

DirMaster/Style Documentation (v3.0.0)

 style64.org/dirmaster/documentation

 Produced by Style. Documentation last updated: August 16th, 2014

DirMaster v3.0

Helping Commodore enthusiasts explore and manage their disk image collections.

About DirMaster

DirMaster is a Windows-based GUI application designed to help Commodore enthusiasts explore and manage their disk image collections. DirMaster supports common (and uncommon) emulator disk image formats (such as .d64, .d81, .d2m, etc), almost every native Commodore archival format (such as .arc, .sda, .lnx, etc), and many native graphic formats (such as koala, doodle, amica, etc). DirMaster was designed to give users a perfect blend of familiar appearance (e.g. natural looking disk image contents using the PETSCII character set) and modern GUI functionality (e.g. drag and drop, opening multiple disk images at once). The first version of DirMaster was released in 2006.

Credits

DirMaster was coded by The Wiz/Style; with features designed by The Wiz/Style and Elwix/Style; icons drawn by saehn/Style. Documentation written by Elwix/Style. DirMaster v3.0 uses the wxWidgets GUI library.

Features

Highlights indicate new features found in v3.0; (*) denotes a prior feature that was notably improved in v3.0.

- disk image support: .d64, .d71, .d81, .d80, .d82, .d2m, **.d4m**, .dhd, **.hdd (read only)**, .dnp, .dfi, **.m2i (read only)**, .t64, .g64 (read only), **.nib (read only)**
- file support: .prg/.p00, .seq/.s00, .usr/.u00, .rel/.r00
- archive support: .arc, .sda, .lnx, .sfx, .ark, **.lbr**, **.spy**, .cvt, .wr3, zip/4 (1!..., 2!..., etc)
- integrated with Windows Explorer via file associations/icons
- **integrated with the Windows Explorer file preview panel**
- all the usual directory editing capabilities like sorting, block size change/correction, file type change, lock, splat, etc...
- most functions activated by keyboard equivalents, including file reordering and filename editing

- multi-level undo (*)
- pervasive drag and drop:
 - copy one or more files from one image to another by drag and drop
 - copy files between different image formats (e.g. d64 to d81, dfi to d71, etc) by drag and drop
 - move/reorder files on the same disk by drag and drop
 - drag and drop files from disk images to the OS file system
 - drag and drop files from the file system onto disk images
- **multiselection with shift+click & ctrl+click; many operations will work on the entire selection**
- invoke your favorite emulator or other external application (up to eight slots) from within DirMaster to run/process a file or disk (*)
- sub directory and partition support for relevant formats (d81, d2m, **d4m**, dnp)
- save disk images as a .txt, .csv (with optional MD5 hash per file), .bmp, **.png, .rtf. and .html**
- visual BAM mode with track/sector view/edit (hex and PETSCII)
- support for the most common error and extended disk extensions
- decompress archives directly from open disk images with one key click
- support for GEOS file types (**new: GeoPaint/GeoPhotoAlbum viewers, GeoWrite to seq conversion**)
- built in viewers for Commodore file types including **raw binary (disassembly and hex viewers)**, SEQ files, **REL files**, BASIC listings (v2.0 and v7.0), **character sets**, and **many popular image formats**
- recursive find: search your disk image collection with * and ? wildcards; **save results to a file**
- batch processing: recursively create .txt, .csv, .bmp, **.png, .rtf. and .html** or extract to prg/seq files
- batch processing: **send extracted files to an external script/executable**
- **experimental OpenCBM support**
- print directory listings for disk envelopes; **two sided directory printing**
- maintain your own favorite 'separators': drag and drop them onto your disk image
- disk forensics:
 - check for cross links
 - **list the sector chain for each file, as well as a list of free chains (potentially recoverable deleted files)**
 - **compare two disks at a sector by sector level**
- **customizable palettes; defaults to the "Pepto" palette but can use any .vpl (VICE) format palette file**

Known Issues

Known issues in the current version of DirMaster will be listed below, and we'll also include outstanding feature requests that we intend to add later.

This documentation is not finished!

Reporting Bugs

First, please check the known issues list above before reporting the same issue. Second, make sure you are using the most recently released version of DirMaster (see "[How to Check for Updates](#)"). If you are experiencing an issue with the most recent version that is not in the known issues list, then please post a bug report to us at <http://style64.org/dirmaster/bug-report-feature-request> rather than (or in addition to) posting on various forums and message boards. Reporting issues directly to us will ensure that we see the issue and that we will follow up with you.

The most actionable bug reports are those with steps for repeating the issue you are having. Be as specific and detailed as possible describing how to reproduce the problem. If you can't provide a reproducible bug, we're still happy to accept your report via the form linked above. If you can provide a disk image or file that is relevant to the issue or is necessary to reproduce the issue, make sure and let us know when you report the bug so that we can arrange to obtain a copy of the file(s) from you directly.

We consider deficiencies in this documentation to be worthy of reports as well - if you find anything to be unclear, missing, or incorrect please let us know. This includes spelling or grammatical errors.

Requesting Features

As with bug reports, go ahead and send us feature requests using the same form (<http://style64.org/dirmaster/bug-report-feature-request>). A number of the features that have been added since v1.0 are the direct result of our users requesting new functionality that furthers the mission of DirMaster. Before sending, please double check the known issues list above as we may also detail pending feature requests there.

User Support

Our bug/feature form is the best way to ensure someone sees your report/request; but if you'd like to meet us more directly to ask questions about DirMaster or our other productions, you're welcome to join our public IRC channel, #style64, on EFNet where we'll try to assist you. When you join, please introduce yourself and be patient as we may not always be immediately available.

System requirements

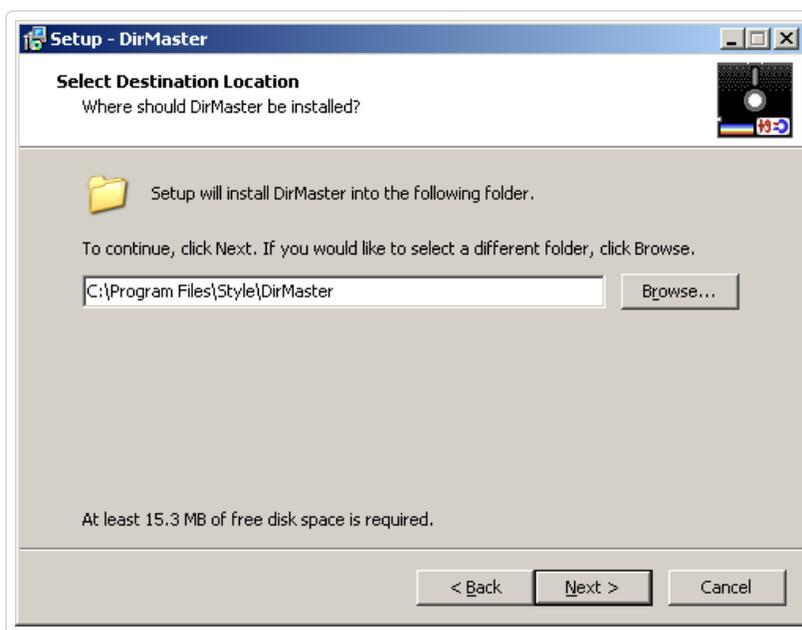
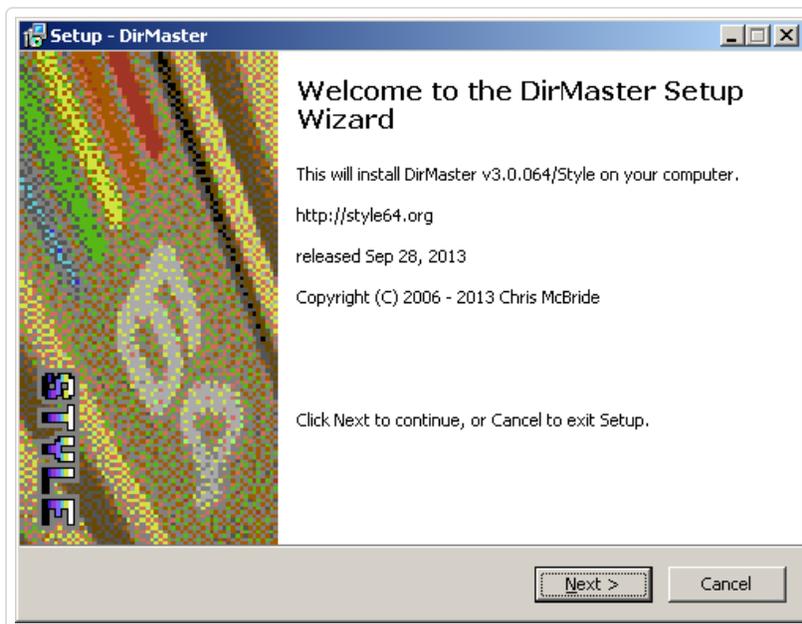
DirMaster v3 requires a Windows XP, Windows 2k, Windows Vista, Windows 7, or Windows 8 operating system. Some features of DirMaster v3 *may* work on Windows 98, but we do not test on that platform; DirMaster v2.2 is the last version that we have heard still functions reasonably on Windows 98.

DirMaster can optionally install a Windows Explorer preview panel extension (see "[Windows Explorer Integration](#)"); this feature requires Windows Vista, Windows 7, or Windows 8.

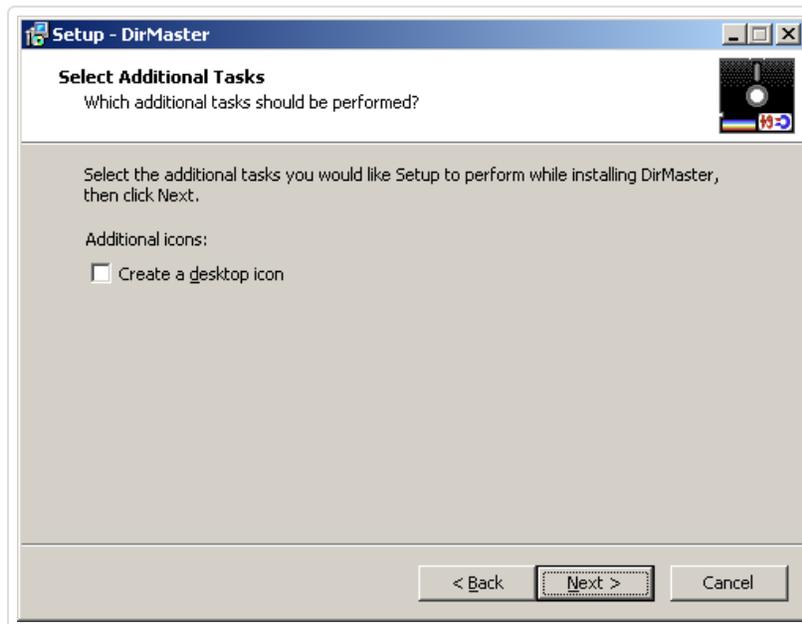
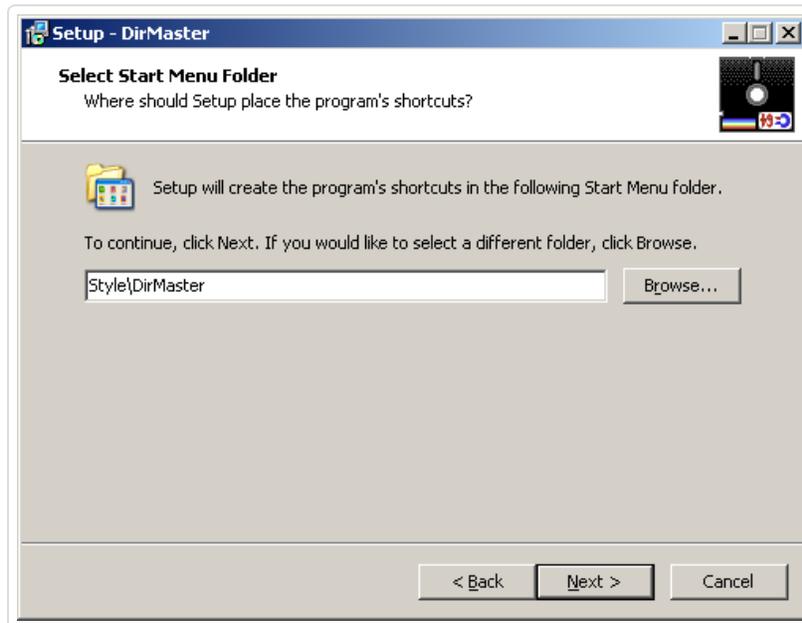
Experimental OpenCBM support in DirMaster v3 is optional and will require connection cables and native hardware devices. DirMaster v3 can communicate with an OpenCBM v4.2 dll library that you provide.

New installs

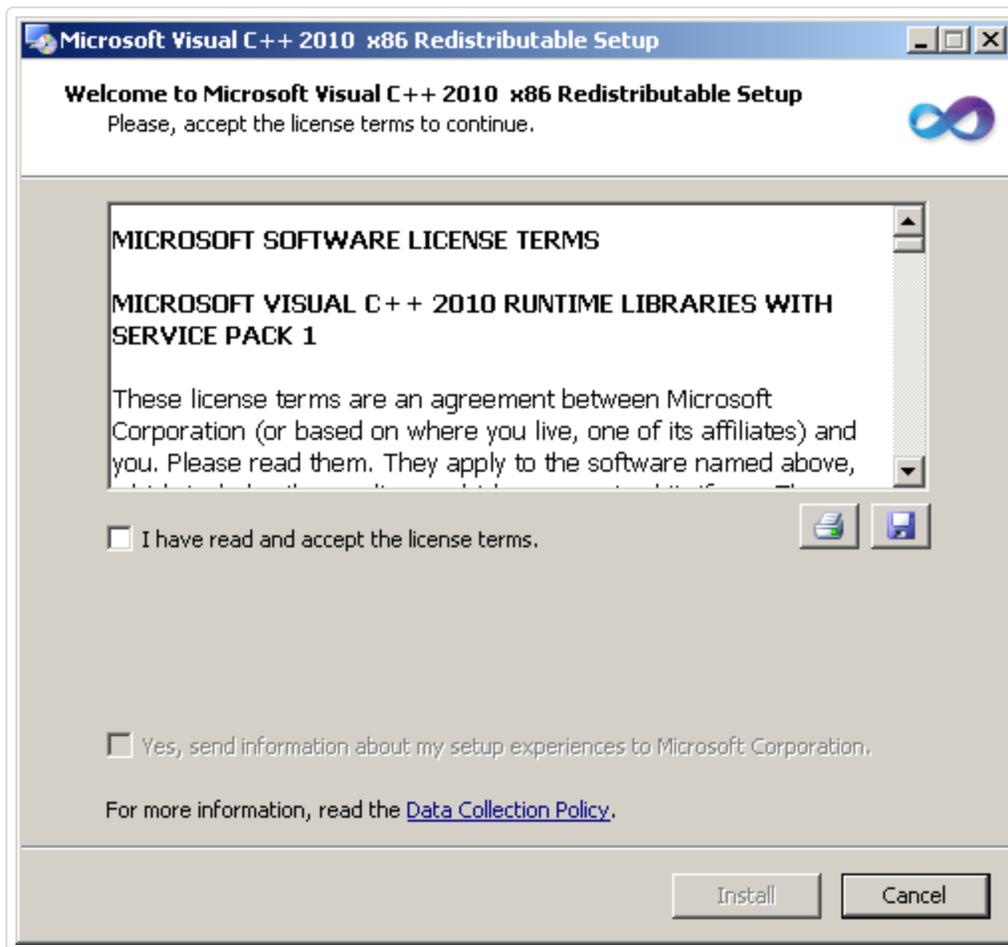
To install DirMaster, download the latest version and run the setup.exe. A User Account Control prompt will likely be shown requesting your permission for setup.exe to make changes to your system; it is safe to allow those permissions. The installer wizard will walk you through selecting a folder to install DirMaster into.



You can also change the default shortcuts folder and optionally create a desktop icon.



On some Windows platforms, it's likely that you may be asked to install the Visual C++ 2010 Runtime Libraries. This is normal, and you will need to finish this installation before DirMaster will work on your system. If you do not see a request to install the libraries, you do not need to download them yourself.



In addition to the DirMaster executable, one TrueType font will be installed in your system font directory. This TrueType font is used by the DirMaster user interface to display accurate PETSCII characters and disk image contents. It is also referenced in the exported .rtf and .html files that DirMaster can create when saving disk images or sequential files. The font can also be used outside of DirMaster in any application! You can learn more about this font at our [C64 TrueType](#) project page.

Upgrading

To upgrade DirMaster, download the latest version and run the setup.exe. A User Account Control prompt will likely be shown requesting your permission for setup.exe to make changes to your system; it is safe to allow those permissions. You may also be notified that the default installation folder already exists and will be prompted for whether you still want to install into that folder - it is safe to do so. When upgrading or reinstalling you may also be asked to reboot the machine in order to replace the Windows Explorer preview panel extension; you will not be required to reboot immediately but you will need to before any new preview panel functionality is available. The remainder of the upgrade installation should proceed as described above for new installs.

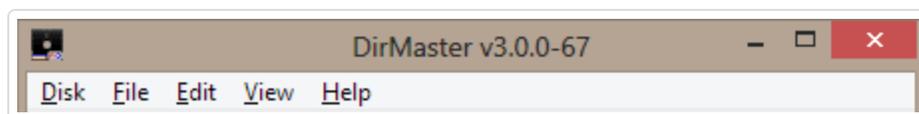
Special Notes on Upgrading from v2

Be sure and review your emulator settings and compare with the suggested settings given in the documentation below ([see here](#)); there are some new configuration arguments, at least one of which is designed to work around previous issues launching files with "non-ascii" characters in VICE emulator. Upgrading to v3 will **not** overwrite your old emulator settings - you must update those as you wish by hand.

This section will walk through and describe the first features that you'll want to look at and configure after installation. Nothing in this section is required to begin using DirMaster and if this is the first time you have used DirMaster you may want to skip ahead to "[Getting Around the UI](#)" and come back here later.

Check What Version You're Using

The DirMaster version number is displayed in the application titlebar at all times:

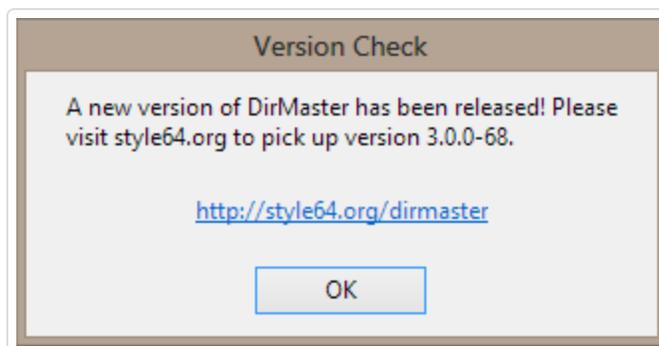


You can also open the About panel from the main menu ().

How to Check for Updates

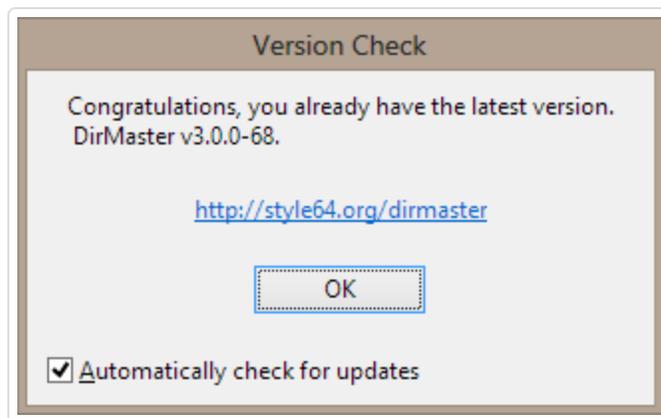
By default, new installations of DirMaster v3 will check for version updates on every launch. If a new version has been released, an update alert will display. From the alert, you can click the link to visit style64.org and download the update.

You can also force an update check from the main menu (). A similar alert will display, and will indicate if a new version has been released or if you are already running the most recent version.



Automatic Update Checking

By default, new installations of DirMaster v3 will check for version updates on every launch. You can disable/enable this behavior from the Version Check dialog accessed through the main menu (). Toggle the checkbox to enable and disable launch time version checks. Click the Ok button to commit your change and dismiss the dialog.



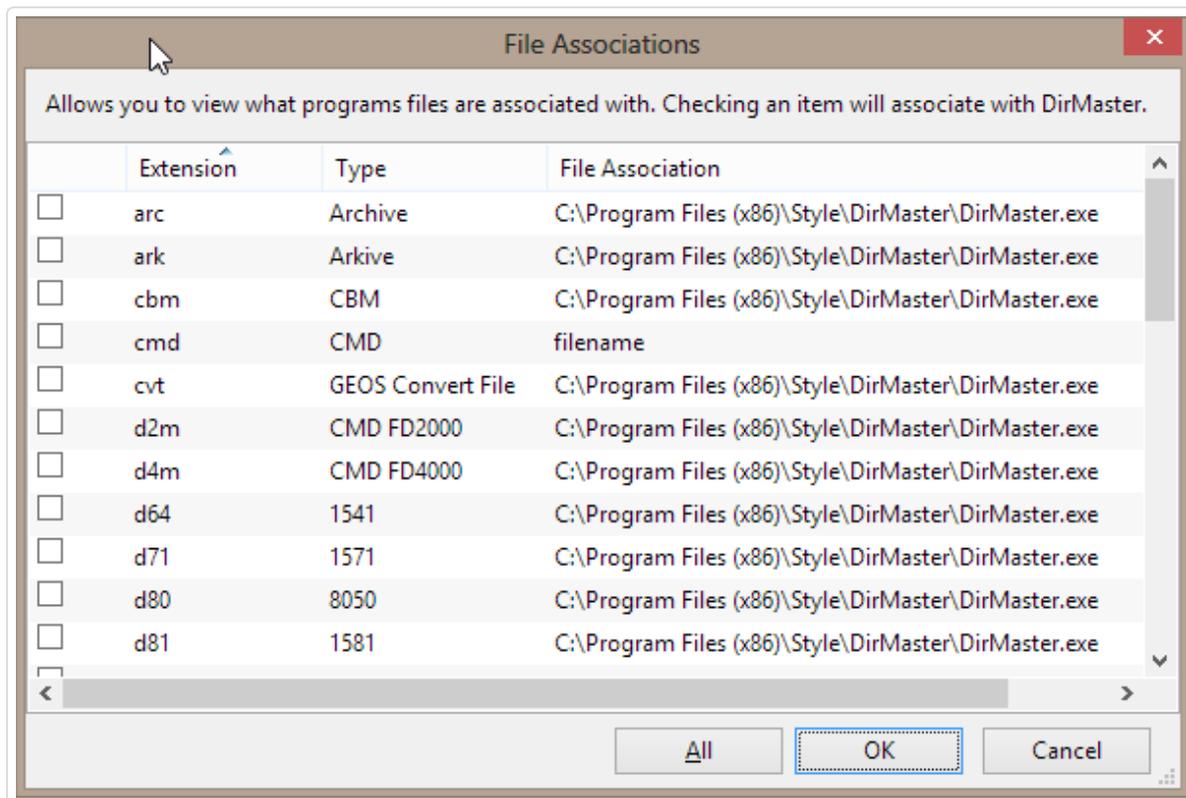
File Associations

DirMaster can associate itself with many popular Commodore filetypes based on extension. Associating files thusly will:

- enable a recognizable icon for files of each type in the Windows Explorer interface
- open files of each type directly in DirMaster when the file is opened via the Windows Explorer interface, launching DirMaster if necessary
- provide a customized file preview available through the Windows Explorer preview panel if you chose to install that option

To manipulate the associated file types, open the File Associations dialog through the main menu (). The list of file types recognized by DirMaster are displayed here, including file extensions. If a file type is associated with DirMaster, then that type's checkbox will be checked and the full path to the DirMaster executable will be shown in the far right column. File types that are not associated with DirMaster will be indicated by an empty checkbox; if another application is associated with that file type, then the full path to that application's executable is shown instead. Since some file extensions are not unique to Commodore disk images and other file types, it will not be uncommon to see other applications associated with those extensions.

See "[File Type Support](#)" for a list of all file types recognized by DirMaster.



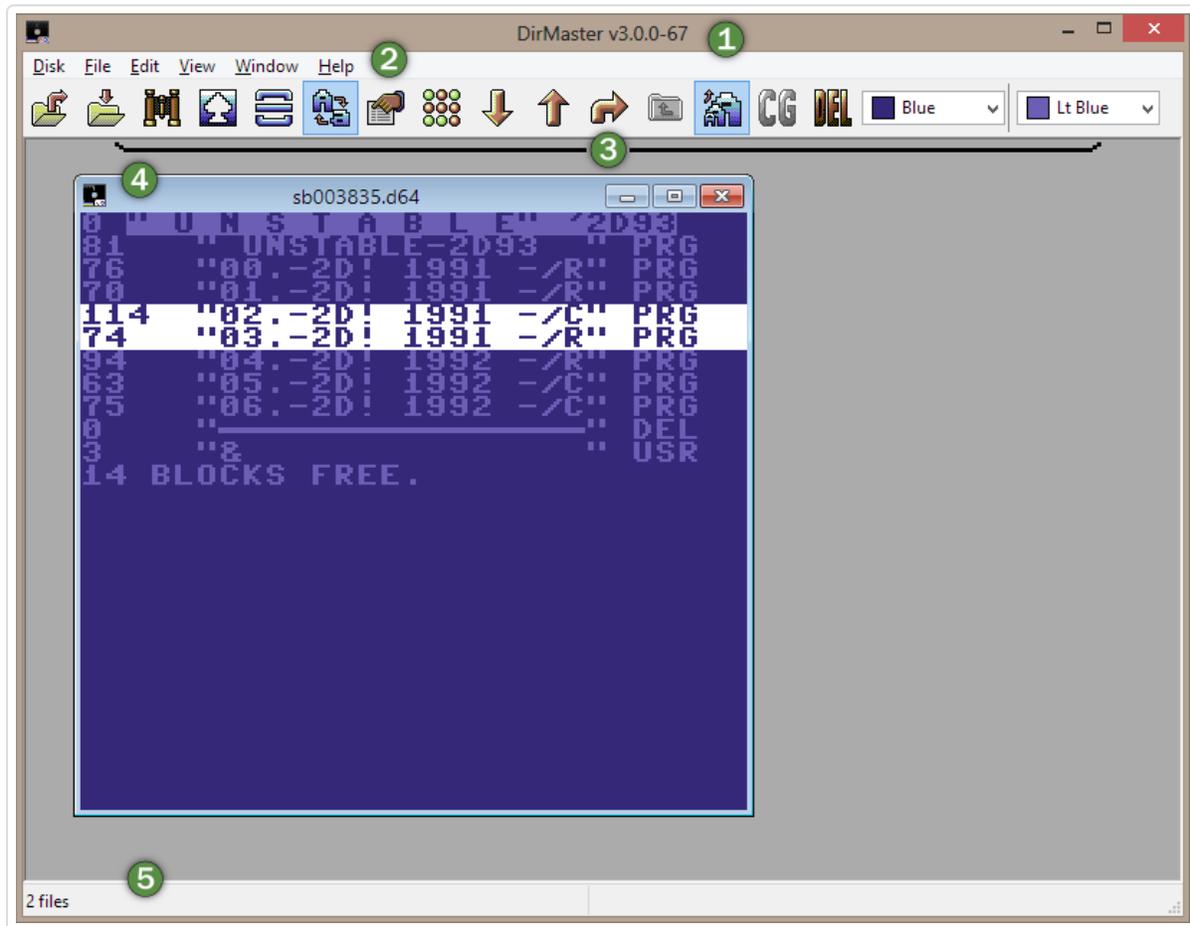
On the File Associations dialog, the All button can be used to quickly associate every file type in the dialog with DirMaster. Click the Ok button to commit any changes you made by toggling checkboxes and click the Cancel button to discard any changes and leave the associations as they were before the dialog was activated.

Depending on your Windows version, it is likely that administrator privileges will be required to change file associations. DirMaster will prompt for those privileges; this is expected behavior and necessary in order to make the Windows registry changes that set the associations.

This section will familiarize you with the basic user interface and help you understand at a high level how to work with disk images and files *within* DirMaster as well as how to work *between* DirMaster and the operating system. Features that adjust the way DirMaster displays disk images and file contents are also covered. If any terminology is confusing, you can skip ahead to "[Terminology Review](#)", then come back here.

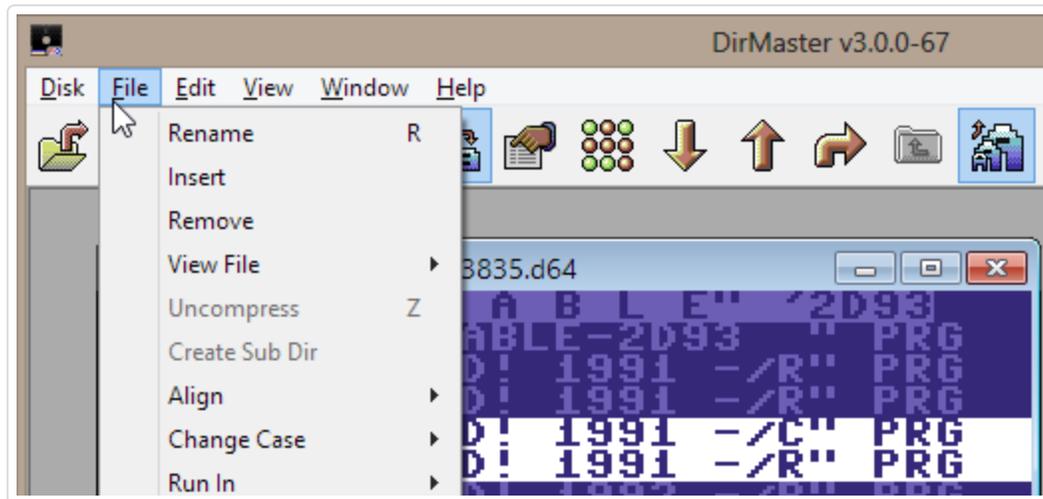
The Main Window

Observe the screen shot below, which shows the main DirMaster window with one disk image open. The numbered markers identify the main elements of the user interface and are explained below.



- 1 The application **title bar**; displays the version number and contains the minimize, expand and close buttons at the far right side of the title bar which affect the entire application
- 2 The **main menu**; most application features that work with disk images and files can be accessed through the main menu and its sub-menus; see below
- 3 The **toolbar**; most toolbar features are also available through the main menu; when you hover the pointer over a toolbar item, a tooltip will give a brief description of the item
- 4 A **disk image sub-window**; multiple disk images can be open at once
- 5 The **statusbar**; is used to convey brief information about whatever the user is currently doing; for example, displaying a count of selected file entries

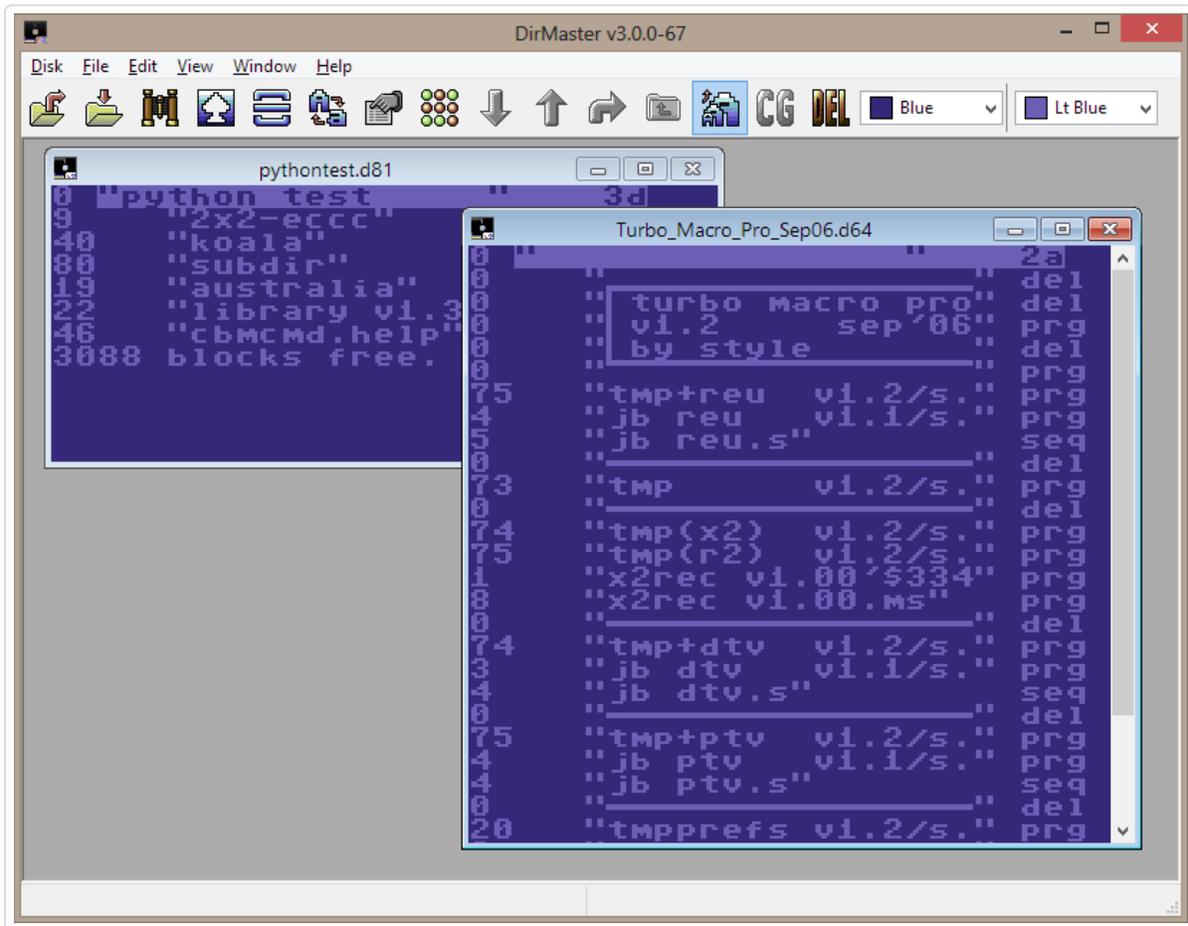
Some of the sub-menu items available through the main menu also have keyboard equivalents. Those keyboard equivalents are displayed to the right side of the sub-menu item. For example, as shown below, hitting the 'r' key will invoke the Rename operation on a file entry. Note that keyboard equivalents are shown in uppercase form for readability, but can be invoked by pressing the plain, unmodified key on the keyboard.



Working with Disk Images

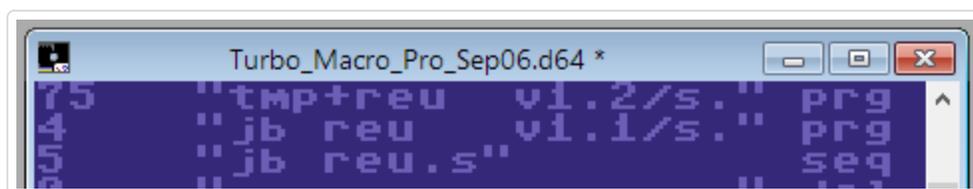
The most common entity you'll work with inside DirMaster is referred to as a "disk image". Disk images represent some kind of physical storage media and act as a container of native Commodore file types. The simplest example of a disk image is a .d64 file which represents a single side of a native 5¼ floppy disk formatted in 1541 compatible mode. A .d64 image would contain a single directory of files entries that point to actual native file data. A more unusual example would be a .dnp image which represents a single CMD partition; that isn't a 'disk' per se but is still a logical container for one or more partitions and native files.

In DirMaster, a disk image is represented by a child window contained inside the main window. DirMaster can have many disk images open at the same time. For example, the screenshot below shows two disk images open in DirMaster; a .d64 image and a .d81 image.



Disk image windows display the contents of the image in a manner as close as possible to how the disk contents would look when examined on native hardware. Floppy disk images therefore show the directory contents very similar to the output obtained by listing the directory from a real floppy disk inside a 1541 disk drive, complete with a disk name/id and blocks free at the bottom. To make disk image manipulation easier, special PETSCII characters that can appear in file names like color codes or cursor movements are suppressed (see "[True Directory Listing\(\\$\)](#)").

The disk image window title bar will show the name of the OS file that the disk image is stored as in the OS's filesystem. An asterisk after the name is displayed when the contents of the disk image open inside DirMaster are different than the contents of the OS file. In other words, the asterisk indicates the opened disk image has been edited or altered in some way. The screen shot below shows how the titlebar of an edited disk image looks:



The disk image's OS file is not changed until you save the image. You should consider an open disk image inside of DirMaster to be a '*soft copy*' of the actual OS file.

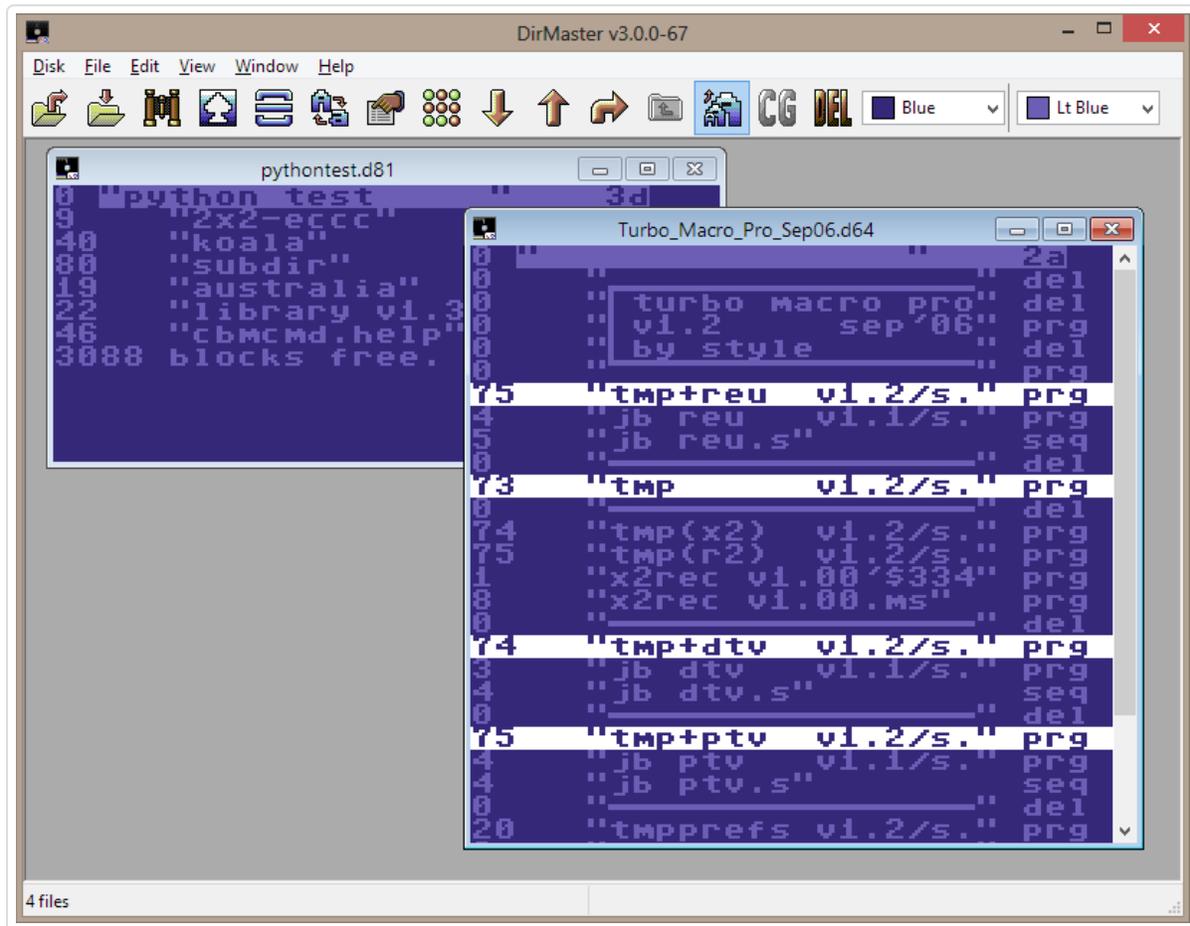
Disk image windows can be moved around anywhere inside the main window and can overlap and partially obscure each other. Scroll bars are shown on the disk image window when all of the contents cannot fit within the bounds of the window, as is the case with the .d64 disk image shown above. You can resize disk image windows by click-dragging the edge of the window. Disk image window controls operate the same way as any operating system window, allowing you to close the disk image, minimize the disk image to a small titlebar thumbnail, or maximize the disk image to fill all available space inside the main window.

Most functions and features that operate on the contents of an entire disk image (or multiple disk images) can be found in the main menu under . Additional functions can be found in the main menu under . See "[Managing Disk Images](#)" and "[Editing Disk Images](#)" for more details on using DirMaster to manage and edit disk images.

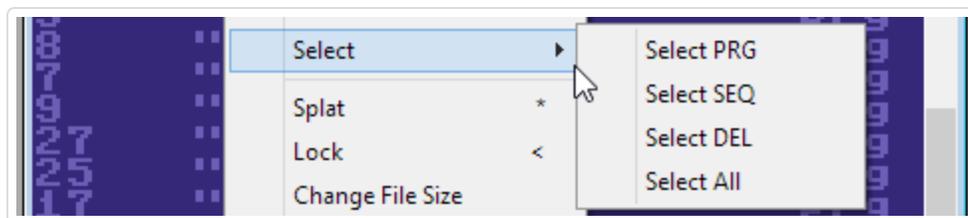
Working with Files

Like physical disks, disk images contain a directory which is a list of file entries. These file entries point to files that correspond with the same files found on the physical media that the disk image represents. In the same manner you might use native utilities on real Commodore hardware to manipulate file entries and files content on a physical disk, you can use DirMaster to manipulate the file entries and file content on the disk image. Where applicable, disk image directories may also contain partitions and sub-directories which can be viewed, manipulated or accessed through DirMaster as well.

Selecting file entries is as simple as single clicking them with the mouse. Selected file entries are highlighted with white backgrounds and reversed text. Multiple selection is possible by holding down the Control key while clicking entries. A set of file entries can be selected at once by holding down the Shift key and clicking. The screen shot below shows a disk image with multiple file entries selected.



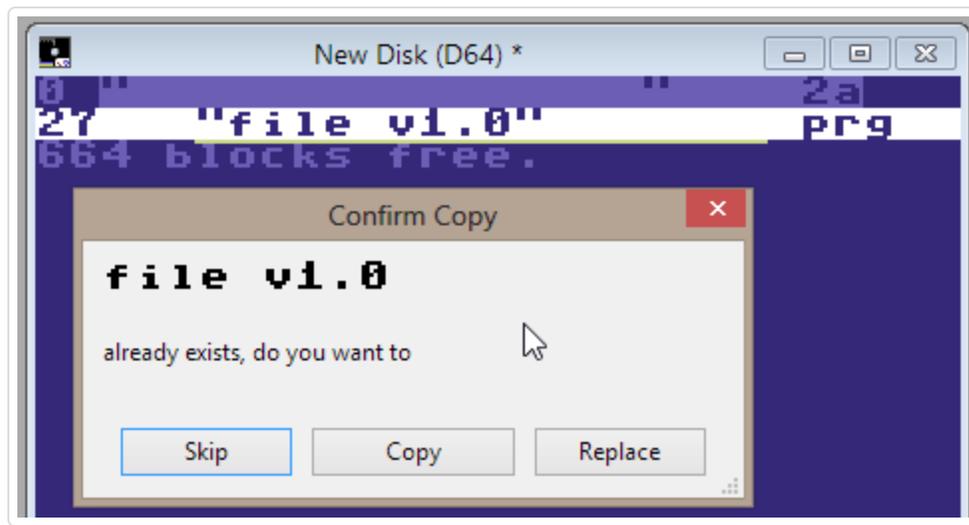
When a disk image is active, you can press Control + A to select all of the file entries. You can also use the right click context menu to select all of the file entries, or all of the file entries of a certain type:



DirMaster enables drag and drop manipulation of files both inside a single disk image and between two different disk images. Simply click and hold the mouse button on a file entry and then move the mouse to drag the file. It is easy to manually reorder files on a disk image by dragging and dropping them into place. When dragging *between* two disk images, the default operation is *copy* - the file or files you have selected and are dragging will be duplicated on the destination disk image and the source disk image will remain unchanged. Holding Shift while dragging files between images will change the operation to *move*, which will first duplicate the files on the destination disk image and then erase the files from the source disk image.

Duplicate File Names

DirMaster will recognize when you are attempting to copy or move a file onto a target disk image that already has a file with the same name. When this occurs, you'll be prompted for the action you want to take:



Click Skip to cancel the copy or move operation for that file; click Copy to proceed with writing the dragged file to the disk image - a file entry with the same name as the existing file will still be created; click Replace to remove the old file entry and write the dragged file and file entry to the disk image - the old file contents are not removed from the disk!

You can control the behavior of duplicate file name handling with a configuration option, **Confirm Copy on Drag/Drop**. This option is available on the Options dialog accessible through the main menu (). For more details on these and other configuration options, see "[Options Dialog](#)".

Most functions and features that operate on file entries can be found in the main menu under . Additional functions can be found in the main menu under . See "[Editing Disk Images](#)" for more details on using DirMaster to edit disk images, file entires and files.

Terminology Review

This documentation attempts to use a consistent terminology when describing the user interface and concepts like disk images, files and so on. This terminology can be confusing especially when referring to disk images and files inside DirMaster versus files that reside in your OS filesystem. Below we list and define these important terms, some of which you've already seen in use:

main window

The application window that contains the main menu, toolbar, and child windows; shown above under "[The Main Window](#)".

child window

Any window that opens inside and remains contained within the main window; for example, disk image windows as shown above under "[Working with Disk Images](#)".

dialog

Any window that opens and can be moved outside of the main window; for example, the PETSCII Charset dialog (see "[PETSCII Charset](#)").

active

"Active" is used to refer to whatever child window or dialog is currently in the foreground and has mouse/cursor focus.

disk image

A disk image is typically a representation of a physical Commodore disk format like a 1541 formatted 5¼ floppy. There are some exceptions, for example, .t64 represents a physical tape cassette while a .dfi represents a virtual disk used by the MMC64 hardware add on. The common requirement for a disk image is that it can contain *multiple* Commodore files with a directory of native filenames. In DirMaster, disk images are displayed using a disk image window.

disk image window

Any window inside the main window that displays a disk image directory and allows manipulation of the disk image.

disk image directory

Disk images contain files; the disk image directory is a list of file entries that point to the files stored on the disk image.

file image

A file image is any representation of a single Commodore file with its native file name intact, for example, a .p00 (an emulator format that contains a single PRG file). In DirMaster, file images are never opened on their own - they are always opened inside the context of a containing disk image.

file entry

"File entry" is used to specifically refer to a single entry in a disk image directory which contains a file name, type, size, and pointers to a file. Some operations can manipulate a file entry but have no effect on the file itself!

file

"File" is used to specifically refer to an actual Commodore file stored on a disk image, encompassing the file's content and whatever additional structure is used by the disk image to store and link the file's content.

OS file

Since "file" is a commonly used term associated with the contents of a disk image, this documentation uses "OS file" as a clarification when referring to any file that is physically stored on your OS's filesystem.

raw file

Used almost synonymously with OS file, but usually in the context of a OS file that contains a plain Commodore file without any other structure attached like a native file name. An example of a raw file would be a .prg file, contrasted with a .p00 file image which contains a native (PETSCII) file name.

OS directory

Since "directory" is a commonly used term to describe the list of files on a disk image, the documentation uses "OS directory" as a clarification when referring to the directory structures in your OS's filesystem.

Working with the OS

Your operating system provides access to your OS files that may contain disk images or other Commodore related content like single program files or single graphics files. DirMaster integrates with your operating system primarily through file associations which indicate that those OS files should be opened via DirMaster. Double clicking associated OS files in the Windows Explorer, for example, will launch DirMaster if it isn't already running and open that OS file inside DirMaster. Opening an OS file that is a disk image (like a .d64) will produce a disk image window and display the disk image directory within it. Opening an OS file that is a file image (like a .p00) will cause DirMaster to first create a disk image window with a blank (empty, as if newly formatted) disk image and then duplicate the file contained in the file image onto the blank disk image. Opening an OS file that is some other Commodore raw file, like a program file (.prg), will cause DirMaster to first create a disk image window with a blank (empty, as if newly formatted) disk image and then duplicate the raw file onto the blank disk image.

Opening Disk Images vs. Files

The difference is important: when you open an OS file that is a *disk image*, you'll be working with a 'soft copy' of the disk image. The open disk image inside DirMaster still references the OS file, but that OS file isn't modified until you deliberately commit your changes by saving the disk image. When you open an OS file that is a *file image* (like a .p00 or .s00 file) or is a Commodore *raw file* (like a .prg or .seq file), you'll be working with a newly formatted disk image that doesn't reference any OS file. That new disk image will contain a copy of the single file. Changes you make to the new disk image inside of DirMaster will never result in a change to the OS file - if you save the new disk image, that will result in a new OS file that contains the disk image. You can also export the file (see "[Exporting Files to the OS](#)") or drag/drop the file (see below) from inside DirMaster back out to the operating system which will either produce a new OS file or potentially overwrite the old OS file if you drop it in the same OS directory.

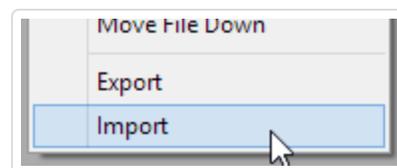
Aside from automatically opening associated files, you can drag and drop OS files from your filesystem to the DirMaster window. Dropping an OS file on the empty area of the main window is the same as opening it. But dropping an OS file onto an already open disk image will be the same as importing the OS file onto the disk image - even if the dropped OS file is a disk image itself.

Importing Files from the OS

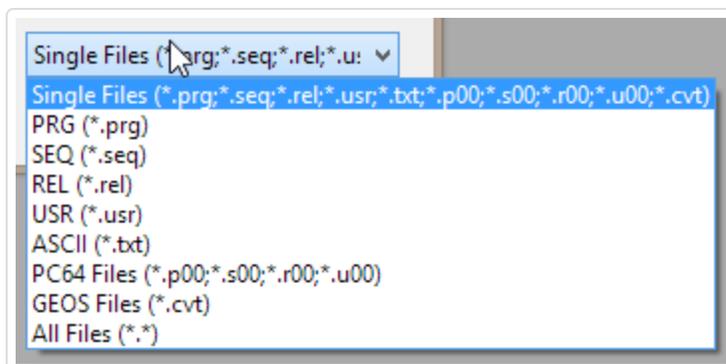
The preceding documentation touched briefly on *importing* files onto existing disk images. There are two ways to import a file from your OS onto a disk image: dragging and dropping

the file from the OS onto an open disk image, and using the main menu to invoke the import feature ().

Using the Import feature will display an Open dialog from where you can browse for the OS file you want to import onto the disk image. Generally, you can import any file image (like a .p00) or raw file (like a .seq). While you *can* import any kind of file, supported file images and raw files are recognized and handled predictably. A selector on the Open dialog lets you limit the file listing to certain types of file images and raw files as shown below:



File images you import will be placed on the disk image using the file type and native file name present inside the file image. Raw files you import will adopt their file type from the OS file extension, and their file name from the OS file name - for example, importing "test file.prg" will produce a file entry named "test file" with the PRG file type.



Dragging and dropping files from the OS onto an open disk image will behave the same way as if you had explicitly imported them using the menu item and Open dialog.

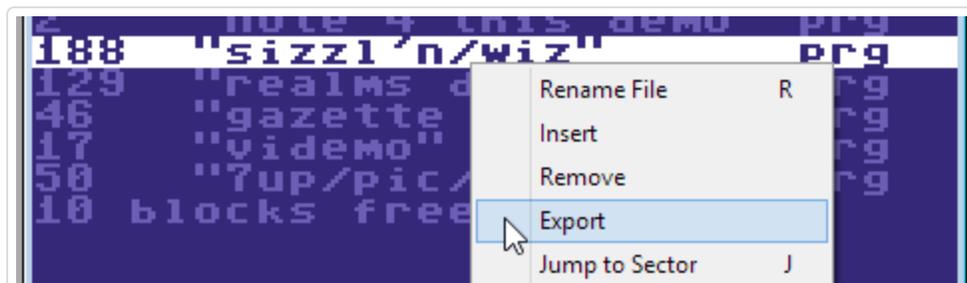
Automatic .txt Conversion

When importing .txt files, DirMaster will automatically convert from ASCII to PETSCII. This applies to .txt files that are drag/dropped onto disk images - they are converted into a sequential file using the PETSCII character set.

Exporting Files to the OS

There are three ways to export a file from an open disk image out to your OS:

- drag and drop the file entry from an open disk image somewhere outside the main window, onto your OS desktop or OS file manager like Windows Explorer
- select the file entry and use the main menu ().
- right click the file entry and use the menu item



When using the Export main menu or content menu, an Export As dialog will display that lets you choose a destination OS directory and file name for the exported file. You will also be able to choose a destination file extension applicable to the Commodore file type of the file you are exporting. For PRG, USR, DEL, and REL files, you'll be able to select either the file image equivalent (like .p00) or the raw file equivalent (like .prg). For SEQ files, you'll be able to choose from several file types:

PETSCII (.seq)	A Commodore sequential file.
HTML (.html)	An HTML file; utilizes a Unicode representation of the PETSCII character set and references the "C64 Pro Mono" font installed by DirMaster to display accurate PETSCII glyphs. Uses a lt blue/blue foreground/background.
ASCII (.txt)	A text file; PETSCII is translated into a best match ASCII representation.
Bitmap (.bmp)	A Windows BMP file; a picture of the entire sequential file content. Uses a lt blue/blue foreground/background.
PNG (.png)	A PNG file; a picture of the entire sequential file content. Uses a lt blue/blue foreground/background.
PC64 (.s00)	A .s00 file which is a file image of a sequential file.
RTF (.rtf)	A Windows rich text file; utilizes a Unicode representation of the PETSCII character set and references the "C64 Pro Mono" font installed by DirMaster to display accurate PETSCII glyphs. Uses a black/white foreground/background.

Files you export as any file image will be written with the file type and native file name of the file entry. Files you export as any raw file will be written with an extension adopted from the file entry, and their OS file name will be derived from from the native file name.

Dragging and dropping files from an open disk image to the OS will behave the same way as if you had explicitly exported them using a menu item and Export As dialog; the default file type will be the *raw file* equivalent.

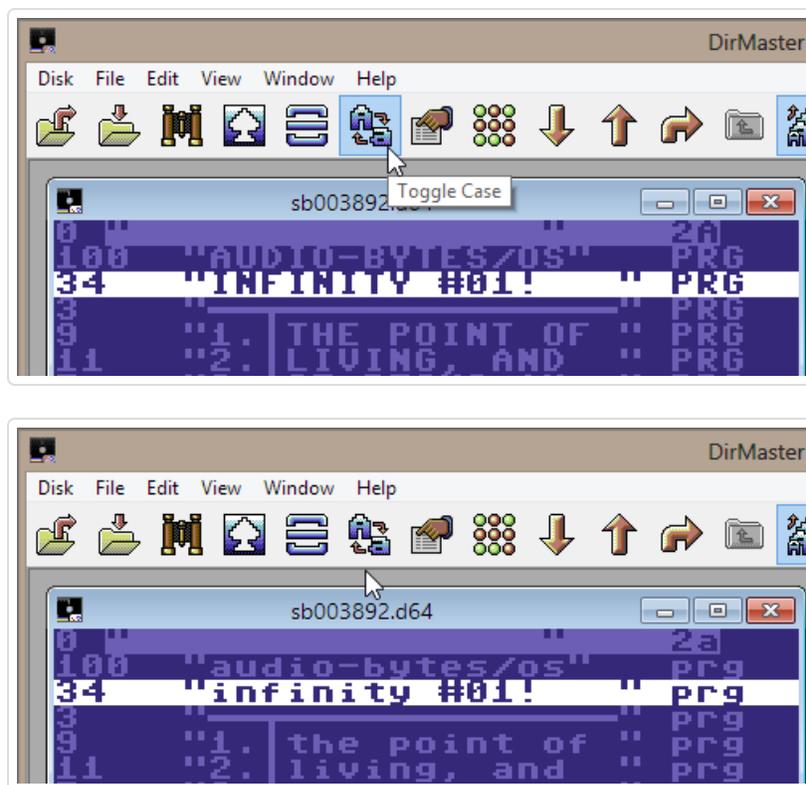
You can also drag and drop files from inside DirMaster out to your OS file system. Just select one or more file entries on an open disk image, and then drag and drop them outside of the main window.

Tweaking the UI

DirMaster offers several ways to change the way the UI looks, particularly with regards to how disk image directories are displayed.

Toggleing Character Mode

Disk image directory displays, and other displays of PETSCII text like sequential file contents or sector bytes are all shown with respect to a current character mode. This mode corresponds with the real Commodore character modes, usually called "upper/graphics" and "lower/upper". You can toggle the current character mode by clicking the Toggle Font toolbar button as shown below:



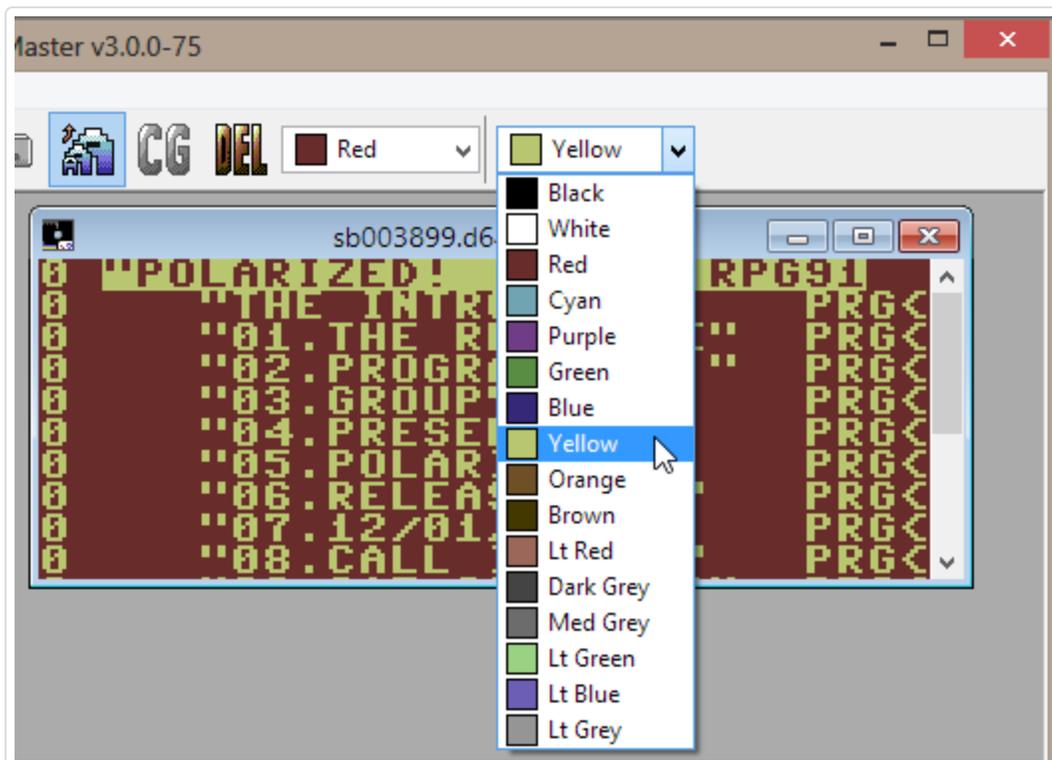
There are two other ways to toggle the character mode:

- Via the main menu; ()
- By pressing Control + Shift

Note that changing the mode affects only the currently active window (like a disk image window). For example, it's possible to toggle the mode on one disk image to upper/graphics mode, while at the same time viewing a sequential file from that disk in lower/upper mode. Changing between the windows will update the toolbar button so keep it in sync with the current mode. Note also that new windows will adopt the mode that the toolbar button was last set to, and not the mode of the currently active disk image!

Changing Colors

DirMaster provides control over the background and foreground colors used when displaying disk image directories and other PETSCII based displays. The color selectors at the right edge of the toolbar are used to set each color (background and foreground in that order); they default to the standard blue/light blue of the C64; the example below shows the color selectors set to red/yellow:



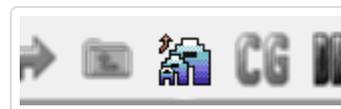
Disk image windows that result from opening a disk image or from creating a new disk image will take their colors from the two color selectors. Other windows, like the Sector Editor and sequential viewer, will inherit their colors from whatever disk image window the sector or sequential file came from.

You can take direct control of the color palette used in DirMaster as well, but there is currently no user interface to do so. Configuration is done via a file as explained in "[Configuring the Palette](#)". The default palette that DirMaster comes with is the well known "Pepto" palette.

Changing Font Size

The font size of PETSCII characters used throughout DirMaster can be adjusted at two levels. First, coarse grained controls allow toggling between a small and a large size; these sizes correspond to 8 pixel (1x1) and 16 pixel (2x2) sizing on the C64. You can quickly toggle between these sizes using the Toggle Font Size button on the toolbar:

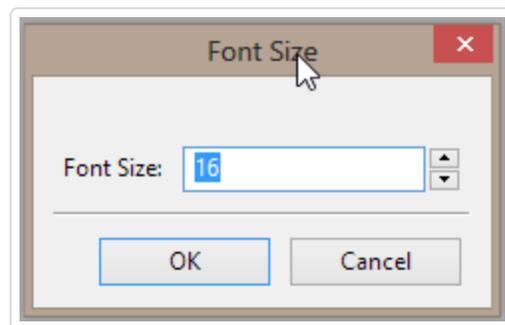
Second, fine grained control to any pixel size is possible by using the Font Size dialog accessed via the main menu ():



Use the text field and/or spinner control to set the font size, then click Ok to accept the new size and dismiss the dialog.

Note that changing the font size affects only the currently active window or dialog. If you make a different window active, the Toggle Font Size button in the toolbar will sync with whatever size that window was set to, or it will reflect the closest match in the event you set

the font size to a value other than 8 or 16. Note also that new windows will inherit the font size that either the toolbar button was last set to, or that was set via the Font Size dialog.



Appearance re: Exported Files

Character mode, colors and font size settings not only affect the visual appearance of the user interface, but they are used where relevant when exporting disk image directories and sequential files as certain other file types. Namely, features that output to .bmp, .png, .rtf and .html will respect these appearance settings and produce output accordingly.

This section will run through most of the ways in which you can use DirMaster to manipulate entire disk images. Important format-specific considerations are also reviewed. None of the features described below will directly change an existing disk image - for details on how to edit disk images and particularly disk image directories, see the next section.

Opening Disk Images

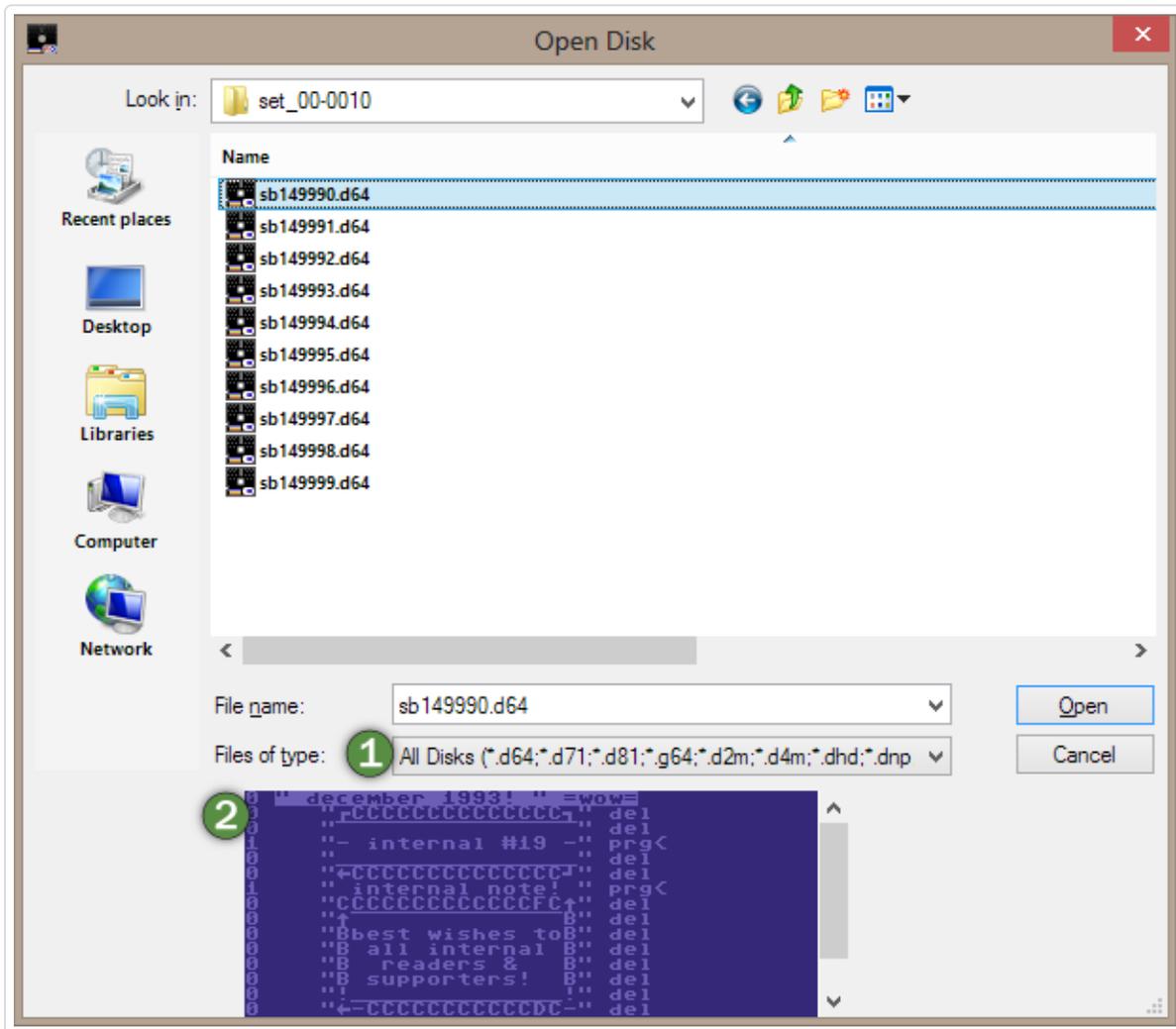
To open existing disk images, you can:

- Drag and drop the OS file of the disk image over the main DirMaster window. You can drag and drop multiple OS files to open them all at once.
- Double click or otherwise launch the OS file of the disk image, which will open the disk image in DirMaster if an appropriate file association has been made (see "[File Associations](#)").
- Invoke the Open Disk dialog through the main menu (); see below for more details about the Open Disk dialog. The keyboard shortcut is Control + O.
- Invoke the Open Disk dialog by clicking the Open toolbar button:

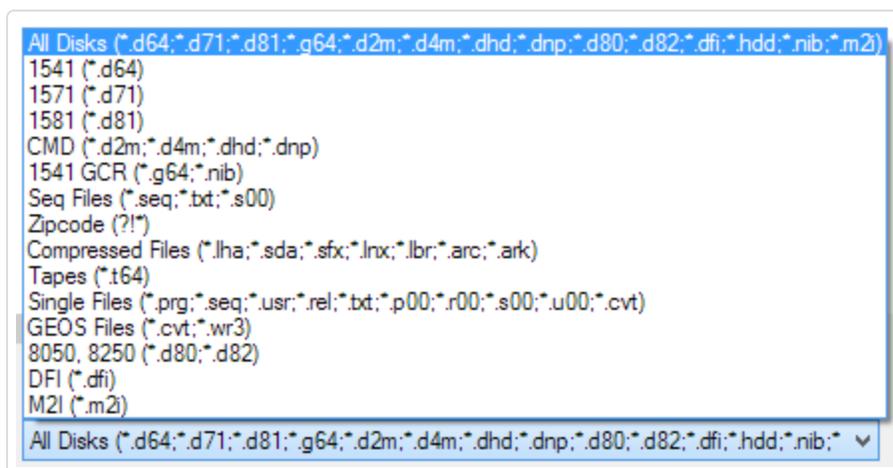


The Open Disk Dialog

The Open Disk dialog functions like any native file open dialog with two additional noteworthy elements.



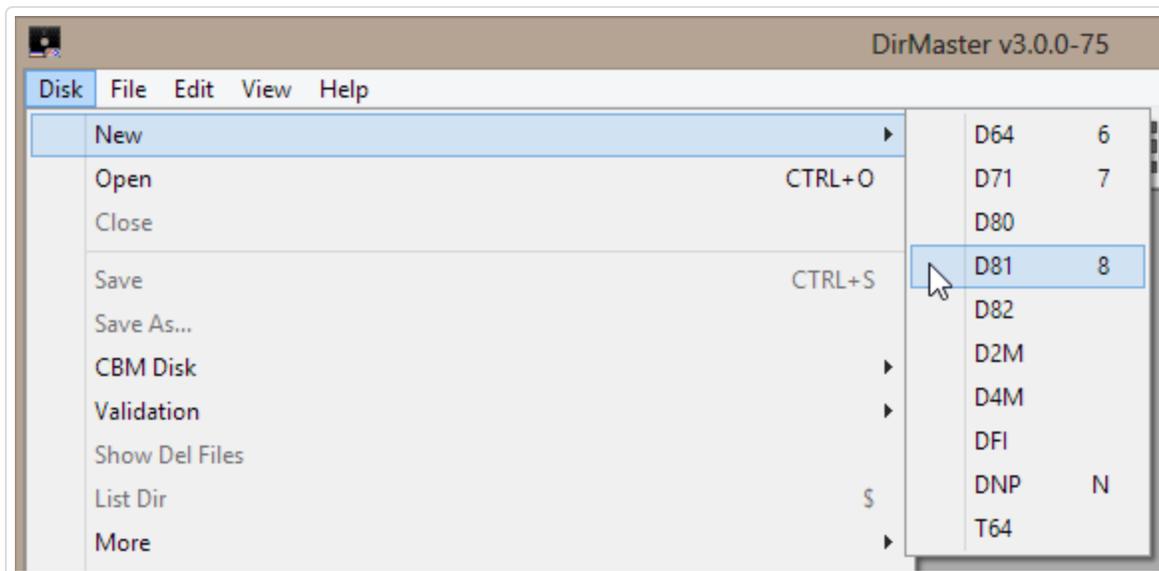
- 1 The **Files of Type** selector; lets you limit the file listing to certain types of Commodore disk images, file images or raw files as shown below:



- 2 The **Preview** panel; will display a PETSCII representation of the disk image directory of the currently selected file in the file browser; use the scrollbar to scan up and down when the disk image directory does not fit entirely within the panel.

Creating Disk Images

You can create new, empty disk images for many of the disk image formats supported by DirMaster. Create new disk images by selecting the disk image type through the main menu ().



As shown above, there are keyboard equivalents for a few of the more popular formats: .d64, .d71 and .d81 and .dnp. For all types other than .dfi, .dnp, and .t64, the new disk image is formatted as though you had issued a standard Commodore DOS new command on actual hardware, such as `open 1,8,15,"n0: , ":close 1`. DirMaster uses a blank disk name and disk id when creating new disk images. For .dfi, the disk image is created with no entries and initial space for 221 disk blocks - DirMaster will automatically resize .dfi images to accommodate their contents. For .t64, which actually represents a tape cassette, the disk image is created with no entries; note that .t64 has no internal structure for size or blocks free and therefore will not display a "blocks free" line at the bottom of its directory listing. For .dnp, which is a CMD native partition, you will be prompted for the initial image size.

Controlling the Format of New Disks

There are two configuration options that control how new disks are created and filled:

- **Fill Byte**
- **Use Fill Byte At Disk New**

These options are available on the Options dialog accessible through the main menu (). For more details on these and other configuration options, see "[Options Dialog](#)".

Saving Disk Images

To save a disk image (including newly created disk images) that you currently have open in DirMaster to a new OS file (a different OS file than you opened it from), you can:

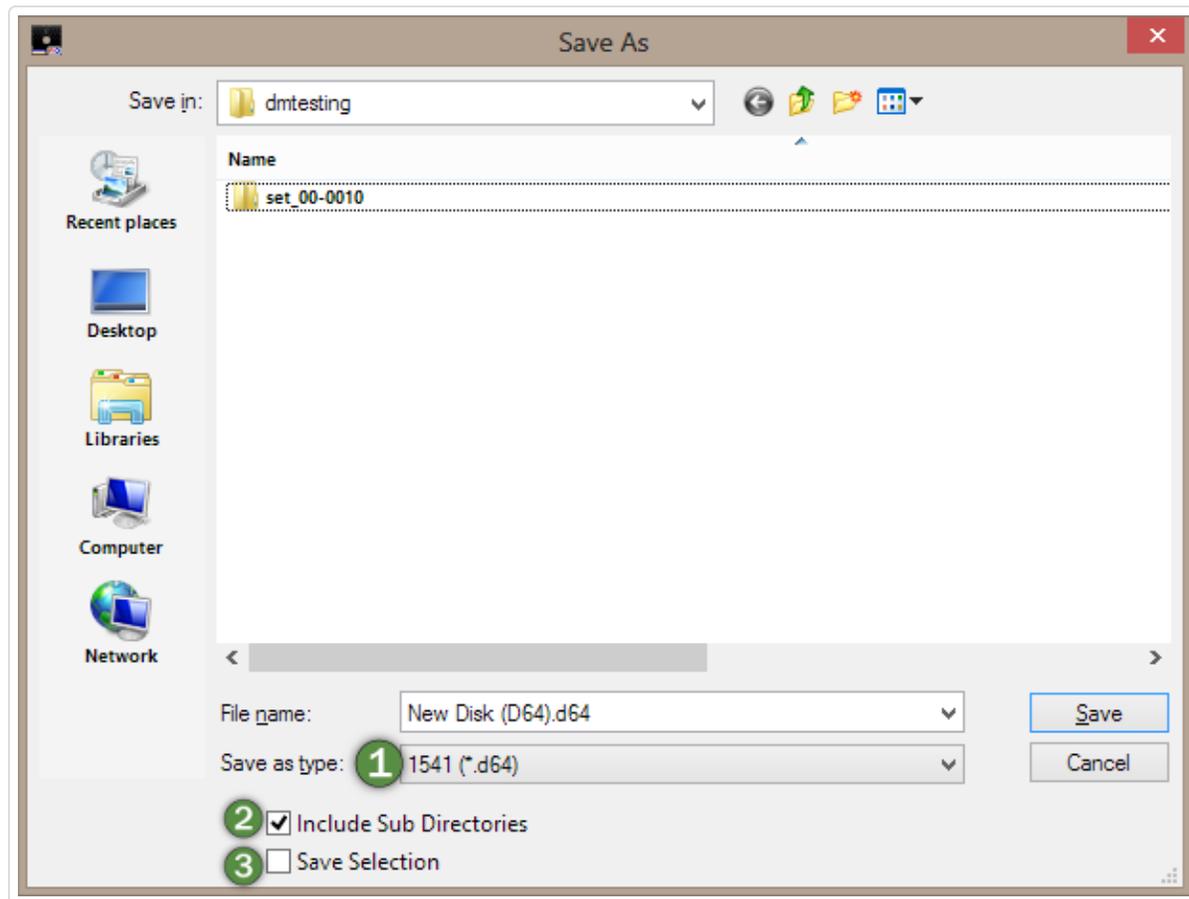
Invoke the Save As dialog through the main menu (); see below for more details about the Save As dialog.

To save an edited disk image that you opened from an existing OS file back to that same OS file, you can:

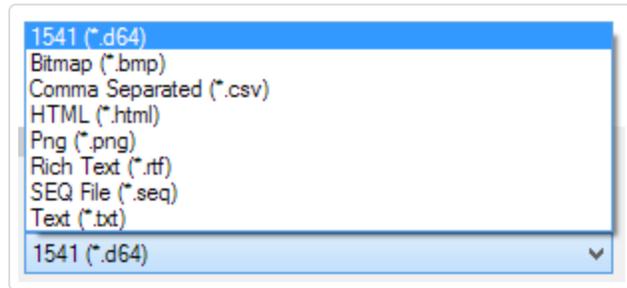
- Save the disk through the main menu (). The keyboard shortcut is Control + S.
- Save the disk by clicking the Save toolbar button:

The Save As Dialog

The Save As dialog functions like any native file save dialog with three additional noteworthy elements.



- 1 The **Save As Type** selector; lets you choose between writing the disk image to an OS file in its standard disk image format and exporting a representation of the disk image directory in one of several alternative output formats as shown below:



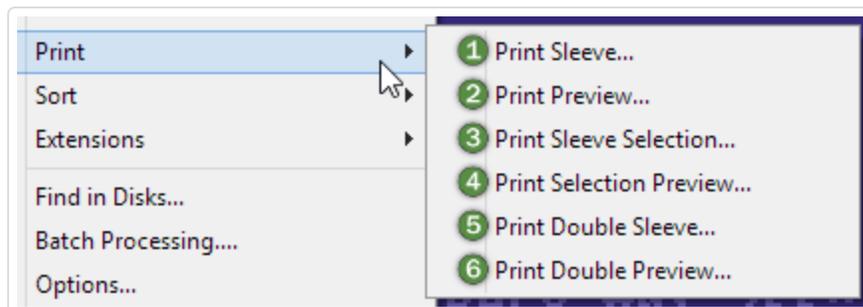
Bitmap (.bmp)	A Windows BMP file; a picture of the full disk image directory identical to how it is displayed within DirMaster. Subdirectories and partitions are not included in the image.
Comma Separated (.csv)	A comma separated file appropriate for opening in a spreadsheet program or by other applications and scripts for further processing. PETSCII is translated into a best match ASCII representation.
HTML (.html)	An HTML file; utilizes a Unicode representation of the PETSCII character set and references the "C64 Pro Mono" font installed by DirMaster to display accurate PETSCII glyphs.
PNG (.png)	A PNG file; a picture of the full disk image directory identical to how it is displayed within DirMaster. Subdirectories and partitions are not included in the image.
RTF (.rtf)	A Windows rich text file; utilizes a Unicode representation of the PETSCII character set and references the "C64 Pro Mono" font installed by DirMaster to display accurate PETSCII glyphs. Uses a black/white foreground/background.
SEQ (.seq)	A Commodore sequential file.
Text (.txt)	A text file; PETSCII is translated into a best match ASCII representation

- 2 The **Include Sub Directories** checkbox; this is only applicable for .csv, .html, .rtf, .seq, and .txt output; it has no effect for .bmp and .png. When checked, the output will contain a recursed directory listing that includes sub-directories and partitions.
- 3 The **Save Selection** checkbox; when checked, the output will contain only those file entries that were selected (highlighted) at the time the Save As dialog was invoked.

Printing Disk Images

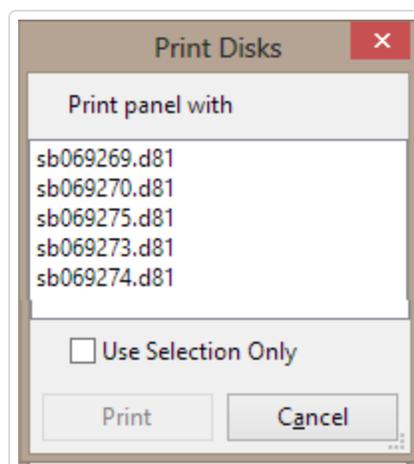
DirMaster can print disk image directories. It comes with a simple built in disk sleeve template onto which the disk image directory is placed; when printing, the filled in sleeve template is sent to whatever print device you've chosen. The printed sheet is suitable for

cutting and refolding into an actual 5¼ disk sleeve; alternatively, the printed section could be cut out and affixed to an existing disk sleeve. Invoke printing via the main menu (); see the Print sub-menu below:



-
- 1 ; inserts the entire directory from the active disk image onto the sleeve template and opens a dialog
-
- 2 ; open a window with a visual preview of what the print output would be
-
- 3 ; inserts *only the selected file entries* from the active disk image onto the sleeve template and opens a dialog
-
- 4 ; open a window with a visual preview of what the print output would be with *only the selected file entries*
-
- 5 ; only enabled when you have two disk images of the same or similar type open (e.g. two .d64 images, or one .d64 and one .g64). Opens a dialog that allows you to select one other open disk image (of the same or similar type):

The **Use Selection Only** checkbox will limit the print output to only those directory entries that were selected on each disk image at the time the print was invoked. Inserts both directories side by side on the disk sleeve template - the directory from the active disk image is always printed on the left side.



-
- 6 ; works as above but finishes by opening a window with a visual preview of what the print output would be.

The sub-window is shown below with one disk image directory:

A file entry is marked as deleted when its filetype is set to "DEL" and its closed flag is unset; on real hardware, this would normally only occur as a result of a Commodore DOS command to remove the file. In DirMaster, you can also mark a file entry as deleted (see "[Deleting File Entries](#)").



It is the closed flag that differentiates a *deleted* file entry from a file entry whose filetype has simply been set as "DEL"; the latter is commonly used for so-called file separators. Standard Commodore DOS behavior is to omit these file entries when listing a directory. When DEL file display has been toggled on, those file entries marked as deleted will be shown in the directory; their filetype displays as "*DEL". It is likely that in addition to these deleted file entries, a series of *unused* file entries will also be displayed. This is because DirMaster does not assume anything about the file entries that it sees with the DEL filetype and blank closed flag. Therefore, whatever unused space is left in the final directory sector will be interpreted as a sequence of "*DEL" files and included in the DEL file display. In the examples below, the disk image is being viewed without DEL File Display in the top screenshot and with DEL Files Display in the bottom screenshot; the bottom screenshot shows a file entry for "load", which is probably an actual *deleted* file, and four more *unused* file entries.

```

sb158406.d64
0 "marble madness" 52a
36 "marble madness" prg
36 "ea" prg
2 "loader" prg
21 blocks free.

```

```

sb158406.d64
0 "marble madness" 52a
36 "marble madness" prg
36 "ea" prg
2 "loader" prg
0 "load" *del
0 "CCCCCCCCCCCCCCCCCCCC" *del
0 "CCCCCCCCCCCCCCCCCCCC" *del
0 "CCCCCCCCCCCCCCCCCCCC" *del
0 "CCCCCCCCCCCCCCCCCCCC" *del
21 blocks free.

```

One possible reason that you may want to toggle DEL file display on would be potential recovery of a file that was deleted. By setting the closed flag (selecting the file entry and pressing the * key to toggle the so-called splat off), and changing the filetype to PRG

(selecting the file entry and pressing the p key) you will make that file entry visible to the Commodore DOS and you can check to see if the file itself is still intact, perhaps by testing it in an emulator.

True Directory Listing (\$)

DirMaster displays disk image directories in PETSCII within each disk image child window in a way that closely resembles how the actual directory appears when listed on real hardware. However, unusual file name contents like PETSCII delete (\$14), color codes, cursor movements, and so on are not shown to keep the interface clean and predictable. But DirMaster can simulate a true directory listing through the main menu () as well as by its keyboard shortcut, \$. The listing is shown in a new sub-window which works identically to the sequential file viewer (see "[SEQ Viewer](#)") and is therefore exportable.

This section will cover all the features provided by DirMaster to actually change disk image contents like directories, file entries, and sector bytes.

Editing the Directory

Renaming Files

Renaming files on a disk image can be accomplished in a few ways:

- by selecting the file entry and using the main menu ()
- by selecting the file entry and pressing the keyboard shortcut, r
- by right clicking a file entry and using the context menu item
- by single clicking over an already selected file entry; another way to think of this is as a slow double click: click once to select, pause, then click again to edit the file name

Each of those triggers will cause the in-place editor field to appear, allowing you to modify the file name:



Hitting return or clicking outside of the in-place editor field will commit the change.

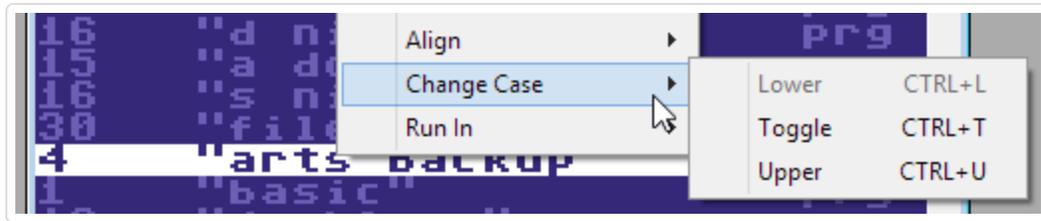
You can use the PETSCII Charset dialog (see "[PETSCII Charset](#)") to help insert special characters that are otherwise not able to be typed with the keyboard.

Changing File Name Case

To change the case of file names, you can:

- use the main menu () and select one of the sub-menu items there to change the case of the selected file entries

- right click a file entry and use one of the sub-menu items under the context menu:



- by using keyboard shortcuts that correspond with the sub-menu items in the main menu and context menu:

control + l	lower; changes the case of every character in the file name to its logical lowercase equivalent
----------------	---

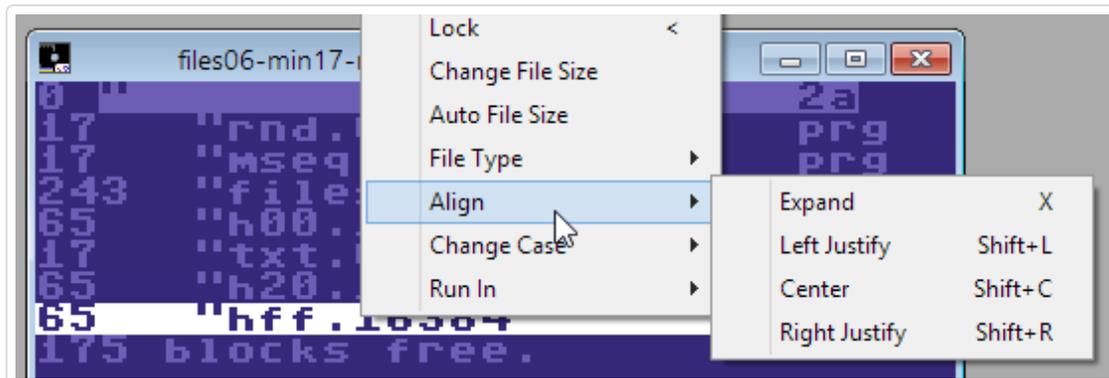
control + t	toggle; changes the case of every character in the file name to its logical opposite case equivalent
----------------	--

control + u	upper; changes the case of every character in the file name to its logical uppercase equivalent
----------------	---

Aligning File Names

To change the alignment of file names, you can:

- use the main menu () and select one of the sub-menu items there to align the selected file entries
- right click a file entry and use one of the sub-menu items under the context menu:



- by using keyboard shortcuts that correspond with the sub-menu items in the main menu and context menu:

x	toggles file name expansion; to expand a file name means to pad the right side with spaces so that it uses all 16 characters
---	--

shift + l	left justify; shifts the file name as far left as possible and pads the right side with spaces out to 16 characters
--------------	---

shift + c	center; centers the file name by padding with spaces on either side out to 16 characters
--------------	--

shift right justify; shifts the file name as far right as possible and pads the left side
+ r with spaces out to 16 characters

Changing File Size

Commodore disk directories have the interesting characteristic of encoding a file size along with the file name for every file entry in the directory; when listing a directory, the real Commodore DOS does not compute an actual size; instead it displays the value encoded in the file entry. Under ordinary circumstances this is kept in sync by the DOS, but the value itself can be edited to any two byte integer. DirMaster provides a convenient way to edit the size without resorting to use of the sector editor, which you can invoke:

- by selecting the file entry and using the main menu ()
- by right clicking a file entry and using the context menu item

Each of those triggers will cause the in-place editor field to appear, allowing you to modify the file size:

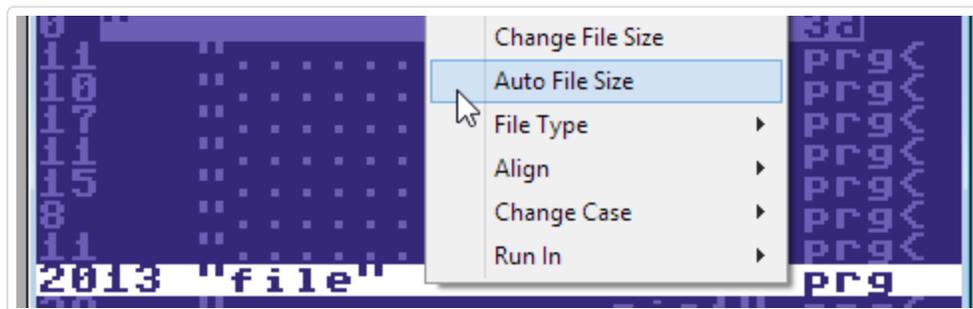


Hitting return or clicking outside of the in-place editor field will commit the change.

Correcting File Size

In the event that a file entry is showing an incorrect size, perhaps by using the feature described above, DirMaster can automatically change the displayed size to reflect the actual block size of the file:

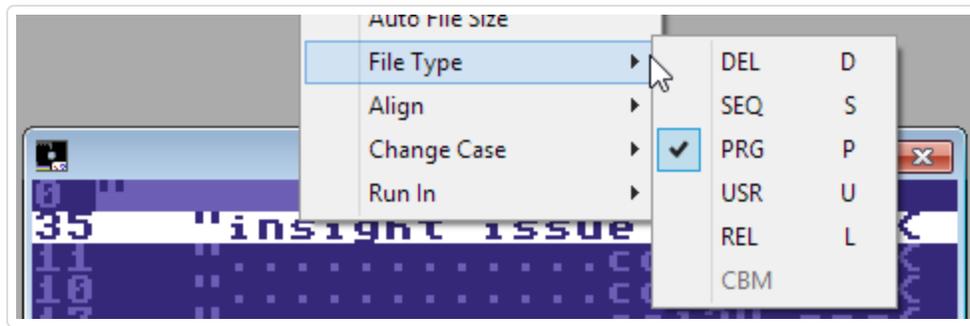
- by selecting the file entry and using the main menu ()
- by right clicking a file entry and using the context menu item



Changing File Types

All Commodore directory listings show the file type of each file entry as a three letter abbreviation to the right of the file name. These types can be changed without altering the actual file content, and DirMaster provides a fast way to do so:

- use the main menu () and select one of the sub-menu items there to change the type
- right click a file entry and use one of the sub-menu items under the context menu:



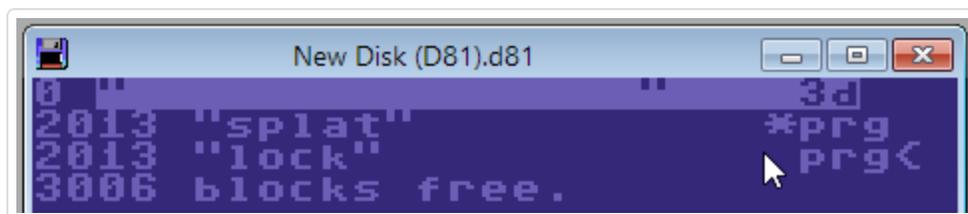
- by using keyboard shortcuts that correspond with the sub-menu items in the main menu and context menu:

d	DEL; changes the file type of the selected file entry to DEL
s	SEQ; changes the file type of the selected file entry to SEQ
p	PRG; changes the file type of the selected file entry to PRG
u	USR; changes the file type of the selected file entry to USR
l	REL; changes the file type of the selected file entry to REL

Changing a file type to REL will not actually create any of the file information and structure that a relative file normally has. Changing a file type to CBM will not actually create a sub-directory or any of the disk structure that a sub-directory normally has.

Toggle Locks & Splats

Commodore DOS recognizes a couple flag modifiers for file types, the "lock" and so-called "splat" flags. The lock flag is meant to prevent the DOS from deleting that file and is indicated in a directory listing with a "<" character that appears after the three letter file type. The splat flag indicates a file that was not closed properly and is indicated in a directory listing with a "*" character that appears before the three letter file type. With DirMaster you can set and clear these flags without resorting to use of the track/sector editor.



Toggle the lock flag by any of the following:

- selecting the file entry and use the main menu ()
- right click a file entry and use the context menu item

- select the file and press the keyboard shortcut, the < key

Toggle the splat flag by any of the following:

- selecting the file entry and use the main menu ()
- right click a file entry and use the context menu item
- select the file and press the keyboard shortcut, the * key

Moving File Entries

DirMaster offers convenient drag and drop control that treats each file entry as a moveable entity. Using the mouse, you can click/hold over a file name, then move the mouse up and down to choose a position to move the file entry to. Release the mouse button to drop the file entry in place. Selecting multiple file entries and moving them together will work as well. Aside from drag and drop control with the mouse, you can also:

- use the main menu (and)
- use the toolbar buttons to move the selected file up and down:
- by using keyboard shortcuts; alt + cursor-up and alt + cursor-down will also move the selected file up and down



Inserting File Entries

To insert a new file entry on a disk image, you can either:

- use the main menu ()
- right click a file entry and use the context menu item

A new file entry will be inserted immediately above the selected file entry. The new file entry starts with an empty file name and the in-place editor field will appear, allowing you to enter a file name (see "[Renaming Files](#)" above). Note: this operation does not write any additional data to the disk image and the new file entry will point to track/sector 0/0.

Deleting File Entries

You can delete file entries from the disk image directory:

- use the main menu ()
- right click a file entry and use the context menu item
- select the file and press the keyboard shortcut, the Delete key

If you have selected multiple file entries, each will be deleted. This operation simulates the Commodore DOS 'scratch' command: the file entry is marked as deleted and therefore hidden from normal directory listings but the file entry remains in the directory track. The file contents also remain intact wherever they were across the sectors that comprised the file

chain. You could use the Compact operation (see "[Compacting the Directory](#)") followed by the Fill Free Sectors operation (see "[Filling Free Sectors](#)") in order to completely remove the file entry from the directory and the file contents from the disk image sectors.

Renaming Disks

Most disk images have a disk name that is displayed at the top of a directory listing. DirMaster provides a way to edit the disk name:

- by using the main menu ()
- by right clicking over the disk name and using the context menu item
- by a slow double click over the disk name: click once, pause, then click again to edit the disk name

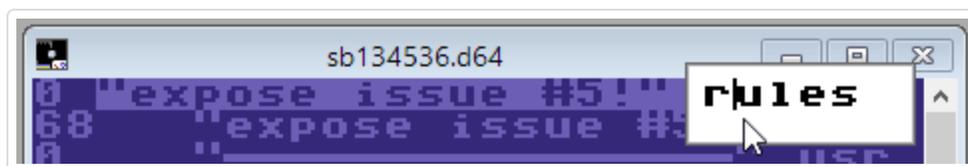
Each of those triggers will cause the in-place editor field to appear, allowing you to modify the disk name:



Hitting return or clicking outside of the in-place editor field will commit the change.

Changing Disk ID

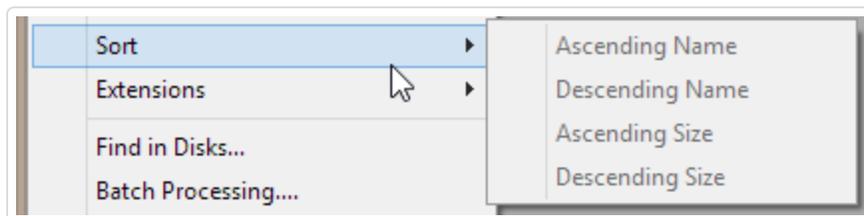
Most disk images have a disk ID that is displayed at the top of a directory listing. DirMaster provides a way to edit the disk ID by using the main menu (); that will cause the in-place editor field to appear, allowing you to modify the disk ID:



Hitting return or clicking outside of the in-place editor field will commit the change.

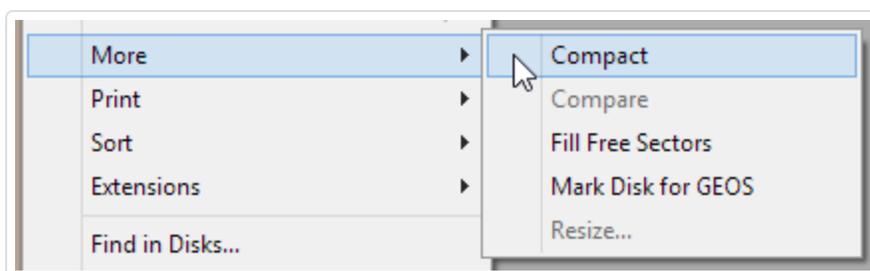
Sorting the Directory

The directory can be sorted by file name or by file size. Sorts can be either ascending or descending. Note that the "visual" file size is used, that is, the file size reported by the file entry - that can sometimes be different than the actual size of the file on disk! Access the sorting operations via the main menu: ()



Compacting the Directory

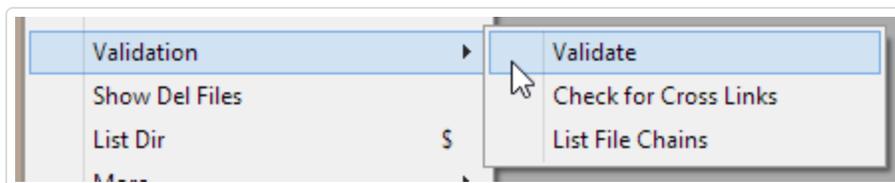
Typically when a file is deleted from a disk, the real Commodore DOS marks the file entry as deleted. However, the file entry itself including the file name continues to occupy space in the directory on the disk image. DirMaster lets you toggle the visibility of these deleted files (see "[Toggling DEL File Display](#)") and provides a feature to compact the directory by rewriting it with only file entries for non deleted files. This permanently removes the deleted file entry from the disk directory. Access the compact operation via the main menu: ()



Full Disk Operations

Validating a Disk

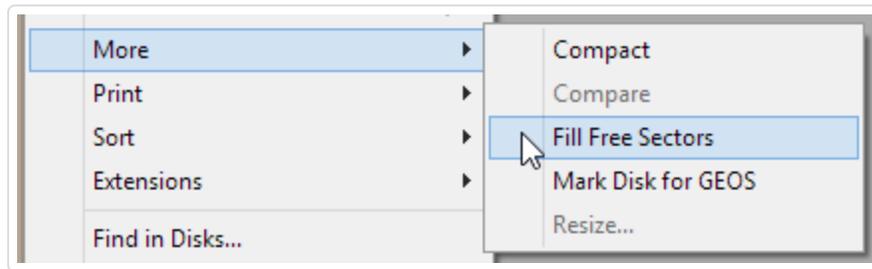
DirMaster can simulate the effects of a validate command on a real disk drive, such as `open 1,8,15,"v0":close 1`. Validation rebuilds the disk's block availability map (BAM) based on the actual used sectors discovered while tracing every non deleted file entry in the directory. Like the real Commodore DOS command, DirMaster's validate operation does not change the directory itself - the file entries in the directory and their block sizes are not altered (even if the block size shown is incorrect). Access the validate operation via the main menu: ()



Filling Free Sectors

Disks used repeatedly with files added and removed will often contain "leftover" data on unused sectors - the remnants of deleted files that have not yet been overwritten by a different file. In the past, this could be especially troublesome when archiving a full disk using the popular "Zip4" utility, which compresses a full 1541-formatted disk side into four files for transport via modem. Of course that's not the only reason to want a "clean" disk; even modern .d64 images are often compressed for storage and/or during transmission via

http/ftp. DirMaster includes a feature for quickly filling the unused sectors (as determined by the disk's BAM) with a fill byte. This fill byte can be set in the Options dialog (see "[Options Dialog](#)"). Access the fill free sectors operation via the main menu: ()



Separators

In days past and even today it was/is popular to decorate directories with file entries that have no other purpose than dividing, grouping, and boxing sets of real file entries, or perhaps conveying some other written information via the directory. These file entries were usually referred to as 'separators' and often used Commodore graphics characters. DirMaster's Separators dialog enables collecting and using your own set of separators. When you install DirMaster for the first time, it will start with a set of common separators.

Invoke the Separators dialog via the main menu () or by clicking the Show Separator Tool button on the toolbar; the toolbar button acts as a toggle, hiding and showing the Separators dialog.

The screen shot below shows the Separators dialog with its right click context menu active:



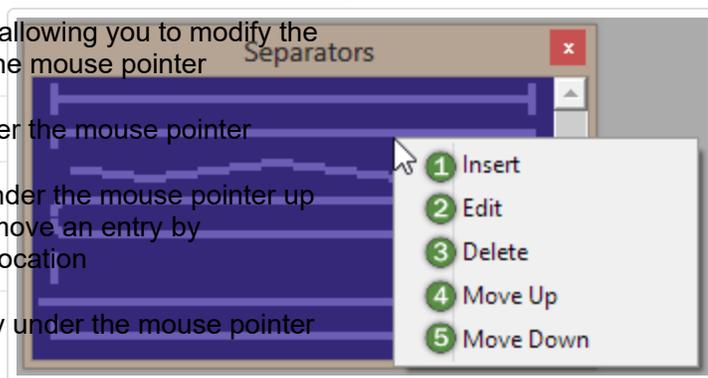
1 **Insert**; inserts a new file entry at the mouse pointer

2 **Edit**; triggers the in-place editor, allowing you to modify the file name of the file entry under the mouse pointer

3 **Delete**; deletes the file entry under the mouse pointer

5 **Move Up**; moves the file entry under the mouse pointer up one slot; note that you can also move an entry by click/dragging it into the desired location

6 **Move Down**; moves the file entry under the mouse pointer down one slot



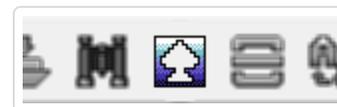
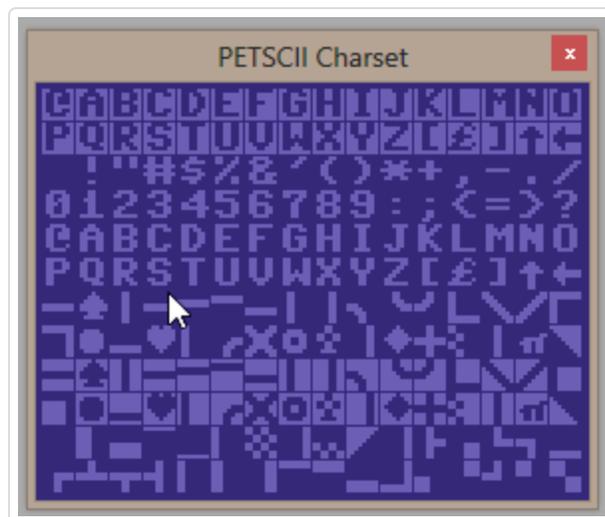
While the Separators dialog is open, you can drag/drop file entries from it onto disk image sub-windows to place copies of the separators in the target directory. Single clicking an entry in the Separators dialog will insert it immediately above the currently selected file entry in the active disk image sub-window. You can also drag files from disks onto the Separators

dialog which will create a separator with the same file name as the file entry you dropped. Your collection of separators is saved when quitting DirMaster and will be available the next time you run it.

PETSCII Charset

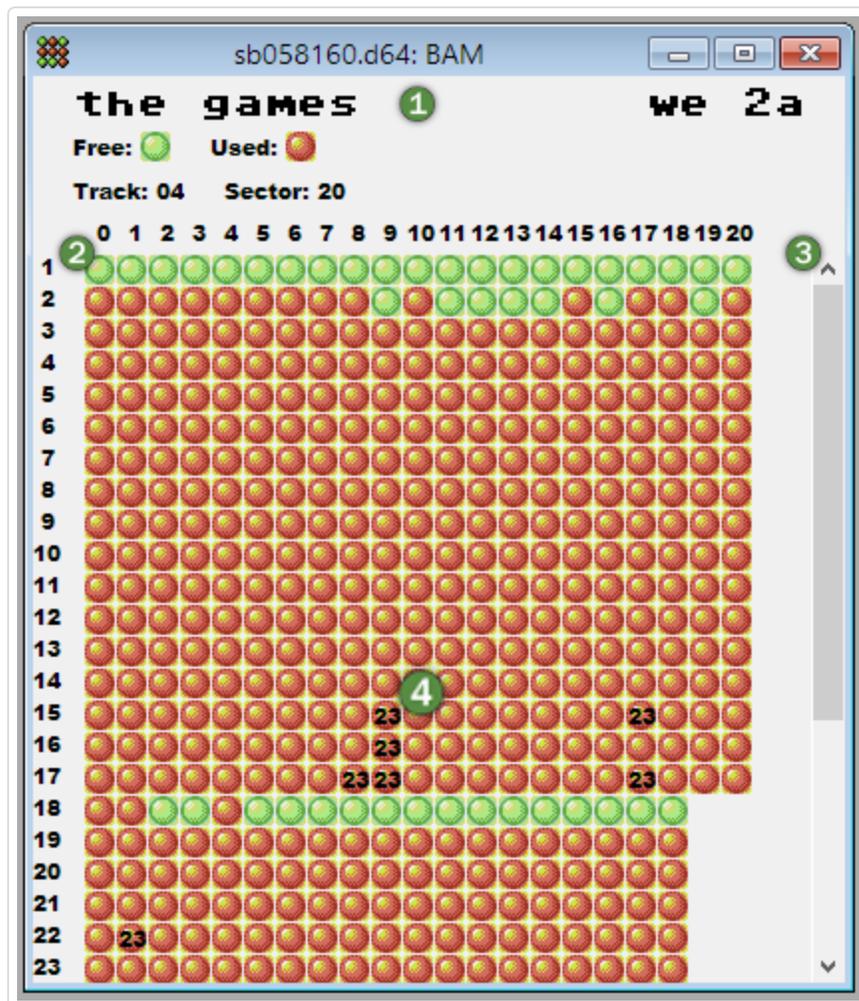
Inputting the full range of PETSCII character codes can be difficult with a PC compatible keyboard. To assist in editing file names, disk names/ids, sector bytes, and so on, DirMaster provides the PETSCII Charset dialog, which you can use to insert any PETSCII character into editable fields. The dialog displays a 16x16 grid of PETSCII characters, and should at all times sync up to the character mode (upper/graphics or lower/upper) of any currently active editing field or character area:

Simply click the character you wish to insert, and it will be placed at the current cursor position where you are editing. Invoke the PETSCII Charset dialog via the main menu () or by clicking the Show PETSCII Tool button on the toolbar.



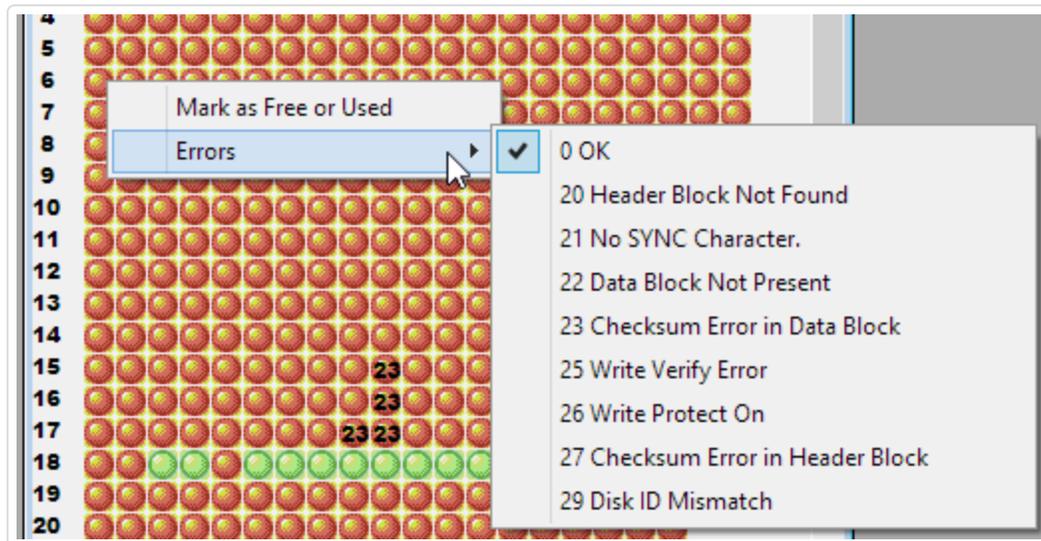
BAM Editor

DirMaster includes a two part capability for examining and editing disk images at a lower level than the directory. This first is the Block Availability Map Editor, or BAM Editor for short. This presents a grid representaton of the track/sector layout of the disk image as shown below:



- 1 disk name/id; the top of the BAM Editor will always display the disk name/id of the disk image that the editor is attached to
- 2 track/sector grid; each grid spot represents one disk block. Green indicates a block that is marked as *free* in the BAM, which would indicate to the Commodore DOS that new file data could be written there; a free block does not necessarily guarantee that there is no data there!
- 3 scrollbar; vertical (as shown) and horizontal (not shown) scrollbars at the right and bottom will let you view the entire BAM
- 4 error codes; some disk images will include error codes for blocks - these usually indicate that the physical media that was imaged had corresponding errors

Left clicking on any grid spot will display the Sector Editor dialog (see below), open to the track/sector that was clicked. If you are working with a disk image that *does not* use the sector error extension, then right clicking on any grid spot will toggle the used flag for that track/sector in the BAM. If the disk image does include the sector error extension then there is a right click context menu that offers additional control over the error codes:

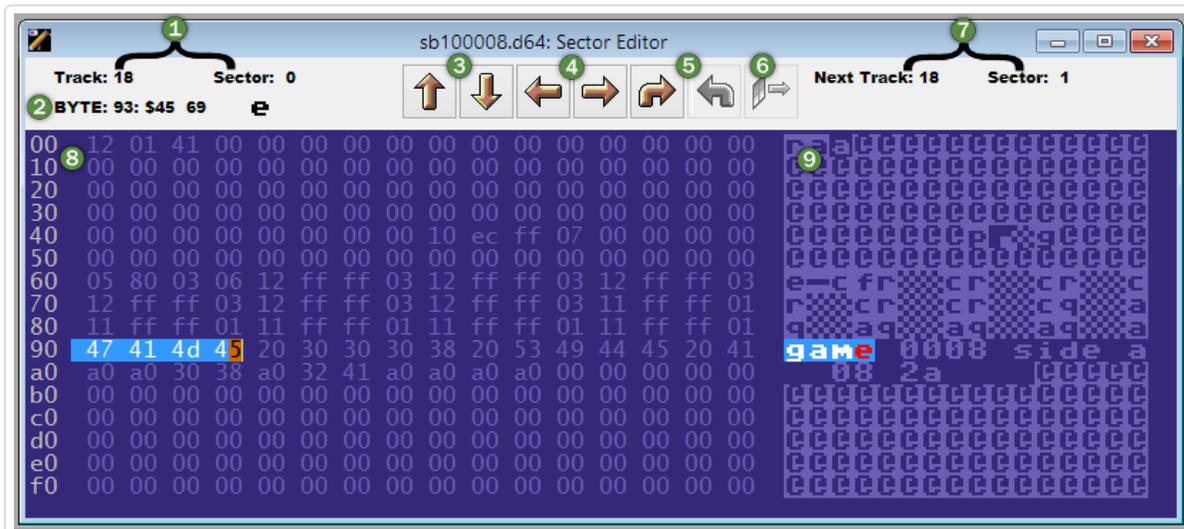


toggles the used flag for that track/sector in the BAM. Clicking any of the sub-menu items under will cause that track/sector it be marked with the error of the selected menu item. Clicking will clear any error code for that track/sector. Note that any changes made to the BAM by toggling used flags or setting/clearing error codes are not saved until you explicitly save the disk image.

You can change the orientation of the track/sector grid in the BAM Editor. By default, tracks are displayed on the vertical axis and sectors on the horizontal axis. A configuration option, **Display Sectors Horizontally in the BAM** is available on the Options dialog accessible through the main menu (). For more details on these and other configuration options, see "[Options Dialog](#)".

Sector Editor

DirMaster includes a lower level disk image editing tool - the Sector Editor dialog. This tool provides most of the same functionality commonly found in native tools that went under various name like "track/sector editors", "disk doctors", and so on. The Sector Editor dialog can be invoked in a few different ways. When using the BAM Editor described above, left clicking any sector grid spot will open the Sector Editor at the corresponding track/sector. This tool can also be opened from a disk image directory by using the J shortcut key ("Jump to Sector") as well as from the main menu (). Once invoked, the dialog will look something like the following screen shot (markers have been added and are explained in detail below):



- 1 The **Track/Sector** display; displays the decimal track and sector numbers of the currently displayed sector.
- 2 The **Byte** display; will reflect information about the currently selected byte in the hexadecimal or PETSCII column. This information is (in order left to right): the hexadecimal byte offset into the sector; the hexadecimal value of the byte; the decimal value of the byte; and the PETSCII representation of the byte.
- 3 The **Previous Track / Next Track** buttons; clicking these will update what sector is currently displayed by decrementing or incrementing the track number respectively. Keyboard equivalents are control + cursor-up and control + cursor-down
- 4 The **Previous Sector / Next Sector** buttons; clicking these will update what sector is currently displayed by decrementing or incrementing the sector number respectively. Keyboard equivalents are control + cursor-left and control + cursor-right
- 5 The **Jump to Next Sector in File Chain / Jump to Previous Sector** buttons; the former button will determine the next logical sector in a file chain (generally pointed to with the first tow bytes in the current sector) and if valid will update the currently displayed sector to that. The keyboard equivalent is J. While navigating tracks/sectors, DirMaster will keep track of previously viewed sectors - the latter button will change the displayed sector to whatever the previously viewed sector was. The keyboard equivalent is shift + J.
- 6 The **Jump to Side Sector** button; when the currently selected byte and the next byte after it comprise a valid track/sector pair (i.e. the values of the bytes form a valid track/sector), this button will be active and clicking it will update the currently displayed sector to where the two bytes point.
- 7 The **Next Track/Sector** display; displays the decimal track and sector numbers of the next logical sector in a file chain, or nothing if there is no valid next sector being pointed to by the current sector.
- 8 The hexadecimal column; the raw hexadecimal bytes that are contained in this sector; a gutter column at the far left displays the byte offset of the first byte in each row. Note that the entire Sector Editor dialog can be resized and therefore the make up of rows and columns displayed here may vary, but at all times should display exactly 256 bytes.

- 9 The PETSCII column; a PETSCII representations of the hexadecimal bytes in the same row - note that the context menu when right clicking in the PETSCII column is different than when right clicking in the hexadecimal column, see below.

Within each of the hexadecimal and PETSCII columns, additional features can be accessed using the right click context menu. Shown below is the context menu for the hexadecimal column:

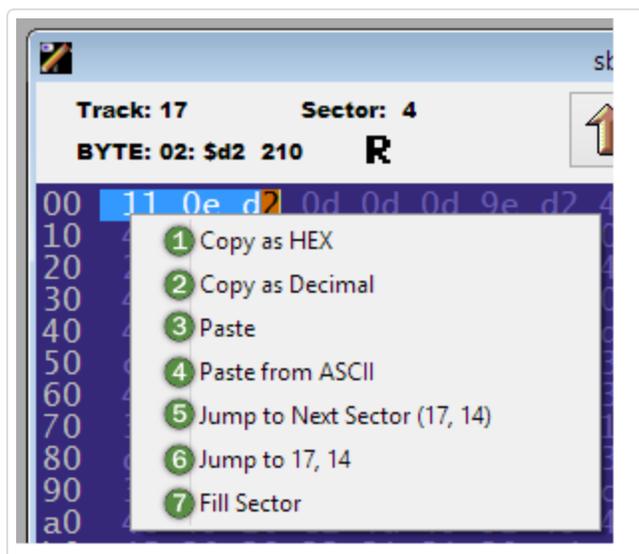
- 1 **Copy as Hex**; puts an ASCII representation of the selected bytes into the clipboard; the bytes are copied as comma separated hexadecimal values; e.g. "\$11, \$0e, \$d2" - the use case this and the following feature addresses is copying sector byte data into another system that expects text input, like a code editor or spreadsheet.

- 2 **Copy as Decimal**; puts an ASCII representation of the selected bytes into the clipboard; the bytes are copied as comma separated decimal values, e.g. "17, 14, 210"

- 3 **Paste**; assumes that the clipboard currently contains PETSCII text, and then pastes the text characters as their corresponding byte value into the sector at the position of the mouse pointer.

- 4 **Paste from ASCII**; assumes that the clipboard currently contains ASCII text, converts each text character into PETSCII, and then pastes the text characters as their corresponding byte value into the sector at the position of the mouse pointer.

- 5 **Jump to Next Sector**; the same as "Jump to Next Sector in File Chain" above - if there is valid the next logical sector, updates the currently displayed sector to that sector.



6 **Jump to tt, ss**; when the context menu is activated and the two bytes under the mouse pointer are a valid track/sector pair, then clicking this sub-menu item will update the currently displayed sector to that track/sector.

7 **Fill Sector**; immediately fills the sector with the currently configured fill byte. This fill byte can be set in the Options dialog (see "[Options Dialog](#)").

The PETSCII column context menu is shown in this screen shot:

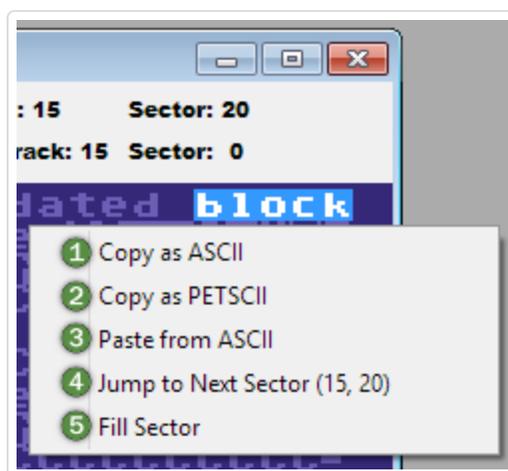
1 **Copy as ASCII**; puts an ASCII representation of the selected PETSCII bytes (converted to ASCII) into the clipboard.

2 **Copy as PETSCII**; puts a PETSCII representation of the selected PETSCII bytes into the clipboard.

3 **Paste from ASCII**; assumes that the clipboard currently contains ASCII text, converts each text character into PETSCII, and then pastes the PETSCII characters into the sector at the position of the mouse pointer.

4 **Jump to Next Sector (tt, ss)**; the same as "Jump to Next Sector in File Chain" above - if there is valid the next logical sector, updates the currently displayed sector to that sector.

5 **Fill Sector**; immediately fills the sector with the currently configured fill byte. This fill byte can be set in the Options dialog (see "[Options Dialog](#)").



If you want to copy and paste sections of byte data within the sector editor, the way to do that currently is to select the bytes you want to copy within the *PETSCII column*, then use the context menu item to copy the bytes. Then you can paste either into the hexadecimal or PETSCII columns using the context menu item. The next effect of this is that you will copy/paste the actual byte values instead of textual representations of the bytes. We may revisit this in the future based on user feedback.

The keyboard can also be used to navigate through sector data; cursor keys will move the byte cursor around the display either in the hexadecimal or PETSCII columns and moving the cursor past the top or bottom of the viewer will scroll the hex dump one line at a time;

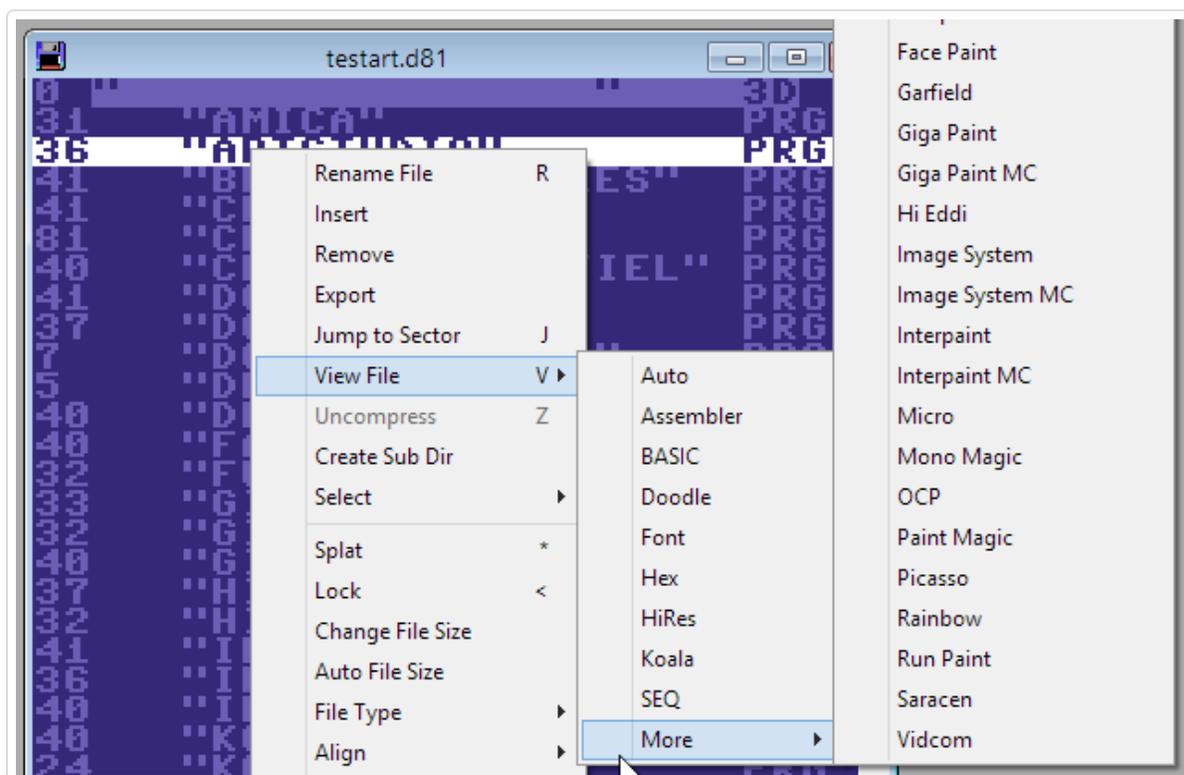
home and end will position the byte cursor at the start and end of the current row respectively.

This section will describe how you can use DirMaster's file viewer features to explore your disk images and their contents in greater detail.

Invoking a Viewer

The file viewer can be invoked by right clicking on a file entry in an open disk image; the context menu item, `V`, will open a sub-menu with several options as shown below; the menu item will open up yet another sub-menu of options specific to graphic/picture viewers.

Selecting one of these sub-menu items will force DirMaster to view the file as though it were of the selected type - *even if it really isn't*.



DirMaster can also auto detect the supported file types shown in the above sub-menus. You can invoke the file viewer with file type detection in one of three ways:

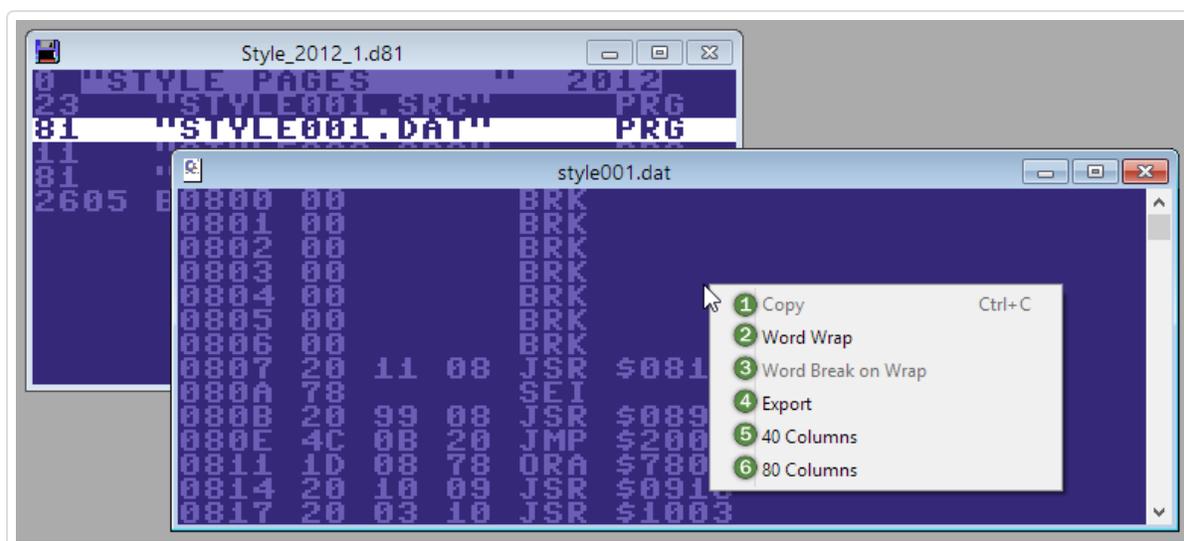
- by a fast double click on the file entry
- by pressing the keyboard shortcut, `v`, when a file entry is selected
- by opening the sub-menu as shown above and selecting the menu item

A file viewer sub-window will be displayed with the file contents presented appropriately for the selected or detected type. For example, right clicking a runnable program file and then invoking will display a picture viewer child window with the data from the program file interpreted as a hires bitmap.

If the file entry's type cannot be determined, then the default behavior is to treat the file as a generic binary file with no type. In that case, a disassembly viewer child window will be displayed with the file's content shown as a disassembly.

Disassembly Viewer

The default viewer presented when no supported file format with a more specific viewer can be detected or when choosing from the sub-menu is the disassembly viewer. This viewer presents the contents of the file as a machine code disassembly listing, with file position to the far left, hex code in the middle, and assembler mnemonics to the right. The file position is computed as though the first two bytes of the file are the load address; the first two bytes are therefore not part of the actual disassembly but can be realized from the first address shown in the viewer.



- 1 **Copy**; puts an ASCII representation of the selected text into the clipboard from where it can be pasted into another application
- 2 **Word Wrap**; toggle word wrapping at the viewer window's right edge
- 3 **Word Break on Wrap**; toggles between wrapping with or without respect to breaking words

- 4 **Export**; invokes a standard save dialog that lets you save the disassembly in one of the following formats:

PETSCII (.seq)	A Commodore sequential file.
HTML (.html)	An HTML file; utilizes a Unicode representation of the PETSCII character set and references the "C64 Pro Mono" font installed by DirMaster to display accurate PETSCII glyphs.
ASCII (.txt)	A text file; PETSCII is translated into a best match ASCII representation
Bitmap (.bmp)	A Windows BMP file; a picture of the full disassembly (including content beyond the bottom and right edges of the viewer window that may be hidden).
PNG (.png)	A PNG file; a picture of the full disassembly (including content beyond the bottom and right edges of the viewer window that may be hidden).
PC64 (.s00)	A .s00 file which is a file image of a sequential file.
RTF (.rtf)	A Windows rich text file; utilizes a Unicode representation of the PETSCII character set and references the "C64 Pro Mono" font installed by DirMaster to display accurate PETSCII glyphs. Uses a black/white foreground/background.

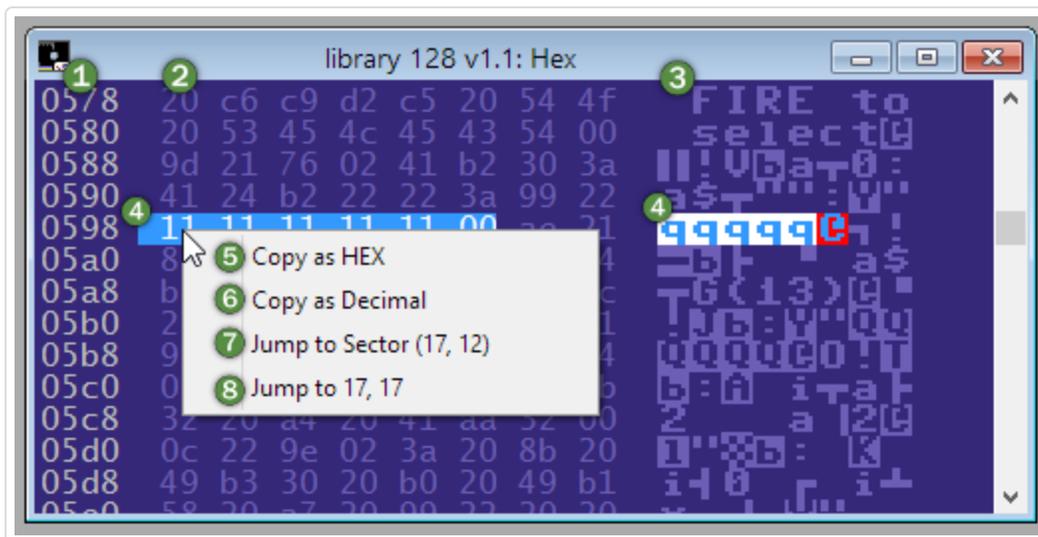
- 5 **40 Columns**; resets the viewer window size to 40 columns wide

- 6 **80 Columns**; resets the viewer window size to 80 columns wide

The keyboard can also be used to navigate through the disassembly listing; cursor-up and cursor-down will scroll up and down one line at a time; page up and page down will scroll up and down one full display page at a time; home and end will position the viewer at the top and bottom of the disassembly respectively.

Hex Dump Viewer

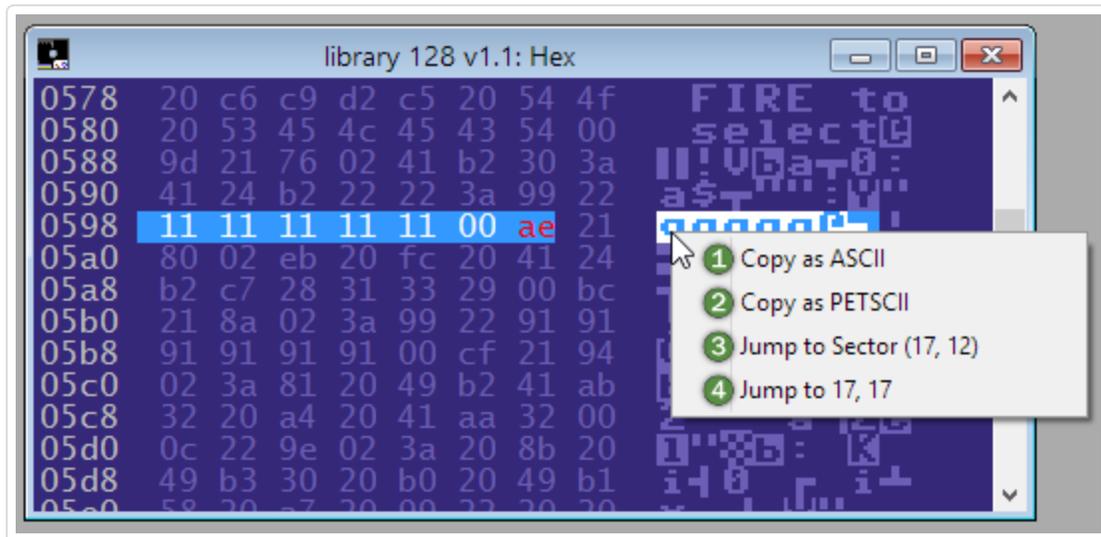
The default viewer presented when choosing from the sub-menu is a hex dump viewer. This viewer presents the contents of the file as hexadecimal byte values to the left with their corresponding PETSCII character values to the right. The file position is computed from \$0000 and the first two bytes of the file are included in the hex dump.



- 1 The file position column; at what position in the file the bytes in that row are displayed from.
- 2 The hexadecimal column; the raw hexadecimal bytes from the file contents.
- 3 The PETSCII column; a PETSCII representations of the hexadecimal bytes in the same row - note that the context menu when right clicking in the PETSCII column is different than when right clicking in the hexadecimal column, see below.
- 4 Selected bytes; you can use the mouse to click/drag and select one more more bytes; You can select from within the hexadecimal or PETSCII columns and the corresponding bytes are highlighted in sync in both panels. A red background is used to mark the current cursor position.
- 5 **Copy as Hex**; puts an ASCII representation of the selected bytes into the clipboard; the bytes are copied as hexadecimal values; e.g. "\$11, \$11, \$11, \$11, \$11, \$00"
- 6 **Copy as Decimal**; puts an ASCII representation of the selected bytes into the clipboard; the bytes are copied as decimal values, e.g. "17, 17, 17, 17, 17, 0"
- 7 **Jump to Sector (tt, ss)**; the current track/sector on which the the byte under the mouse pointer resides is reflected in the label of this sub-menu item; clicking it will invoke the sector editor opened to that track and sector on the disk image.
- 8 **Jump to tt, ss**; when the context menu is activated and the two bytes under the mouse pointer are a valid track/sector pair, then clicking this sub-menu item will invoke the track/sector editor opened to that track and sector on the disk image.

The keyboard can also be used to navigate through the hex dump; cursor keys will move the byte cursor around the display either in the hexadecimal or PETSCII columns and moving the cursor past the top or bottom of the viewer will scroll the hex dump one line at a time; page up and page down will scroll up and down one full display page at a time; home and end will position the byte cursor at the start and end of the current row respectively. The keyboard shortcuts j and g invoke the **Jump to Sector (tt, ss)** and **Jump to tt, ss** operations, respectively.

The context menu when clicking in the PETSCII column is shown below:



- 1 **Copy as ASCII**; puts an ASCII representation of the selected bytes into the clipboard
- 2 **Copy as PETSCII**; puts a PETSCII representation of the selected bytes into the clipboard; this is also the default behaviour when using the standard copy keyboard shortcut, control + c, whether the cursor/mouse is active on the hexadecimal or PETSCII columns
- 3 **Jump to Sector (tt, ss)**; works the same as the operation described for the hexadecimal column above
- 4 **Jump to tt, ss**; works the same as the operation described for the hexadecimal column above

SEQ Viewer

The default viewer presented when choosing from the sub-menu is a sequential file viewer. This viewer presents the contents of the file as PETSCII character text. *Most* PETSCII control codes like color changes and reverse on/off are respected, but cursor movements, home, and other cursor positional codes may not translate into identical output as when viewed on a real Commodore.



- 1 **Copy**; puts an ASCII representation of the selected text into the clipboard from where it can be pasted into another application

- 2 **Word Wrap**; toggle word wrapping at the viewer window's right edge

- 3 **Word Break on Wrap**; toggles between wrapping with or without respect to breaking words

- 4 **Export**; invokes a standard save dialog that lets you save the sequential file in one of the following formats.

PETSCII (.seq)	A Commodore sequential file.
HTML (.html)	An HTML file; utilizes a Unicode representation of the PETSCII character set and references the "C64 Pro Mono" font installed by DirMaster to display accurate PETSCII glyphs.
ASCII (.txt)	A text file; PETSCII is translated into a best match ASCII representation
Bitmap (.bmp)	A Windows BMP file; a picture of the entire sequential file content
PNG (.png)	A PNG file; a picture of the entire sequential file content
PC64 (.s00)	A .s00 file which is a file image of a sequential file.
RTF (.rtf)	A Windows rich text file; utilizes a Unicode representation of the PETSCII character set and references the "C64 Pro Mono" font installed by DirMaster to display accurate PETSCII glyphs. Uses a black/white foreground/background.

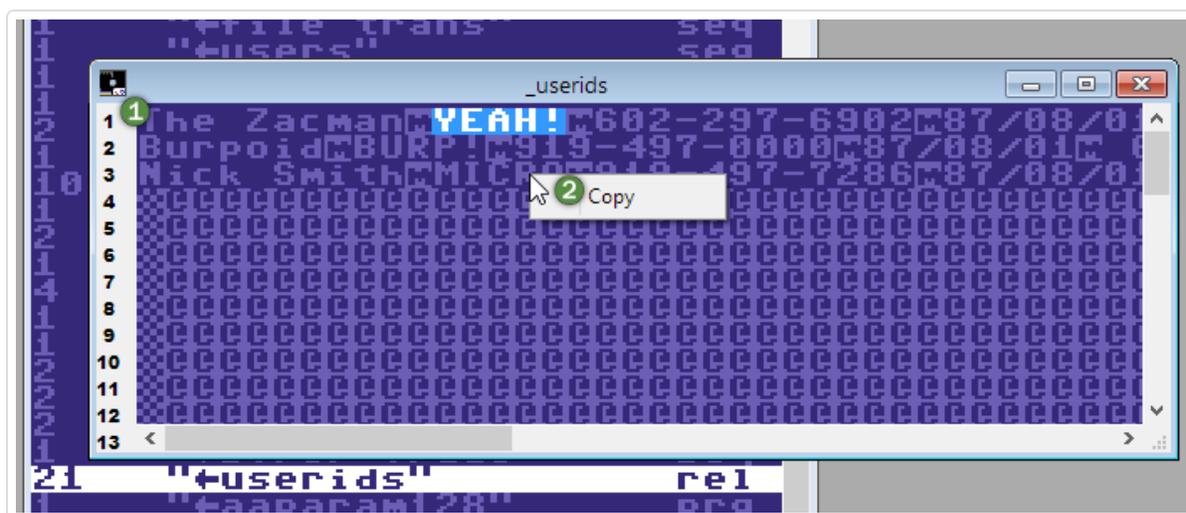
- 5 **40 Columns**; resets the viewer window size to 40 columns wide

6 **80 Columns**; resets the viewer window size to 80 columns wide

The keyboard can also be used to navigate through the sequential file; cursor-up and cursor-down will scroll up and down one line at a time; page up and page down will scroll up and down one full display page at a time; home and end will position the viewer at the top and bottom of the sequential file respectively.

REL Viewer

The default viewer presented when choosing to view any file entry of the **"REL"** file type is a REL record viewer. This viewer interprets the REL file structure as a set of records, which it displays one per line in the viewer window. The goal is to enable the user to make sense of the REL record and field contents, which would be more difficult when viewing the REL file bytes through a standard track/sector editor. The REL viewer is shown below with the right click context menu open:

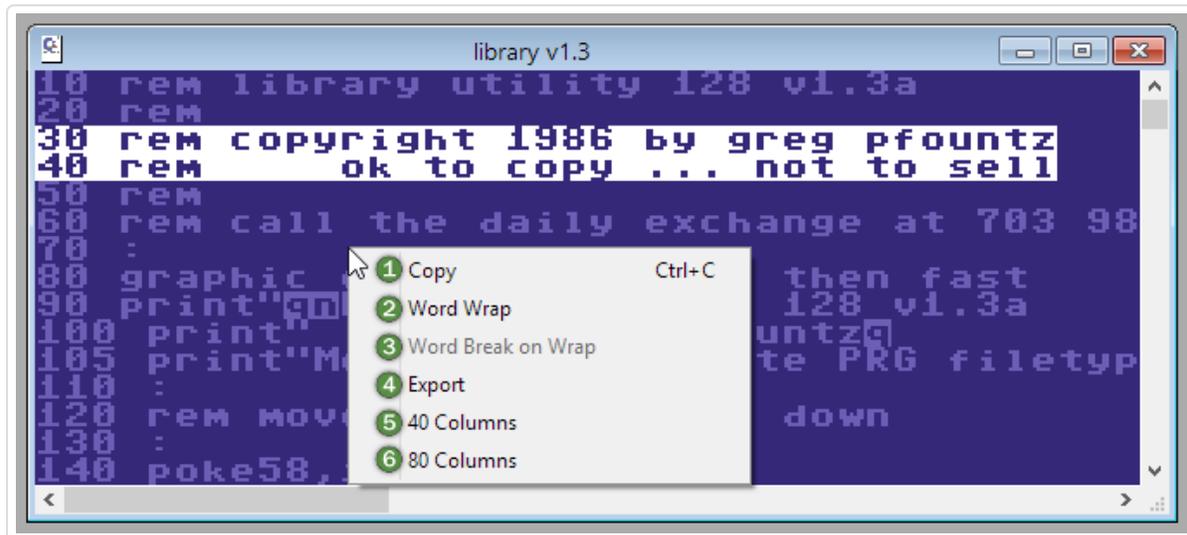


1 Record index column; the numbers at the left of the REL viewer sub-window denote the record index for that row

2 **Copy**; puts a PETSCII representation of the selected text into the clipboard from where it can be pasted into another application

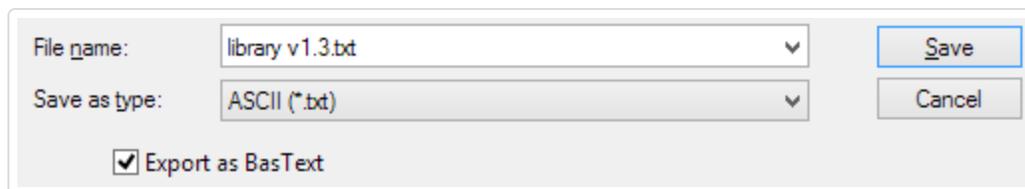
BASIC Viewer

The default viewer presented when choosing from the sub-menu is a BASIC file viewer. This viewer presents the contents of the file as a detokenized BASIC listing using PETSCII character text. Currently, standard BASIC 2.0 (c64) and BASIC 7.0 (c128) are supported.



- 1 **Copy**; puts an ASCII representation of the selected text into the clipboard from where it can be pasted into another application
- 2 **Word Wrap**; toggle word wrapping at the viewer window's right edge
- 3 **Word Break on Wrap**; toggles between wrapping with or without respect to breaking words

- 4 **Export**; invokes a standard save dialog and lets you save the BASIC listing in one of the following formats. The save dialog also has an extra **Export as BasText** checkbox which is checked by default; when checked, exported .txt listings will contain 'BasText' token replacements for PETSCII characters with no direct ASCII equivalent. When unchecked, a best guess ASCII character is output instead. The **Export as BasText** checkbox has no effect for other the export options.



PETSCII (.seq)	A Commodore sequential file.
HTML (.html)	An HTML file; utilizes a Unicode representation of the PETSCII character set and references the "C64 Pro Mono" font installed by DirMaster to display accurate PETSCII glyphs.
ASCII (.txt)	A text file; PETSCII is translated into a best match ASCII representation; uses 'BasText' tokens when Export as BasText is checked
Bitmap (.bmp)	A Windows BMP file; a picture of the BASIC listing content
PNG (.png)	A PNG file; a picture of the BASIC listing content
PC64 (.s00)	A .s00 file which is a file image of a sequential file.
RTF (.rtf)	A Windows rich text file; utilizes a Unicode representation of the PETSCII character set and references the "C64 Pro Mono" font installed by DirMaster to display accurate PETSCII glyphs. Uses a black/white foreground/background.

- 5 **40 Columns**; resets the viewer window size to 40 columns wide

- 6 **80 Columns**; resets the viewer window size to 80 columns wide

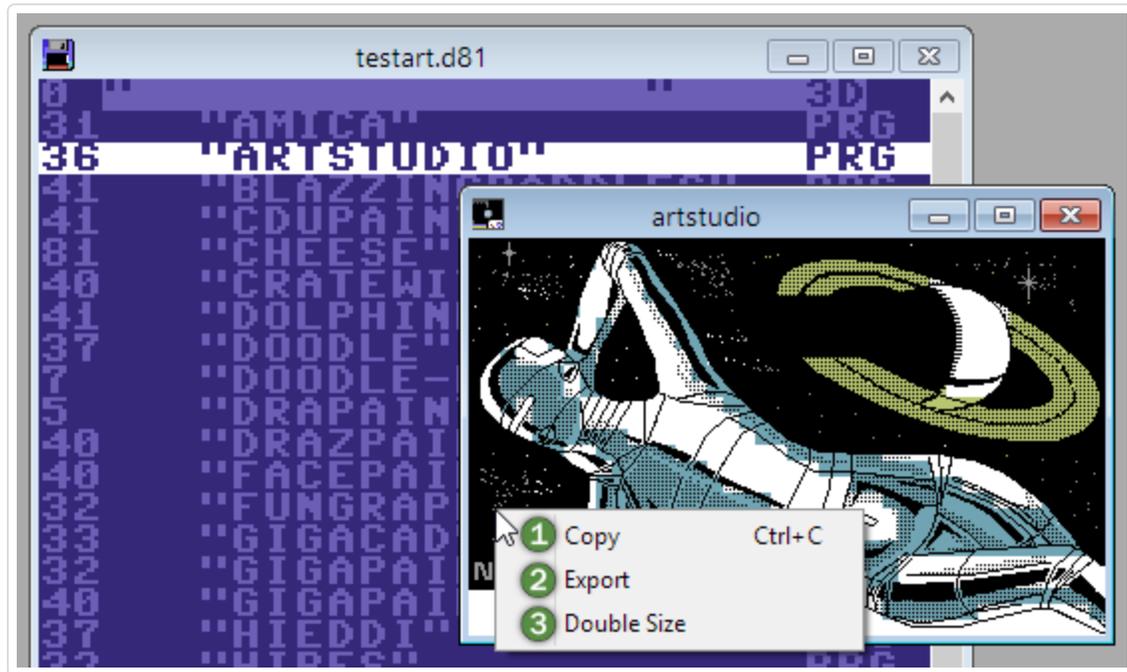
The keyboard can also be used to navigate through the BASIC listing; cursor-up and cursor-down will scroll up and down one line at a time; page up and page down will scroll up and down one full display page at a time; home and end will position the viewer at the top and bottom of the BASIC listing respectively.

Picture Viewer

DirMaster can recognize and display many picture (i.e. full screen/logo) related graphics formats like Koala, Amiga Paint, and so on (see "[File Type Support](#)" for a full list). To view an image file, you can:

- Use the v keyboard shortcut or select from the sub-menu and let DirMaster autodetect the image format
- Force DirMaster to display the file interpreted as certain format by choosing , , or a menu item from the sub-menu under the sub-menu

DirMaster will open a picture viewer window with the picture rendered using the current palette. The example below shows the picture viewer window with its right click context menu activated:



- 1 **Copy**; puts the picture into the clipboard from where it can be pasted into another application
- 2 **Export**; invokes a standard save dialog where you can export the picture as either a .bmp or .png file
- 3 **Double Size**; toggles between a standard size and a double size picture viewer sub-window

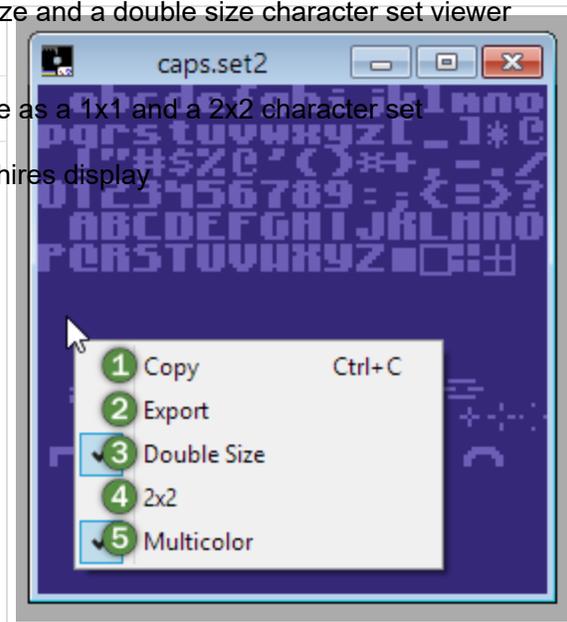
Character Set Viewer

DirMaster can also recognize and display some character sets. At this time only 1x1 and 2x2 character sets are directly supported. When a character set is detected or forced by choosing from the sub-menu;, DirMaster will open a character set viewer sub-window with a 16x16 character grid display as shown below with the right click context menu activated:

- 1 **Copy**; puts a picture of the character set into the clipboard from where it can be pasted into another application
- 2 **Export**; invokes a standard save dialog where you can export a picture of the character set as either a .bmp or .png file

- 3 **Double Size**; toggles between a standard size and a double size character set viewer sub-window
- 4 **2x2**; toggles between interpretation of the file as a 1x1 and a 2x2 character set
- 5 **Multicolor**; toggles between multicolor and hires display

Although other character set sizes are not directly supported, a goal of this feature is enabling users to make sense of a file and help identify its content as some kind of character set - other character set sizes displayed using this viewer will still provide a very good visual clue as to what they are even if the individual characters are not displayed "properly".



This section will explain how easy and fast it is to extract disk images and files from Commodore archive formats. For a list of the supported archive formats, see "[File Type Support](#)". Also note that DirMaster does not provide any ability to create or edit archive files; support is limited to viewing contents, searching for files, and extracting files.

Extracting from the File System

To open and uncompress a raw file that is a supported archive format, you can:

- drag and drop the OS file that is the archive over empty space inside the main window. You can drag and drop multiple OS files to open them all at once.
- double click or otherwise launch the OS file that is the archive, which will open the archive in DirMaster if an appropriate file association has been made (see "[File Associations](#)").
- invoke the Open Disk dialog through the main menu (), browse for an OS file that is an archive, and open it. You will need to change the **Files of type**: selector from the default to **Compressed Files** in order to see archive file types - see "[Opening Disk Images](#)" for more information.
- alternatively invoke the Open Disk dialog by clicking the Open toolbar button:



If the OS file you are opening is a recognized and valid archive format that DirMaster can identify and parse, the result will be a new disk image child window with the uncompressed files from the archive extracted onto the new disk image. The default behavior is to create a blank .d64 image and place the uncompressed files on it. When the uncompressed files

require more space than allowed by a .d64 (664 blocks, 144 files), then a .d81 image is created instead. In the event the files cannot fit on a .d81 (3160 blocks, 296 files), then a .dnp image sized as large as necessary is created as a final fallback.

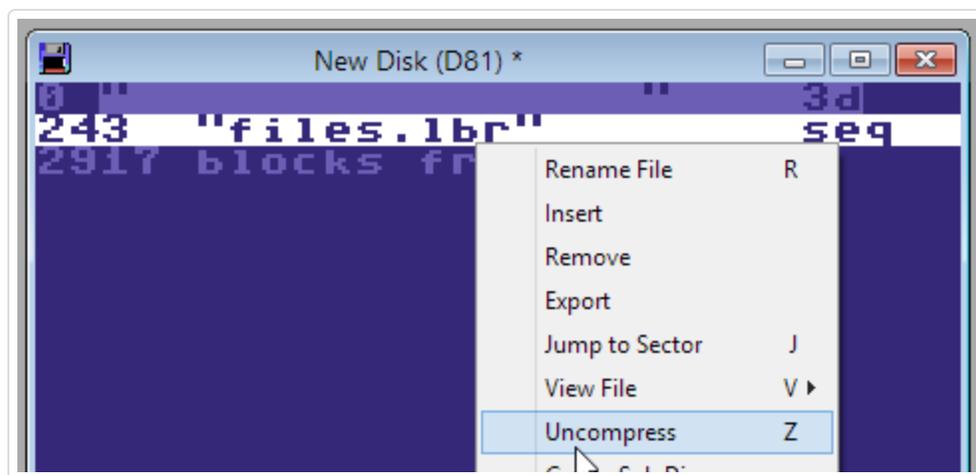
Be careful to note where you are drag/dropping the archive; if you drop it over an open disk image window instead of over empty space in the main window, you will copy the file onto that disk image instead of extracting it.

You can change the default disk image for uncompressed archive files from .d64 to .d81 through the Options dialog accessible through the main menu (). For more details, see "[Options Dialog](#)".

Extracting from a Disk Image

DirMaster can also extract the files from an archive that is on an open disk image. When you see a file entry that you suspect is an archive, you can either:

- right click the file entry and use the context menu item
- select the file and press the keyboard shortcut, z



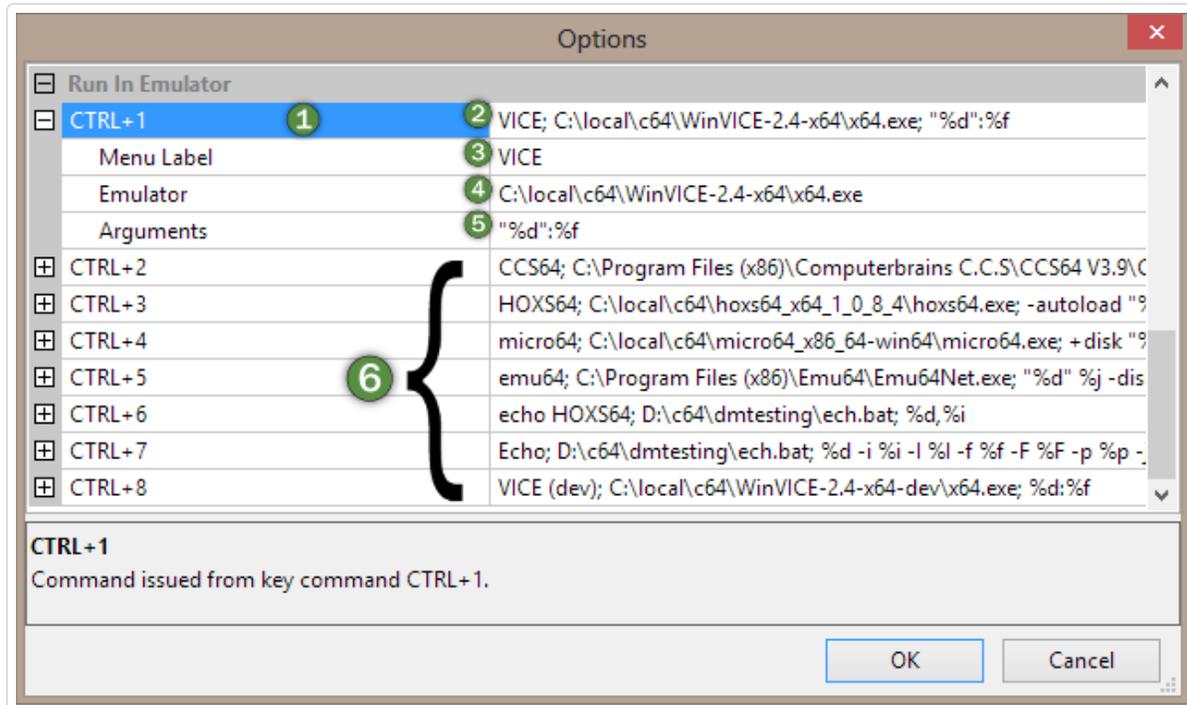
If DirMaster recognizes the file as a supported archive format, then it will extract directly from the disk image. For archives that span multiple files, you can trigger the extraction by selecting or right clicking any one of the files that are part of the archive; for example, any one of the four files of a zip4 archive. The extracted files are placed on a new disk image - either a .d64, .d81, or .dnp depending on the size of the extracted files as described above. The archive file(s) on the source disk image are not modified.

It is not necessary to save a disk image before uncompressing archives from it. Therefore it is possible to, for example, extract a zip4 archive which will open a new disk image with the uncompressed content, and then select another archive file from that new disk image, like a .lnx, and extract it to yet another new disk image.

This section will explain how to configure DirMaster to integrate with emulators (and other third party applications).

DirMaster provides for a quick way (via keyboard or mouse) to expose a currently selected file entry from an open disk image to an external application. The most common use case for this feature is running a program file ("PRG") with an emulator, but the integration capability applies to any external application that can be launched with a command line. Other use cases could include disassemblers, graphics editors, and so on.

There is room for eight (8) different integration slots. Configuring these slots is done via the **Run In Emulator** section of the Options dialog accessed via the main menu (). Observe the Options dialog below which shows the **Run In Emulator** section open with all eight slots used; the numbered markers are explained below.



- 1 The first slot; use the open/close toggles (they look like a boxed plus/minus) at the left to show or hide the details for each slot. You will not be able to edit the keyboard shortcut, but you can change the text in each of the fields numbered 2-5.
- 2 The slot configuration rule; this is a merged field of the three specific configuration fields immediately below it (numbered 3-5). You can edit the text here, and the changes made will be reflected in the other configuration fields. You can also edit the other configuration fields, and merged rule will update as well. Semicolons are used to delimit the specific values in the merged configuration rule.
- 3 The **Menu Label** field; the value entered here provides the visual/text cue elsewhere in the application (e.g. the context menu when right clicking a file entry).
- 4 The **Emulator** field; this must be a full path to an executable file that can be invoked with command line arguments. Again, it is not necessary that this application be a Commodore emulator. Any application or executable script (.bat, python script, etc) could work. You do not need to surround the path with quotation marks; DirMaster will add quotation marks as needed when invoking the executable.

5 The **Arguments** field; any command line arguments to pass to the executable. You can and should add quotation marks as appropriate. A number of "printf-style" format specifiers can be used to pass the OS file name of the disk image, the file name of the selected file entry, and so on. Note that at any time you can see a list of the allowed format specifiers within the application by clicking into the **Arguments** fields; the help panel at the bottom of the options dialog will display them. The allowed specifiers are:

-
- %d Replaced by the full path/filename of the OS file that the active disk image was opened from. Quotation marks *are not* assumed here and you should specify those in the **Arguments** field if necessary.

 - %i Replaced by the selected file entry's 0-based index in the disk image directory. Hidden file entries (deleted files) *are not* included in the index determination.

 - %l Replaced by the selected file entry's 0-based index in the disk image directory. Only file entries with the 'PRG' filetype are included in the index determination.

 - %j Replaced by the selected file entry's 0-based index in the disk image directory. Hidden file entries (deleted files) *are* included in the index determination.

 - %f Replaced by the selected file entry's filename, converted to ASCII where quote marks (") are replaced by the question mark (?).

 - %F Replaced by the selected file entry's filename, converted to ASCII where quote marks (") as well as all non ASCII characters are replaced by the question mark (?).

 - %p The selected file entry is first exported to an OS file in the operating system's temporary file directory; the format specifier is then replaced by the full path/filename of the exported temporary OS file. Quotation marks *are not* assumed here and you should specify those in the **Arguments** field if necessary.

6 The remaining slots; each of the other slots can be opened and configured the same as the first. Slot keyboard equivalents are control + [slot #], where [slot #] is 1 through 8.

Launch-or-View

When you have selected a file entry in an active disk image window, hitting return will act as a quick 'launch-or-view' function: if the selected file is runnable, DirMaster will proceed as though you had invoked the first emulator slot (same as hitting control + 1); otherwise DirMaster will proceed as though you had invoked the file viewer (same as hitting v) and display the file contents in the most appropriate viewer window (see "[Viewing Files](#)").

Emulator invocation in DirMaster v3 has been tested against several emulators and with different scenarios including when the path and/or file name of OS files contains spaces; before reporting issues with invoking an emulator, we suggest trying the following rules for the **Arguments** field when configuring a slot for each emulator:

VICE (v2.x)	<code>"%d":%F</code> There was no support in VICE for indexed file selection at the time we tested, and using %F should give the best success rate when launching files with unusual filenames
CCS64 (v3.9)	<code>"%d,%i"</code>
HOXS64 (v2.x)	<code>-autoload "%d" %I</code>
micro64 (v2013)	<code>+disk "%d"</code> There was no support in micro64 for directly selecting and running a file on a disk image at the time we tested; instead you can mount a disk at which point you must select the file to run though the micro64 interface.
Emu64 (v4.x)	<code>"%d" %j -diskmount</code>

DirMaster includes a search feature that can recurse through very large disk image collections and return a list of disk images containing file entry matches. This section will explain how to use DirMaster to search disk image collections.

Search Speed

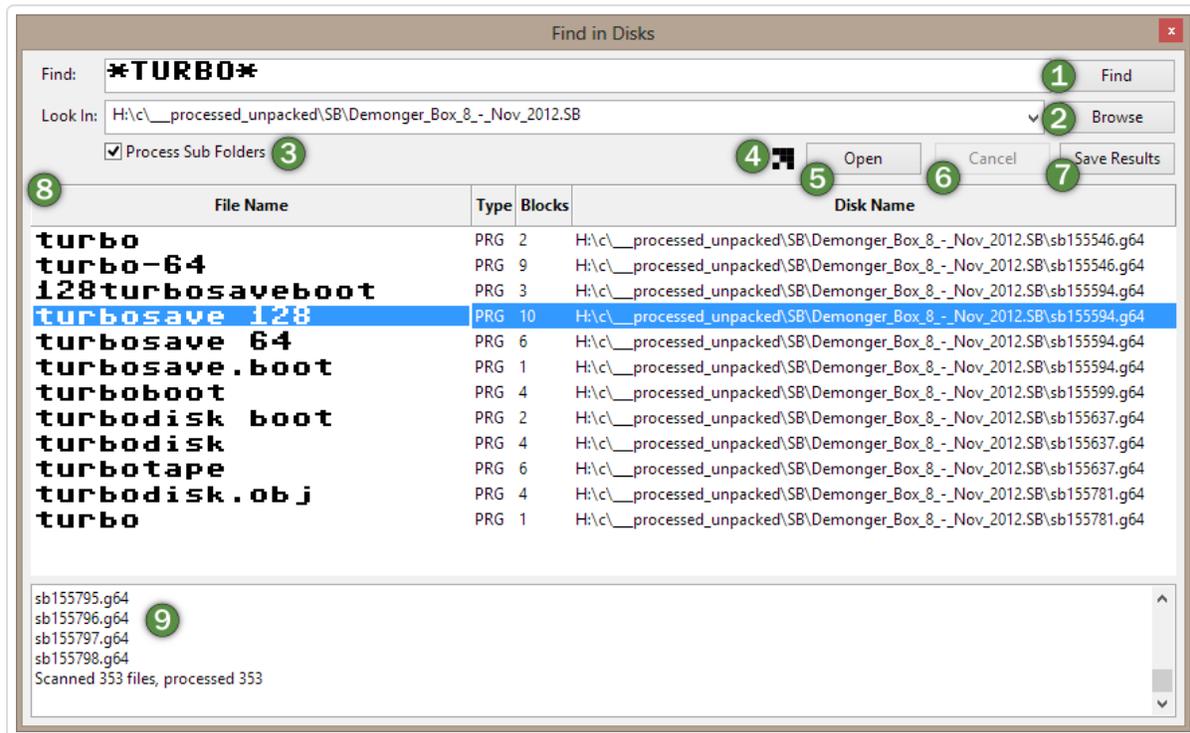
Search speed will of course vary, but as an example, DirMaster matched file entries at a rate of roughly 225 disk images per second from a USB 2.0 device and i7 processor and at a rate of roughly 1,000 disk images per second from a RAM disk and i7 processor.

Find in Disks Dialog

The Find in Disks dialog can be accessed through the main menu () or by clicking the Find in Disks toolbar button.

Observe the dialog; the numbered markers identify the main elements of the dialog interface and are explained below.





- 1 The **Find** field and button; enter your search text in the field and click Find to begin a search.
- 2 The **Look In** field and browse button; searches will begin in whatever OS directory you enter into the **Look In** field. Click the browse button to invoke a file selection dialog with which you can navigate to and select the OS directory you wish to start finding disk images in.
- 3 The **Process Sub Folders** checkbox; when checked, DirMaster will recurse all OS directories found in and below the OS directory given in the **Look In** field. When unchecked, the search is limited to only the contents of the OS directory given in the **Look In** field.
- 4 An **Activity** spinner; animates whenever a search is active.
- 5 The **Open** button; highlight a file/disk image result row in the results table (see below) then click Open to open that disk image in DirMaster.
- 6 The **Cancel** button; will terminate an active search.
- 7 The **Save Results** button; click Save Results to save the list of matched files/disk images to either a comma separated (.csv) or rich text (.rtf) file.
- 8 The **Results** table; all files found that matched your Find text are listed here, one row per matching file, along with the file type and size in native disk blocks. The last column shows the full path of the OS file that contained the matching disk image/file entry.
- 9 The **Log** text area; contains a running list of files that were searched, errors, and a final summary of searches vs. matches.

Advanced Searching

The **Find** field uses the PETSCII character set in upper/graphics mode. Keyboard input will insert the best match in PETSCII, but you can have full control over inserted characters by using the PETSCII Charset dialog (see "[PETSCII Charset](#)"). This makes it trivial to search for files containing unusual PETSCII characters (e.g. back arrow). In addition, the asterisk ("*") and the question mark ("?") act as wildcard characters: asterisk will match zero or more characters and question mark will match only one character.

Be aware of wildcard usage or non usage - in order to match file names that contain a given word anywhere in the name (for example, "style") you would need to surround the word with asterisks, like "*style*". If you searched for "*style" you would match only file names that end in "style". If you searched for "style" with no wildcards, then you would only match file names that exactly match "style" - five characters long.

Note that DirMaster's Find feature will match all file entries that it can; this means that it may match multiple files on a single disk image. Keep this in mind especially if you are saving match results to a file for further processing.

Supported File Types

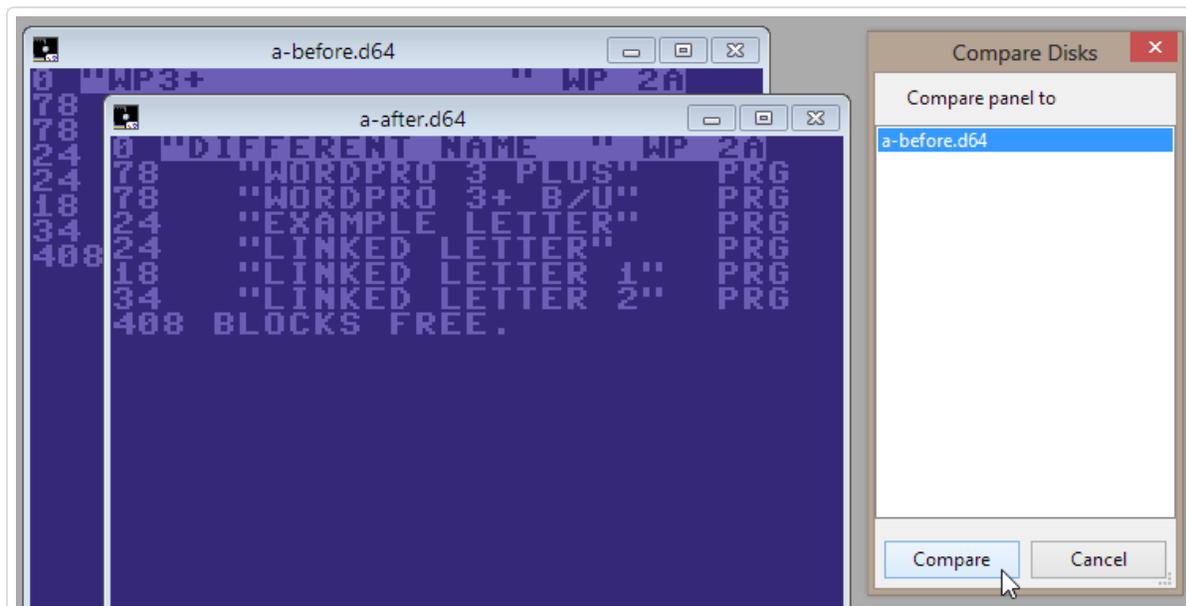
DirMaster can search file entries in all disk image and file image formats as well as raw files that are supported archiver formats (see "[File Type Support](#)"). To put it another way, any supported file type that contains file entries using native file names will be searched for matches.

DirMaster will search through OS files that are Commodore archiver formats and that are present in the search path (and down through recursive OS sub-directories if specified); but DirMaster will not search inside the Commodore archiver format files that are inside other disk images. For example, if the OS file "this.lnx" is in the search path and is a valid Lynx archive, DirMaster will match against file names inside the "this.lnx"; if the file "disk.d64" is in the search path and contains the file entry "that.lnx", then DirMaster could match against the "that.lnx" *file name*, but not against any of the file entries inside the "that.lnx" *file*.

This section will familiarize you with three key features useful for deeper exploration of disk image contents and assisting with potential data recovery.

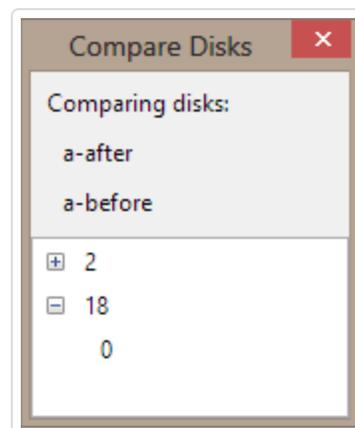
Comparing Disk Images

DirMaster can compare the contents of two disk images. This comparison is performed byte vs byte and only disk images of the same type can be compared. To begin, open the disk images you wish to compare and make one of them active (e.g. bring that disk image window to the front). Then invoke compare via the main menu (). This opens the Compare Disks dialog which will list all other open disk images of the same type as the current disk image:



Click Compare to start the comparison. The results window will provide a tree view that contains every track/sector within which different bytes were found; differing sectors are grouped by track. For example, the results below indicate that track 2 has one or more sectors that are different (you could open the tree row to view which sectors) and that track 18, sector 0 is also different.

You can single click the sector tree row (in the above example, the row with "0" visible under the "18") to open two Sector Editor dialogs - one for each disk image you just compared. Each Sector Editor will be open to the track/sector you selected from the comparison results. From there you can visually inspect the sector contents to see what the differences are.



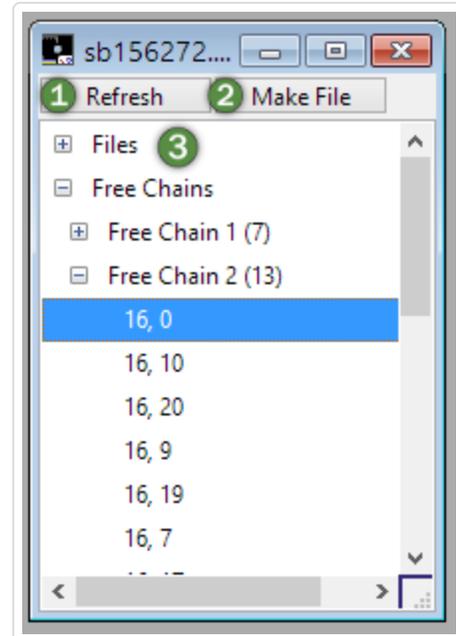
Hidden File Chains

If you've ever performed a fast new or some other directory-wiping operation that leaves the contents of the rest of the disk intact, the List File Chains operation accessed via the main menu () may be useful. When invoked, DirMaster builds a list of all file chains - that is, every list of sectors that appear linked to one another from a starting sector. A File Chains dialog is then presented with a tree view of these chains.

There are three types of chains that will be shown. A list of **Files** which will show you the chained sectors for every (non deleted) file entry, including one chain that represents the directory itself; a list of **Free Chains** which will display a list of chains detected on the disk that do not belong to any file entries; and a list of **Free Single Blocks**, which are single

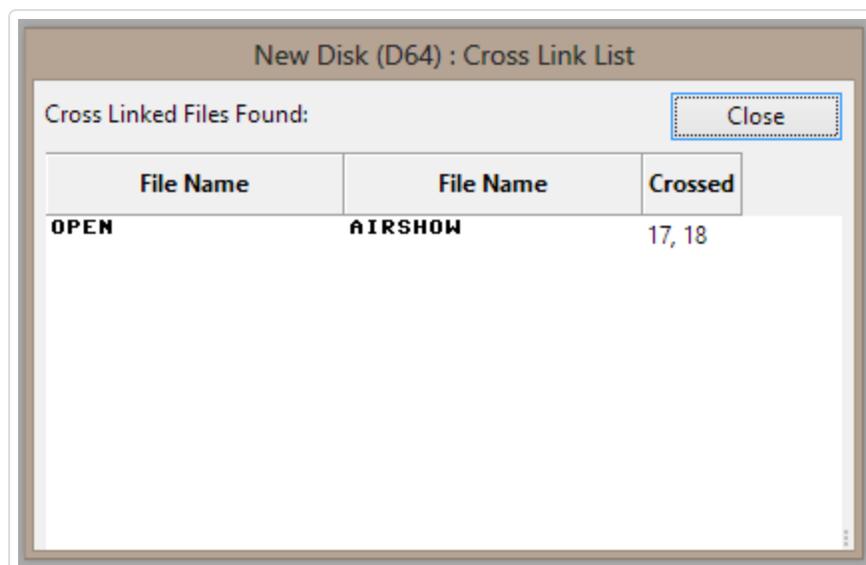
sectors that do not belong to any chain. The example below shows the File Chains dialog with the **Free Chains** node opened, showing a list of chains under it. The chain labeled "Free Chain 2 (13)" has been opened to reveal the list of track/sectors that comprise the chain. The (13) refers to the number of blocks in the chain. The first sector has been selected.

- 1 The **Refresh** button; click to recompute and refresh the list of file chains. You would use this if you are making changes to the disk image and want to see those changes reflected here.
- 2 The **Make File** button; this button is enabled whenever a sector from a row under **Free Chains** has been selected. Clicking Make File will produce a new file entry with the file name "free chain tt-ss", where tt and ss are the track and sector of the first track/sector in that chain. The newly created file entry will point to the first track/sector of the chain. You would use this to create an accessible file on your disk image from a chain that you think contains valuable data or a file that you want to be able to load, copy or manipulate later.
- 3 The file chains tree view; lists all the chains and single sector blocks as described above.



Cross Links

A potential disk image integrity issue is cross linked files; that is, two or more files on the disk image that share one or more sectors in their file chain. DirMaster can detect and report these via the main menu (). In the example below, the Cross Link List dialog is displaying one cross link, between the files "open" and "airshow"; the files are crossed at track/sector 17,18.

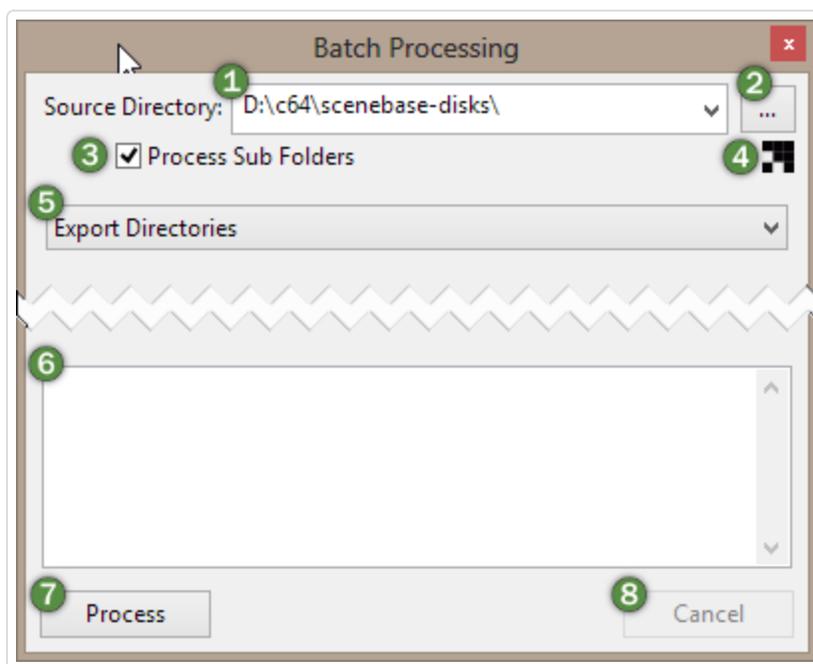


This section will cover the powerful batch processing capabilities in DirMaster that can help you work with very large disk image collections spread across several OS directories/sub-directories.

Batch processing in the context of DirMaster is the act of traversing and optionally recursing an OS directory, performing an action on every OS file discovered that is a supported disk image, file image or raw file archive format. Like the "Find In Disks" capability, batch processing runs in the background and you can use the other features of DirMaster while a batch process is active. There are three main types of processing actions:

- **Exporting Directories:** for each OS file processed, export its directory (if disk image) or list of contained files (if file image or raw file archive) in one of several possible output formats.
- **Extract Files:** for each OS file processed, extract the Commodore native files it contains into the OS file system.
- **Decompress Files to D64s:** for each OS file processed, if it is a raw file archive, extract and uncompress the Commodore native files it contains onto a new disk image, saving the resulting disk image into the OS file system.

The screen shot below shows the *common elements* of the Batch Processing dialog; these elements and controls are present regardless of the processing action being taken and are marked in the screen shot. The cut out area is where the action-specific controls would appear; processing actions and their controls are shown and explained in the sections following this one.



-
- 1 The **Source Directory** field; batch processing begins on the OS files contained in the OS directory specified here; the pop down control at the right interior of this field will display a history of previously used source directories
-

-
- 2 The browse button; click it to browse for and select a source directory

 - 3 The **Process Sub Folders** checkbox; when checked, DirMaster will recurse all OS sub-directories found in and below the **Source Directory**. When unchecked, the processing is limited to files in the **Source Directory**.

 - 4 An activity spinner; animates whenever processing is active

 - 5 The action pop up; switch this to the processing action you wish to perform; the space immediately below the pop up will change to reflect the chosen action (see below)

 - 6 The **Log** text area; contains a running list of OS files that were processed

 - 7 The **Process** button; click to commence the processing with whatever parameters that were set using the action specific controls immediately below the action pop up

 - 8 The **Cancel** button; click to stop the processing at the earliest possible point (actions already started on a file will complete but no new files will be processed)

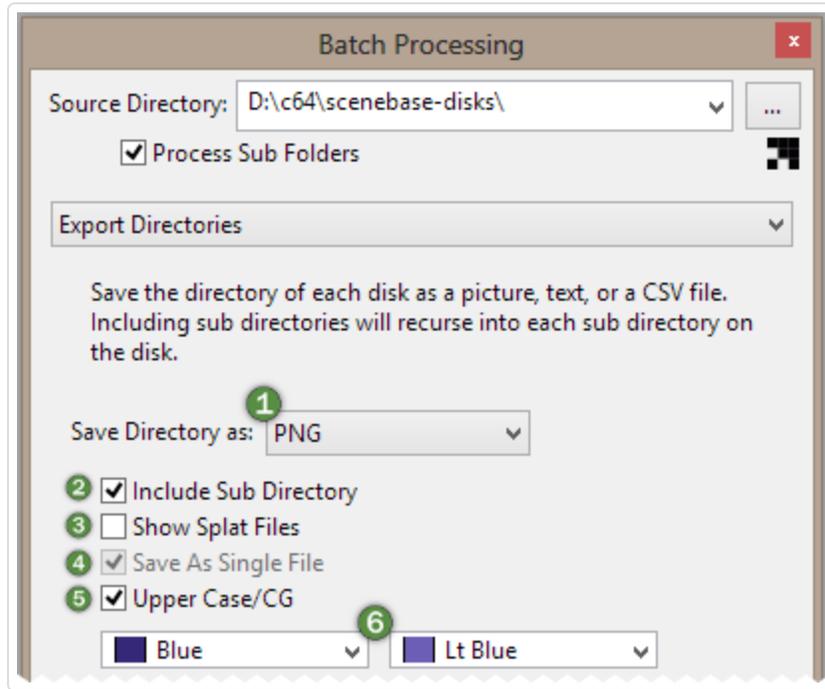
Export Directories

When exporting directories, DirMaster produces output for each disk image or archive file that is processed. Output formats are the same set of formats that are possible when saving a disk image: .bmp, .csv, .html, .png, .rtf, .seq and .txt. The directory export itself is written as the best visual representation of the directory possible given the target format. For formats like .png, the visual representation will be identical to the way the directory looks within DirMaster itself (which should also be - in most cases - identical to the way the directory looks when listed on actual hardware). Other formats like .txt and .rtf are obviously more limited. Comma separated is specifically geared toward providing an easily parsed format for importing into other systems.

MD5 Hashes in .csv Output

Exports to .csv format offer one additional option not directly configurable on the Batch Processing dialog - that is including an MD5 hash of each disk image and each file entry. This can be enabled by checking the "Output MD5 hash in CSV" option found on the Options dialog accessible through the main menu ().

The screen shot below denotes the configuration controls that appear for and are specific to batch exporting of directories. The common controls at the bottom of the Batch Processing dialog are omitted from the screen shot; those controls and the common controls at the top of the dialog are described above.

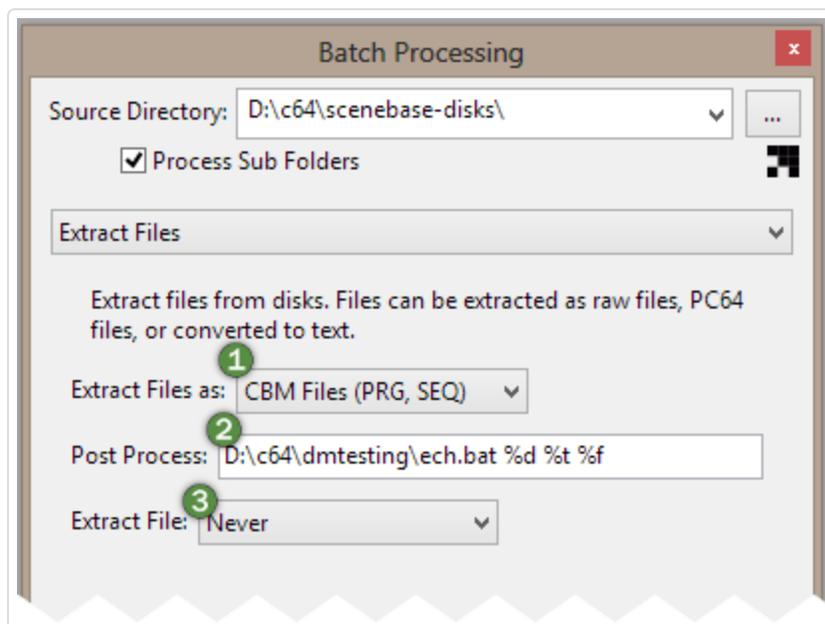


- 1 The **Format** selector; use this to chose what format to export disk image and archiver directories to.
- 2 The **Include Sub Directory** checkbox; when checked, the directory export will recurse into sub directories and partitions.
- 3 The **Show Splay Files** checkbox; when checked, the directory export will include DEL file entries.
- 4 The **Save As Single File** checkbox; when checked, only one file will created which contains all disk image and archiver directory exports. This checkbox is only available for .csv, .html, .rtf, .seq and .txt format exports.
- 5 The **Upper Case/CG** checkbox; when checked, directories are written as if Upper/CG mode were in effect.
- 6 The **Color** selectors; for formats where color is applicable (.bmp, .png, .html and .seq) these selectors control the foreground and background colors used in exported directories.

Extract Files

DirMaster's batch processing can extract individual files from disk images, saving those files to the file system. All disk image formats supported by DirMaster can have their files extracted, however only PRG and SEQ files are candidates for extraction. In the simplest use case, DirMaster will automatically extract every PRG and SEQ file, but the configuration controls specific to extracting files provide some additional control over whether extracted files are written to the file system as well as whether any additional processing is performed on each extracted file.

The screen shot below denotes the configuration controls that appear for and are specific to batch extraction of files. The common controls at the bottom of the Batch Processing dialog are omitted from the screen shot; those controls and the common controls at the top of the dialog are described further above.



- 1 The **Extract Files as** selector; use this to chose what format extracted files should be written as:

CBM Files	Extracted files are written as raw files with either .prg or .seq extensions. No changes are made to the file's contents before writing.
PC64 Files	Extracted files are written as file images with either .p00 or .s00 extensions. Each file image adheres to the respective PC64 format and contains the original file name inside the image. No changes are made to the file's contents before writing.
SEQ as ASCII	PRG files are not extracted. Extracted SEQ files are converted to ASCII and written to the file system with a .txt extension.

- 2 The **Post Process** field; if unset, does nothing. If set, the content of this field should be a full path/file pointing to a runnable target (exe, bat, or other runnable script). As files are extracted during the batch processing, each is written to disk and then the runnable target is invoked with three arguments in order:
1. A double quoted full path/file to the disk image the extracted file resided on
 2. A three letter file type argument, either PRG or SEQ
 3. A double quoted full path/file to the extracted file

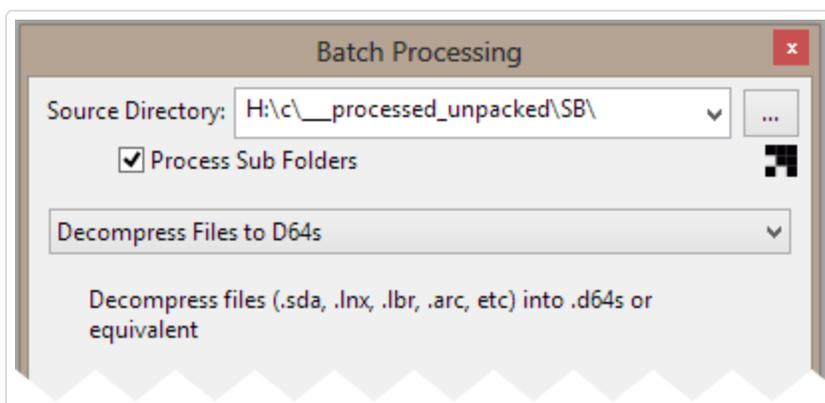
The goal of this feature is to provide a simple built in mechanism for processing the contents of disk images using self built or third party tools and scripts. When combined with the "Extract File" selector described below, it can also be used to filter or selectively extract files based on the return value of the invoked target.

- 3 The **Extract File** selector; the choice here determines whether or not an extracted file is *left* in the file system or deleted. All PRG/SEQ files are extracted initially (or only SEQ files in the case that "SEQ as ASCII" was selected) - but their final disposition depends on a combination of this selector and the "Post Process" target, if provided.

Always	Extracted files are always left in the file system regardless of the setting of the "Post Process" field.
Never	Extracted files are always deleted from the file system regardless of the setting of the "Post Process" field.
Post Process Success	Each extracted file is kept if the return value of the "Post Process" runnable target invocation is non-zero; otherwise the file is deleted.

Decompress Files to D64s

DirMaster's powerful archiver file decompression feature is also available for use while batch processing a collection of files. When using this batch mode, all archiver file images discovered in the batch will be decompressed and their contents copied onto new disk images written to the file system. The new disk images are typically .d64 but the same logic is applied here for choosing a disk image format as described under "[Extracting Archives](#)". The screen shot below shows the Batch Processing dialog when this mode is selected, but as you may note there no other configuration options associated with batch decompression.



DirMaster supports disk images that are formatted and structured for use in the GEOS operating system for C64/C128 as well as a few GEOS-specific file and archive formats. This section will detail those features specific to GEOS support.

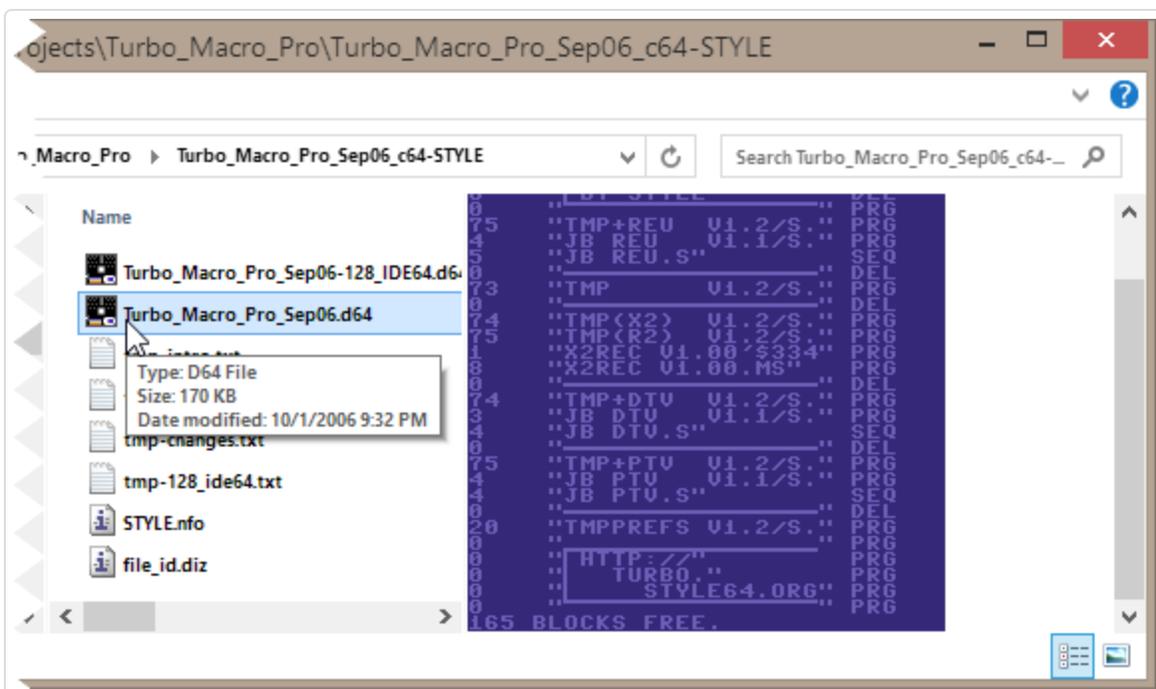
Documentation pending.

DirMaster v3 includes experimental OpenCBM support, and this section will explain how to use that to pull disk images into DirMaster from real Commodore disk drive devices.

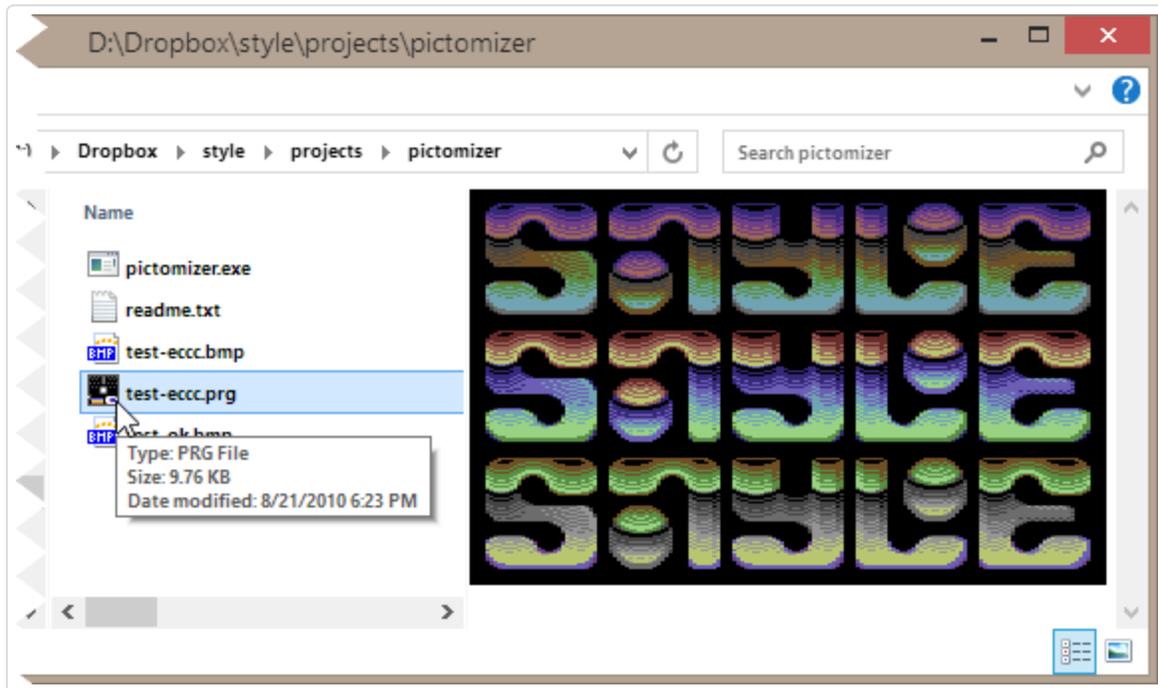
Documentation pending.

DirMaster v3.0 comes with an additional feature for users running Windows Vista, Windows 7, or Windows 8: Explorer Preview Pane integration.

If you chose to install the Explorer Preview integration (see "[New installs](#)"), then the preview pane will provide an appropriate snapshot of any file supported by DirMaster. This includes disk images, sequential files (.seq or .s00), archive contents (.ark, .lnx, etc) and many popular graphic formats (koala, etc). File recognition for previewing is a combination of supported file types (you must have added the extension to the file associations; see "[File Associations](#)") as well as, in some cases, recognized contents. For example, to preview the contents of a graphic picture file, the file would need to be named with a ".prg" extension, that extension would need to be associated with DirMaster, and the .prg would actually need to contain an intact and recognized image format like koala. To preview the contents of an archive, the file would need to be named with, for example, a ".lnx" extension, and so forth. See below for an example of the preview pane displaying a disk image:



As well as an example showing a graphic picture:



This section will describe the command line options provided by DirMaster.

You can invoke DirMaster from a command line interface in order to open one or more supported files; if you haven't changed the default executable name, this would look something like the following example that attempts to open two OS files (observe the use of quotation marks when the path/file contains a space):

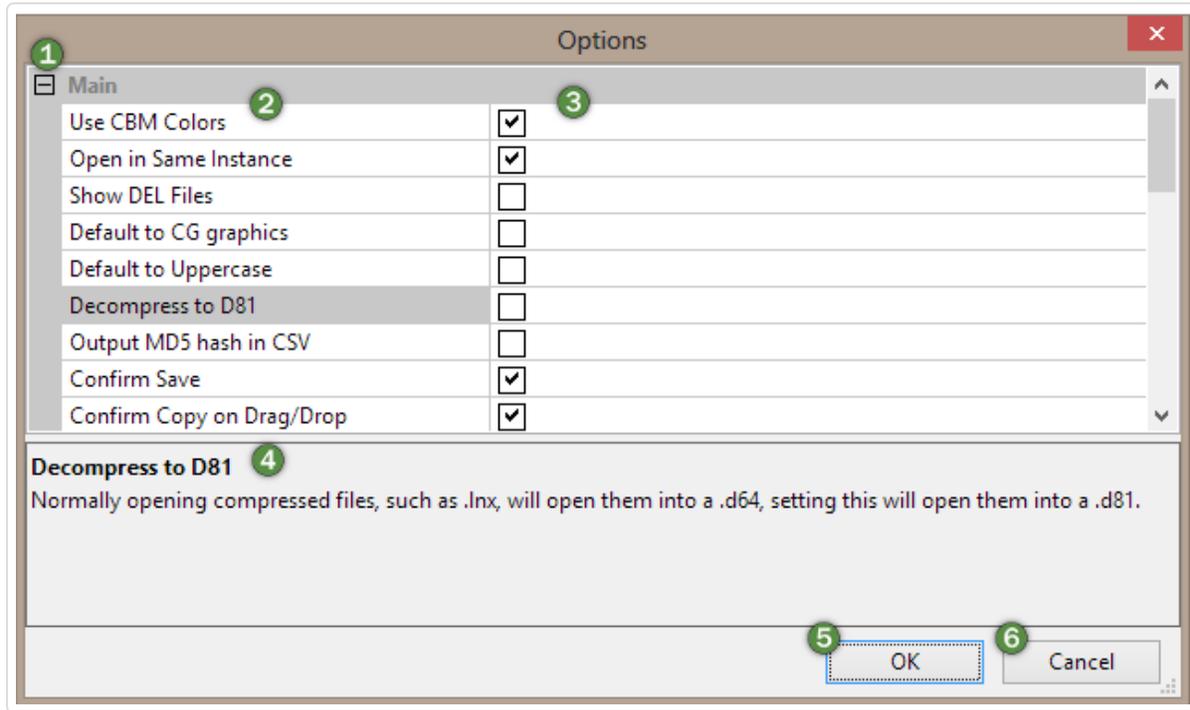
```
dirmaster.exe disk.d64 "c:\disks\second disk.d81"
```

You can pass a list of one or more OS file names with or without paths where multiple file names are separated by one or more spaces. Each file name can optionally be enclosed in quotes. DirMaster will attempt to open all OS files passed on the command line. Whether a new instance of DirMaster is started depends on the setting of the "Open in Same Instance" option (see "[Options Dialog](#)"). When this option is checked, opening files from the command line will cause them to open in an already running instance of DirMaster, if there is one.

This section will explain how to access and change configuration parameters that control how DirMaster works and which were not already explained in detail in preceding sections.

Options Dialog

The primary interface for configuring DirMaster is the Options dialog which can be invoked via the main menu (). Observe the dialog; the numbered markers identify the main elements of the dialog interface and are explained below.



- 1 The collapse gutter; the options as a whole are split into two large groups: the "Main" options and the "Run in Emulator" options (see "[Emulator Integration](#)"); some options within those groups are further grouped. The gutter contains collapse/expand markers to control the display of those groups.
- 2 The **Option Name** column; option names are shown here; you can click on an option name in this column to highlight it and display a short description in the Option Description panel (see below). In the screen shot above, the "Decompress to d81" option is highlighted.
- 3 The **Option Setting** column; contains a setting field or fields for configuring the option. In the screen above the displayed option settings are all check boxes but other options could be text fields, color selectors, and so on.
- 4 The **Option Description** panel; clicking an Option Name will display the corresponding description in this panel.
- 5 The **OK** button; click this to save changes you've made to the options.
- 6 The **Cancel** button; click this to cancel any changes and close the Options dialog.

The table below lists every option found in the "Main" section of the Options dialog. For more details on the "Run in Emulator" section, see "[Emulator Integration](#)".

Use CBM Colors	When checked, DirMaster uses the selected CBM colors for background/foreground text instead of system colors.
Open in Same Instance	When checked, DirMaster will reuse the current application instance when opening new files (e.g. double clicking an associated file in Windows Explorer) instead of starting a new application instance.

Show DEL Files	When checked, DirMaster will display DEL file entires in directory windows; this has the same affect as the corresponding menu and toolbar controls (see " Toggling DEL File Display ").
Default to CG graphics	When checked, DirMaster will to the best of its ability output actual C/G characters and control codes (colors, etc) when displaying SEQ files instead of outputting the equivalent printable PETSCII character.
Default to Uppercase	When checked, DirMaster will open new directory windows and other windows in upper/graphics mode instead of lower/upper mode.
Decompress to D81	When checked, DirMaster will uncompress archives by creating and outputting files to new .d81 disk image instead of a new .d64 disk image; see also " Extracting Archives ".
Output MD5 hash in CSV	When checked, DirMaster will compute and output an MD5 hash for each file it processes when producing .csv file output, for example during a batch operation; see " Exporting Directories ".
Confirm Save	When checked, DirMaster will display a confirmation prompt before closing any disk image has been changed; when unchecked DirMaster immediately closes the image and ignores any unsaved changes.
Confirm Copy on Drag/Drop	When checked, DirMaster will display a confirmation prompt when attempting to copy a file onto a disk image where a file with the same name already exists; when unchecked DirMaster copys the file, therefore resulting in two file entries with the same name on the destination image.
Font Size	Accepts an integer value and sets the font size in pixels; any value below 8 could be difficult to read and any value not evenly divisible by 8 will not exhibit correct (read: square) aspect ratios. The value set here is also controlled/changed when using the Toggle Font Size button on the toolbar or the Font Size dialog accessed via the main menu (see " Changing Font Size ")
Use Custom Interleave	When checked, DirMaster will use a custom interleave (see the "Interleave" option below) when writing files to disk instead of the standard interleave that emulates disk drive behavior (e.g. 10 for a 1541 (.d64)
Interleave	Accepts an integer value as the interleave to use when writing files. (<i>grouped under "Use Custom Interleave"</i>)
Progressive Interleave	When checked, DirMaster will search for free sectors at track/sector 1,0 instead of emulating typical disk drive behavior (e.g. beginning at the middle of the disk ala 17,0).
Detokenize to BasText	When checked, DirMaster will display so called BasText tokens (e.g. <code>{red}</code>) when displaying SEQ files and BASIC listings instead of outputting the equivalent printable PETSCII character.
Display Sectors horizontally in the BAM	When checked, DirMaster will display sectors from left to right and tracks from top to bottom, instead of tracks from left to right and sectors from top to bottom.
Fill Byte	Accepts an integer value from 0 to 255 which DirMaster will use when filling sectors (see " Filling Free Sectors ").

Use Fill Byte At Disk New	When checked, DirMaster will use the "Fill Byte" (see above) as the initial byte that all sectors are filled with for new disk images instead of emulating typical disk drive behavior by using an appropriate canonical 'new sector'.
Text	Color selector that sets the foreground color used when displaying directories, SEQ files, and so on. Works the same as the corresponding tool bar color selector (see " Changing Colors ").
Background	Color selector that sets the background color used when displaying directories, SEQ files, and so on. Works the same as the corresponding tool bar color selector (see " Changing Colors ").
Headless HTML	When checked, DirMaster will omit any header/wrapper (e.g. <code><HTML></code> , <code><BODY></code> tags) from HTML directory exports. The net effect of this is to produce only the <code><DIV></code> block containing the actual directory, which is useful when the intention is to (programmatically) include the exported HTML in a dynamic page or similar system.
Embed CSS in HTML	When checked, DirMaster will output a <code><SCRIPT></code> tag with CSS styling necessary for "proper" display of disk directories in all supported browsers. When unchecked, DirMaster will instead output a <code><LINK></code> tag pointing to the filename provided in the "CSS Link" option (see below). (<i>grouped under "Headless HTML"</i>)
CSS Link	Accepts any text that is used as the target of an outputted <code><LINK></code> tag when "Embed CSS in HTML" is not checked. (<i>grouped under "Embed CSS in HTML"</i>)

Configuring the Palette

DirMaster supports palette customization although there is no direct GUI interface for manipulating the palette. By palette, we are specifically addressing the sixteen Commodore 64 colors that DirMaster uses to display directories, hex dumps, sector data, seq files, pictures, and so forth. Any palette is only a rough approximation of an actual c64 display, but if you have a preferred palette you can enable that in DirMaster.

You'll need a .vpl format palette file - this is the same file format used by VICE emulator. Simply rename any proper .vpl file to `dirmaster.vpl`, then move or copy that file into the same directory where the installed DirMaster.exe resides. Quit and restart DirMaster and your palette should be active. In the absence of a `dirmaster.vpl`, DirMaster defaults to the same colors as the popular "pepto" palette.

File Type Support

All file types supported by DirMaster in some way are shown below. Each is classified at a high level according to the way DirMaster handles the files of that type:

- **disk image:** a container for Commodore files with native (PETSCII) file names; usually representative of physical media
- **file image:** a container for a single Commodore file with a native (PETSCII) file name

- **archive**: a container for Commodore files with native (PETSCII) file names; usually a direct copy of an archive file produced by Commodore software
- **raw file**: bytes with no accompanying native file name; usually a direct copy of some Commodore file (.prg, .seq, etc)

.d64	disk image	representation of one side of a 5¼ floppy disk formatted for a 1541-compatible drive
.g64	disk image	representation of one side of a 5¼ floppy disk formatted for a 1541-compatible drive; image replicates the real floppy disk at the GCR level Read Only
.nib	disk image	representation of one side of a 5¼ floppy disk formatted for a 1541-compatible drive; image replicates the real floppy disk at the GCR level and includes additional data about the physical media like track density Read Only
.d71	disk image	representation of both sides of a 5¼ floppy disk formatted for a 1571-compatible drive in double sided mode
.d81	disk image	representation of a 3½ disk formatted for a 1581-compatible drive Sub-directories
.d80	disk image	representation of a 5¼ floppy disk formatted for an 8050-compatible drive
.d82	disk image	representation of a 5¼ floppy disk formatted for an 8250-compatible drive (uses both disk sides)
.d2m	disk image	representation of a 3½ disk formatted for a CMD FD2000-compatible drive Partitions Sub-directories
.d4m	disk image	representation of a 3½ disk formatted for a CMD FD4000-compatible drive Partitions Sub-directories
.dhd	disk image	representation of an entire CMD HD hard drive Partitions Sub-directories
.hdd	disk image	representation of an entire IDE hard drive formatted for use with the IDE64 Read Only
.dnp	disk image	representation of a CMD native partition, which is a file system structure used on CMD devices like the FD2000 and CMD HD Sub-directories
.t64	disk image	representation of a tape cassette formatted for use in, e.g., a 1530 Datasette
.dfi	disk image	a custom format devised for use with the MMC64 series of expansion cartridge
.m2i	disk image	a custom format devised for use with the MMC2IEC/SD2IEC series of hardware addons Read Only

Note: all file images are Read Only

.cvt	file image	representation of a single GEOS file; produced by Convert
.p00	file image	representation of a single PRG file; originated with PC64 emulator
.r00	file image	representation of a single REL file; originated with PC64 emulator
.s00	file image	representation of a single SEQ file; originated with PC64 emulator
.u00	file image	representation of a single USR file; originated with PC64 emulator

Note: all archives are Read Only

.arc	archive	archive format produced by ARC 64 and ARC 128
.ark	archive	archive format produced by Arkive
.lbr	archive	archive format produced by Library and Library 128
.lnx	archive	archive format produced by Lynx and Lynx variants
.lha	archive	archive format produced by LHArc from the CS-DOS package
.sda	archive	archive format produced by Arc 64 and ARC 128 with a self dissolving stub
.sfx	archive	archive format produced by tools in the CS-DOS package
.spy	archive	archive format produced by Spyne
.wr3	archive	archive format produced by Wraptor v3.0
1! (zip4)	archive	archive format produced by Zip-Pack

Note: all raw files are Read Only

.prg	raw file	raw data derived directly from a PRG file
.rel	raw file	raw data derived directly from a REL file
.seq	raw file	raw data derived directly from a SEQ file, presumably in the PETSCII character set
.usr	raw file	raw data derived directly from a USR file
.txt	raw file	presumably ASCII text

Other Supported Files

DirMaster also supports a number of Commodore specific file formats for graphics/pictures, both in the application (see "[Picture Viewer](#)") and through the Windows Explorer preview panel (see "[Windows Explorer Integration](#)"). The currently supported formats are: Koala

(incl. packed), Doodle (incl. packed), Amica (incl. packed), Centauri, OCP, DrazPaint (incl. packed), GigaPaint (hi/mc), MonoMagic, Image System (hi/mc), Interpaint (hi/mc), HiEddi, Art Studio, Paint Magic, Picasso, Rainbow, Run Paint, Saracen, Blazing Paddles, Micro, CDU, Artist, Vidcom, FacePaint, Cheese, Garfield, Dolphin Ed