DIMG

User Manual

Document Number: `A000-UM02`

April 2010

This document provides configuration and usage guidelines for DIMG, a portable graphical disk imager.

Revision/Update Information:	This is an updated manual.
Operating System:	Microsoft Windows NT 5.0+; Linux
Software Version:	DIMG Version 1.1

This document was prepared using LATEX 2_ε.

Contents

List of Figures

Chapter 1

Getting Started

DIMG is a utity for imaging floppy disks. It allows you to create a file which represents the exact content of the source disk which can then be restored onto another disk creating an exact copy.

In addition to basic reading and writing of disk images, DIMG provides the ability to view the contents of disk images as if they were archives. This allows files to be extracted from a disk image without having to write them to disk first.

Figure 1.1 shows DIMGs main window. In the center of the main window is the file view area. This shows files present in the currently open disk image. Figure 1.6 shows the DIMG main window again with an MS-DOS 6.22 boot disk image loaded.

1.1 Installation

1.1.1 Windows

On Windows, DIMG has no specific installation requirements. It will automatically initialise all settings required to default values on first run.

DIMG is made of the following four main components on Windows. All are required for proper functionality.

1. DIMG
2. DQWinDrv
3. fat12fs
4. FPraw
5. FPzip

If you wish for DIMG to remain self-contained ("portable") and prevent it from storing its settings anywhere on your computer, create a file called `DIMG.ini` in the same directory as `DIMG.exe`. DIMG will use this file for storing its settings rather than the windows system registry. Doing this allows you to run DIMG from a portable storage device without it leaving anything behind on any computers it is run on.

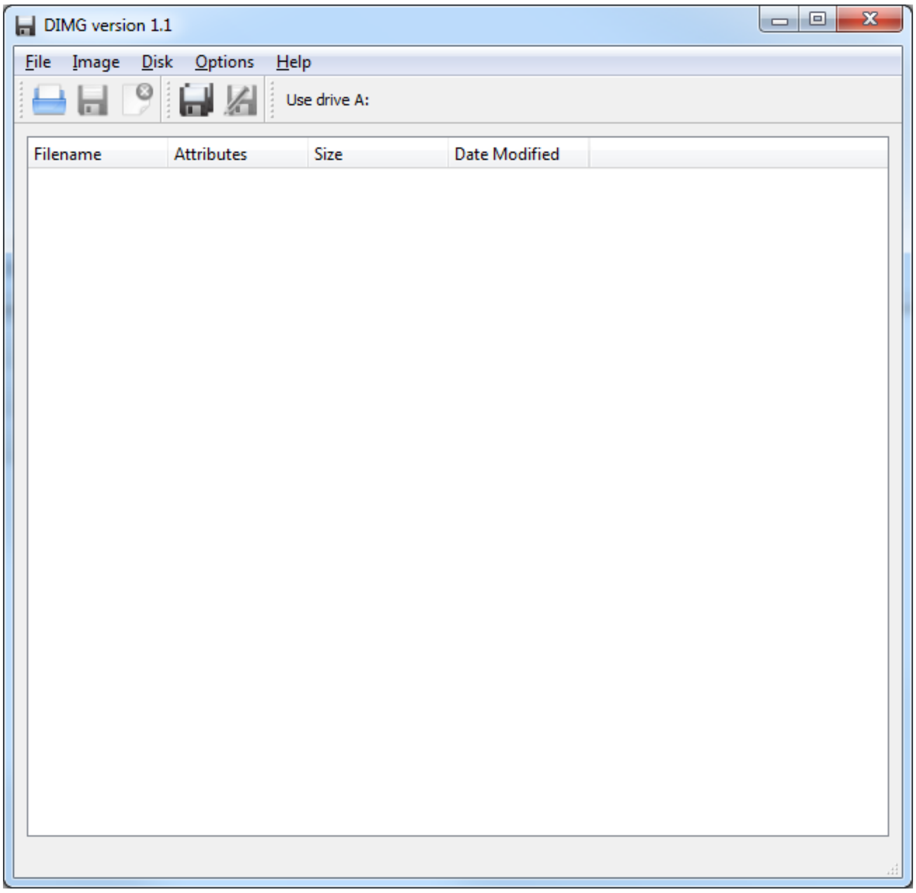


Figure 1.1: The DIMG Main Window under Windows 7

If DIMG is run without `DIMG.ini` present it will store its settings under `HKEY_CURRENT_USER\Software\zxnet\DIMG` in the system registry. When removing DIMG you may also wish to remove this key.

1.1.2 Linux

DIMG is made of four components:

1. DIMG
2. DQDLibDsk
3. fat12fs
4. FPraw
5. FPzip

All four components require Qt 4.6 or higher. Additionally, DQDLibDsk requires that LibDsk be installed on the system.

To install DIMG on linux simply cd into the source directory and run `qmake`. QtDIMG.pro is the project file used to build DIMG. Should you wish to just build QFATreader an alternate project file (QFATreader.pro) is also provided. Once you have run `qmake`, run `make`. This will produce a set of binaries in either the release or debug directory. You can then install these binaries into `/opt/dimg` by running `make install`.

DIMG will store all of its settings in the configuraiton file `~/.config/zxnet/DIMG.conf`.

1.2 Initial Configuration

Under Windows, DIMG requires no initial configuration. Under Linux, however, you must tell DIMG where to find your systems floppy drive(s) before you can proceed.

1.2.1 Adding Drives on Linux

Some initial setup is required on Linux systems as DIMG is not able to automatically detect the systems floppy drives when using the DQDLibDsk driver (the default). To do this, select "Configure Driver" from the Disk menu (Figure 1.2) in the Main Window.

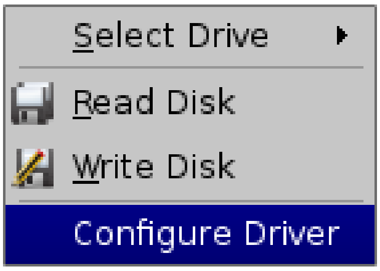


Figure 1.2: Configure Driver in the Disk Menu

This will bring up the DQDLibDsk configuration screen (Figure 1.3 where you can manually add all floppy drives you wish to use on the system. To do this, Click on the Add button which will bring up a prompt for a "Floppy Drive file" (Figure 1.4). This is where you can enter the name of the floppy drive (eg, `/dev/fd0`) or a raw image file which will be used as a virtual floppy drive. When you click OK the specified drive will be added to the list of floppy drives in the configuration screen (Figure 1.5).

When the program is next started the specified floppy drive(s) will appear in the Drive Select menu.

1.3 Working with Disk Image Files

To open a disk image you can either pass the filename you wish to open to DIMG on the commandline (for example, `./dimg /tmp/file.img`) or select Open from the File menu.

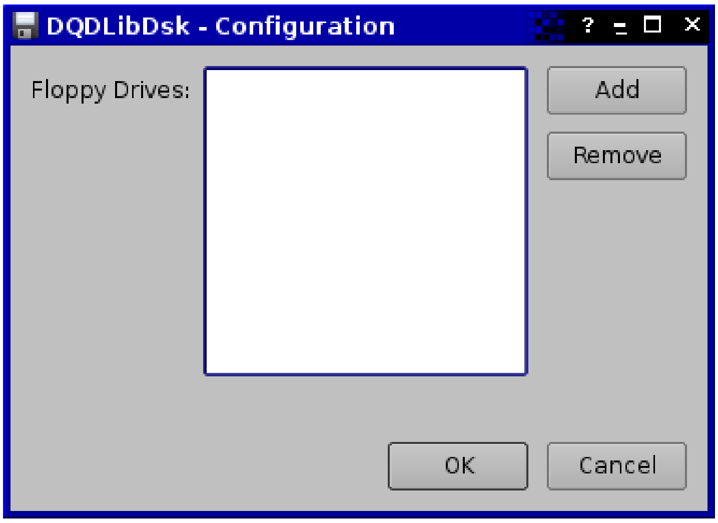


Figure 1.3: DQDLibDsk Driver configuration screen with no drives configured

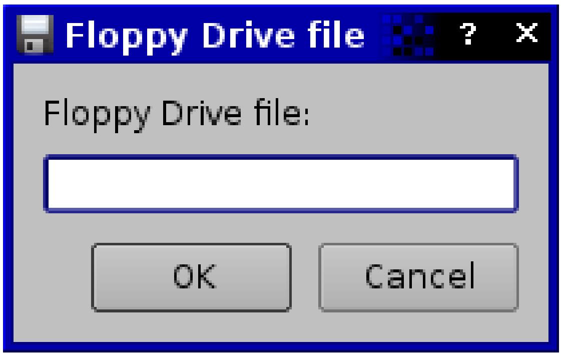


Figure 1.4: DQDLibDsk Driver Prompt for Floppy Disk File

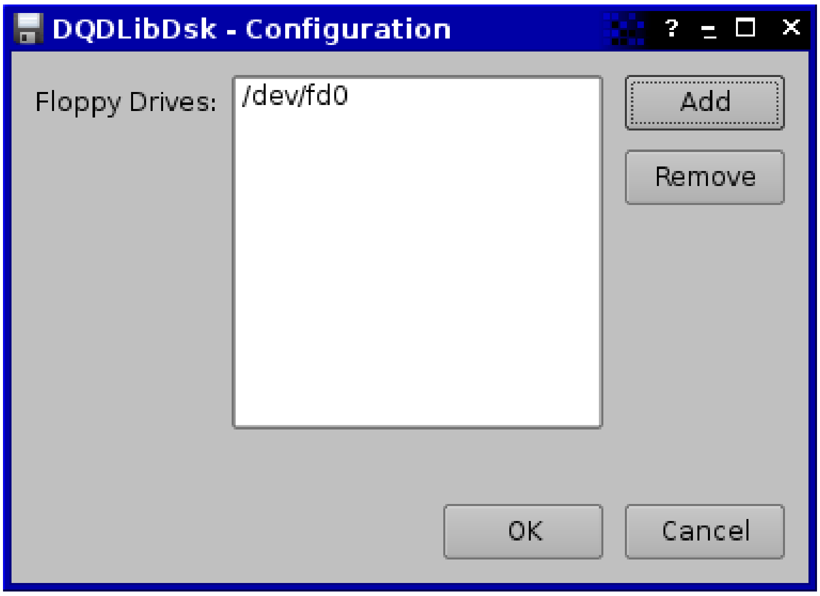
Selecting Open from the File menu will display a standard file open dialog which you can use to locate a supported disk image file. Only raw and compressed (.IMZ, zipped) disk image files formatted with the FAT-12 filesystem are currently supported. Trying to open disk images with other filesystems or corrupt filesystems may require the FAT-12 filesystem plugin to be disabled to prevent malfunction.

Support for disk images generally follows support for disk types - as DMF disks are unsupported trying to open a DMF-format raw disk image will cause bad things to happen (program crashes).

Once a supported disk image has been opened its contents will be displayed in the main window such as in Figure 1.6. The name of the file loaded is shown in the title bar and the images volume label is shown in the status bar at the bottom of the main window. In this case, the volume label is `STARTUP`.

To write an image to a disk see Section 2.3.

The image can be closed again by choosing *Close Image* from the File Menu. This will close the disk image and return the program to how it looked before

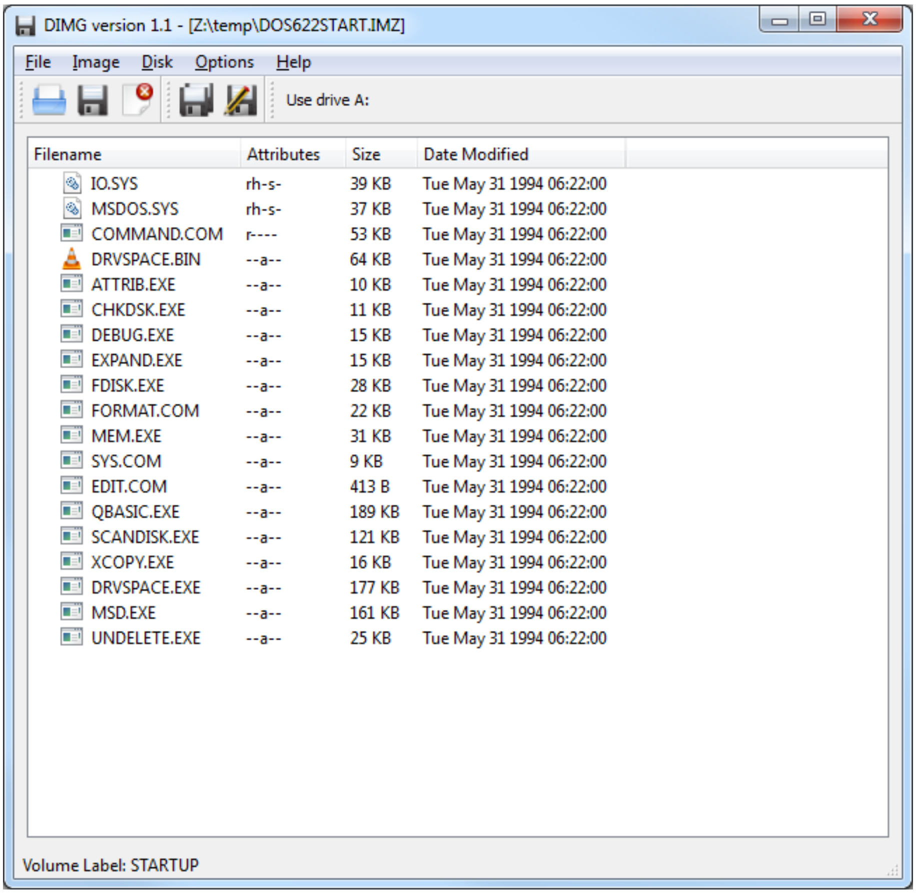


DQDLibDsk - Configuration

Floppy Drives: /dev/fd0

Figure 1.5: DQDLibDsk Driver configuration screen with drive added

the image was open (Figure 1.1).



Filename	Attributes	Size	Date Modified
IO.SYS	rh-s-	39 KB	Tue May 31 1994 06:22:00
MSDOS.SYS	rh-s-	37 KB	Tue May 31 1994 06:22:00
COMMAND.COM	r----	53 KB	Tue May 31 1994 06:22:00
DRVSPACE.BIN	--a--	64 KB	Tue May 31 1994 06:22:00
ATTRIB.EXE	--a--	10 KB	Tue May 31 1994 06:22:00
CHKDSK.EXE	--a--	11 KB	Tue May 31 1994 06:22:00
DEBUG.EXE	--a--	15 KB	Tue May 31 1994 06:22:00
EXPAND.EXE	--a--	15 KB	Tue May 31 1994 06:22:00
FDISK.EXE	--a--	28 KB	Tue May 31 1994 06:22:00
FORMAT.COM	--a--	22 KB	Tue May 31 1994 06:22:00
MEM.EXE	--a--	31 KB	Tue May 31 1994 06:22:00
SYS.COM	--a--	9 KB	Tue May 31 1994 06:22:00
EDIT.COM	--a--	413 B	Tue May 31 1994 06:22:00
QBASIC.EXE	--a--	189 KB	Tue May 31 1994 06:22:00
SCANDISK.EXE	--a--	121 KB	Tue May 31 1994 06:22:00
XCOPY.EXE	--a--	16 KB	Tue May 31 1994 06:22:00
DRVSPACE.EXE	--a--	177 KB	Tue May 31 1994 06:22:00
MSD.EXE	--a--	161 KB	Tue May 31 1994 06:22:00
UNDELETE.EXE	--a--	25 KB	Tue May 31 1994 06:22:00

Volume Label: STARTUP

Figure 1.6: DIMG Main Window showing a DOS 6.22 disk image loaded

Chapter 2

Imaging Disks

The primary function of DIMG is to image disks. This includes both reading a disk to image and writing an image to a disk. At this stage DIMG only supports working with Floppy Disk images. This version has only been tested with 3.5" high density disks but will probably work with other varieties. In particular, DMF formatted disks are not currently supported.

2.1 Selecting a Drive

Before you can image a disk you must select the drive containing the disk you wish to image. To do this select the desired drive from the Select Drive submenu of the Disk menu (See Figure 2.1).

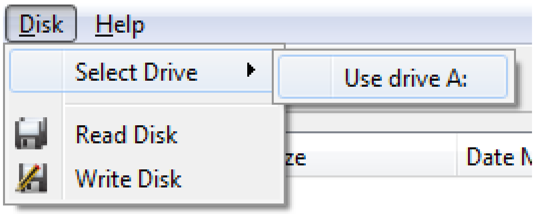


Figure 2.1: Select Drive Menu

If the Select Drive menu is empty or the desired drive is missing, see Section A.1 (Troubleshooting Missing Drives)

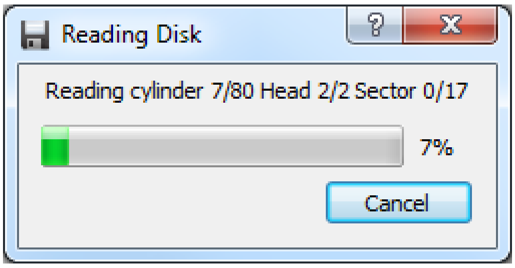
The drive you select will remain selected until either you select a different drive or restart the program. All image read or write operations will be performed on this drive for as long as it is selected.

2.2 Reading

Before a disk can be read into an image file the drive containing the disk you wish to image must be selected (See Section 2.1).

To read a disk into an image file, choose *Read Disk* from the *Disk* menu (See Figure 2.1). If a disk is present in the selected drive DIMG should start reading the image immediately provided your user account has access to the selected drive.

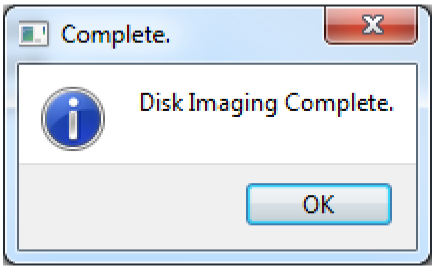
When DIMG is reading a disk into an image file it will show a progress dialog like the one shown in Figure 2.2. Clicking the Cancel button will cancel the read process.



Reading cylinder 7/80 Head 2/2 Sector 0/17

Figure 2.2: Reading Disk into Image File

When imaging has completed a message will appear (Figure 2.3) and the filesystem on the disk will be displayed.



Disk Imaging Complete.

Figure 2.3: Imaging Complete Message

An image of the disk now resides in memory. If you wish to keep the disk image you have just created you must then save it by choosing *Save* from the *File* menu.

2.3 Writing

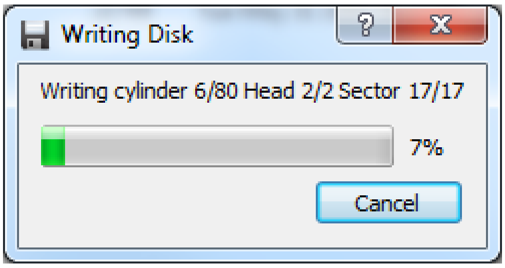
The basic process for writing a disk image to disk much the same as reading a disk into memory and then saving it to a file - just in reverse. Essentially, all you must do is:

1. Open the disk image you wish to write.
2. Choose the drive you wish to write it to.
3. Start the write process.
4. Wait for it to finish.

Before you can write a disk image to disk you need to have that disk image open. You can either load a disk image file into memory (See section 1.3) or you can read a disk directly into memory (Section 2.2).

Next you must select the drive you wish to write the disk image to (Section 2.1). Make sure you select the correct drive as any existing data on the disk in that drive will be erased. Once this is done you may start the actual disk imaging process.

Lastly, choose *Write Disk* from the Disk menu. A progress dialog (Image 2.4) will appear and, should no errors occur, a “Disk Imaging Complete.” message (Image 2.3) will eventually appear. The disk in the selected drive should now be an exact copy of the original source disk or disk image file.



Writing cylinder 6/80 Head 2/2 Sector 17/17

Figure 2.4: Writing a disk

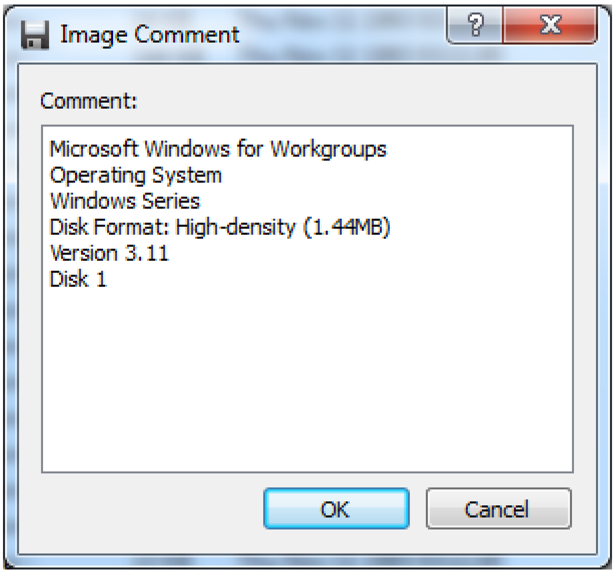
If errors have occured, try replacing the disk you are writing to or clean the drive. Despite what error messages may say, this version of DIMG is not capable of retrying reads or writes. If a read or write fails you have to start again.

2.4 Comments

When using compressed (`.IMZ`) disk image files, DIMG supports giving the file a comment. To view or set the comment on a compressed disk image file, choose the *Comment* option from the *Image* menu. This will bring up the Comment dialog (Figure 2.5) where you can view or set the comment. If the disk image is not saved as a `.IMZ` file you will not be able to set the comment.

After changing the comment you must save the file for the change to take effect. If you later re-save the disk image as a raw disk image (normally a `.IMA`, `.IMG` or `.FLP`) file the comment will not be saved in the new file.

The comment can not be set on a new disk image before you have saved it. You can, however, get DIMG to prompt you for a comment when ever you save as a compressed image file. Instructions on how to do this are covered in section 4.2.1.



Comment:

Microsoft Windows for Workgroups
Operating System
Windows Series
Disk Format: High-density (1.44MB)
Version 3.11
Disk 1

Figure 2.5: The Comment dialog

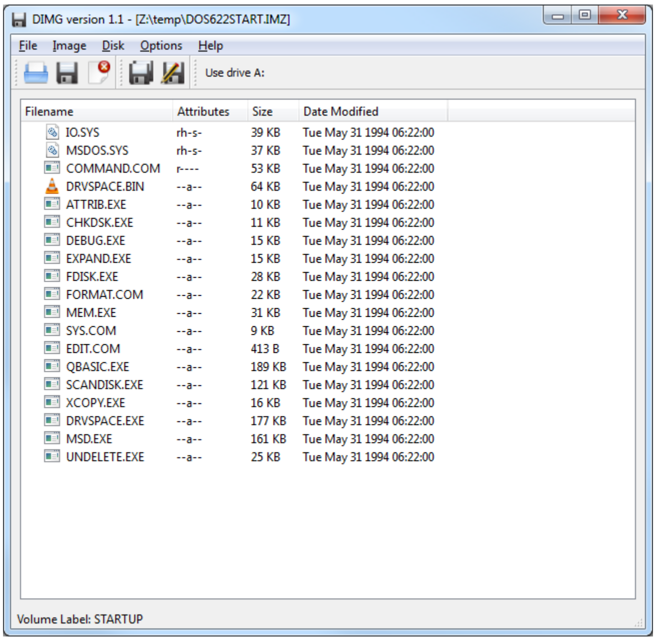
Chapter 3

File Management

This chapter explains how you can treat a disk image file like you would an archive/zip file. This includes browsing its contents and extracting files without having to write the image back to disk.

3.1 Browsing

When you open a disk image file or read in a disk with a **valid** and **non-corrupt** (important) FAT-12 filesystem on it (most PC disks use this filesystem) you will see all of the files and directories present on the disk in the main window (Figure 3.1). To see the contents of a directory, click on the expand symbol next to the folder icon.



Filename	Attributes	Size	Date Modified
IO.SYS	rh-s-	39 KB	Tue May 31 1994 06:22:00
MSDOS.SYS	rh-s-	37 KB	Tue May 31 1994 06:22:00
COMMAND.COM	r----	53 KB	Tue May 31 1994 06:22:00
DRVSPACE.BIN	--a--	64 KB	Tue May 31 1994 06:22:00
ATTRIB.EXE	--a--	10 KB	Tue May 31 1994 06:22:00
CHKDSK.EXE	--a--	11 KB	Tue May 31 1994 06:22:00
DEBUG.EXE	--a--	15 KB	Tue May 31 1994 06:22:00
EXPAND.EXE	--a--	15 KB	Tue May 31 1994 06:22:00
FDISK.EXE	--a--	28 KB	Tue May 31 1994 06:22:00
FORMAT.COM	--a--	22 KB	Tue May 31 1994 06:22:00
MEM.EXE	--a--	31 KB	Tue May 31 1994 06:22:00
SYS.COM	--a--	9 KB	Tue May 31 1994 06:22:00
EDIT.COM	--a--	413 B	Tue May 31 1994 06:22:00
QBASIC.EXE	--a--	189 KB	Tue May 31 1994 06:22:00
SCANDISK.EXE	--a--	121 KB	Tue May 31 1994 06:22:00
XCOPY.EXE	--a--	16 KB	Tue May 31 1994 06:22:00
DRVSPACE.EXE	--a--	177 KB	Tue May 31 1994 06:22:00
MSD.EXE	--a--	161 KB	Tue May 31 1994 06:22:00
UNDELETE.EXE	--a--	25 KB	Tue May 31 1994 06:22:00

Volume Label: STARTUP

Figure 3.1: The contents of a DOS-6.22 disk image

Four columns are shown in the file area of the main window. These are Filename, Attributes, Size and Date Modified. The contents of all these except perhaps attributes should be fairly obvious.

The attributes column shows which attributes the file has set. These attributes are:

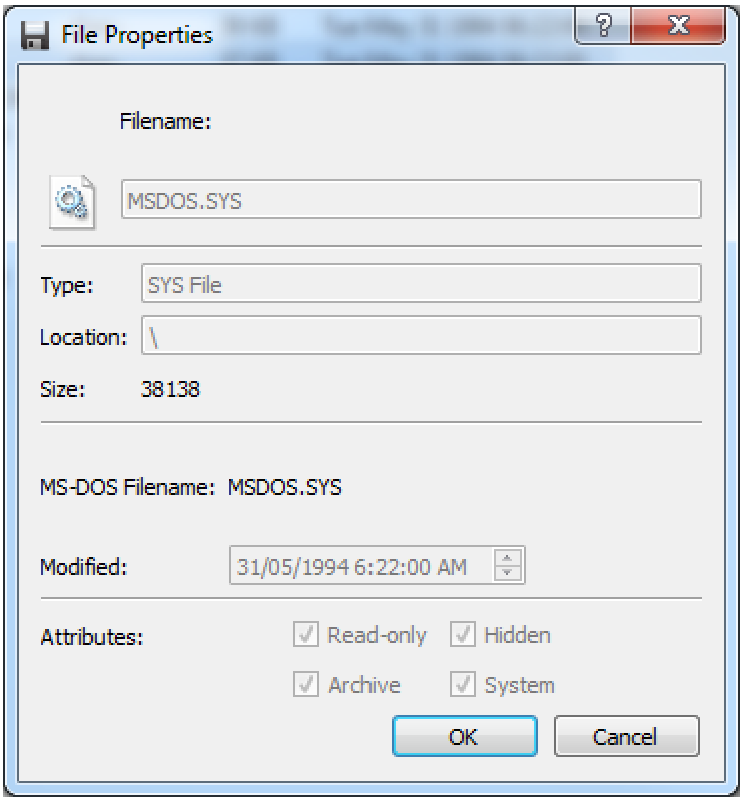
Attribute letter	**Attribute Name**
r	Read-Only
h	Hidden
a	Archive
r	System
d	Directory

3.2 Extracting Files

Like an archive, all of the files you see in the disk image can be extracted to a location on your computer without having to write the disk image out to a disk first. To extract a file, right click on the desired file and choose Extract File. A save dialog will appear where you can choose where to extract the file to and what to name it.

3.3 Viewing File Details

To view the full details of a file, right click on the desired file and choose the *File Properties* option. This will display a properties dialog similar to the one in Figure 3.2.



File Properties

Filename: MSDOS.SYS

Type: SYS File

Location: \

Size: 38138

MS-DOS Filename: MSDOS.SYS

Modified: 31/05/1994 6:22:00 AM

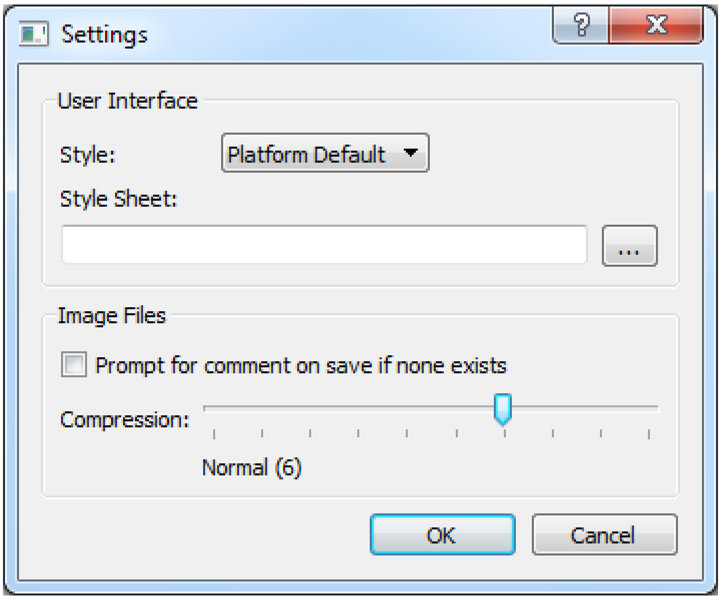
Attributes: ☑ Read-only ☑ Hidden ☑ Archive ☑ System

Figure 3.2: File Properties

Chapter 4

Settings

A number of basic program settings can be changed using the settings dialog (Figure 4.1) which can be accessed from the *Options* menu. To change more advanced settings the configuration file must be edited manually. See Appendix B for more details on the advanced settings available and how to change them.



Settings

User Interface

Style: Platform Default

Style Sheet:

Image Files

□ Prompt for comment on save if none exists

Compression: Normal (6)

Figure 4.1: Settings Dialog

4.1 User Interface

Two user interface related settings can be changed - the programs Style and its Style Sheet.

4.1.1 Style

The style controls what the program looks like at a very basic level. It exists to make the program fit in on different operating systems. Six options are provided:

1. Platform Default - this makes DIMG look like a Windows 7 program on Windows 7, a Windows XP program on Windows XP, a MacOS X program on MacOS X, etc.

2. Cleanlooks - this style is similar to the default theme on the Gnome desktop environment under Linux.

3. Plastique - this style is similar to the default theme on later releases of the KDE 3 desktop environment under Linux.

4. Windows - make DIMG look like a Windows 9x/NT/2000 program everywhere!

5. CDE - try to make DIMG fit in with the Common Desktop Environment (CDE) - a common UNIX desktop environment.

6. Motif - style DIMG as if it were a Motif program, a common look on UNIX systems and some older linux desktop environments.

After changing the style DIMG must be restarted for the changes to take effect.

4.1.2 Style Sheet

Style Sheets provide a flexible way of changing how the program looks. They use a similar syntax to Cascading Style Sheets commonly used in HTML documents on the World Wide Web. Consult the Qt documentation on how to write suitable style sheets. At the time of writing, such documentation could be found at `http://doc.trolltech.com/4.6/stylesheet.html`. To set the style sheet, click on the button to the side of the stylesheet text box to browse for one or enter its full filename directly into the text box. To unset it, delete the contents of the text box.

Like with styles, the program must be restarted for this setting to take effect.

4.2 Image Files

These settings control DIMGs handling of image files.

4.2.1 Prompt for Comment

If enabled, DIMG will prompt you for a file comment when ever you save a disk image as a new image file in a format that supports comments. The prompt dialog looks like the one in Figure 4.2. If you press cancel on this dialog the image file will be saved without a comment. If you press OK the image file will be saved with what ever comment was entered.

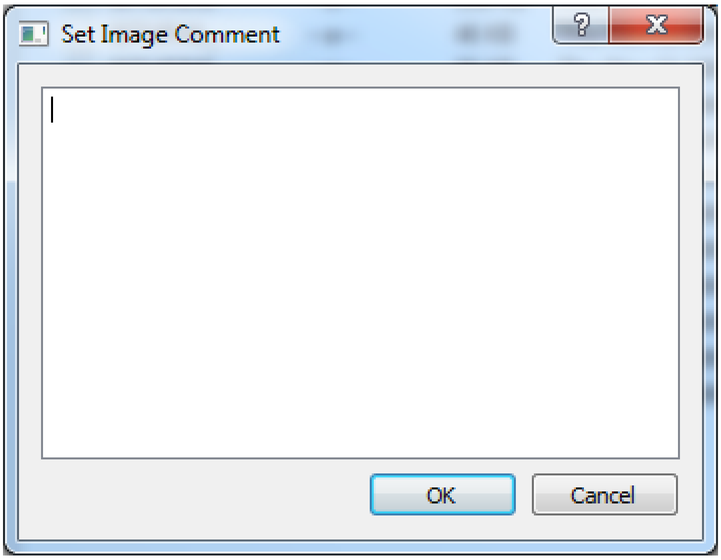


Figure 4.2: Image Comment prompt dialog

4.2.2 Compression

This slider sets the compresison level to use when saving in a disk image format that uses compression. On this slider, the far left (0) provides no compression at all, 1 provides minimal compression and 9 (far right) provides maximum compression. The higher the compression the slower saving and loading is. You are unlikely to notice any performance difference between the options on any computer DIMG runs on but if you have to open a compressed disk image on a very very slow old computer you may want to use a lower compression level. Level 6 should be a good normal value but if you only plan to use the image files on modern computers you may as well pick level 9.

Chapter 5

Advanced Topics

5.1 Plugin Architecture

DIMG has been designed with a modular architecture to allow various components to be upgraded or replaced without requiring major changes to other components.

At this time there are four main components:

1. DIMG - the main program
2. Filesystem plugin - Provides access to a disk images filesystem
3. Drive Access plugin - Provides access to the systems floppy drives.
4. Image Format plugin(s) - One or more plugins to provide support for image file formats.

At this stage only one Filesystem plugin and one Drive Access plugin can be active at a time. That means that DIMG can only support a single filesystem at a time - if you replace the FAT12 filesystem plugin with an HFS one you will no longer be able to see the contents of FAT12 disks.

5.1.1 Drive Access Plugins

The Drive Access Plugin is responsible for providing access to the systems floppy drive. It handles all reading and writing of disks. Currently there are two available Drive Access Plugins:

DQDLibDsk

A portable drive access plugin which uses LibDsk to do most of the work. This plugin has some serious performance issues on Windows systems and so is only provided with Linux versions of DIMG.

DQWindrv

This Windows-only Drive Access Plugin is a lot simpler than DQDLibDsk but provides significantly better performance. It is the default Drive Access Plugin for Windows systems.

5.1.2 Filesystem plugins

Filesystem plugins provide access to the filesystem on a disk image allowing files to be extracted as if it were an archive. At this stage only one Filesystem plugin is provided - FAT12fs. This plugin provides access to the FAT12 filesystem commonly used on PC-formatted floppy disks.

5.1.3 Image Format plugins

Image Format plugins allow DIMG to support new image file formats without having to adjust the program itself. Out of the box, DIMG provides two such plugins - FPraw and FPzip.

FPraw

FPraw supports generic raw disk image files. These can be handled by most disk imaging utilities and indeed many only support this format. When FPraw tries to load a disk image it will actually fully load it into memory if the file is small enough - this setting can be changed in the configuration file. See Appendix B for more details on the configuration file. FPraw does no format checking at all - if you tell it to try and open a word document it will happily do so. Because of this, FPraw is the last plugin to be tried when attempting to open an image file.

FPzip

FPzip provides support for compressed disk images which typically have the `.IMZ` extension such as those produced by WinImage. Unlike FPraw, FPzip is capable of storing a *comment* with the disk image and always (regardless of size) fully loads the uncompressed image into memory. To see how to use the comment feature, see Section 2.4. If FPzip is present it will be used as the default image format handler and .IMZ will be the default file format when saving.

The files FPzip handles are just standard Zip files except they usually have a `.IMZ` extension. FPzip will only open zip files which have a single `.IMA` file in them - if there is more than one file or the file has a different extension FPzip will not load the disk image at all.

5.2 Plugin Loading

On startup DIMG tries to load plugins using a number of default filenames trying each one until something loads. The filenames for some plugin types depend on the operating system in use and the type of build (debug or release). The format of filenames can be summarised in table 5.1.

So the DQDLibdsk (DQlibdskdrv) plugins filename on windows for a debug build would be `DQlibdskdrvd.dll`. On linux for a release build it would be `libDQlibdskdrv.so`.

Platform	Build Type	Prefix	Suffix	Extension
Windows	Debug		d	.dll
	Release			
UNIX	Debug	lib	d	.so
	Release			

Table 5.1: Plugin filename components

5.2.1 Filesystem Plugins

For Filesystem plugins DIMG will first try to load the `fat12fs` plugin moving on to `fsplugin` should it not be found. Should niether plugin exist it will display a File Open dialog allowing you to browse for a Filesystem plugin.

Applying the rules in table 5.1 a windows release build would first try to load fat12fs.dll and then fsplugin.dll.

5.2.2 Drive Access Plugins

The loading of Drive Access Plugins is similar to that of Filesystem Plugins (see section 5.2.1) except for a few minor differences.

The first plugin filename DIMG tries depends on the platform. It tries to load `DQlibdskdrv` on UNIX platforms or `DQWindrv` on Windows. On both platforms it will then attempt to load `dqplugin` before displaying a File Open dialog allowing you to browse for a Drive Access plugin.

Unlike with filesystem plugins, this process can be overridden by specifying the plugin name in the DIMG configuration file. For more details on how to do this see Section B.1, DQDriver Location. Should the specified plugin fail to load, DIMG will fall back to the default behaviour (trying the two default filenames before prompting).

5.2.3 File Format plugins

When starting, DIMG automatically scans the Image Format Plugin directory (`./plugins/imageformats/` by default) for anything that looks like an Image Format plugin and loads what ever it can.

Image Format plugins come in two types: Raw plugins and standard plugins. Raw plugins are normal image format plugins which report themselves as being able to handle raw image files. As there is generally no way of telling if any particular file is a valid raw image file, the raw plugin is expected to load what ever it is told to without attempting any validation. Because of this, raw plugins are the very last to be tried when attempting to open a disk image. Anything that does not report itself as handling raw image files (that is, all standard plugins) will be attempted first.

For the DIMG to work properly there must be at least *one* compatible raw plugin present in this Image Format plugin directory.

Appendix A

Troubleshooting

A.1 Missing Drives

On Windows, DIMG attempts to automatically detect all supported drives on startup. If the desired drive is not visible it may be that it is not supported or was not connected when DIMG started.

On Linux DIMG is unable to automatically detect supported drives. Drives must be manually configured - see Section 1.2.1 for instructions on how to do this.

Problem	**Possible Cause**	**Action**
The Select Drive menu is empty	No supported drives are present (Windows)	Attach a supported drive and restart DIMG
	No drives are configured (Linux)	Configure drives on the system (Section 1.2.1)
One drive not available in Select Drive menu while others are	The drive may not be supported	Use a different drive type
	Drive was attached after DIMG was started (Windows)	Restart DIMG

Appendix B

Configuration File

This appendix describes all configuration options directly available in the programs configuration file. Some, but not all, of these configuration options may be available through the GUI.

If `DIMG.ini` is present in the programs directory, DIMG will always use this file for loading and storing its configuration data. If this file is *not* present, DIMG will use what ever is normal for the operating system it is running on. Under windows, this means using the system registry. Under UNIX it will be something like `~/.config/zxnet/DIMG.conf`.

If configuration is stored in the Registry under windows, configuration keys will live under `HKEY_CURRENT_USER\Software\zxnet\DIMG\`.

If configuration is stored in a file the file is a standard INI file. The first segment of the Key is the group with the remaining segments forming the Key. For example, the key `State/Dirs/LastOpenDir` would appear in the INI configuration file as:

```
[State]
Dirs\LastOpenDir = "C:\somewhere\"
```

Note that DIMG does not automatically create keys in the configuration file. If a key is not present the program will use its hard-coded default value or ignore the setting it controls entirely. If a setting is changed from the GUI and the key does not exist it will be created.

B.1 Core

DQDriver Location

Key:	`Plugins/DQDriver`
Type:	String
Example Value:	`dqnull.dll`

Description

The name of the drive access plugin to load. Overrides the built-in default filename.

Image Format Plugin directory

Key:	`Plugins/ImageFormatDir`
Type:	String
Example Value:	`plugins/fileformats/`

Description

The location to scan for Image Format plugins.

Last Open Directory

Key:	`State/Dirs/LastOpenDir`
Type:	String
Example Value:	`C:\somewhere\`

Description

The directory that was last used to open an image file.

Last Save Directory

Key:	`State/Dirs/LastSaveDir`
Type:	String
Example Value:	`C:\Somewhere\`

Description

The directory that was last used to save an image file. If this key does not exist it will not be created and the `State/Dirs/LastOpenDir` will be used instead.

UI Style

Key:	`UI/Style`
Type:	String
Example Value:	`PLATFORMDEFAULT`

Description

The UI style to use. Valid values are one of:

1. PLATFORMDEFAULT
2. CLEANLOOKS
3. PLASTIQUE
4. WINDOWS
5. CDE
6. MOTIF

The setting is *not* case sensitive.

UI Stylesheet

Key:	`UI/StyleSheetFile`
Type:	String
Example Value:	`C:\DIMG\style.txt`

Description

The Qt Stylesheet to apply to the program on startup.

B.2 File Handling

Maximum Cache Size

Key:	`FileHandling/maxCacheSize`
Type:	long
Example Value:	`5242880`

Description

The maximum disk image size to cache in memory. If the disk image is larger than this size it will be read and written to directly (changes will be made instantly). Some Image Format plugins may ignore this setting and cache the image in memory anyway.

Compression Level

Key:	`FileHandling/CompressionLevel`
Type:	integer
Example Value:	`6`

Description

The compression level to use for disk image file formats that support compression. `0` is no compression at all, `1` is minimum compression and `9` is maximum compression.

Ask for Comment

Key:	`FileHalding/AskForComment`
Type:	bool
Example Value:	`true`

Description

If true, the user will be prompted for a comment when ever saving a disk image as a new file in a format that supports a comment.

B.3 DQDLibDsk Drive Access Plugin

Track-At-A-Time

Key:	`DQDLibDsk/taatr_enable`
Type:	bool
Example Value:	`true`

Description

Try to enable Track-At-A-Time reads to get better performance. LibDsk does not currently seem to properly implement this ability so enabling this may have unpredictable behaviour.

UNIX Floppy Drives

Key:	`DQDLibDsk/drives`
Type:	Comma-separated String List
Example Value:	`/dev/fd0,/dev/fd1`

Description

A comma-separated list of UNIX floppy drives. This setting is ignored on non-UNIX systems.