

## **Index**

[introduction](#)

[concepts](#)

[installation](#)

[first tryout](#) (1-page primer for the RTFM inhibited)

[file menu](#)

[backup menu](#)

[verify menu](#)

[report menu](#)

[restore menu](#)

[DVD menu](#)

[help menu](#)

[toolbar buttons](#)

[editing backup jobs](#)

[script files](#)

[large backup jobs](#)

[technical notes](#)

[change log](#)

## dkop Introduction

**dkop** is a **Linux** utility program for copying disk files to recordable DVDs. This is a **free open source program** licensed under the GNU General Public License v.2.

Three kinds of backup are available: full, incremental, and accumulate. A **full backup** copies all specified files and leaves no other files on the DVD. An **incremental backup** copies only new or modified files to a prior dkop DVD, bringing it up to date. This is normally much faster than a full backup. Unmatched files on the DVD are also deleted, so that the DVD is left exactly matching the disk. An **accumulate backup** is like an incremental backup, but unmatched files are not deleted.

You **select files** to be copied using a **GUI**. You can navigate through the file system and select files or directories to include or exclude at any level in the hierarchy. These choices can be saved in a **backup job file** to automate recurring backups. If files are added or deleted within an included or excluded directory, the next dkop run will include these changes automatically. You need to revise the backup job only if you make new exceptions.

DVDs can be verified three ways: full, incremental, and thorough. A **full verify** reads the entire DVD and reports any files having read errors. An **incremental verify** reads only those files that have been newly written by a preceding backup job. This is usually much faster while still offering a high level of security. A **thorough verify** reads every file on the DVD and makes a bitwise comparison with the corresponding disk files. This provides an additional assurance that hardware and software are working correctly.

You can **report** all files in a backup job, or all files on a DVD. You can **search** for specific files using wildcards. You can **compare** a DVD with the corresponding backup job, listing all differences: files that have been created, deleted, or modified since the DVD copy was made. This comparison can be done at three levels: a detailed list of files, a directory level summary, or a job level summary.

For disaster recovery or file transfer, dkop has a **file restore** capability. You can select and restore DVD files to their original directories or anywhere else.

**One DVD:** Incremental backups update the same DVD from a prior full backup. This simplifies both backup and restore: you do not need to track sets of full and incremental backup DVDs, and you do not need to restore files from multiple DVDs in correct sequence.

**Backup history log:** Searchable log files are generated with time / date, DVD label, and files copied. You can search the log to find all DVDs with copies of a desired file, using wildcards to simplify the search (e.g. find \*joeblow/\*/myfile\*).

A **script file** can be used to automate backups or run dkop from a shell script.

**Speed:** Incremental backup and verify can take less than a minute if the updated files are within 30 megabytes or so. For larger jobs, the DVD speed determines the time required. With 4x media, backup + verify runs about 150 megabytes per minute.

The **download tar file** contains the source code, a binary executable that may work out of the box, and a complete user guide, including technical information about how dkop works. If the binary does not work, you will need to compile from the source code. This process is simple and is described below under "dkop installation".

## dkop Concepts

The files in a backup job are specified with include and exclude statements. These have filespecs with optional wildcards placed almost anywhere. Examples:

```
include /home/*           # add user files
include /root/*           # add root files
include /shared/*/documents/* # add shared document files
exclude */mp3/*           # exclude files in mp3 directories
exclude */.Trash/*        # exclude trash files
```

The first include adds all files owned by users in their home directories and sub-directories. The second include adds all files owned by root. The third include adds all files under the /shared top directory that also have an intermediate directory named /documents. The two exclude statements exclude files within /.Trash and /mp3 directories at any level.

**GUI interface:** The above statements are normally generated using a standard Gnome file selection dialog. The process is documented in the section on editing backup jobs.

### file selection logic:

```
loop:
  get next control statement, if EOF quit
  if include: add all matching files to backup file set
  if exclude: remove all matching files from backup file set
loop-end
```

**Note that excludes are effective only against prior includes.** They have no effect on following includes, which are processed afterwards. See the section on editing backup jobs. Restriction: include statements must include at least the first directory name (top-level) without wildcards (the GUI file-chooser does this automatically).

### limitations

- max. 200,000 files in a backup job (compile time constant)
- must run as root user to copy protected files (or mount DVDs on some systems)
- supports DVD media only (not CD media)
- not useful for disk imaging (operating system backup)

## dkop Installation

This is a standard Gnome GTK+ application, requiring the same libraries as other GTK applications such as Gimp and Gthumb.

The following install procedure is illustrative only. Many variations are possible.

01. download the tar file dkop.xx.tar.gz
02. choose a directory (this example: /opt/dkop)
03. open a terminal window
04. \$ su root # get root privileges
05. \$ mkdir /opt/dkop # create directory
06. \$ mv dkop.xx.tar.gz /opt/dkop # move tar file into directory
07. \$ cd /opt/dkop # change to directory
08. \$ tar -xzf dkop.xx.tar.gz # unpack tar file
09. \$ ./dkop.x # test if executable runs
10. \$ ./dkop-build.sh # if not, rebuild executable
11. If necessary, install missing packages and repeat build.
12. \$ chown -R root:root \* # make root owner of all files

If the executable runs "out of the box", you do not need to do anything after step 9. If it does not run, use the build script (step 10) to build a compatible version for your system. Recent versions of the Gnu C++ compiler and GTK2 libraries are required. If either is missing, look for g++ and libgtk2.0-dev in your package manager. They should be available even if your system is KDE based. The executable was built using Ubuntu 6.06 (released June 2006). If your Linux distribution is older, dkop may not run "out of the box" with the older libraries. Rebuild dkop so that it links to whatever libraries you have. This may work fine.

### Files included in the dkop distribution:

dkop.cpp	application source code
dkop.x	executable file
dkop-build.sh	shell script to build the executable
dkop-guide.pdf	user guide and help file (this file)
icons/xxx.png	icon files for toolbar buttons
zlib.cpp	collection of GTK and utility functions
zlib.h	C++ header file

After installing dkop, please perform the **first tryout** exercise (next page). This may be all you need at first (if you are like most people and RTFM is a drag). You can enhance your security and ultimately save time if you read this whole ugly document.

### License and Warranty

Dkop is a free program licensed under the GNU General Public License, version 2 (from the Free Software Foundation). Dkop is not warranted for any purpose whatsoever, but if you find a bug, I will try to fix it.

### Origin and Contact

Dkop originates from the author's web site at: <http://kornelix.squarespace.com/dkop>  
Other web sites may also offer it for download. Keep in mind that modifications could have been made. If you have questions, suggestions or a bug to report, you can contact me at: [kornelix@yahoo.de](mailto:kornelix@yahoo.de)

## dkop first tryout

The following short exercise will check that dkop functions correctly on your system and help you become familiar with dkop usage.

1. Load a recordable DVD, but **do not insert the tray** (another program may seize control of the DVD drive, depending on your preference settings).
2. Start dkop (this example assumes executable location is: /usr/dkop/ )
  - if privileges not needed: `$ /usr/dkop/dkop.x`
  - if privileges are needed: `$ sudo /usr/dkop/dkop.x`
3. Select button: [ edit job ]
4. Change the DVD device if needed for your system (default: /dev/dvd)
5. Change the mount point if needed for your system (default: /media/dvd)  
(or add the directory /media/dvd to your file system)
6. Select full backup and full verify
7. Erase the default backup job shown (select and delete, or use [ clear ] button)
8. Select the button [ file chooser ] at the bottom
9. Navigate through the directories and select some files/directories to be copied
  - double-click a directory to open it and enable selection within that directory
  - select one or more files/directories, using left-mouse (or shift+left-mouse)
  - use the [ include ] button to include all selected items in the backup job
  - use the [ exclude ] button to exclude items previously included at a higher level
  - use the [ include ] button to include items previously excluded at a higher level
  - use the buttons at the top to go back up the directory hierarchy
  - use the [ hidden ] button to toggle the display of hidden files
10. select the [ done ] button when finished selecting files
11. Inspect the generated include and exclude statements
  - these may be edited directly if desired (e.g. erase mistakes or redundancies, change the order, or make additions or revisions)
  - re-enter the file chooser dialog if wanted - new choices will be appended
  - cycle between the editor and file chooser as much as needed
12. Select button [ OK ] when done editing the job
13. If errors are shown, select [ edit job ] and fix (remember that exclude statements must follow relevant include statements - excludes are exceptions to prior includes)
14. Select menu: Report > get backup files. Inspect the counts. Be sure the total byte count is within the DVD capacity. Look for zero counts, indicating possible errors. Re-edit the job if needed.
15. Select button: [ run job ]. DVD tray should load and backup should begin. Verification should follow automatically. Check that the error count is zero.
16. Save the job file if desired: menu: File > save job
17. Select button: [ quit ]
18. Next steps: play with incremental backups and reports

## Detailed Usage Instructions

### File Menu

#### **open job**

Open a previously saved backup job file for re-use (edit, run). The default location for job files is /home/user/.dkop (or /root/.dkop).

#### **open DVD**

Open the backup job file on the currently loaded DVD. This file was saved on the DVD when the last backup job was run on that DVD.

#### **edit job**

Opens an edit dialog for the current backup job (from the last job file opened, or from a prior edit). Note: if no file has been opened, internal default data will be used as a starting point.

#### **show job**

List the current backup job data and diagnose any errors.

#### **save job**

Save the current backup specifications in a job file. Default is the same file that was last opened, but you may select any file.

#### **run job**

The current backup job is executed. Backup and verify modes are taken from the job.

#### **run DVD**

The backup job file stored on the DVD is executed. Backup and verify modes are taken from the DVD job. Whenever a backup is performed, the current job file (including any edits that were made) is copied to the DVD.

**Note:** what is copied to the DVD is the current job, not menu commands given manually. Thus, if you load a job file which specifies incremental backup, and then do a full backup using the menu command, the backup job stored on the DVD will still specify incremental. To change the job written to the DVD, edit the job before starting the backup.

#### **quit**

Exit program.

## **Backup Menu**

### **full**

The current backup file set is copied to the DVD fully. All files are copied unconditionally. The DVD is initialized beforehand (fast). For large jobs, additional DVDs will be requested as needed. If growisofs aborts the job (declaring the DVD to be "unknown type" or "not formatted"), the menu command `DVD > format` may fix the problem.

### **incremental**

The current backup file set is copied to DVD incrementally. New and modified files (since the DVD was created or updated) are copied. Files that already match their corresponding disk files are not copied. Any "extra" DVD files (not in the backup file set) are deleted. At the end, the DVD is 100% identical to the backup file set, with the possible exception of files modified during the backup run. See the technical notes for more details about how matching files are recognized and skipped over.

### **accumulate**

Same as incremental, but without DVD file deletions.

## **Verify menu**

### **full**

All files on the DVD are read and checked for errors. DVDs need this extra level of protection, since poor media quality has been a problem. If errors are detected, clean off the fingerprints or discard the DVD. If errors happen on more than 1% of your media, consider getting a new drive (typically \$50) or changing media brands.

### **incremental**

New files on the DVD are read and checked for errors. "New" means any files written by an immediately prior incremental or accumulate backup. Files not touched are not checked.

### **thorough**

All DVD files having the same modification time as their corresponding files on disk are read and compared with the disk. There should be no differences. This verifies that both dkop and the hardware is working correctly. Files that are expected to be different (different mod times) are read and checked for errors, but not compared with the disk.

## Report menu

### get backup files

The backup job include and exclude statements are listed, along with the file and byte counts that are added or removed by each statement. Look for zero counts, indicating a possible error. The disk directories are read and the list of files included in the backup job is saved in memory. This data is used to determine what files are different between the disk and DVD and must be copied for an incremental backup. The file list is static and is not updated by disk activity. The list of "new" files that are checked with an incremental verify is also reset with this command.

### diffs summary

Report the total number of files in each category:

new	on disk, but not on the DVD
modified	on both, but not the same content
deleted	on the DVD, but not on disk
unchanged	on both, with the same content

Differences between the disk and DVD may be caused by disk updates (file additions, deletions, updates, or moves), or by changes to the job file itself.

### diffs by directory

Each directory having differences between the disk and DVD is reported, along with counts of new, modified, and deleted files. The total bytes for new and modified files is also given.

### diffs by file

All files that are different between the disk and DVD are listed in alphabetic sequence within groups for new, modified, and deleted files.

### list backup files

All files in the backup file set are listed in alphabetic sequence. Use this to check that the correct files are being backed-up.

### list DVD files

All files on the DVD are listed in alphabetic sequence.

### find files

Enter a search pattern with optional wildcards (e.g. /home/dir\*name/file\*name). All matching file names on the disk (in the backup job file set) are listed. All matching file names on the DVD are listed. All backup log files are also searched, and those containing the target file(s) are listed (by date / time and DVD label). These files correspond to backup jobs, one-to-one. Use this method to locate all backup copies of a given file or group of files, sorted from oldest to newest. A file may be present in multiple log files for multiple incremental backups made to the same baseline full backup, but it actually exists only once on the DVD, in its latest version.

### view backup hist

All backup history log files are listed (up to 200). These correspond to backup jobs, one-to-one, and contain a list of files copied to the corresponding DVD. The most recent 20 log files are put into a dialog for selection. Select one of these from the dropdown list, or modify the input to select an older file. The text editor gedit is invoked to display the log file. You can page up and down and search for strings using gedit.

Backup log file names are formatted as follows: `dkop-hist-yyyyymmdd-hhmm-dvdlabel`

Note that one DVD having a full backup and one or more incremental backups will have a log file for each backup, showing those files copied for each backup. A file may be present in multiple log files for multiple incremental backups made to the same baseline full backup, but it actually exists only once on the DVD, in its latest version.

### save screen

The main window, where messages and reports are written, is saved in an ordinary text file.



## **Restore menu**

### **setup DVD restore**

Specify the copy-from location (on the DVD), the copy-to location (on disk), and the files to be restored. The copy-from location is the topmost DVD directory of a tree of files to be restored.

example: /home/joeblow/documents # note that mount point is omitted

The copy-to location is an existing disk directory where the tree of files will be copied-to.

example 1: /home/joeblow/documents

example 2: /home/joeblow/documents/restored

In example 1, the restored files will go back to the same place they were when backed-up. In example 2, they will go to a new place.

Files to be restored are specified the same way as in a backup job (see the section below on using the file selection dialog). Use the button [ file chooser ] to start the dialog.

If you need to restore multiple trees of files, you can do this in multiple runs, or you can simply begin the tree at a higher level and use the file selection dialog to specify multiple sub-trees.

### **list restore files**

After performing the setup, use this function to list all matching files on the DVD that will be restored, exactly where they will be restored. You should check this list carefully to be sure you are restoring the correct files to the intended locations.

### **restore files**

When you are satisfied with the restore job specification, use this menu to perform the restore. You will see a running log of the activity. Use the kill button to stop the job if desired.

## **DVD menu**

### **set DVD device**

The DVD device and mount point may be set independently of the backup job. The DVD device and mount point for the current backup job is modified. No mounting is done.

### **set DVD label**

Set the DVD label that will be used for a subsequent backup job. The default is to keep the same label that the DVD already has. A full backup will have :A to :Z appended to the label, to handle up to 26 DVDs within one backup job. An incremental backup will not have the appended letter. The DVD mount command will show this label. If this is not done, "dkop" is the default label.

### **mount DVD**

#### **eject DVD**

Mount a DVD, or eject the DVD tray. Root privileges may be required. If the mounted DVD has been used for dkop before, the date-time of the last backup to this DVD is displayed, and the dkop usage count (no. of backups done to this DVD). This should normally go into the hundreds before the DVD gets tired and starts having errors (which the verify function will detect and warn you about).

### **reset DVD**

This does a hardware reset to the DVD drive. This is sometimes useful if a drive gets locked-up and cannot be ejected using either the dkop eject command or the tray button. This sometimes happens when there is a DVD error or a backup is killed in mid-process.

### **erase DVD**

Writes zeros to the entire DVD surface. This takes 10-20 minutes, depending on the DVD drive speed and medium. This works only for DVD+RW and DVD-RW media. DVD-R media are write-once and cannot be erased. See the note below on privacy and data protection.

### **format DVD**

This uses the dvd+rw-format utility to format a DVD+RW, -RW, or -RAM medium in a few minutes. The entire DVD is not erased. See the note below on privacy and data protection. If Backup > full refuses to start, this format command may fix the problem.

## **Help menu**

### **contents**

Display the help file (this file).

### **about**

Display the dkop program version and date.

## **Toolbar buttons**

### **edit job**

Shortcut to the backup job editor (same as menu File > edit job)

### **run job** and **run DVD**

The current job, or the job on the DVD, is executed.

### **pause** and **resume**

The currently running job or menu function may be paused and resumed. Use this to inspect output on the fly.

### **kill job**

The currently running function is killed. You may need to wait a while for the function to die and screen output to cease. If a backup job is killed, growisofs will gracefully exit in a few seconds, leaving the DVD in an undetermined status. Only the big time-consuming functions have been implemented as separate threads and are killable.

### **clear**

The main window, where messages and reports are written, is cleared.

### **quit**

Exit the application. If the job file has been edited and not saved, you will be given an opportunity to save the changes.

## Editing backup jobs (see screenshot on next page)

Select menu: File > edit job or button: edit job

Fill-in the following items in the dialog box:

DVD device	/dev/dvd
mount point	/media/dvd
capacity GB	4.0
backup mode	check full / incremental / accumulate
verify mode	check full / incremental / thorough

The DVD device and mount point examples shown above may not work on your computer. Add the directory /media/dvd if it does not already exist, or use whatever is already there for DVDs. The DVD capacity may be set from 1.0 to 8.0 GB (for double layer DVDs). Full backups will be limited to this amount. The default of 4.0 GB leaves a leftover space of about 0.7 GB for incremental updates. See the technical notes for more details about this.

## File selection dialog

You may edit the backup file set (the include and exclude statements) directly in the text window. You may also use the browse button to get a standard file selection dialog, with additional buttons: hidden, include, exclude. The hidden button toggles the display of hidden files (file names with leading dots, like .gnome). Select one or more directories or files, using left-mouse or shift+left-mouse, then press the include or exclude button. The selected files/directories will be written into the text window as include or exclude statements. If you select a directory, the entry is modified to add a wildcard at the next level:

selecting directory /aaa/bbb/cc → include /aaa/bbb/cc/\*

You may alternate between editing the text window and using the file-chooser dialog. When you are done, press OK to accept. The include/exclude data will be validated to the extent possible. Go back and re-edit to fix any problems. To change the sequence, cut and paste in the text window. When you are done, use the report functions "get backup files" and "list backup files" **to verify that you have the correct files!**

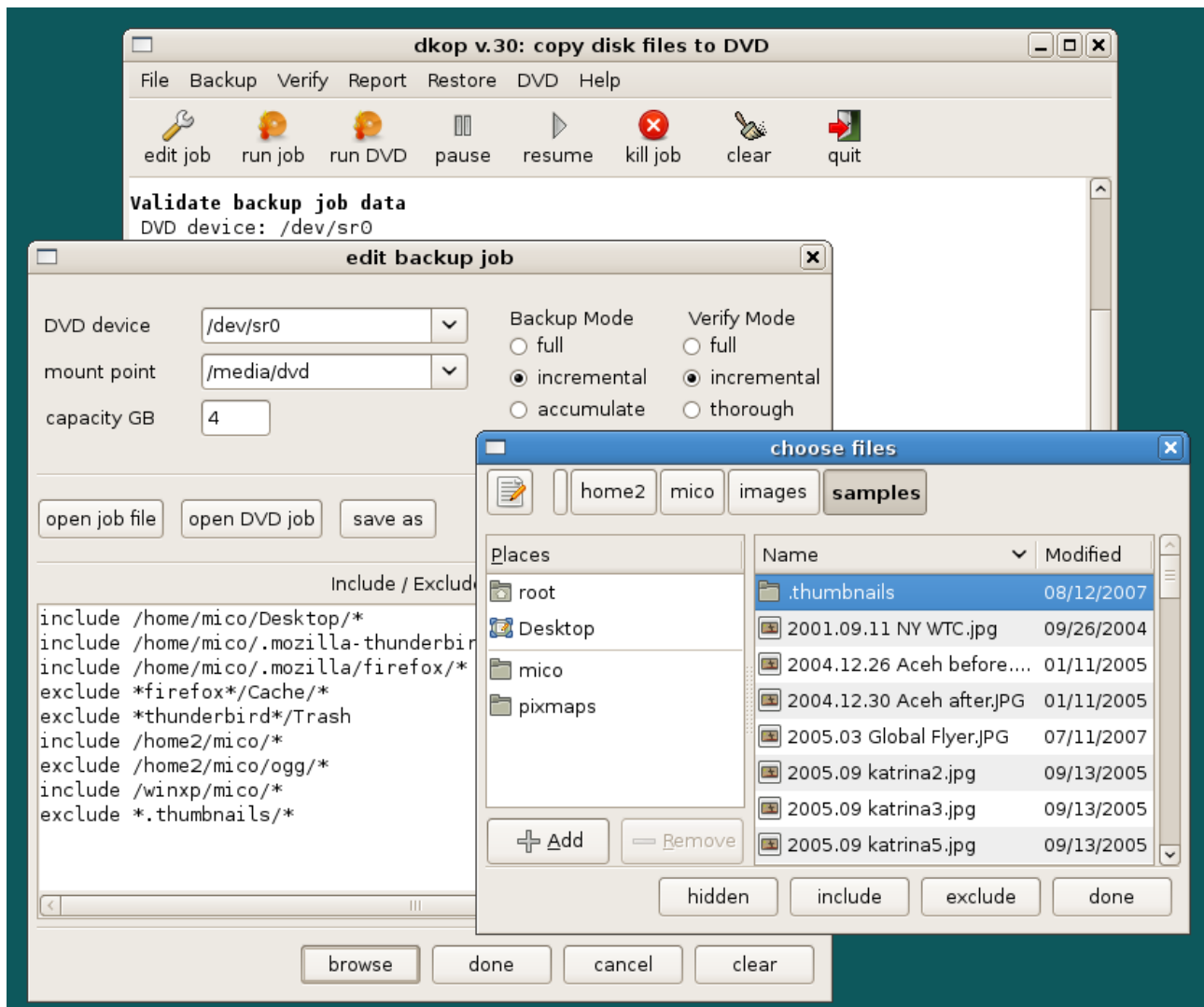
The include and exclude control statements allow precise control of the backup file set:

include /aaa/bbb/*	# include file tree under /aaa/bbb/
exclude /aaa/bbb/cc/*	# exception: exclude /cc/ subtree
include /aaa/bbb/cc/xxx.yyy	# exception: include file /cc/xxx.yyy

The file-chooser dialog may be used to quickly converge on the desired results. The editor may also be used to make adjustments.

Because of wildcards, newly added files within the scope of existing include or exclude filespecs are automatically comprehended. In the above example, if a new file is added somewhere within the /aaa/bbb/ tree, it will be automatically included in the next backup job, unless of course it is in the excluded /aaa/bbb/cc/ subtree.

## dkop job edit, file selection dialog



The file > edit job menu command (or toolbar button) pops up the left box. This can be edited directly: click anywhere in the text area and start writing. The right box is the choose files dialog, which is started with the browse button in the left box. Choose files using the right box, and the left box records your choices. You can navigate around the directory hierarchy and select any number of files or directories. The hidden button toggles the display of hidden files. Click one of the include or exclude buttons to get the selected files added to or removed from the backup list. Selecting a directory is an implied selection of all contained files, thus the selection appears as directory/\* in the list of selected files. To make an exception, go down one level, choose files, and select the opposite include or exclude button. You can refine the file selections manually if desired. It is sometimes handy to use wildcards in the directories to make more general and compact selection criteria, e.g. "exclude \*thunderbird\*/Trash" will omit trashed mail even if the overlying directories change (they do) and even for multiple users.

You can add comments (or disable an include / exclude line) by putting # in column 1.

## Script Files

A script is a text file with a series of commands that can be run as a batch job. All dkop menu commands can be scripted. This enables dkop to be automated and run via `cron` (you would still need to leave a DVD in the drive, or be present to change DVDs if a large job is run). Script files were implemented in order to automate the testing of dkop, but you may find it otherwise useful.

The format of the records in the script file is as follows:

```
menu1 > menu2 > parameter    # comment
```

The menu names must match the interactive menu names exactly, including case.

To run a script file: `$ /usr/dkop/dkop.x -script /pathname/scriptfile`

Here is a sample script file to get you familiar with the possibilities:

```
File > open job > jobfile1
Report > get backup files
DVD > mount DVD
Report > differences-detail      # report changed files
Backup > incremental            # back them up
Verify > full                   # verify all files
Report > differences-counts     # should be zero
File > save screen > dkop.log    # save a log file
DVD > eject DVD
File > quit
```

The toolbar buttons may also be used, e.g.

```
button > pause    # press resume to continue
```

(at this point you may use the menu interactively and then resume the script by pressing the resume button).

The command `exit` may be used to end the script file and return to interactive mode. Script file EOF does the same thing.

## Large Backup Jobs (more than one DVD)

This is supported in a limited manner starting with release v.14. A full backup may be larger than one DVD, and you will be asked to load additional DVDs as needed. If a job is being run (rather than the menu `backup > full`) each DVD will be verified (if specified in the job) before the next DVD is requested. An attempt is made to fit all files derived from a single `include` statement on the same DVD, if possible. This allocation is made after excluded files have been removed from the backup file set. The job file written to each DVD reflects the files actually copied to that DVD, so that subsequent incremental updates may be done individually on each DVD, e.g. as follows:

```
File > open DVD
Backup > incremental
Verify > incremental
```

Note that if an `include` statement is too big to fit on one DVD, this strategy will not work as expected. The backup job file on the first DVD will have the big `include` statement, but additional DVDs used for this same `include` will not. If possible, break up the large `include` statement into smaller ones.

## Technical Notes

**DVD auto-mount:** I recommend you **disable auto-mount of removable media** in your preference settings. This sometimes causes mount failures because some other application grabs the DVD drive (even though dkop issued a mount command). Code has been tweaked to get around the problem, but this is not very robust. I notice that some other applications make the same recommendation.

**Running dkop as root:** dkop will only copy files for which the user has read access. If files belonging to root or other users are to be copied, you must run dkop as root. Use "su" or "sudo", or log in as root (see the note below about making a launcher to handle this). The growisofs manpage says it will not work using "sudo". However, using dkop with "sudo" (which starts growisofs as a subprocess) apparently works fine. On some systems mounting a DVD requires root, making dkop also require root.

**"suid" permission:** dkop cannot run with the "suid" permission bit: the GTK library refuses to initialize. The GTK authors believe it is impossible to maintain good security if non-root users are allowed to use GTK apps with root permissions. If root privileges are needed, run dkop as root: `$ sudo dkop.x`

**Blank DVDs:** the first time a DVD is used, do a full backup. A blank DVD will not mount, but a full backup will still work and make the DVD mountable thereafter.

**Flakey DVDs:** drive and media combinations sometimes have compatibility problems, resulting in media errors. The newest drives (2006+) are much better at adjusting to media variations. I have **never yet** seen a DVD that passes a dkop verify and then becomes unreadable later, but of course this could happen, so make regular backups and avoid overdependence on a single DVD.

**Media errors:** If the dkop verify function runs into a read error, the DVD drive may lock-up for a minute or more while retrying the failed read hundreds of times. Give the "kill" command and wait for the drive to give up. If the DVD is dirty, clean it and try again. Otherwise throw it out.

**Backup strategy:** I suggest you buy a stack of DVD+RW media (< \$1 each) and run a daily incremental backup, cycling through the DVDs one at a time. Take one from the bottom (oldest), do the backup, and replace on the top (newest). If you accidentally delete a file, you can still recover it even after many backup cycles. Additionally, I suggest you make a monthly full backup to DVD+R media and save these indefinitely.

**Privacy and data protection:** to protect your private data on discarded DVDs, you should destroy them. A few seconds in a microwave oven will completely destroy the metallic recording layer (with spectacular visual effects). Do not inhale the fumes.

### Command line arguments:

```
$ dkop.x -job jobfile          # load job file
$ dkop.x jobfile               # load job file
$ dkop.x -run jobfile          # load job file and run it
$ dkop.x -script scriptfile    # run script file
```

The -run and -script commands are intended for shell scripts. The -job command is more useful for a desktop launcher, leaving the user free to elect the backup mode or make other changes in the job before execution. If the jobfile name contains blanks, quotes are required, e.g.

```
$ dkop.x -job "my dkop job"
```

**Desktop launcher:** a desktop icon / launcher may contain a command like this:

```
gksu /usr/dkop/dkop.x -job myjob.job
```

"gksu" will ask for the root or administrator password and run the job as root.

**Deleted DVD files:** growisofs is used to perform the file copies. It can replace existing DVD files with new versions, but it does not delete files. For incremental backups, dkop replaces deleted files with null files (zero length). Full backups do not have this issue, since the DVD is initialized. If you recover files from a dkop DVD using a shell copy command with wildcards, or drag-and-drop of an entire directory, you may get some unwanted null files. If this happens, it is easy to get rid of them like this:

```
$ rm -i $(find /dir1/.../dirN -empty)
```

(remove all empty files in a directory tree, with confirmation of each)

Note that if you use dkop restore, these null files are invisible and are **not** restored.

**Incremental backups:** a DVD file is considered identical to its corresponding disk file if their lengths and modification times are the same. Incremental backups exclude such files. If the modification times differ by less than 0.001 seconds they are considered equal. A thorough verify will read and compare the files unconditionally.

**File names containing "=" :** mkisofs requires that "=" in file names be replaced with "\\=". DVD files end up with "\\=" replacing the original "=". The file-chooser dialog in dkop file restore shows "\\=" instead of "=", but the files will be correctly restored with "=" only.

**Restoring file owner and permissions:** For some reason, mkisofs does not preserve owner and permissions for directory files copied to DVD, although data files are preserved. To get around this, dkop copies a special file to the DVD with the data needed for file restores.

**Special dkop files on DVD:** directory dkop-data is written to the DVD with three files:

datetime	backup date-time and DVD usage count
filepoop	owner and permissions for all backed-up files and directories
jobfile	a copy of the backup job specs last used on this DVD

These are ordinary text files which you can view with an editor.

**Special file types:** pipes, devices, and sockets are not copied. **Symlinks** are copied, but their target files are not. This works OK as long as both are included in the backup or restore file set, which should logically be the case, since it makes no sense to copy one without the other. If a symlink is copied without its target file, the menu Reports > differences will correctly show disk files which are not on the DVD. Symlinks are sometimes broken (link to non-existent files), and this seems to be a frequent and normal condition in Linux.

**Killing growisofs** (killing a backup job in progress): this will sometimes leave the DVD in a condition that growisofs refuses to deal with. If you decide to abort a backup job (e.g. to revise the job specs and start over), you may get this condition. You should retry a **full backup** on this DVD. If growisofs still refuses, format the DVD (dkop menu), then try the full backup job again.

**Duplicate files:** If job file "include" statements overlap, resulting in duplicate files in the backup set, this is reported and the backup is terminated.

**GTK thread locking:** the functions zlock() and zunlock() are used to surround GTK function calls and make them thread-safe. These locking functions have also been coded to do nothing if called from the main() thread, and to detect and avoid redundant locking (a fatal bug) if there are nested calls.

**Microsoft Windows:** DVDs created with dkop use the standard ISO-9660 file system, which can be read by Microsoft Windows XP.

**Getting DVD drive and medium information:** here are two useful commands:

```
$ ddevinfo -q all -n /dev/dvd      # DVD drive information
$ dvd+rw-mediainfo /dev/dvd       # DVD medium information
```



**Gentoo:** This is the only version of Linux I have tested (so far) that required a source code change to dkop. The bash `find` command did not accept the `-L` parameter, but `-follow` (outdated) can be used instead.

**Incremental backups:** new and updated files are written to a new "session" on the DVD, along with new directory files which may reference data files in both the old and new sessions. Nothing is changed in the old sessions. Thus, incremental backups consume more space on the DVD even if the corresponding disk files are not any bigger. For DVD+R and DVD-R media (write once), only one full backup may be made, and as many incremental backups as can fit in the remaining space. For DVD+RW and DVD-RW media (rewritable), a new full backup will initialize the DVD and recover all space. These DVDs can be used until they wear out. I have exceeded 100 uses on one test DVD+RW medium and it still works fine. The default dkop DVD capacity of 4.0 GB leaves about 0.7 GB for incremental backups. This is an editable field in the backup job, so you may set this margin to suit yourself.

**growisofs progress tracking:** Growisofs (`mkisofs` / `genisoimage`) outputs a "% done" value every few megabytes. Dkop uses this number to dead-recon the current position in the list of files to be copied, and the resulting file is echoed to the main window. The update frequency is typically less than once per file, so some file names will be bypassed. Large files (megabytes) will stay on the screen for several update cycles. For full backups, the math is straightforward. For incremental backups, growisofs starts off with:

$$\% \text{ done} = 100 * (\text{initial DVD bytes used}) / (\text{final DVD bytes used})$$

Dkop assigns this value to the first file in the backup list. The last file is assigned 100%, and the rest are interpolated using accumulated bytes.

**(poor) Linux error codes:** Linux error codes can be misleading. If an attempt is made to open a file that is already open and therefore locked, the error code translates to "no such file or directory". The error codes are the same for an attempt to mount an empty tray or a corrupted DVD. The same is true for an attempt to mount a DVD that is already mounted, or a blank DVD. Dkop outputs messages of its own that mention the multiple possibilities. Likely there are other cases, and hopefully this will improve over time.

**Backup history files:** (v.29) A history file is generated for every backup job run.

location: `/home/username/.dkop/` or `/root/.dkop/`

file name: `dkop-hist-yyyymmdd-hhmm-dvdlabel`

The file name corresponds to the date and time of the backup and the DVD label. A history file contains a list of all the files copied to that DVD at that time. Thus, a DVD used for a full backup and two incremental backups will have three corresponding history files, each one containing those files copied by the respective backup job. A full backup spanning multiple DVDs will have multiple history files, one per DVD, and the labels will be appended with :A to :Z to handle up to 26 DVDs for one backup job. History files accumulate and are not automatically deleted. When 200 files are reached, the `find` files and view backup history reports produce warnings. Delete the oldest files or move them elsewhere. The 200 limit is a compile time constant: `maxhist`. This could be set much higher if desired (and if you have so many DVDs before you re-use them).

**DVD label:** (v.30) The new menu `DVD > set DVD label` is for an optional DVD label input, which you can use as part of your media management system. A subsequent backup job will write this label to the DVD, and the DVD mount command will show the label.

Recommendation: for full backups, set the label to match what is written on the DVD (with a soft pen). For incremental backups, leave the label unchanged.

**mkisofs / genisoimage errors:** (v.34) If a disk file is deleted after growisofs begins, the DVD will be defective: directory entries for the missing files and all following files will point to garbage (which may even be readable). The error reported by `mkisofs` / `genisoimage` is ignored by growisofs. Dkop now scans growisofs output for the ignored errors and un-ignores them.

**KDE quirk:** Toolbar buttons are missing their text by default. You can change this with the KDE menu Settings > Appearances > Style > Toolbar, but this affects all applications.

### Tested Linux Distributions

The following table summarizes my testing of different Linux distributions:

exe      the pre-built executable dkop.x worked out of the box  
build    dkop.x had to be rebuilt using dkop-build.sh

distribution	exe	build	desktop	comments
Ubuntu 6.06 thru 7.10	y		gnome	my development system
Fedora 5, 6, 7	y		gnome	
Suse 10.1, 10.2	y		KDE	
Mandriva 2006	y		KDE	DVD tray auto-load did not work
Gentoo 2006.1		y	gnome	bash find command had to be changed
Mepis 6.0	y		KDE	
PCLinuxOS 2007	y		KDE	
Debian 4.0		y	gnome	
Freespire 1.0.13			KDE	unable to download package libgtk2-dev
Damn Small Linux			Fluxbox	frustrating to work with, so I gave up
elive revolution			evolution	system was unusable (crashes, lockups)

### DVD media types:

DVD+RW      good for many (hundreds?) of full and incremental backups  
DVD-RW      good for many (hundreds?) of full and incremental backups  
DVD+R      good for one full and many incremental backups  
DVD-R      good for one full and many incremental backups  
DVD-RAM      good for many (thousands?) of full and incremental backups  
                 (reputedly the most reliable medium)

### Tested drive and media combinations (using Ubuntu 6.06)

drive ↓      media →	DVD+RW	DVD-RW	DVD+R	DVD-R	DVD-RAM
Benq DW1640	ok	ok	ok	X	X
Benq DW1650	ok	ok	ok	ok	X
LG GSA-4082B	ok	ok	ok	ok	ok *
LG GSA-H10A	ok	ok	ok	ok	ok *
Plextor PX-716SA (SATA)	ok	ok	X	X	X
Samsung SH-S162	ok	ok	ok	ok	X

\*DVD-RAM exhibited strange behavior: For full backups, the DVD drive stayed busy for a long time after growisofs was done. For incremental backups, the DVD reported differences relative to the disk, until ejected and remounted, then the differences were gone. Apparently data is being cached and written to the DVD when it is unmounted, but the logic has holes.

## change log

2007.11.20 v.34

- trap mkisofs / genisoimage errors not reported in growisofs status (makes bad DVD look good)

2007.11.15 v.33

- new GTK requirement: `if (! g_thread_supported()) g_thread_init(0);`

2007.11.12 v.32

- stop propagation of DVD label suffix (dkop:A:A:A...) if same DVD used for multiple full backups
- locale related library changes

2007.10.15 v.31

- fix significant bug in thorough verify: files not compared 100%
- list the include and exclude records at the start of the history log file
- don't lose comment records in job file
- eliminate "end" record in job file (end is EOF)

2007.08.29 v.30

- move DVD label setting from job edit to separate menu item
- DVD mount no longer overwrites a previously set label for a subsequent backup

2007.08.23 v.29

- stop exotic files names with special characters like `\r` from trashing output formats
- optional user input of DVD label
- provide searchable backup log file (time / date, DVD label, list of files copied)
- query to find all backup DVDs containing desired files / directories, using wildcards

2007.07.13 v.28

- accept job file on command line (without `-job`) (per Linux convention)
- allow retry of a failed DVD within a multiple-DVD backup job
- correct progress tracking for multiple-DVD jobs

2007.07.04 v.27 .27b

- back to using bash for DVD mount (undocumented black magic that works better)
- follow Linux convention for application files in `/home/user/.dkop/` or `/root/.dkop/` (move your job files here for convenience)
- improve clarity of command outputs and reports
- allow show / hide of hidden files in job edit file chooser dialog

2007.05.28 v.26

- add file load and save convenience buttons to job edit dialog
- increase file limit from 100K to 200K files (memory usage +15 MB)

2007.05.13 v.25

- replace menu and toolbar macros with new zlib functions

2007.04.25 v.24

- minor report improvements
- add report: disk:DVD differences by directory
- allow file size exceeding 2 GB

2007.04.08 v.23

- fix bug: thorough verify was trying to check files created after last DVD update
- better progress monitoring: backup displays file names as they are written to DVD
- set margin for comparing file modification times to 1 millisecond instead of 1 second
- remove new owner option from restore (no need to duplicate Linux core capability)
- use c-lib functions for file restores instead of shell commands (simpler, faster)

2007.03.26 v.22

- detect mount EROFS status (read-only file system) and call it success

2007.03.26 v.21

- unmount DVD before eject (avoid drive hung busy with some Linux kernels)
- use `$ wodim -abort` to reset hung-up DVD drives (newest Linux systems only)

2007.03.19 v.20

- clarify output from menu: DVD and mount point
- use mount() function for DVD instead of shell command - better status information

2007.02.02 v.19

- job editor: select DVD and mount point from list of known options (instead of typing them in)
- (19.1) incremental backup: eliminate duplicate report of files and DVD differences

2007.01.10 v.18

- use generic monospace instead of Courier 10 font (which is not always available)
- rationalized menus and toolbar buttons for an easier, more conventional interface
- removed Joliet file system from DVD: Windows XP can read a standard DVD.
- converted dialogs to zdialog functions
- added icons to toolbar buttons

2006.11.30 v.17

- added tool tips to buttons

2006.11.28 v.16

- added 3 sec. delay between backup and verify and between mount retries: DVD drive may still be busy after growisofs completes, causing following mount command to fail.
- reorganized library functions into zlib.cpp and zlib.h
- added "edit" button for faster access to backup job editor
- corrected bug causing verify fopen() error for file names having trailing blanks

2006.11.01 v.15

- added menu: set DVD device and mount point independently of job file.
- bug fix: report of disk-DVD differences was ignoring a manual DVD change.
- GTK 2.6.10 issue: hidden files are no longer shown in the file-chooser dialog.  
A button was added to allow the user to view and choose these files if desired.

2006.09.26 v.14

- added support for large backup jobs using multiple DVDs
- corrected GTK coding error that was slowing down report output
- added button to clear window in backup job editor
- added command to set DVD device independently of backup job file
- verify (thorough) reports files modified during execution (no file compare error)
- default backup job changed to: include /home/userid/\*
- made minor improvements in output formats and user guide

2006.09.12 v.13

- exit script file if a command fails
- fix bug in thorough verify: DVD file compared to itself instead of disk file (ouch)
- detect growisofs failure from its log messages (may not emit a bad status)

2006.09.05 v.12

- reorganize menu names and groups
- add menu command for quick-formatting DVDs
- add menu command for drive hardware reset (may unlock hung drive)
- do not retry failed mount for full backup (likely a blank DVD)
- add growisofs undocumented parameters to improve robustness (sometimes rejects DVD already having ISO-9660 file system)
- improve disk-DVD differences report: report file names by category: unmatched disk files, unmatched DVD files, files with different content

2006.08.23 v.11

- improve error reporting (DVD mount failures)
- DVD mount status 8192: meaning? insert extra eject and mount retry
- build script: check that GCC compiler and GTK+ library are installed
- workaround for GTK multi-thread seg. faults (gtk\_threads\_enter() etc. not enough)
- fix bug: restore owner failure for filenames containing single quote character
- additions to user guide, technical notes

#### 2006.08.09 v.10

- correct zlock() thread problems (add mutex lock for global variables)
- suppress error messages (e.g. broken links) for excluded files
- ignore status from shell "find" command
  - non-zero status can happen even if intended files are found
  - errors are visible in terminal window, e.g. "permission denied"
- minor additions to user guide

#### 2006.08.18 v.09

- added script file capability: all menu commands can be scripted
- improved robustness in the use of GTK within threads
- command line parameters revised as follows:
  - backup command -job filename -script filename
- exclude statements pertain to **ALL** prior include statements, not just the last one
- improve diagnostics for problem files ( stat() function returns error)

#### 2006.07.13 v.08

- minor improvements to user guide / help file
- added an option to change ownership for files restored from DVD
- removed option to restore directory permissions and made this automatic instead (revising permissions is outside dkop scope - see technical notes in user guide)
- eject is now unmount + eject (eject alone sometimes does not work)
- made restore more impervious to special chars in file names [ \$ " = ]
- /tmp/dkop-xxxx files now have PID in name, allowing parallel users of dkop
- /tmp/dkop-xxxx files are now deleted automatically when dkop exits

#### 2006.07.10 v.07

- add index and hyperlinks to help file, dkop-guide.pdf
- make help function run as subprocess (view help and run dkop in parallel)
- use acroread | gpdf | evince (first one available) to display help file
- speed up include files processing (SearchWild(), use find instead of readdir)
- report byte counts for included and excluded files
- if command line job file is relative, make absolute /pathname/filename (gtk\_file\_chooser\_set\_filename() does not apply cwd)
- fix bug: restore of files having '\$' in pathname failed (use cp 'path' instead of "path")
- detect duplicate files (from overlapping include records) (avoid growisofs failure)
- better formatting for report > in/ex file counts
- added verification of DVD device and mount point

#### 2006.07.01 v.06

- added toolbar with buttons for clear, kill, pause, resume
- minor tightening of job verify logic after loading or editing
- added menu function: show job file
- removed auto mount of DVD (confusing)
- avoid redundant remounts of DVD (detect when no change, avoid time waste)

#### 2006.06.24 v.05

- new menus: full backup + verify, incremental backup + verify, accumulate backup + verify
- remove autoload of file default.job - unnecessary and confusing
- skip special files that do not backup and restore - pipes, sockets, devices
- fix bug: restore directory owner & permissions omitted some directories
- symlinks are backed-up and restored, but not target files (review user guide topic)
- fix SearchWild() bug: search for file without wildcards failed (should find 0 or 1 file)

#### 2006.06.15 v.04

- save on DVD: backup job file, backup date-time, DVD usage counter
- new feature: run backup job file stored on DVD itself
- detect broken symlink files, diagnose and ignore (message was "error: success")
- revised GTK menu macros to use "string" instead of #string

#### 2006.06.10 v.03

- finished escaping of '=' in file names (mkisofs requirement)
  - disk and DVD file names are equal if disk with '=' matches DVD with "\\="
  - file restore function replaces "\\=" with '='
  - user sees "\\=" in restore function file-chooser dialog
- file restore bug: invalid copy\_from produced diagnostic message naming copy\_to location
- auto locate help file in same directory as dkop executable
- fixed bug in zmainloop(): menu lockup from use of pause function
- simplified lock and pause/resume logic to make more robust

#### 2006.06.01 v.02

- added file restore capability
- added directory owner and permissions recovery
- corrected bug in byte count calculation (1 file and 0 bytes to restore)
- eliminated trailing blanks in filespecs (caused failures)
- corrected bad detection of open() error (fid < 0)
- replace '=' in file names with "\\=" (mkisofs requirement)  
(caused failures with strange gnome file names)
- stopped use of fprintf() for writes to scratch files ('%' in filename interpreted as format)
- bug: open of /tmp files fails if they do not exist beforehand

#### 2006.05.01 v.01 initial release

- backup job: open, edit, save
- backup: full, incremental, accumulate
- verify: full, incremental, thorough
- disk / DVD differences reports