therefor all your archive has to fit on a single volume! This will be fixed in a later release.

Toggle use of remote mode

To choose whether to use remote access or not. Usually switched off.

Set hostname to connect to

Here you must specify the machine that hosts the backup device.

Set user to use for login

For logging in to the remote machine, KBackup needs a user id. This user must have access to the backup device and the relevant programs.

Verify whether settings work

This will try to access the remote device to check whether your option settings work.

MultiBuf Options–Menu

In this menu you can set options related to the MultiBuf program which is responsible for handling multivolume archives. It does however handle all archives, so better have a look at the relevant section (10).

The options that you might like to change are the blocksize and the number of blocks. The blocksize is the atomar unit of data MultiBuf deals with. Some tape drivers (e.g. `ZFTAPE`) require this to have a certain size, 10240 by default. The number of blocks determines how big the memory buffer is that each instance of MultiBuf allocates. For more information, see also the sections 9.2.2 and 9.2.2 about double buffering.

Usually, you should not change any of these options. If you encounter problems with the MultiBuf program, you can disable its use here.

Set Directory for Temporary Files

Here you set in which directory KBackup stores temporary files. This should point to your "`/tmp`" directory.

Set command for viewing online manual

Here you can specify the command to use for displaying the online versions of this manual. Depending on your taste this can be set to use the plain text version or a hypertext browser for the HTML version.

9.2.3 Select device for backup–Menu

In this menu you can select which device to use for your backups. You can either choose one device from the menu or specify your own file name. *This does not install any driver!* You need to have a driver for your device in your kernel, so KBackup can talk to its device file.

You can also choose between different ways of how to treat the device. KBackup currently supports three different ways of archive treatment, which can be set with the Set Device Type –option:

The new Linux floppy driver supports high volume floppy formats of up to about 1.9 MB per 3.5" HD floppy, or 3.8 MB per 3.5" ED floppy. If the corresponding entries are set in your `/etc/fdprm` file, they can be used from KBackup . It will automatically call the `setfdprm(1)` program for you. For more information on Linux floppy formats, see the documentation of the `fdutils` –package, available from: ftp://sunsite.unc.edu/pub/Linux/system/Misc/fdutils-4.1.lsm

Attempt to autodetect tape device

This option requires a tape cartridge in your tape device and will scan your system for valid tape devices and try to find out the one to use with KBackup . This should solve the problems of all those people who tried using KBackup with a rewind-on-close device. After examining your system, KBackup will prompt you with a list of usable devices.

Caution: This will write to the tape and overwrite any data on it!

Check tape drive for compatibility

This will check the currently selected device for compatibility with KBackup , that is, it will verify that it is a non-rewind-on-close device. If you cannot read your archives properly, check your device with this function. *Caution:* This will write to the tape and overwrite any data on it!

Set device type

With this switch you can toggle between different ways of treating the device. You can choose between `TAPE`, `FLOPPY`, `BLOCKDEV` and `FILE`.

- `TAPE:` This should be selected when you are writing to a tape device.
- `BLOCKDEV:` This format is new since KBackup Version 1.2. With this setting, KBackup will format and mount the device and then start writing to it. This gives you all the advantages of KBackup , like a backup header with information about the backup and the contents directory, together with a *much* improved reliability compared to the `FLOPPY`–format. If you are using floppy disks with this option, you *must* low-level format them, using the `fdformat(1)` or `superformat(1)` commands (under Linux). You do not need to write a filesystem on it, this will be done by KBackup . When selecting this option, you also must set the filesystem type and the formatting command.

  I recommend using this format for any removable media, like floppies or removable hard-drives [3] or cartridges. It does *not* work with tapes, it is intended for block devices that can be mounted. *The BLOCKDEV support was new in version 1.2.0 and is therefore less tested than TAPE of FLOPPY. If you should expect any trouble using it, please contact the author. If it works for you, I'd like to know as well.*
- `FILE:` This is will make use of the Filename for backup files setting. KBackup will write the archive to a couple of files starting with the given name and having distinctive extensions. Note that any existing backup with that name will get overwritten. I have created this archive type to allow people to put multiple backups onto one filesystem, e.g. on a mounted removable hard-drive. A typical use would be to `mount(8)` your medium at `/mnt` and set the filename to `/mnt/Foo-Archive`. If you want to add another backup to the disk, you just need to change the archive name and the backups will not interfere with each other.
- `FLOPPY:` With this option, KBackup writes directly to the floppy. This is most reliable when you are using `afio(1)` – better do not try to use `tar(1)` with floppies. However, writing straight to a floppy is not to safe

[3] e.g. the IOmega ZIPdrive

under LINUX and there is no way for KBACKUP to write a header or a directory of contents to the floppy. So you just have a sequential archive, like when using `afio(1)` directly.

However, using this format is not recommended, better use the `BLOCKDEV`–format which is more reliable.

Enter your own device/archive file

Here you can enter a device name of your own choice.

`/dev/fd0` and `/dev/fd1`

Select the first or second floppy drive for your backups.

Blockdev filesystem type

This is only of interest, if you are using the `BLOCKDEV` device type.

Here you need to specify the filesystem type that you want to use on mountable block devices. See the following section for more details and a discussion of which filesystem to use.

Blockdev make filesystem command

This is only of interest, if you are using the `BLOCKDEV` device type.

Before using any medium, KBACKUP will issue this command to make a filesystem on it. This filesystem must be the same as specified in the Blockdev filesystem type , otherwise KBACKUP cannot mount the device.

When using floppy disks, you can low-level format them and make a filesystem on them in one step, like e.g. this:

```
fdformat /dev/fd0H1440 && mkfs -t minix /dev/fd0H1440 1440
```

Instead of specifying the device name directly, you can use "`\$device`", which will get replaced by the actual device name.

For floppy disks, I recommend using the "`minix`" filesystem because it gives you the biggest capacity. If you are using `superformat(1)` instead of `fdformat(1)` , you can write up to 1.9 Mb onto a single HD floppy disk.

For other media, you should probably use the "`ext2`" filesystem which is the most advanced and secure one available under LINUX .

Set floppy format–Menu

In this menu you can choose a format for your floppy disks. *Warning:* this setting will not get checked before it is actually used, it *must* be a valid entry from your "`/etc/fstab`" file which will be set using the `setfdprm(1)` command.

9.2.4 Set waiting time for device

It appears that some drivers (e.g. the QIC02–driver) cannot cope with commands following each other to quickly. Therefor KBackup has this option of waiting a few seconds after each tape operation.

Set device for resetting driver

If your tape driver implements a device to reset the driver, like e.g. the `QIC-02` driver does, you can set its name here. For the `QIC-02` driver, this is `/dev/tape-reset` .

9.2.5 Remote mode settings–Menu

What is it?

With the remote mode enabled, you can access another computer's devices for use with KBackup . **IMPORTANT:** On the remote computer, MultiBuf *must* be installed, or the remote mode won't work. It is best, to install the complete KBackup on both machines, including also the tools used by it.

Toggle use of remote mode

Normally, this is set to "NO", so to access your local devices. If set to "YES", KBackup will try to access the devices on the remote host.

Set hostname to connect to

Set this to the name of the host, who's backup device you want to access.

Set user to use for login

Here you have to specify the username to use for logging in to the remote host. This user must have access to the backup device and must not be prompted for a password, which can be arranged by setting up a `.rhosts` or an `/etc/hosts.equiv` file.

Verify whether settings work

This will try to log on to the remote computer using the settings specified in the menu. If it is successful, a backup will be likely to succeed as well. If not, you have no chance to write a backup, so you need to change your settings.

9.2.6 Archive Type Menu

Here you choose which archive type to use. KBackup uses either `tar(1)` –format or `afio(1)` –format.

`tar(1)` has been around since the early days of Unix , while `afio(1)` is newer and not installed on every system. The advantage of `afio(1)` is that it is smarter about corrupted archives, so if you have a read error somewhere in your backup, it will still continue and recover as many files as possible, while `tar(1)` would just abort. Further on `afio(1)` does compression file by file which is highly recommended for reasons of safety. The default format is `AFIO`.

Use Afio

Obviously, this makes KBackup use `afio(1)` for handling archives.

Use Tar

Obviously, this makes KBackup use `tar(1)` for handling archives.

Set additional command line parameters for Afio

This allows you to pass additional command line parameters to `afio(1)` . KBackup by defaults sets this to `-T 3k` to compress only files which are larger than at least 3 kbytes.

Set additional command line parameters for Tar

This allows you to pass additional command line parameters to `tar(1)` .

9.2.7 Advanced Options–Menu

The options in this menu deeply affect the operation of KBackup . You should know what you are doing before making any changes.

Backup Only Files Specified

If set to "YES", KBackup will only backup those files that are specified in the list of files to include, that you can edit from within the Advanced Options –menu.

Edit List of Files to Include in Backup

If you have activated the option to backup only the files specified, you can use this menu entry to edit the list. A typical situation is when you want to save several complete directory trees at once. The list of files might then look like this:

```
/etc
/home
/usr/local
```

Initialise List of Files to include

This creates a default list of files, based on the other selection criteria.

Set Find Expert Mode

Well, if you know about the many options of `find(1)` , you can activate this option and use the following one to give the options `find(1)` will use to generate the list of files to back up. *Warning: If you activate this, all other options about file selection will get ignored! The shell patterns and include/exclude lists will not get interpreted either.*

Set Find-Options

If the option above this one is active, you can set the options to use with `find(1)` .

Set Option to Manually Edit List of Files

If this is set to "YES", before doing a backup, KBackup will present you a list of files and let you edit it. Only the files remaining in the list will get processed.

Toggle Directories–Only Flag

If set to "YES", the log files will only show directories, not the actual files. The same applies for the list of files you can edit before backing up or restoring. By setting this option, you can significantly reduce the size of your logfiles or contents directories.

Usually, this should be set to "NO".

Chapter 10

The MultiBuf Program

10.1 What is MultiBuf ?

MultiBuf is an support program written especially for use from within KBackup . It is covered by the same copyright and conditions of use that apply to KBackup .

It is intended to be a reliable and transparent handler for multivolume archives, superior to both `afio(1)` and `tar(1)` .

In addition, it provides the capability of double–buffering to keep your tape drive streaming continuously, therefor greatly extending the life of both your tapes and your drive.

One of its most most advanced features is, that it automatically determines the length of your tapes. You can write multiple volume archives to a series of tapes having different lengths. You can also use MultiBuf to automatically change tapes if you have an automated tape changer. [1]

The information in this section is only of interest, if you want to understand how it works or use it as a standalone program for some other purpose. You do not need to understand this to use KBackup , as the use of MultiBuf is fully automated. However, if you are concerned about compatibility, the data format used or restoring files without KBackup , continue to read this section.

10.2 How does it work?

MultiBuf can be run in two modes, "write–mode" (default) and "read–mode". In "write–mode", it reads data in small blocks from `stdin` and writes it in bigger

[1] This simply involves redefining the commands used to interact with the user. Instead of prompting for the next tape, they can autoload it. If you are interested in this feature, please contact the author, so we can fit this feature into the official release. I'd be glad to help you with this in any way.

chunks ("superblocks") of several blocks to the output device specified. In "read–mode", it does the contrary, reading big chunks of data from the device and outputs the data to `stdout`.

Whenever writing a big data chunk fails, MultiBuf assumes that it has reached the end of the backup medium (e.g. end of tape) and executes one of two user-definable commands. On successful execution of this command, it optionally executes a second command and then continues writing to the device with the last block that did not fit onto the medium before. During restoration of the data, it does more or less the same, prompting the user when it cannot read a complete chunk of data. *It is important to mention, that for writing and reading data, you need to use the same block sizes and number of blocks per superblock!*

To improve reliability and error checking, MultiBuf can write so-called "sequence information headers" to the tape. These headers tell MultiBuf about the right order of backup media and thus prevents you from using the tapes in the wrong order. Further on, this enables MultiBuf to distinguish between an "end of medium" and an "end of file", sparing you some unnecessary and possibly confusing dialogs. Also, this increases the number of files per backup on a tape from currently three to five, because two additional short end–markers get written. On mountable block devices MultiBuf will write the end markers as separate small files.

10.3 The Multivolume Archive–Menu

This section describes the Multivolume Archive –menu of KBackup , from which you can change the MultiBuf related options.

10.3.1 Set MultiBuf block size

This determines the smallest block of data, MultiBuf will deal with. It *should* be an integral multiple of the block size of your tape. For `ZFTAPE` users, this should be set to the blocksize that `ZFTAPE` requires, which is 10240 by default.

10.3.2 Set number of MultiBuf blocks per call

To avoid unnecessary overhead, MultiBuf always writes (or reads) a couple of data blocks at once to (from) tape. Making this value large slightly reduces CPU usage, but makes it waste a bit of space at the end of every tape. I suggest using a value that is both significantly lower (when multiplied with the block size) than your tape's capacity as well as much bigger than the relatively small block size. The standard value is 100, giving you (with a block size of 1024) a total buffer size of 100 Kbytes. So you can be sure to waste no more than 100 K per tape and still reduce CPU overhead by two orders of magnitude.

10.3.3 Use sequence information in MultiBuf

This has two effects:

- It enables MULTIBUF to distinguish between the end of a file on the tape, like e.g. the end of the archive or contents directory, from the physical end of the tape. If you disable this switch, MULTIBUF will ask you whether to continue with the next medium whenever it detects an end of file. *So you have to explicitly tell it "NO" after it has read the contents directory or after having read all the archive!*
- It writes some sequence information to the tape and thus prevents you from using your tapes in the wrong order.

10.4 MultiBuf Command Line Options

When used with the option **`--help`**, MULTIBUF will show you a screen le this:

```
MultiBuf $Revision: 1.1.1.1 $      (C) 1995-1997 by Karsten Ballder
                           (Ballueder@usa.net)

USAGE: /valid options:
   -r, --read             : set read mode
   -w, --write            : set write mode (default)
   -v, --verbose          : verbose progress reporting
   -b, --blocksize        : set size of each data block in bytes
   -n, --nblocks          : set number of blocks in each superblock
   -m, --multibuf         : enable multibuffering with this many buffers
   -c, --command "shell-command"  : set command to execute at end of medium
   -C, --command2 "shell-command" : set command to execute at begin of medium
   -R, --retry-command "shell-command" : set command to execute when finding
                                          wrong sequence number
   -1, --end-only         : only execute command at end of media
   -2, --begin-and-end    : exeute commands at end and begin of media
   -s, --settings         : display settings
   -x, --exit             : exit after processing options (for testing)
   -h, --header           : add header/sequence information to data
   -E, --no-endmarker     : do not use an endmarker
   -F, --file-endmarker   : write endmarker as a file on a mounted medium
   -p, --disable-padding  : disable padding of incomplete input blocks
   -D, --disable-filemark-fix : for debugging purpose only
   -V, --version          : print version information

   return codes: 0=success, 1=read error, 2=write error, 3=wrong sequence
```

Appendix A

Frequently Asked Questions

A.1 Introducing the FAQ

This FAQ is not very long at the moment, because most of the questions about KBackup versions 1.0 and 1.1 do not apply any longer. They concerned bugs removed in this version.

However, I am confident that I will receive lots and lots of questions about this version as well, so I have decided to start a FAQ on it. This will be constantly being updated, the latest version of it will always be available via the KBackup homepage[1]. If you have any questions, mail them to me at:
`frascone@users.sourceforge.net`
or use your web–browser:
send mail.

A.2 How can I. . . ?

A.2.1 How can I choose multiple directories?

Just use the list of files to include and give the directory names relative to the given parent directory. A typical setup would be to set the parent directory to "`/`" and the list of files to include to something like

```
home
etc
usr/local
```

A.2.2 Exclude a directory tree from backup?

To exclude a directory and all its sub-directories from the backup, you can list it in the exclude list (Configuration/File selection –menu). This was broken in versions before 1.2 but is fixed now. Just specify the name of the directory relative to your parent directory. E.g. if your parent directory is "`/`" and you want to exclude "`/proc`", just specify "`proc`" only. This would exclude the whole process–filesystem.

A.2.3 Restore my files manually?

Well, that is fairly simple. To restore from a tape, you can do this (assuming that "`/dev/tape` " is a link to your tape device):

First you have to "`cd` " to the directoy to where you want to restore the files. Then:

```
mt rewind               # to rewind your tape
cat /dev/tape           # this will show you the archive header
cat /dev/tape >/dev/null # this will skips the contents directory
```

Depending on whether you used `AFIO` or `TAR` format, do one of the following (using double buffering and decompression):

```
multibuf -r -h /dev/tape | dd obs=4096k | afio -i -vzZk - # for AFIO
multibuf -r -h /dev/tape | dd obs=4096k | tar -xzvf -     # for TAR
```

[1] http://kbackup.sourceforge.net/

The options for `afio(1)` and `tar(1)` depend on whether you used compression or not. See their manpages for details.
For floppy disks, using `afio(1)` is rather simple, just type:

```
afio -i -vF -s 1440 -G /dev/fd0H1440
```

For higher capacity floppy formats, you need to adjust the numbers; without compression, drop the "`-G`" option.
For mountable block devices, this is tricky. You *can* restore them manually. It goes like this:

```
multibuf -r -h -F -c ``...'' -C ``...'' /mnt/Archive | afio -i -vzZ -
```

The arguments to the "`-c`" and "`-C`" must be set to mount and unmount the volume at "`/mnt` ", for example like this:
`-c ``umount /mnt ; echo Next Volume...; read ''`
`-C ``mount /dev/fd0 /mnt -t minix''`

A.2.4 Restore a system no longer booting?

If you cannot boot your system any longer, the first thing you need to get is a Bootdisk–set. This is something you should have prepared just after installing your system. In case that your main LINUX partitions are no longer accessible, not even from a boot disk, you need to recreate them and then you can use KBACKUP to restore your data onto them.

A.2.5 How can I use my removable hard disk?

Use the `BLOCKDEV` archive type and configure the appropiate commands for making a filesystem. This should work with any removable hard disk. If your typical backups are smaller than the medium, you should better use the `FILE` archive type which allows you to fit multiple backups on one disk.

A.2.6 How can I do automatic backups?

Using the `--auto-backup Config` option, it is easy to perform automatic backups or use KBACKUP from a shell script. Just create a configuration, e.g. "`myAUTO`" and call KBACKUP like this:

```
kbackup --auto-backup myAUTO
```

Then it will perform a backup without asking you for anything. It is simple to create a short shell script to check for the right tape, e.g.

```
#!/bin/sh
mt rewind
tape_number=`cat /dev/tape`
required_tape_number=`cat /var/adm/tape_number`
if [ "$tape_number" != "$required_tape_number" ]
then
   exit 1;
fi
echo `expr $required_tape_number + 1` >/var/adm/tape_number
kbackup --auto-backup myAUTO
```

If you prepare a couple of tapes with one file on each of them, containing a serial number, this script can check whether the tape in the drive has the right serial number for the next backup. Of course, you need to prepare the tapes in advance and initialise the file for the serial number properly. With such a mechanism it is guaranteed, that you won't accidentally overwrite yesterday's backup.

A.2.7 How can I use an auto–loader?

Pretty simple, the best thing to do is to proceed like in the FAQ above (How can I do automatic backups?) and just insert a command for changing the tape in the shell script. So before calling KBACKUP , you would just let the auto–loader change the tape and then start the backup. Unfortunately, then you can only do backups that fit on a single tape.
If you want to write longer (multi–volume) backups, you need to know MULTIBUF how to change tapes. In the Configuration –Menues you find an entry for the commands MULTIBUF will execute at the end of a tape. Here you need to enter a command for changing the tape, and everything should work fine.

A.3 Can I. . . ?

A.3.1 Can I make backups of my DOS partitions?

YES. KBACKUP can make backups of any filesystems mounted. To make a backup of your DOS (or whatsoever) partitions, simply mount them to an appropiate place and go ahead.

A.3.2 Can I make backups over the network?

YES. Well, probably. On the KBackup Homepage you can find a version of KBACKUP supporting remote devices. Get it and read its `README` file for details.

A.4 Common problems

A.4.1 I cannot read or verify any tape archive.

Well, maybe you are using the rewind-on-close device. You *must* use the non-rewind-on-close device (e.g. "`/dev/nrm*` "). See also the "No valid KBackup header" question in this section.

A.4.2 I cannot restore my old archives.

If you have trouble restoring your old KBackup archives, generated with earlier version of the software, contact the author.

If you are using KBackup version 1.2.x with $x < 3$, upgrade to a more recent version. KBackup should detect the old archive format and restore it properly.

A.4.3 I get "No valid KBackup header"

If KBackup cannot find a valid header on your backup medium, although you have written an archive to it using KBackup , than there are two possible reason for this:

1. you are using a tape device that rewinds after closing a file. KBackup requires you to specify a device that does *not* rewind on close. These devices often carry an "n" in their name. If you try using a device with rewind-on-close feature, KBackup will overwrite its header with the directory and afterwards overwrite the directory with the actual archive, instead of writing the files one after another.

2. You might not have started your backup at the beginning of the tape. The automatic verify after a doing a backup will rewind the tape before verifying the backup, it will *not* automatically rewind the tape before writing the backup (for reasons of security). So try to manually rewind the tape before writing the backup.

In version 1.2.3, I have introduced an option to check a tape device for compatibility with KBackup and also autodetect the proper device file.

A.4.4 KBackup does not work with FTAPE?

Well, there might be lots of reasons for this. First, try whether it works if you disable MultiBuf support. If that does not help, re–enable it and try playing with its block sizes and also with the block size for writing to tape in the configuration menu. Especially when using ZFTAPE , this parameter must fit the parameter chosen in ZFATPE which is 10240 bytes by default.

A.4.5 I want to pass additional options to `tar` or `afio`.

Someone asked to support the "`-T`" option of `afio` . It's pretty simple, in the Archive Type , you can choose between `afio` and `tar` and also specify additional options to pass to them. The default setting for `afio` is "`-T 3k`" so to compress only files smaller than 3 kbytes.

A.4.6 I cannot use KBackup as a normal user.

Well, ehem, that's one of the dark sides of this program. :-)
I never thought of anyone else but root wanting to use this program, but you can anyway. The default for KBackup is to write all its data to "`/var/KBackup/`". This directory hierarchy gets created with root access only. If you want to run KBackup as a normal user, you need to either give him write–access to these files and directories, or copy everything to a different place and edit the "`.kbackrc`" file in his home directory to point to the new place. Of course, you also need to make sure he has write access to the backup device and that all required programs are in his path (they should be anyway, I believe.).

A.4.7 KBackup does not find any device?

A typical question:

> Installed KBackup - attempting to make work with an internal Iomega "ditto" drive... but doesn't seem to like /dev/tape or any of the other drivers on the system.

Well, that's the wonderful world of FTAPE. If loading the kernel module with the proper driver does not help, try buying a *real* tape drive instead of a toy. :-)

A.4.8 Remote operation does not do multiple volumes.

If you want to write multivolume archives to a remote device, you need to run KBackup in tty-mode. Furtheron you need to manually copy the temporary files "`/tmp/kbtmpXX/mbuf*`" to the remote mashine. This will change in a later release, but currently these two operations will not work together otherwise.

Appendix B

Copyright and Conditions of Use

GNU GENERAL PUBLIC LICENSE
Version 2, June 1991

B.1 Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software–to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

B.2 TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgement or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

B.3 NO WARRANTY

11. 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.

12. 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 PERMITTED ABOVE, 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.

END OF TERMS AND CONDITIONS

Index