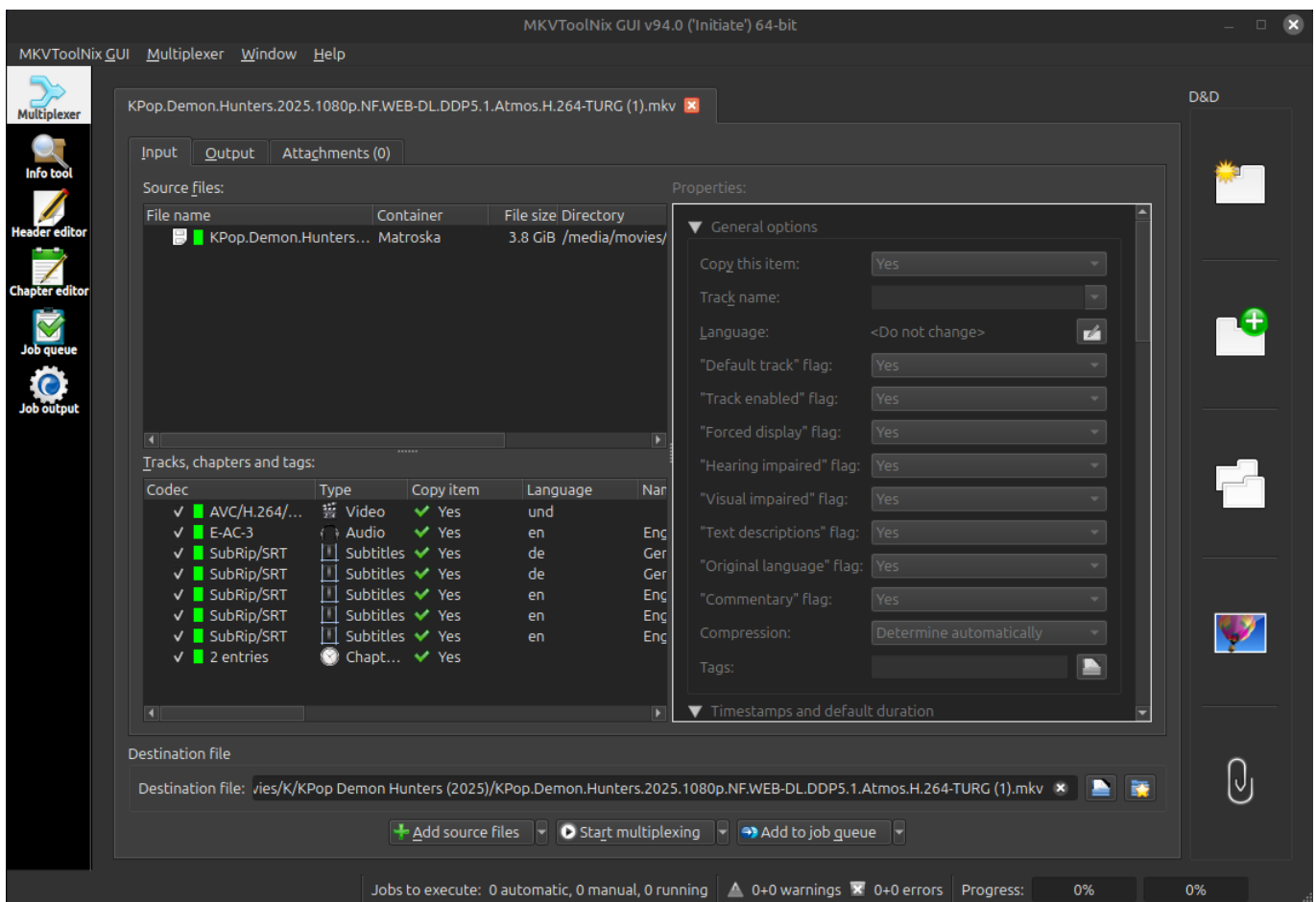


Merge/Combine/Join Video

GUI tool for mkv files. Older version is in default debian/ubuntu repositories, but latest can be installed in Mint 22 Wilma using the following, afterwards associate mkv files with mkvtoolnix:

```
sudo wget -O /etc/apt/keyrings/gpg-pub-moritzbunkus.gpg
https://mkvtoolnix.download/gpg-pub-moritzbunkus.gpg

cat <<EOF | sudo tee /etc/apt/sources.list.d/mkvtoolnix.list
deb [arch=amd64 signed-by=/etc/apt/keyrings/gpg-pub-moritzbunkus.gpg]
https://mkvtoolnix.download/linuxmint/ wilma main
deb-src [arch=amd64 signed-by=/etc/apt/keyrings/gpg-pub-moritzbunkus.gpg]
https://mkvtoolnix.download/linuxmint/ wilma main
EOF
sudo apt update
sudo apt install mkvtoolnix mkvtoolnix-gui
```



MPG/MPEG files:

```
sudo apt-get install mpgtx
mpgjoin vid1.mpg vid2.mpg vid3.mpg -o combinedvideo.mpg
```

WMV/MP4 files (same codec/properties):

```
ls -la > list.txt
#edit list and sort/format lines as
# file 'video1.wmv'
# file 'video2.wmv'
# file 'video3.wmv'

#or

ls -l *.wmv | sort -n | sed "s/\(.*\) /file '\1'/" > list.txt

ffmpeg -f concat -safe 0 -i list.txt -c copy outputfilename.wmv

#or

ffmpeg -f concat -safe 0 -i video1.wmv video2.wmv video3.wmv -c copy
outputfilename.wmv
```

<https://stackoverflow.com/questions/49371422/how-to-merge-two-videos-without-re-encoding/49373401#49373401>

Script for multiple conversions based on filename pattern. Converts all related files that have a number at the end and start with 1:

```
#!/bin/bash
SAVEIFS=$IFS
IFS=$(echo -en "\n\b")
for f in *1.wmv
do
    shortfile=${f::-5}
    echo "file "$shortfile"1.wmv" > list.txt
    echo "file "$shortfile"2.wmv" >> list.txt
    echo "file "$shortfile"3.wmv" >> list.txt
    ffmpeg -f concat -safe 0 -i list.txt -c copy $shortfile.wmv
    rm list.txt
done
IFS=$SAVEIFS
```

Script to merge files:

[video_merge.sh](#)

```
#!/bin/bash

# Check if a parameter is provided
if [ -z "$1" ]; then
    echo "Usage: "$(basename $0)" <files>"
    exit 1
fi

TMPFILE=$(mktemp)
```

```

for input in "$@";
do
    if [ -f "$input" ]; then
        if [ "$container" == "" ]; then
            container="${input##*}"
        fi
        if [ "$container" != "${input##*}" ]; then
            echo "Files need to be in same format to merge!"
            echo "Abandoning merge"
            if [ -f "$TMPFILE" ]; then
                rm $TMPFILE
            fi
            echo;read -rsn1 -p "Press any key to continue . . .";echo
            exit
        fi
        echo "file '$input'" >> $TMPFILE
        container="${input##*}"
        filename=${input%.*}
    fi
done

output="$filename.mergedprogress.$container"
sort -n -o $TMPFILE"_sort" $TMPFILE

echo "The following files will be merged in this order:"
cat $TMPFILE"_sort"
echo;read -rsn1 -p "Press any key to continue . . .";echo

cmd='ffmpeg -f concat -safe 0 -i "'"$TMPFILE"_sort" -c copy
"'"$output"''''
eval $cmd
if [ $? -eq 0 ];
then
    actualheight=$(ffprobe -v error -select_streams v -show_entries
stream=height -of csv=p=0:s=x "$output")
    #output2="${output%.*}."$actualheight"p.$container"
    output2="${output//.mergedprogress./.}$actualheight""p.merged.}"
    mv "$output" "$output2"
    echo "Output file: $output2"
    rm $TMPFILE
    rm $TMPFILE"_sort"
else
    echo "An error occured. File not converted properly!"
    echo "Full ffmpeg command:"
    echo "$cmd"
fi

echo;read -rsn1 -p "Press any key to continue . . .";echo

```

Nemo action to call this merge script (to be put in ~/.local/share/nemo/actions:

video_merge.nemo_action

```
[Nemo Action]
Name=Video Merge
Comment=Video Merge
Exec=video_merge.sh %F
Icon-Name=stock_down
Selection=m
Extensions=mp4;wmv;avi;mkv;mov
Quote=double
EscapeSpaces=true
Terminal=true
```

alternative merge script with chapters, inspired by <https://gist.github.com/bdurrow/b51470869dd72b2333407dbfcb947801> has some issues with single quotes in filenames or paths.

video_mergechapters.sh

```
#!/bin/bash
#http://redsymbol.net/articles/unofficial-bash-strict-mode/
set -euo pipefail
IFS=$'\n\t'

cd "$1"
echo "$1"
## Script to merge all mp4 videos in current directory (recursively 2 levels)
## And update chapter marks to retain the folder/filename

## Script for merging videos

temp_dir=$(mktemp -d)
function finish {
    rc=$?
    if [[ $rc != 0 ]]; then
        echo
        echo "FAILED!"
    fi
    echo -n "Cleaning up "
    rm -rf "${temp_dir}"
    echo ".....[ DONE ]"
    exit $rc
}
trap finish SIGHUP EXIT

current_dir=$(pwd)
bname=$(basename ${current_dir})
final_mp4=${bname}.mp4
input_list=${temp_dir}/${bname}-input_list.txt
```

```

file_list=${temp_dir}/${bname}-file_list
meta_file=${temp_dir}/${bname}-metadata.txt

#Hopefully this will work for either BSD or GNU sed
extended_match="-r"
echo "" | sed ${extended_match} 's|foo|bar|' 2>/dev/null ||
extended_match="-E"

if [ -e "${final_mp4}" ]; then
    echo "${final_mp4} already exists, please remove it."
    echo;read -rsn1 -p "Press any key to continue . . .";echo
    exit 1
fi

echo -n "Generating file lists "
find . -maxdepth 2 -type f -iname '*.mp4' | sort -V |
    sed -e 's|^\.||' | sed -e "s|'|\"'\\"'" |
    tee "${file_list}" |
    awk "{printf \"%file '%s'\n\", \"$0}" >
"${input_list}"
echo ".....[ DONE ]"

cat "${input_list}"
echo;read -rsn1 -p "Press any key to continue . . .";echo

## chapter marks
#Do this first so we fail early
#TODO: Test ('=', ';', '#', '\\') are escaped
ts=0
echo -n "Generating chapter marks "
ffmpeg -i "$(head -1 "${file_list}")" -f ffmadata "${meta_file}" -v
quiet

cat "${file_list}" | while read file
do
    ds=$(ffprobe -v quiet -of csv=p=0 -show_entries format=duration
"${file}")
    # echo "$ds"
    escaped_title=$(echo ${file} | sed ${extended_match} -e
's|([=#\\]|\\|g' -e 's|. [Mm][Pp]4$||' )
    echo "[CHAPTER]" >> "${meta_file}"
    echo "TIMEBASE=1/1" >> "${meta_file}"
    echo "START=${ts}" >> "${meta_file}"
    ts=$(awk "BEGIN {print ${ts}+${ds}; exit}")
    echo "END=${ts}" >> "${meta_file}"
    echo "TITLE=${escaped_title}" >> "${meta_file}"
done
echo ".....[ DONE ]"

echo -n "Merging the files "
ffmpeg -f concat -safe 0 -i "${input_list}" -i "${meta_file}" -

```

```
map_metadata 1 -codec copy "${final_mp4}" -v quiet
echo ".....[ DONE ]"

echo "Job Completed."

echo;read -rsn1 -p "Press any key to continue . . .";echo
```

Notes re subtitles: note: mp4 format does not support ass or srt subtitles, but mov_text subtitles. ffmpeg can convert them. formatting from ass subtitles will be lost.

Embed srt subtitle file into mp4 as additional user-controllable track:

```
ffmpeg -i infile.mp4 -i infile.srt -c copy -c:s mov_text outfile.mp4
```

set metadata of first subtitle stream to english:

```
ffmpeg -i infile.mp4 -i infile.srt -c copy -c:s mov_text -metadata:s:s:0
language=eng outfile.mp4
ffmpeg -i infile.mp4 -i infile.ass -c copy -c:s mov_text -metadata:s:s:0
language=eng outfile.mp4
```

extract subtitle from video:

```
ffmpeg -i Movie.mkv -map 0:s:0 subs.srt
```

Multiple subs require mapping, otherwise first sub will be overwritten First add another input: -i input2.srt. Second, map that as 2nd stream: -map 2:0. Finally, select encoder for 2nd subtitle stream (the same as the first one): -c:s srt. The complete example\

```
ffmpeg -i input.mp4 -f srt -i input.srt -i input2.srt -map 0:0 -map 0:1 -map
1:0 -map 2:0 -c:v copy -c:a copy -c:s srt -c:s srt output.mkv
```

convert srt to ass or ass to srt:

```
ffmpeg -i input.srt input.ass
ffmpeg -i input.ass input.srt
```

Remove audio from a video file:

```
ffmpeg -i input_file -c copy -an output_file

# remove all but a specific language audio streams
ffmpeg -i input_file -map 0 -map -0:a -map 0:a:m:language:fre -c copy
output_file
# loop to remove all but turkish audio streams:
for f in *.mkv; do ffmpeg -hide_banner -i "$f" -map 0 -map -0:a -map
0:a:m:language:tur -c copy "${f%.*}.tur.mkv"; done
```

From:

<http://wuff.dyndns.org/> - **Wulf's Various Things**

Permanent link:

<http://wuff.dyndns.org/doku.php?id=howto:merge-video&rev=1755981117>

Last update: **2025/08/23 21:31**

