

mdvtool

The MIST QL core primarily uses microdrive images in QLAY format. Unfortunately this is not the primary format of QL files available in the net. Many are in ZIP format instead and must be unpacked on the QL itself to correctly deal with special header flags.

Thus **mdvtool** was written.

It can be used to erase all data from a given MDV image and to write the contents of a ZIP file instead.

The resulting image can then be used with the MIST QL core or QL Microdrive emulators which use the same format.

It can also be used to create new empty MDV images.

Just unzip the Windows binary (called mdvtool) to a suitable folder on your hard drive, then you can use the following pages as reference for the commands to be used. The examples assume that mdvtool, the MDV image files and any individual files or zip files are all in that same folder.

If you enter the command **mdvtool** with no parameters, you'll get a short list of available commands as follows:

Commands	Function
create	- create a new MDV image
dir	- list MDV comments
check_files	- check file integrity
file_chains	- list chain of sectors for each file
check_mapping	- check the sector mapping
show_mapping	- show physical/logical sector mapping
export file_name	- export a file from the MDV image
erase	- erase the MDV image
name	- image name
rand random	- set the random number "fingerprint" for the MDV image
import file_name	- import a file to the MDV image
zip_import file_name	- import an entire ZIP archive
write file_name	- write the MDV image

In Windows, the commands take the following general format:

mdvtool source_file command parameter write output_file

For most commands, you need to specify an MDV image file to work from (source_file above) and a new one to write the results out to (output_file above). The write command tells mdvtool to write the result out to the MDV image file called output_file, which should have the filename extension .mdv . If output_file is the same name as source_file, it overwrites the original source_file MDV image.

Create A New MDV Image File

Create a new MDV image of 255 sectors (253/255 usable) by using mdvtool with a create command and a write instruction:

mdvtool create write blank.mdv Create a new MDV image called blank.mdv, with the default medium name of "MD"

If you wish to give the new MDV image a medium name, you can add a **name** command after the **create** to give the new MDV image a name of up to 10 characters long. If the **name** command is not used, a default name of "MD" is used.

mdvtool create name A_BLANK write blank.mdv Create a new MDV image called blank.mdv with a medium name of A_BLANK

Use the command with care – it is easy to overwrite an existing MDV file by mistake.

Specify A Random Number For The MDV Image

If you'd like to specify a particular random number in the formatting of an MDV image, it is possible to use the **rand** command to allow mdvtool to do so. This 16-bit random number is used as a kind of "fingerprint" by some protected QL software to prevent unauthorised copying. It's also used by a QL with physical microdrives to help detect change of microdrive cartridges.

Mdvtool blank.mdv rand 12AB write blank.mdv change the image's random number to the hexadecimal value specified

The random value can be specified preceded by 0x for clarity to indicate hexadecimal value:

mdvtool blank.mdv rand 0x12AB write blank.mdv change the image's random number to the hexadecimal value specified

Check List Of Files On An MDV Image

Use the **dir** command after the name of the MDV image file you wish to check:

mdvtool games.mdv dir

This will give a listing from both the directory file and file headers, to show the medium name, random number value, total sectors, sectors used, sectors free and a list of files listed from both the directory file and file headers.

Add (Import) A File To An Existing MDV Image

Use the **import** command to add a QL file called example_txt into an existing MDV image called games.mdv :

mdvtool games.mdv import example_txt write games.mdv add the example_txt file to the MDV image called games.mdv and write it out with same MDV image name

Copy A File Out Of An MDV Image

Using the **export** command we can copy a file out of an MDV image. Suppose we have an MDV image called Games.mdv and we wish to extract a file called manual_txt from it.

mdvtool games.mdv export manual_txt export file called manual_txt from games.mdv to current directory

Erase An MDV Image

To erase an MDV image, we can use the **erase** command to remove all files from within a specified MDV image and write it out either to the same or a different MDV image.

mdvtool games.mdv erase write games.mdv erase all files from **games.mdv** and write the result back out to the same MDV image

mdvtool games.mdv erase write newer.mdv erase all files from **games.mdv** and write the result out to a new MDV image called **newer.mdv**

Copying Content Of A Zip File Into An MDV Image

Many QL programs you can download from the web are in zip file format. They must be extracted on a QL file system to correctly deal with the executable file headers. We've all been there: extracted a QL program in a non-QL file system such as Windows and the program will no longer EXEC, most probably giving a "bad parameter" error when you try. This facility comfortably bypasses this issue by allowing files from zip files to be put directly into an MDV image without loss of executable file headers by being temporarily placed in a non-QDOS filing system.

Mdvtool can write the contents of such a downloaded zip file directly into an MDV image in order to preserve any executable file headers, provided of course that the content of the zip file will fit into the MDV image, which has a maximum capacity of 253/255 sectors.

For this example we have downloaded a game called puzzle.zip and wish to add it to a MDV image called games.mdv :

mdvtool games.mdv zip_import puzzle.zip write games.mdv import content of puzzle.zip into games.mdv and write out to the same MDV image file.

We can write the result out to a different MDV image. This would be useful if we wish to add files, but test the result first to see if the imported files work in the emulator using the MDV image. Then if it doesn't work we still have the original MDV image to go back to.

mdvtool games.mdv zip_import puzzle.zip write games2.mdv import content of puzzle.zip into games.mdv and write out to a new MDV image file called games2.mdv

Probably a better way of doing this, especially if we want all the zip files to be imported into separate MDV image files (each game being in its own MDV image) is to create a blank MDV image to use as a base and then write out the imported files to a new MDV image:

Step 1:

mdvtool create write blank.mdv create a blank MDV image called "MD"

Or, you could add a **name** command to give it a particular medium name if you wish:

mdvtool create name blank_mdv write blank.mdv create a blank MDV image called blank_mdv

Step 2:

mdvtool blank.mdv name TANK zip_import tank.zip write tank.mdv

Broken down into sections, the second step consists of:

blank.mdv – this is used as the base MDV image to create others from

name TANK – the newly created MDV image will have the medium name TANK

zip_import tank.zip – import all files from the zip file called tank.zip

write tank.mdv – this writes the result out to a new MDV image called tank.mdv

If the base (original) MDV image is not empty, you can add an **erase** command to empty the copy before anything else is done – that way the original MDV image is not changed as long as you write it out to a new MDV image:

mdvtool blank.mdv erase name TANK zip_import tank.zip write tank.mdv

Broken down into sections, this consists of:

blank.mdv – this is used as the base MDV image and copied to work from

erase – any files in the copy are erased, making the copy blank

name TANK – the now blank MDV image is given the medium name TANK

zip_import tank.zip – import all files from the zip file called tank.zip

write tank.mdv – this writes the result out to a new MDV image called tank.mdv

Note that the original MDV image is not affected in any way (unless you write the result out to the same name as the original).

Once you have completed all steps required, simply copy the MDV image file to a location on your hard drive where your emulator will be able to use it.