gnmediahelper

a Windows tool that helps making basic operations of FFmpeg easier usable.

Table of Contents

Index 0

1 Purpose

gnmediahelper is a Win32 command line tool that can be used in a Windows cmd.exe shell or called inside a .bat or other Windows script file.
Using a command line tool requires some knowledge in DOS usage.

FFmpeg is a trademark of Fabrice Bellard the initiator of the FFmpeg project (https://www.ffmpeg.org)
ffmpeg.exe is an open source free GNU command line tool to convert and modify media files (video, audio, images).
ffplay.exe is an open source free GNU command line media player.
ffprobe.exe is an open source free GNU command line media analysis tool.

gnmediahelper only task is to call the free media conversion tool ffmpeg.exe . gnmediahelper self does not convert or analyse media files.
ffmpeg.exe can be downloaded and used free on your computer. ffmpeg.exe is a very mighty tool to convert media files (video, audio, images) and apply modification using filters during the conversion but unfortunately it is very difficult to use it. Even a programmer like me must search in internet for examples to solve certain tasks that could be done with FFmpeg .
Command line tools are surely always a bit more difficult to use than applications with dialogs but ffmpeg.exe syntax is extreme difficult compared to most other command line tools simply because it supports complex tasks and the usual tasks for users are a bit lost in complex syntax.
Using ffmpeg.exe in scripts by programmers is not easy (e.g. writes errors and normal information to same output that it is difficult for a script to distinguish).

gnmediahelper tries to simplify most important tasks and allows to combine them and repeat them with many files (depending on the license features you have purchased).
gnmediahelper supports defining user operations. You could have found your optimal FFmpeg command line for a certain task and in most cases you can define a user command with parameters and options that your operation can be done simpler in future.

2 License

gnmediahelper is a commercial application. Program sources are not open. You can try demo free for 14 days. After this you must purchase a license or delete the program.
ffmpeg.exe is a free application that you can download from https://www.ffmpeg.org (read license at their web page)
Surely you can use ffmpeg.exe without gnmediahelper.exe
It is allowed to run ffmpeg.exe or call ffmpeg.exe from other applications without costs according to their license.
gnmediahelper application does not include any code from FFmpeg in its source code. It is independent development from this project. gnmediahelper is no GNU project.
gnmediahelper licenses cost some price. If you don't want to support our work you can surely use ffmpeg.exe directly without any costs and invest your time self.

gnmediahelper runs different

- without license (runs as demo with some limitations)
- with a single operation license (supports standard operations and combi operations for single input files)
- with a batch operation license (supports also single, combi, user and batch operations)

A license includes free upgrades for 2 years. Later released program versions can be used with a license upgrade. Demo runs when using a program version that is not included in your license.

Hint: Upgrading to batch operation license later might be possible but the price could be higher than purchasing the batch license directly.

3 Demo

The gnmediahelper demo can be used without installed license and without costs for 14 days. After the demo period you must purchase a license or delete the program.

User operations can be tried with some demo user operations op:mspaint, op:pixelize, op:scaledgreywithborder, op:louder.
batch processing (convert many files and folders) can be tried with limit: 2 converted files per folder.

To use the features with other files and more files at once you would need to purchase and install a gnmediahelper license (single or batch).

order page: https://www.gnmidi.com/gnorderen.htm

4 Disclaimer

The results of gnmediahelper depend on the used tool ffmpeg.exe
gnmediahelper only calls ffmpeg.exe and does not create media files self.
gnmediahelper can not produce different or better results than ffmpeg.exe produces.
gnmediahelper cannot speed up any FFmpeg operation and it is not the task of gnmediahelper to improve ffmpeg.exe .
Different ffmpeg.exe versions exist and might produce different results. It is recommended only to use ffmpeg.exe versions that we tested.
Sometimes ffmpeg.exe might create a result that looks correct for ffmpeg.exe and can be displayed by their media player ffplay.exe but other players or programs can not read the files.
Certain media formats have limitations and causes error message when using settings that are not usable with the format (e.g. some video formats do not allow odd pixel width or height or cannot handle very small sizes).
gnmediahelper does not know limitations of such formats and using them will cause error during conversion.
It is possible to define user operations self. Minor errors in the defined FFmpeg command line might cause that the operation does not work or produce an error.
ffmpeg.exe analyzing or conversion of video (and sometimes also audio) might take very long time. E.g. calculating the exact duration or modifying all frames of a compressed video must read the video till end.
ffmpeg.exe developers try to keep the command line syntax compatible between versions but that does not work always.
Some operations that were supported in older versions are deprecated in newer version and need other command line syntax to achieve similar results.
we suggest a 32bit and a 64bit Windows version compiled and published in 2023 and tested with gnmediahelper. gnmediahelper can not find out if an older ffmpeg.exe version worked identically with a command.
We have also seen a ffmpeg.exe version that started very slow (some seconds for collecting libraries or whatever it did, maybe also virus checker always checked the loaded binaries and cost too much time for using that version).
We have also noticed that certain operations have run successful with ffmpeg.exe and the result was not playable or only some part of the media was playable.
Some ffmpeg.exe versions treat text files as valid media files with duration 0:00 and newer versions display an error (stream not found).
Results can be different depending on used ffmpeg.exe version.

5 DOS tutorials

gnmediahelper.exe is a command line tool (also ffmpeg.exe).
It requires some knowledge about:

- starting Windows cmd.exe MS-DOS box
- changing current directory (cd ...) and getting listing entries of a directory (using dir command)
- absolute (c:\myfolder\image.jpg) and relative path names (..\..\myfolder\image.jpg) of files
- using path that contains spaces (e.g. "..\myfolder\images\pink flowers.bmp")
- wildcards and ? for matching some file names (e.g. c:\myfolder*.jpg)
- adding path of gnmediahelper folder to system environment PATH variable (e.g. PATH=c:\utils;c:\gnmediahelper;c:\utils\ffmpeg\bin)
- starting a program like gnmediahelper.exe with parameters

gnmediahelper does not require knowledge about .bat programming or filename modifiers or special DOS commands (like IF, FOR, GOTO, SHIFT ...)

Here you could find a tutorial about using cmd.exe
https://www.computerhope.com/issues/chusedos.htm

A usual gnmediahelper session looks like this in cmd.exe

```
C:\> y:
Y:\> mkdir result
Y:\> cd "\mymusic\The Beatles"
Y:\mymusic\beatles> dir *.mp3
05.02.2013  22:59         6.799.242 The Beatles - Hey Jude.mp3
04.02.2013  04:51         2.011.512 The Beatles - Yesterday.mp3

Y:\mymusic\beatles> gnmediahelper "The Beatles - Yesterday.mp3" y:\result op:fadein
10
Y:\mymusic\beatles> dir y:\result\*.mp3
09.01.2024  11:39         2.010.595 yesterday with fadein.mp3

Y:\mymusic\beatles> gnmediahelper "y:\result\yesterday with fadein.mp3" op:open
```

Hint: the beginning of each line till > is displayed by cmd.exe (the current drive and directory)
Hint: when writing the beginning of an existing file or directory path cmd.exe allows to complete the full name with TAB key and automatically quotes the path if it contains spaces.

6 First steps

unzip gnmediahelper.zip into a new folder that is not protected by Windows (do not use c:\programs or c:\program files or desktop ...).
Depending on your computer you need a 64 bit or 32 bit ffmpeg package (warning: 64bit application will not run on a 32 bit computer, other direction works).

The gnmediahelper folder contains a file links_english.htm that can be displayed with your browser (simply open the file). It contains links to FFmpeg download pages and a list of ffmpeg.exe version numbers that were tested successfully with gnmediahelper. Here you can also find link to homepage, download page, order page of gnmediahelper application.

When using an other ffmpeg package than suggested ones it is not guaranteed that all commands are still same (ffmpeg syntax or behavior might change with versions).

Unzip the ffmpeg into a new folder (best into a sub folder of gnmediahelper.zip). It should contain a bin

folder that contains ffmpeg.exe, ffplay.exe, ffprobe.exe

Start the ffmpeg.exe with following command line in the bin folder: ffmpeg -help
This shows the ffmpeg syntax and its complexity. If it displays an error about 64bit not supported then you must use the 32bit ffmpeg package.

Use gnmediahelper to choose this ffmpeg.exe with the dialog:
```
gnmediahelper -chooseffmpeg
```

It can also be done without dialog:
```
gnmediahelper -setffmpegpath c:\gnmediahelper\ffmpeg\bin\ffmpeg.exe
```

(use the correct full path where you have ffmpeg.exe)

Add your gnmediahelper folder to system environment PATH variable with system dialog (use semicolon ; between two path names) and store it when the new PATH is correct.
PATH c:\gnmediahelper;...

if you want you can also add the path of the ffmpeg bin folder but it is not required for using gnmediahelper.

If not adding gnmediahelper folder to PATH then you need always specify the full path of gnmediahelper
e.g. `c:\gnmediahelper\gnmediahelper input.jpg output.png op:convert`

Hint: the PATH setting does not change the PATH in already open command shells. You need to close and open a new cmd.exe

With working PATH set you can write
```
gnmediahelper input.jpg output.png op:convert
```
(independent what your current folder is).

7 Usage and examples

```
gnmediahelper -help
gnmediahelper -help | more
```
shows usage of program including all operations (a bigger list)

```
gnmediahelper -help sys
```
shows usage of program and operation usage for operations that contain sys in name e.g. system

```
gnmediahelper -examples=*
gnmediahelper -examples=* > examples.txt
gnmediahelper -examples=* | more
```
shows examples for all operations (where examples are available)
this is a long text use redirection (> examples.txt) or more pipe

```
gnmediahelper -examples op:scale 320 op:greyscale op:open
```
show examples for operations op:scale, op:greyscale op:open and any operation where the parameter is contained in operation name (e.g. x would match *x*)

```
gnmediahelper -examples scale
```
shows examples for op:scale and op:scaleaspect

```
gnmediahelper -examples *bright*
```
shows examples for op:brightness, wildcards are allowed in filter

8 General syntax

```
gnmediahelper [options] input [output] op:operation1 operationparameters
op:operation2 ...
```

Available options:
-help or -h shows usage of the command and available operations
-setffmpegpath path once the ffmpeg.exe path must be set that gnmediahelper calls the correct ffmpeg.exe version.
-chooseffmpeg once the ffmpeg.exe must be chosen by using a file dialog.
-examples operations gives usage and examples for specified operations (wildcards * is allowed)
-logfile=logfilepath.log (default in user application data)
-openlog opens the log file by system text editor (by default: notepad), log file contains information about done operations and error messages. For user operations it contains output from ffmpeg.exe
-clearlog deletes the log file before operations are started
-debug enables more logging output for user operations (shows how ffmpeg.exe was called and what output it returned including possible errors and warnings).
-overwrite suppresses asking if an existing output file can be overwritten by the program (risk by user, backup is recommended).
-nodeleteoldtemporaryfiles
do not delete old temporary files at beginning of this program use (will be done at a later time).
This option is only necessary if calling gnmediahelper again in sub processes with temporary results of the caller gnmediahelper process.
-allfiles also convert non-media files during batch conversion

The input is required and could be an existing file path or an input folder or an input folder with file mask (batch operations if license supports batch operations).

The output can be omitted (or minus -) if the last operation displays the result by a viewer or displays

information about the result.
A specified output can be a file name or an output file mask with file extension or an output folder with output file extension (batch operations).

Filenames that contain spaces must be enclosed using double quotes "...".
Warning: If a file path contains non-ascii characters (e.g. é or spanish y or special characters from other countries) then the file might not be usable. Using only ASCII characters in file path is suggested.

Operations always start with **op:** and use an identifier for operation name.
All parameters between two op: arguments or till last one are operation parameters for the operation before.
  gnmediahelper input.mp3 output.wav op:operation1 params1 ... op:operation2 params2
An operation does not require to have parameters and some operations could allow default options.
Operation parameters can be named options or unnamed parameters. Order of named options is not important. Order of unnamed parameters is essential for their meaning.

A **named option** can be written as (with same meaning, option can be any name that the operation supports e.g. width, padcolor):
```
-option
-option=value
-option:value
option=value
option:value
/option
```

Unnamed parameters can be
```
param1
-1
240
filename.ext
c:\path\filename.txt
```

Usage of operations
`gnmediahelper`
  displays usage of all operations
`gnmediahelper | find "scale"`
  displays usage of operations that contain scale in usage

Examples of operations
`gnmediahelper -examples op:scale op:size`
  displays usage and examples for specified operation names if the operation name is known

Log file
in file gnmediahelper.log text file important notices about conversions or errors are added

open the log file in a text editor with
`gnmediahelper -openlog`

At end of the text file you find information about the most recent operations.

9 Single operation

has only one op:operationname and parameters till last argument.
Default operation is op:info and shows information about input file.
The operation writes result into output file if specified and not equal to minus -
some operations use default option values when the named argument or the unnamed parameter is missing

Output can be written as path*.ext if the result file should be written to a given folder and other file extension should be created (that will use the same input name).

examples for single operations

```
gnmediahelper input
gnmediahelper input op:info
```
displays information about input.

```
gnmediahelper input.jpg output.png op:scale 320 240
gnmediahelper input.jpg output.png op:scale height:240 width=320
gnmediahelper input.jpg output.png op:scale -height=240 -width=320
```
scales input file input.jpg to width 320 and height 240 and writes result to output.png (format PNG).

```
gnmediahelper input.jpg output.png op:scale 320
gnmediahelper input.jpg output.png op:scale width=320
gnmediahelper input.jpg output.png op:scale -width:320
```
scales input file input.jpg to width 320 and assumes that the missing argument height is calculated from aspect ratio of input image and writes result to output.png (format PNG).

```
gnmediahelper input.jpg output\*.png op:scale 320 240
```
scales input file input.jpg width 320 and height 240 and writes result to output\input.png (if folder output can be created or already exists).

10 Combi operations

If more operations are specified then these operations are applied serial to produce output as last operation result.

```
gnmediahelper input.jpg op:scale width:320 height:240 op:addborder borderwidth:5
bordercolor:white op:open
```
first scales input.jpg to 320x240 to a temporaryfile1
then adds border with given color to temporaryfile1 and writes to temporaryfile2
then opens temporaryfile2 by a system viewer
then deletes temporaryfile1
temporaryfile2 will be deleted at next program start (viewer might be open for longer)

```
gnmediahelper input.jpg output.png op:scale width:320 height:240 op:addborder
borderwidth:5 bordercolor:white
```
first scales input.jpg to 320x240 to a temporaryfile1
then adds border with given color to temporaryfile1 and writes to temporaryfile2
then asks user if user allows to overwrite file output.png (if file was existing)
then copies temporaryfile2 to output.png (if user allowed overwrite)
then deletes temporaryfile1 and temporaryfile2

Hint: if an operation is a view operation (like op:open or op:ffplay) then the temporary file will be deleted at next program start so that the viewer can still view the file even when conversion has been completed.

11 Batch operations

Notice: this feature requires a batch license

Batch processing means that many files from an input folder (matching a file mask) are converted at once to an output folder (with optional output file extension).
gnmediahelper creates same folder structure as found in the input folder in the output folder (including

sub folders).
Be sure to specify an output folder that does not contain files. By default gnmediahelper will not overwrite existing output files and ask user for every existing file about permission to overwrite file. For own risk you can use option -overwrite that the program does not ask anymore (be sure to have a backup, operations can not be undone!).
Batch processing can be done with single operation or combi operations.
Warning: Batch processing with video or audio can take long time.
With Ctrl+C (Strg+C) you can abort the application (temporary files might remain in folder %temp% when aborting).

Hint: batch operation ignores many files that usually do not contain video/image/audio data (e.g. *.txt, *.zip).
gnffmpeg.ini uses a setting that defines which file extension should be accepted for input files (use only lower case).

```
[supportbatchfileextension]
.dmp=no
.ogg=yes
```

Hint: Use option -allfiles to convert all files matching the input file mask even (ignores the supportbatchfileextension table)
Hint: Use an input file extension in the input file mask to convert matching files with this extension (ignores the supportbatchfileextension table)

```
gnmediahelper inputfolder outputfolder op:scale width:320 height:240
```
scales all files to size 320x240 (only works for images and videos)
for existing file inputfolder\images\camera\vacation.jpg it creates outputfolder\images\camera\vacation.jpg
errors for other files are written to log file

```
gnmediahelper inputfolder\*.* outputfolder\*.png op:scale width:320 height:240
```
scales all files to size 320x240 and converts to format png (only works for images and videos)
for existing file inputfolder\images\camera\vacation.jpg it creates outputfolder\images\camera\vacation.png
errors for other files are written to log file

```
gnmediahelper inputfolder\*.jpg outputfolder\*.png op:scale width:320 height:240
```
scales all jpeg files to size 320x240 and converts to format png
for existing file inputfolder\images\camera\vacation.jpg it creates outputfolder\images\camera\vacation.png

The **log file contains error messages** for all failed conversions. Use -openlog option for opening the log file and see the entries at end of the text file.

Example: use your own single file modifying script and batch convert with gnmediahelper (use .exe, .bat, perl .pl, gnscript from https://www.gnmidi.com) to convert an input file to an output file

```
C:\> gnmediahelper -debug input\*.txt output\modifiedtext\  op:system gnscript.exe
modifytext.gnscript %input% %output%
```

Hint: use -debug to get more information about the system calls (it should not exit with error 1)
Hint: use -allfiles to convert non-media files if specifying only an input folder
Hint: create a user operation to integrate your own operation using the simple option and parameter syntax.

12 User operations

Notice: this feature requires a batch license except for the pre installed demo user operations after first start (op:mspaint, op:pixelize, op:scaledgreywithborder, op:louder)

Additional ffmpeg.exe or system operations can be defined in text file gnffmpeg.ini in your personal documents folder.
The command line parameters for a new command can be set and placeholders %input% %output% are filled automatically. Options and parameters can be defined so that the new command can use parameters in a comfortable way.

The standard operations could be overloaded by defining a user operation with the same name e.g. op:scale could be redefined if you have an individual way to implement that operation.
User operations can also be used in combi and batch operation.

Valid user operations are also listed in usage and examples if the user operation has added usage and example entries.

```
gnmediahelper -examples op:pixelize
```

Example: integrate your extern commandline tool as user operation into gnmediahelper

add into gnffmpeg.ini text file a new section:

```
[op:modifytext]
cmd=c:\myscripts\modifytext.gnscript %input% %output%
exe=gnscript.exe
outputfiletype=text
isviewer=no
ignoreexitcode=no
usage=op:modifytext input.txt output.txt
example1=gnmediahelper  myinput.txt myoutput.txt op:modifytext
example2=gnmediahelper c:\input\*.txt c:\output\*.txt op:modifytext
```

then you can call the user operation as batch operation

```
C:\> gnmediahelper -debug input\*.txt output\modifiedtext\  op:modifytext
```

13 Standard operations

Hint: %temp% is system variable that holds the path of the temporary folder.

op:open

```
usage: op:open                open result file by system
examples:
gnmediahelper "pink flowers.bmp" op:open
gnmediahelper "pink flowers.bmp" op:greyscale op:open
```

op:open is a viewer operation. The temporary result file will be kept for longer than the application runtime so that the viewer which might run longer than gnmediahelper and still can access it. The temporary file will be deleted at next program start.
op:open uses system to start an application that is registered for opening this file extension.
Use file explorer to register applications for opening a file with given file extension.

op:ffplay

```
usage: op:ffplay [startsecond]   open result file ffplay.exe (if existing)
```

```
gnmediahelper "cat leo.mp4" op:ffplay 0:35
gnmediahelper "Guenter Nagler - Cold as Ice.mp3" op:changevolume 20% op:ffplay
```

op:ffplay is a viewer operation. The temporary result file will be kept for longer than the application runtime so that the viewer which might run asynchronous can access it for longer. The temporary file will be deleted at next program start.

op:ffplay uses ffplay.exe to open a media file. ffplay.exe will be automatically searched in the folder from where ffmpeg is used and in the PATH.
If ffplay.exe can not be found then you could edit gnffmpeg.ini text file and add ffplay.exe=path\ffplay.exe as line below the ffmpeg.exe path line

Optional parameter startsecond can be a number (seconds) or a time (mm:ss). Default startsecond is 0 (start of media).

op:info

```
usage: op:info [append:file]               detailed info about file (default)
gnmediahelper "pink flowers.bmp" op:info
gnmediahelper "pink flowers.bmp"
gnmediahelper "pink flowers.bmp" op:scale 700 op:info
gnmediahelper c:\users\user\pictures op:info append:%temp%\info.txt
```

op:info displays the information that ffmpeg.exe prints about the given file. It usually contains (quick) duration, media size, format, bit rate. It could also contain error messages about the file (especially unknown format or invalid files).

op:verify

usage: op:verify [append:file] error info about file (default)
gnmediahelper "pink flowers.bmp" op:verify
gnmediahelper "pink flowers.bmp"
gnmediahelper c:\users\user\pictures op:verify append:%temp%\verify.txt

op:verify reads the input file and verifies if it is a valid media file that ffmpeg.exe can decode.
The output is
valid: filename
or
invalid: filename error messages
or
dangerous: filename seems to contain executable

Hint: using batch license whole folder tree can be verified and results written to the text file specified in parameter append.
Hint: it cannot verify if files can be read by other applications or viewers than ffmpeg.exe
Hint: it is very dangerous if a media file (e.g. filename.mp4) contains executable data because most Windows systems start the hidden executable (e.g. trojan) instead of play invalid file (Microsoft what the hell have you done???).

op:quickduration

```
usage: op:quickduration [-ms][-time] [append:file]        (might be inexact)
gnmediahelper "cat leo.mp4" op:quickduration
gnmediahelper "cat leo.mp4" op:quickduration -ms
gnmediahelper "Guenter Nagler - Cold as Ice.mp3" op:copypart 0:10 0:15
op:quickduration -time
gnmediahelper c:\users\user\videos op:quickduration append:%temp%\quickduration.txt
```

op:quickduration gets the duration information from the ffmpeg.exe info about the file.
The option -ms converts duration time to milliseconds. The option -time is default if -ms is not used.

The result will be displayed on screen and written to the output text file if output is specified and not minus (-).
The optional argument append file might be useful in batch operations. The information will be appended to the file so that at end all collected information of valid files can be found in the file.

op:quickbitrate

```
usage: op:quickbitrate [append:file]        (bitrate might change during file)
gnmediahelper "cat leo.mp4" op:quickbitrate
gnmediahelper c:\users\user\videos op:quickbitrate append:%temp%\quickbitrate.txt
```

op:quickbitrate gets the bitrate information from the op:info output. The bitrate used during the streams might bit different or change (variable bitrate).
For a video only the video bitrate is displayed. op:info also shows bitrate of audio streams.
The operation only delivers information for video and audio files.
The optional argument append file might be useful in batch operations. The information will be appended to the file so that at end all collected information of valid files can be found in the file.

op:duration

```
usage: op:duration [-ms][-time] [append:file]        (might take long time)
gnmediahelper "cat leo.mp4" op:duration
gnmediahelper "cat leo.mp4" op:duration -ms
gnmediahelper "Guenter Nagler - Cold as Ice.mp3" op:copypart 0:10 0:15 op:duration -
time
gnmediahelper c:\users\user\videos op:duration append:%temp%\duration.txt
```

op:quickduration gets the duration information by analysing the video or audio file. This can take long time for big media files.
The option -ms converts duration time to milliseconds. The option -time is default if -ms is not used.
The result will be displayed on screen and written to the output text file if output is specified and not minus (-).
The optional argument append file might be useful in batch operations. The information will be appended to the file so that at end all collected information of valid files can be found in the file.

op:size

```
usage: op:size [append:file]
gnmediahelper "pink flowers.bmp" op:size
gnmediahelper "pink flowers.bmp" op:scale height:700 op:size
gnmediahelper c:\users\user\pictures op:size append:%temp%\size.txt
```

op:size gets the media size (width x height) in pixels from the ffmpeg info. Only images and videos have this information.
The result will be displayed on screen and written to the output text file if output is specified and not minus (-).
The optional argument append file might be useful in batch operations. The information will be appended to the file so that at end all collected information of valid files can be found in the file.

op:getpixel

```
usage: op:getpixel x y [-append:file]
gnmediahelper "pink flowers.bmp" op:getpixel 0 0
gnmediahelper "cat leo.mp4" op:getpixel x:100 y:150
gnmediahelper c:\users\user\pictures op:getpixel 0 0
append:%temp%\gettopleftpixelrgb.txt
```

op:getpixel converts an image to format PPM or extracts first frame of a video to format PPM and reads the RGB value from the pixel at x,y .
Hint: if the input file has already image format PPM (.ppm) or PGM (.pgm) then conversion is not

necessary.
The result will be displayed on screen and written to the output text file if output is specified and not minus (-).
e.g. file pink flowers.bmp pixel at 0,0 has color RGB(85,128,23) #558017
The option append file might be useful in batch operations. The information will be appended to the file so that at end all collected information of valid files can be found in the file.

op:convert

```
usage: op:convert [-videobitrate=kb] [-audiobitrate=kb] [-fileext=.ext]
gnmediahelper "pink flowers.bmp" "pink flowers.png op:convert
gnmediahelper "pink flowers.bmp" *.png op:convert
gnmediahelper "cat leo.mp4" "cat leo.mp3 op:convert audiobitrate:48
gnmediahelper "cat leo.mp4" - op:convert fileext:.mp3 op:open
gnmediahelper c:\users\user\pictures\*.jpg %temp%\resultpictures\*.png op:convert
```

op:convert converts between different file formats. Optional the bitrate for existing audio and video can be changed. The bitrate must be specified as number in kb/s. ffmpeg.exe might use an other bitrate for a video audio stream and the bitrate can change during the output file. The fileext option should be used if no output filename is specified or if in a combination of operations the convert operation should produce a certain format defined by the file extension (e.g. if input is a video then it could be converted to a video or audio format, op:convert without given fileext would in this case use file extension of output or the input file extension by default).

Hint: option -fileext=.ext can be used with every operation. A given fileext option is ignored when the operation is a view operation or if it is the last operation that is no view operation in a combination of operations (which uses automatically output file extension if given).

This command is for converting between media types:
video to video
video to audio
audio to audio
audio to video (result only contains audio track)
image to image
image to video (result contains one frame, result can be viewed with ffplay.exe, not with media player)

op:copypart

```
usage: op:copypart startms endms
gnmediahelper "Guenter Nagler - Cold as Ice.mp3" "Guenter Nagler - Cold as Ice.wav
op:copypart 0 0:10
gnmediahelper "cat leo.mp4" result.mp4 op:copypart startms:10000 endms:0:15
gnmediahelper "cat leo.mp4" op:copypart startms:10000 endms:0:15.500
op:quickduration
gnmediahelper c:\users\user\videos\*.mp4 %temp%\shortvideos\*.mp4 op:copypart
startms:0 endms:0:05
```

op:copypart copies a part of a video or audio between start and end time to result.
The start and end time can be specified as milliseconds or with time format mm:ss[.milliseconds] (e.g. 1:15, 3:22.400).
Both times must be specified. Start time must be smaller than end time.
Hint: Specify a high end time e.g. 60:00 to copy from start time till end

op:crop

```
usage: op:crop x y width height
gnmediahelper "pink flowers.bmp" "pink flowers.png op:crop 0 20 300 200
gnmediahelper "pink flowers.bmp" *.png op:crop 0 0 height:100
gnmediahelper "cat leo.mp4" "cat leo.mov op:crop x=50 width=200 height=150
```

```
gnmediahelper c:\users\user\pictures\*.jpg %temp%\resultpictures\*.png op:crop 30 30
```

op:crop copies a rectangular area from a picture or video.

op:scale

```
usage: op:scale width[%] height[%]
gnmediahelper "pink flowers.bmp" "pink flowers.png op:scale 300% 200
gnmediahelper "pink flowers.bmp" *.png op:scale height:100 width=50%
gnmediahelper "cat leo.mp4" "cat leo.mov op:scale width=400
gnmediahelper c:\users\user\pictures\*.jpg %temp%\resultpictures\*.png op:scale
height=600
```

op:scale changes the width and height of an image or video. The width and height can be specified by pixel or percent.
if only width or height is specified then the other parameter will be calculated from the aspect ratio of the original file.
Some media formats do not accept very small sizes and only support even pixel size so that it produces an error message if the width or height is not usable.
Hint: This operation does not scale down to a size smaller than 10 pixel.
Hint: For .mp4 and .mov this operation automatically rounds sizes up to next even pixel size.

op:rotate

```
usage: op:rotate degrees         (90 or -90 or 180 degrees)
gnmediahelper "pink flowers.bmp" "pink flowers.png op:rotate -90
gnmediahelper "pink flowers.bmp" *.png op:rotate degrees=180
gnmediahelper "cat leo.mp4" "cat leo.mov op:rotate 90
gnmediahelper c:\users\user\pictures\*.jpg %temp%\resultpictures\*.png op:rotate +90
```

op:rotate rotates an image or video by 90, 180, -90 degrees. Degree values +360 or -360 are allowed and have same meaning (e.g. -90 is same as 270).
Other degrees are not supported. 90 rotates to the right. -90 rotates to the left.

op:flip

```
usage: op:flip direction         (h or v)
gnmediahelper "pink flowers.bmp" "pink flowers.png op:flip v
gnmediahelper "pink flowers.bmp" *.png op:flip direction=h
gnmediahelper "cat leo.mp4" "cat leo.mov op:flip horizontal
gnmediahelper c:\users\user\pictures\*.jpg %temp%\resultpictures\*.png op:flip -
direction=vertical
```

op:flip mirrors an image or video by a given side
h horizontal
v vertical

op:addborder

```
usage: op:addborder borderwidth bordercolor
gnmediahelper "pink flowers.bmp" "pink flowers.png op:addborder 10 white
gnmediahelper "pink flowers.bmp" *.png op:addborder borderwidth:30
gnmediahelper "cat leo.mp4" "cat leo.mov op:addborder bordercolor=yellow
borderwidth=20
gnmediahelper c:\users\user\pictures\*.jpg %temp%\resultpictures\*.png op:addborder
-bordercolor=black 30
```

op:addborder adds a border with given pixel size and given border color around the image or video.
Hint: depending on the destination format the color could be modified by image/video compression near the border and picture (e.g. jpeg or mpeg is no loss less compression for pixels).

op:copyframeattime

```
usage: op:copyframeattime ms
gnmediahelper "Guenter Nagler - Cold as Ice.mp3" "Guenter Nagler - Cold as Ice.png
op:copyframeattime 0:10
gnmediahelper "cat leo.mp4" result.bmp op:copyframeattime ms:10000
gnmediahelper c:\users\user\videos\*.mp4 %temp%\firstimage\*.jpg op:copyframeattime
ms:0
```

op:copyframeattime copies a single frame image from a video at given time in milliseconds or time (mm:ss).
It can take more time to find and extract the image near the end of a big video.
The output file extension defines the output format of the image (e.g. .jpg for JPEG format, .bmp for BMP format, .png for PNG format ...).
Beginning of video has time 0.
An error occurs if the time is outside of the video duration.

op:copyframeatframenumber

```
usage: op:copyframeatframenumber framenr
gnmediahelper "Guenter Nagler - Cold as Ice.mp3" "Guenter Nagler - Cold as Ice.png
op:copyframeatframenumber 10000
gnmediahelper c:\users\user\videos\*.mp4 %temp%\firstimage\*.jpg
op:copyframeatframenumber 0
```

op:copyframeatframenumber copies a single frame image from a video with given frame index.
It can take more time to find and extract the image near the end of a big video.
The output file extension defines the output format of the image (e.g. .jpg for JPEG format, .bmp for BMP format, .png for PNG format ...).
Beginning of video has frame index 0.
An error occurs if the frame index is not in the video.

op:scaleaspect

```
usage: op:scaleaspect width[%] height[%] padcolor
gnmediahelper "pink flowers.bmp" "pink flowers.png op:scaleaspect 1000 1000 gray
gnmediahelper "pink flowers.bmp" *.png op:scaleaspect padcolor:black height:800
width=150%
gnmediahelper "cat leo.mp4" "cat leo.mov op:scaleaspect width=400
gnmediahelper c:\users\user\pictures\*.jpg %temp%\resultpictures\*.png
op:scaleaspect height=640 width=480 padcolor=red
```

op:scaleaspect changes the size of an image or video to a given width and height by keeping the aspect of the original image/video and filling the remaining part with padcolor.
Width and height may be specified as pixel or percent.
If only width or height will be specified then the other size will be calculated by the aspect ratio of the original image/video. In this case no padding will be necessary.
Hint: depending on the destination format the padcolor could be modified by image/video compression near the picture content (e.g. jpeg or mpeg is no loss less compression for pixels).
Default padcolor is black.

op:copythumbnail

```
usage: op:copythumbnail ms width[%] height[%]  padcolor
gnmediahelper "cat leo.mp4" "pink flowers.png op:copythumbnail 0:10 320 240 gray
gnmediahelper "cat leo.mp4" *.png op:copythumbnail ms=0 padcolor:black height:800
width=150%
gnmediahelper "cat leo.mp4" "cat leo.mov op:copythumbnail 0:35 width=400
gnmediahelper c:\users\user\videos\*.* %temp%\thumbnails\*.png op:copythumbnail 0
height=480 width=640 padcolor=black
```

op:copythumbnail copies an image from a video at given time. The image will be scaled to given width and height by keeping aspect ration of original image and padding with padcolor.
Width and height may be specified as pixel or percent.
If only width or height will be specified (e.g. width=-1) then the other size will be calculated by the aspect ratio of the original image/video. In this case no padding will be necessary.
Default padcolor is black.
Hint: depending on the destination format the padcolor could be modified by image/video compression near the picture content (e.g. jpeg or mpeg is no loss less compression for pixels).

op:reverse

```
usage: op:reverse
gnmediahelper "cat leo.mp4" reverse.mp4 op:reverse
gnmediahelper "Guenter Nagler - Cold as Ice.mp3" reverse.mp3 op:reverse
gnmediahelper "Guenter Nagler - Cold as Ice.mp3" op:reverse op:reverse op:open
gnmediahelper c:\users\user\music\*.mp3 %temp%\reverse\*.mp3 op:reverse
```

op:reverse reverses an audio or video (from back to start). Audio and video tracks are reversed if existing.
Hint: with reversing twice you can test that the audio/video is again correct (but file does not need to be identical encoded as original file).

op:concatvideo

```
usage: op:concatvideo [listfile] [inputfiles]     input must be a text list file
with input files
gnmediahelper  videolist.txt output.mp4 op:concatvideo
gnmediahelper "cat leo.mp4" op:concatvideo second.mp4 third.mp4
```

op:concatvideo merges two or more video files to a merged result (plays the parts one after one).
Hint: it is suggested to use same width/height in all input files to avoid that input videos are scaled without your control.
The operation can be done with a list text file as input file (the text file contains one video filename or path per line) or for few files input file can be the first part and parameters of the operation specify the other parts.
Hint: This operation does not work audio files or images.

op:concataudio

```
usage: op:concataudio [listfile] [inputfiles]     input must be a text list file
with input files
gnmediahelper  medley.txt output.mp3 op:concataudio
gnmediahelper "Guenter Nagler - Cold as Ice.mp3" output.mp3 op:concataudio "Guenter
Nagler - Am I Really Human.mp3"
```

op:concataudio merges two or more audio files to a medley (plays audio in series).
The operation can be done with a list text file as input file (the text file contains one audio filename or path per line) or for few files input file can be the first part and parameters of the operation specify the other parts.
Hint: This operation does not work video files or images.

op:brightness

```
usage: op:brightness percent         0..200%
gnmediahelper "pink flowers.bmp" "pink flowers.png op:brightness 120%
gnmediahelper "pink flowers.bmp" *.png op:brightness percent=120
gnmediahelper "cat leo.mp4" "cat leo.mov op:brightness 80
gnmediahelper c:\users\user\pictures\*.jpg %temp%\resultpictures\*.png op:brightness
130
```

op:brightness changes the brightness of an image or video by a percentage of the brightness used in

original file.

op:saturation

```
usage: op:saturation percent         0..300%
gnmediahelper "pink flowers.bmp" "pink flowers.png op:saturation 120%
gnmediahelper "pink flowers.bmp" *.png op:saturation percent=120
gnmediahelper "cat leo.mp4" "cat leo.mov op:saturation 80
gnmediahelper c:\users\user\pictures\*.jpg %temp%\resultpictures\*.png op:saturation
230
```

op:saturation changes the saturation (color intensity) of an image or video by a percentage value.

op:contrast

```
usage: op:contrast percent         -200..200%
gnmediahelper "pink flowers.bmp" "pink flowers.png op:contrast 120
gnmediahelper "pink flowers.bmp" *.png op:contrast percent=-120
gnmediahelper "cat leo.mp4" "cat leo.mov op:contrast 80
gnmediahelper c:\users\user\pictures\*.jpg %temp%\resultpictures\*.png op:contrast
130
```

op:contrast changes the contrast of an image or video by a percentage value.

op:greyscale

```
usage: op:greyscale
gnmediahelper "pink flowers.bmp" op:greyscale op:open
gnmediahelper "pink flowers.bmp" *.png op:greyscale
gnmediahelper "cat leo.mp4" "cat leo.mov op:greyscale
gnmediahelper c:\users\user\pictures\*.jpg %temp%\resultpictures\*.png op:greyscale
```

op:greyscale changes the colors of an image or video to grey scale.
Hint: The resulting image file might be encoded using a grey color space depending on the used output file extension (only stores grey intensities).
Using the grey encoded image with color operations (e.g. addborder bordercolor:red) might keep the image gray after the operation. This can be prevented by forcing to product RGB colored temporary result files. /fileext=.ppm always produces a RGB file even if the content is grey or black white only.
e.g.

```
gnmediahelper "pink flowers.bmp" *.png op:greyscale /fileext:.ppm op:addborder 5
yellow
```

op:changevolume

```
usage: op:changevolume expr      (150%  or -5dB)
gnmediahelper "Guenter Nagler - Cold as Ice.mp3" op:changevolume 50% op:open
gnmediahelper "cat leo.mp4" result.mp4 op:changevolume +3dB
gnmediahelper "cat leo.mp4" result.mp4 op:changevolume -db:-3
gnmediahelper "cat leo.mp4" result.mp4 op:changevolume -percent:-20
gnmediahelper c:\users\user\music\*.mp3 %temp%\silent\*.mp3 op:changevolume 80%
```

op:changevolume changes the volume of an audio or video by percentage or by decibels.
Hint: this operation does not set a certain volume, only change it louder or more silent.

op:fadein

```
usage: op:fadein seconds
gnmediahelper "Guenter Nagler - Cold as Ice.mp3" op:fadein seconds:5 op:open
gnmediahelper "cat leo.mp4" result.mp4 op:fadein milliseconds:5500
gnmediahelper c:\users\user\music\*.mp3 %temp%\fadein\*.mp3 op:fadein seconds:10
```

op:fadein fades in audio and video at beginning of the media for given seconds.

Audio increases volume.
Video increases brightness.

op:fadeout

```
usage: op:fadeout seconds
gnmediahelper "Guenter Nagler - Cold as Ice.mp3" op:fadeout seconds:5 op:open
gnmediahelper "cat leo.mp4" result.mp4 op:fadeout milliseconds:5500
gnmediahelper c:\users\user\music\*.mp3 %temp%\fadein\*.mp3 op:fadeout seconds:10
```

op:fadeout fades out audio and video at end of the media for given seconds.
Audio decreases volume.
Video decreases brightness.
Hint: This operation requires to analyse the exact duration first which can take much time for big media files.

op:system

```
usage: op:system [-isviewer] exepath params      (use %input% and %output%)
gnmediahelper "pink flowers.bmp" op:system -isviewer mspaint.exe %input%
gnmediahelper "cat leo.mp4" op:system c:\utils\videoconverter.exe -options %input%
%output% op:greyscale op:open
```

op:system calls an installed application that can be started by command line and optional parameters.
Option -isviewer must be specified as first option if the application will use the (temporary) input file when gnmediahelper already exited.
gnmediahelper would automatically delete temporary results at exit so that the application could not use the file anymore.
By using -isviewer the deleting of a temporary result will be done later at next start of gnmediahelper.
The application will be searched in gnmediahelper.exe folder and in folders specified by the system PATH variable if no path is specified.
mspaint.exe will probably be found in system path.
c:\windows\system32\mspaint.exe will be directly used from this folder (if existing and Windows system permits).

14 Defining user operations

Notice: this feature requires a batch license

User operations can be defined in gnffmpeg.ini text file in your personal folder.
After first start of gnmediahelper.exe this file should be created and should contain demo user tools op:mspaint and op:pixelize definition.

The **operation name** inside [...] is the user operation name and must begin with op:
The operation name only allows using letters, digits and underscore character _
The key value assignments following the section header [...] define the operation. The order is not important but must be before next [...] section starts.

cmd=...
contains the command line without the ffmpeg.exe or system exe.
placeholder **%input%** will be replaced by the used input file (already quoted if the path or file name contains spaces e.g. "cat leo.mp4"). %input% is always required.
placeholder **%output%** will be replaced by the used output file (already quoted if the path or file name contains spaces). %output% might be optional if the tool is a viewer.
input and output will often be temporary file names.
Imporant: if you need a percent character % in the command line then you must use **%%** a single

percent character starts a placeholder %....%
The cmd parameter can contain option placeholders %option:name=defaultvalue% and parameter placeholders %param1:name=defaultvalue% (see below about options and parameters)

exe=
exe is by default the ffmpeg.exe tool that was set once (value is always empty or the exe key not assigned).
exe can be "combi" if the user operation defines a combination of other operations.
exe can be an other executable (.exe, .bat, ...) if an other application is to use (e.g. mspaint.exe).

isviewer=no
is a boolean value (1, 0, true, false, yes, no) that tells that the operation does not produce an output (usually a viewer). Default is no.
Hint: viewer operations should better be used as last operation. gnmediahelper copies the input file to a temporary output file for the case that after following operations continue to process.

ignoreexitcode=no
is a boolean value (1, 0, true, false, yes, no) that tells that the specified executable exit code should be ignored (maybe does not exit with 0 for success and other for error).

outputfiletype=inputfiltype
is optional output ffile type(default is inputfiletype) and can have the values: inputfiletype, text, audio, video, image (inputfiletype means that the operation produces same ffile typeas the input file has)

usage=...
you can add usage line to describe the syntax of your user operation. Use [-optionname=defaultvalue] or paramname=defaultvalue for available options and parameters.
e.g.
```
usage=op:myoperation [-padcolor=black] width height
```

examples1=...
you can add exampleNN lines to give some examples for using the user operation. First example has number 1.
e.g.
```
example1=op:myoperation 320 200
example2=op:myoperation -padcolor=#ff0000 width:640 height:-1
```

using **options** in the cmd definition:
an option is always named and can optionally have a default value. A default value will be used if the operation arguments does not set the option value.

`%option:name%` (option has no default value and must be specified in the operation arguments)
`%option:name=defaultvalue%`
`%option:name=%` (empty text is default value of this option)

option usage in the command line:
```
name:value
name=value
-name=value
-name:value
/name=value
/name:value
```
Hint: it is not important where the option is used in the operation arguments since it always is referred by name

option example:
if a user operation op:fill uses a ffill colorthen this could be defined as an optional option with default color

```
[op:fill]
cmd=... %option:fillcolor=yellow% ...
```
and the option can be used in your command line
`gnmediahelper input.png output.png op:fill 0 0 320 200` (uses default fillcolor yellow)
`gnmediahelper input.png output.png op:fill -fillcolor=black 0 0 320 200` (uses fillcolor black)

Hint: cmd might contain the same option several times (default value should be same at all locations)

using parameters in the cmd definition:
a parameter can be specified by name or the nth unnamed argument is used.

`%paramNN%` the NNth unnamed argument (e.g. %param1%) has no default value and must be specified in operation args
`%paramNN=defaultvalue%` the NNth unnamed argument can be specified as NNth unnamed operation argument (e.g. first argument) or defaultvalue is used instead
`%paramNN:name%` the NNth unnamed argument can be used as NNth parameter or a named parameter (same usage as named options) can be specified anywhere in the arguments. An error occurs if NNth argument is missing and also no option with this name is used anywhere
`%paramNN:name=defaultvalue%` the NNth unnamed argument can be used as NNth parameter or a named parameter (same usage as named options) can be specified anywhere in the arguments. The defaultvalue will be used if NNth argument is missing and also no option with this name is used anywhere

Hint: param1 is the first unnamed argument in the operation arguments
Hint: when mixing named parameters and unnamed parameters the order of parameters must be kept and default values by missing parameters can only be used in the last arguments (not in middle).

parameter example:
an operation op:myscale requires parameters x, y, width and optional height
```
[op:myscale]
cmd=... %param1:x% %param2:y% %param3:width% %param4:height=-1% ...
```

the parameters can be used in command line
`gnmediahelper input.png output.png op:myscale 0 0 320 200` (all parameters are used without name)
`gnmediahelper input.png output.png op:myscale x:0 0 320` (param1 is used with name at correct parameter position, param4 is missing and default value -1 will be used
`gnmediahelper input.png output.png op:myscale width:320 height:200 -x=0 -y=0` (using all parameters by name)
Hint: the named parameter x is used at correct position 1 and has value 0, the second parameter is used unnamed with value 0 and position refers to y, the third parameter is used unnamed with value 320 and position refers to width.

`gnmediahelper input.bmp output.bmp op:myscale 0 x:30 400 500` is wrong use because first parameter 0 already refers to param x and x:30 too, an error occurs that tells about parameter x used twice
Hint: better use parameters at correct position (with or without name) or use all parameters only with names in any order.

placeholders:
The %...% placeholders in the cmd definition will be replaced by values from user arguments or default value of an option.
%input% and %output% are automatically quoted if they contain spaces. It must be done self when an other option or parameter requires quotes

e.g.
```
[op:setcaption]
cmd= ...caption="%param1:text"...
gnmediahelper input.png output.png op:setcaption "text:hello world"
```

%% will be automatically replaced by single %
Hint: for two adjacent placeholders %option:x%%option:y% the %% in middle is not a percent character but inside or outside a placeholder it is a percent character e.g. %option:percent=120%%%

Hint: use double quotes if an argument contains spaces e.g. a path name "pink flowers.bmp"

Hint: using named options is to prefer. When using pparametersin a user operation then

example user operation calling an executable:
```
[op:mspaint]
cmd=%input%
exe=mspaint.exe
isviewer=yes
ignoreexitcode=yes
usage=op:mspaint image.ext
example1=gnmediahelper image.bmp op:scale 150% op:addborder 5 red op:mspaint
example2=gnmediahelper image.png op:mspaint op:scale 150% op:mspaint op:addborder 5
red op:mspaint
```

the executable mspaint.exe will be searched in the folders from the PATH variable.
the user operation is a viewer (produces no output) and the exit code is not important.

Hint: by default gnmediahelper assumes that a user operation produces same file type (same file extension) as input had. if output file extension is available this will be used.
e.g. `gnmediahelper input.wmv op:myoperation op:open` assumes to produce a .wmv video
e.g. `gnmediahelper input.wmv output.jpg op:myoperation` assumes to produce a .jpg image

example user operation using ffmpeg.exe:
```
[op:pixelize]
cmd=-i %input% -vf
"[in][blurin]feedback=x=%param1:x=0%:y=%param2:y=0%:w=%param3:width=10000%:h=%param4
:height=10000%[out][blurout];[blurout]pixelize=width=%option:blocksize=16%:height=%o
ption:blocksize=16%[blurin]" %output%
usage=op:pixelize [-blocksize=16] x y width height
example1=op:pixelize 100 150 width=320 height=200
example2=op:pixelize width:200 height:200 x:0 y:50 blocksize=8
```

Hint: op:pixelize demo user operation is available in gnffmpeg.ini after first start.
This operation allows to specify a rectangle area (pixels) of an image or video and optional the block size of pixelation. The area will become unsharp by pixelation. Each block gets the average color of the area.

Hint: use a high value for width and height to set the area till right and bottom side

combi user operation definition

```
[op:operationname]
cmd=op:operation1 ... op:operation2 ...
exe=combi
```

the cmd contains some operations with parameters
The cmd can use %option% and %param...% which can be used as arguments in the combi operation

example for a combi user operation definition

```
[op:scaledgreywithborder]
cmd=op:scale %option:width=200%%% op:greyscale op:addborder %option:borderwidth=5%
%option:bordercolor=yellow%
exe=combi
isviewer=false
usage=op:scaledgreywithborder [-width=200%] [-borderwidth=5] [-bordercolor=yellow]
example1=gnmediahelper "pink flowers.bmp" op:scaledgreywithborder op:open
example2=gnmediahelper "cat leo.mp4" videowithborder.mp4 op:scaledgreywithborder -
width=640 borderwidth:5 bordercolor=white
```

Important

15 Derive a standard operation by a user operation definition (for experts only)

Notice: this feature requires a license

if gnffmpeg.ini defines a user operation with a name of an existing standard operation then this operation replaces the standard operation.
This can be used to define an operation with your own optimizations or options. This behaves like a user operation and options and parameters must be defined and used in command.

16 Derive a standard operation using the options and parameters of the original operation definition with a modified commandline (for experts only)

Notice: this feature requires a license

implementations of the standard operations use command line templates with **placeholders** formatted like %placeholdername%. %% is the percent character self.
The **templatecmd** for an operation is configurable by setting in gnffmpeg.ini text file (in personal documents folder).

The priority of finding a templatecmd is (from high to low):
1. use templatecmd from section {operationname}_mediaformat_{mediaformatname}
2. use templatecmd from section {operationname}_mediatype_{mediatypename}
3. use templatecmd from section {operationname}_general
4. use internal templatecmd suggested by gnmediahelper program as default

{operationname} is the operation name e.g. op:scale
{mediaformatname} is the usually the input file extension (without dot), and jpg for .jpg and .jpeg and tiff for .tif and .tiff
{mediatypename} is video or audio or image or text from input

The command line template must be stored as field key **templatecmd** and must contain all required placeholders. The application would tell if a required placeholder is missing. %input% is always required.

```
[op:greyscale_general]
templatecmd=-i %input% -vf "format=gray,eq=brightness=0.2" %output%
```

this would convert to greyscale and automatically make it brighter for all video/image input formats.

```
[op:greyscale_png]
templatecmd=-i %input% -vf "format=gray,eq=brightness=0.2" %output%
```

this would convert to greyscale and automatically make it brighter for png images only. Other files will use other template depending of the priority list. This allows to use an alternate command line for different input file types (e.g. use a compression option for jpg format).

```
[op:greyscale_mediatype_image]
templatecmd=-i %input% -vf "format=gray,eq=brightness=0.2" %output%
```

this would convert to greyscale and automatically make it brighter for images only. Video files will use other template depending of the priority list.

Hint: defining different derived op:open variants could be used to open files depending on media type (video/image/audio) or even file type (e.g. jpg, tif, .wav) with different applications. The standard implementation lets Windows system choose the application for opening a file.

Hint: if you want to add new options to a command you must define a new user command

Deriving user operations
This deriving method also works for user operations.
Define a user operation and optionally templatecmd settings for different mediatype or mediaformat (user operation op:pixelize could use different templatecmd implemented differently for .jpg and .mp4). The option and parameter definitions are always used from user operation definition cmd.

16.1 Template names and placeholders used by standard operations (experts only)

This table lists the placeholders used by the standard implementation. For deriving one of these commands you also need to have an idea how the ffmpeg.exe arguments must be used to do the operation with your modification ideas. See ffmpeg.exe -help and FFmpeg documentation pages about this. This document does not help you with this task.

op:open

template cmd "op:open" uses placeholder %input%

op:ffplay

template cmd "op:ffplay" uses placeholders:
%ffplay%
%input%
%startsecond%

op:info (no template available)
op:quickduration (no template available, uses op:info)
op:quickbitrate (no template available, uses op:info)
op:duration (no template available)
op:size (no template available, uses op:info)
op:getpixel (no template available, uses op:convert)

op:convert

template cmd "op:convert" uses placeholders
%input%
%videobitratestart%
%videobitratekb%
%videobitrateend%
%audiobitratestart%
%audiobitratekb%
%audiobitrateend%
%output%
Hint: the part including %videobitratestart% till %videobitrateend% will be removed if option videobitrate is empty else only these placeholders are removed and %videobitratekb% is a number (kb/s).
Hint: the part including %audiobitratestart% till %audiobitrateend% will be removed if option audiobitrate is empty else only these placeholders are removed and %audiobitratekb% is a number (kb/s).

op:copypart

template cmd "op:copypart" uses placeholders:
%start%
%input%
%duration%
%output%
Hint: %start% and %duration% are time strings like mm:ss or mm:ss.milliseconds

op:crop

template cmd "op:crop" uses placeholders:
%input%
%cropw%
%croph%
%cropx%
%cropy%
%output%

op:scale

template cmd "op:scale" uses placeholders:
%input%
%scalewidth%
%scaleheight%
%output%

op:rotate

template cmd "op:rotate90" uses placeholders:
%input%
%output%

template cmd "op:rotate-90" uses placeholders:
%input%
%output%

Hint: rotate90 is used for rotation by 90 degrees clockwise
Hint: rotate-90 is used for rotation by 90 degrees counterclockwise
Hint: for rotation by 180 degrees the rotate90 is done twice.

op:flip

template cmd "op:flipv" uses placeholders:
%input%
%output%

template cmd "op:fliph" uses placeholders:
%input%
%output%

Hint: flipv is used for flipping vertically (top/bottom)
Hint: fliph is used for flipping horizontally (left/right)

op:addborder

template cmd "op:addborder" uses placeholders:
%input%
%newwidth%
%newheight%
%padx%
%pady%
%padcolor%
%output%

op:copyframeattime

template cmd "op:copyframeattime" uses placeholders:
%input%
%start%
%output%

op:copyframeatframenumber

template cmd "op:copyframeatframenumber" uses placeholders:
%input%
%framenumber%
%output%

op:scaleaspect (has no template, it uses operations scale and addborder)

op:copythumbnail (has no template, it uses operation copyframeattime and scaleaspect)

op:reverse

template cmd "op:reverse" uses placeholders:
%input%
%output%

op:concatvideo

template cmd "op:createts" for each input file uses placeholders:
%input%
%output%
Hint: output is a temporary .ts file

template cmd "op:concatts" uses placeholders:
%tslist%
%output%"
Hint: tslist is a temporary text file that contains ts of all input files. the tslist path is not quoted.

op:concataudio

template cmd "op:concataudio" uses placeholders:
%inputlistfile%
%output%

op:brightness

template cmd "op:brightness" uses placeholders:
%input%
%brightnessfactor%
%output%

Hint: brightnessfactor is a value between -1.0 and +1.0 (negative gets darker)

op:saturation

template cmd "op:saturation" uses placeholders:
%input%
%saturationfactor%
%output%
Hint: saturationfactor is a value between 0.0 and 3.0 where 1.0 is 100%

op:contrast

template cmd "op:contrast" uses placeholders:
%input%
%contrastfactor%
%output%

op:greyscale

template cmd "op:greyscale" uses placeholders:
%input%
%output%

op:changevolume

template cmd "op:changevolume" uses placeholders:
%input%
%volumeexpression%
%output%

Hint: volumeexpression can be +3dB (for 3 decibel higher) or -0.10 (for 10 percent more silent). The volumeexpression is not quoted.

op:fadein

template cmd "op:fadein" uses placeholders:
%input%
%seconds%
%output%

op:fadeout

template cmd "op:fadeout" uses placeholders:

%input%
%seconds%
%output%

op:system (has no templatecmd)

"op:system" uses placeholders:
%input%
%output%

16.2 Supported mediatypes by file extension

gnmediahelper only knows the supported mediatype for most used file formats (e.g. .jpg, .avi, .mp3, .mp4 ...) and not the full list of formats supported by ffmpeg.exe

More information about this can be added to gnffmpeg.ini text file in a section
[SupportedMediaTypes]
.ext=mediatype

mediatypes can be one or more media type names (video, audio, image, text) separated by comma

```
[SupportedMediaTypes]
.jpg=image
.avi=video,audio
```

Usually it is enough to specify a file extension known by ffmpeg.exe in the output filename (even if gnmediahelper does not know this file extension).
When omitting the output file in command line or option fileext in an operation then it could occur that gnmediahelper needs to choose a result file extension self which might be difficult without having the mediatype information about an unknown file extension. The program could warn about using default file extension .mp4 for an operation when no output file extension for the operation is available.
Using option fileext:.ext in an operation also helps.

mediatype

16.3 Mediaformat from file extension

In most cases the mediaformat is identical to the file extension (without dot). This is also the default that gnmediahelper assumes.
In few cases certain file extension variant is still containing the same data format e.g. .jpeg is jpg same as .jpg

in text file gnffmpeg.ini such differences could be added to section MediaFormatFromFileExtension:

```
[MediaFormatFromFileExtension]
.jpeg=jpg
```

This mediaformat is then used in a derived standard command template special for the given mediaformat.