

# VLC

## VLC no audio

Change output module in Tools → Preferences → Audio from Auto to Pulseaudio or Alsa.

alternative edit config file:

```
vi ~/.config/vlc/vlcrc
```

```
# Audio output module (string)
aout=pulse
```

## Associate VLC to webm videos

Gnome uses two lists to register applications to file types located in:

- /usr/share/applications/mimeinfo.cache
- /usr/share/applications/defaults.list
- ~/.local/share/applications/mimeinfo.cache

```
#manually
sudo vi /usr/share/applications/mimeinfo.cache
sudo vi /usr/share/applications/defaults.list
vi ~/.local/share/applications/mimeinfo.cache

#or via sed:
sudo sed -i 's#video/webm=#video/webm=vlc.desktop;#'
/usr/share/applications/defaults.list
sudo sed -i 's#video/webm=#video/webm=vlc.desktop;#'
/usr/share/applications/meminfo.cache
sudo sed -i 's#audio/webm=#audio/webm=vlc.desktop;#'
/usr/share/applications/meminfo.cache
sed -i 's#video/webm=#video/webm=vlc.desktop;#'
~/.local/share/applications/meminfo.cache
sed -i 's#audio/webm=#audio/webm=vlc.desktop;#'
~/.local/share/applications/meminfo.cache
```

## VLC Playlist from file

These scripts make VLC behave like MPC-HC, ie when you open a file with VLC in a folder, it will automatically add all the files in this folder to the playlist (starting with the opened file, then sorted

alphabetically), so that you can use Previous and Next to navigate through the folder.

<https://github.com/x-hgg-x/Utils/tree/master/VLC%20Create%20playlist%20from%20file>

## Usage

**Notes** Due to internal behaviour of VLC, you cannot replace current playlist by another, only append it. Therefore the script must kill existing instances of VLC to launch a new playlist, so it works only when VLC is in one-instance mode.

You may want to activate the loop option in VLC preferences if VLC is configured to play the next track automatically. (Tools → Preferences → Show all settings → Playlist → Repeat current item)

or edit `~/.config/vlc/vlcrc`:

```
vi ~/.config/vlc/vlcrc
```

```
# Repeat current item (boolean)  
repeat=1
```

## Linux

Copy `vlc.sh` in `~/.local/bin` directory and mark it executable, then copy the desktop file of VLC `/usr/share/applications/vlc.desktop` to `~/.local/share/applications/vlc.desktop` and change the field `Exec` to the path of `vlc.sh` (remove argument `--started-from-file %U`). Known extensions are stocked in variable `$extensions`.

```
sed -i 's#Exec=/usr/bin/vlc --started-from-file  
%U#Exec=/home/'$USER'/~/.local/bin/vlc.sh#'  
/usr/share/applications/vlc.desktop >  
~/.local/share/applications/vlc.desktop
```

note: source from github adjusted to sort numbers as Versions like nemo file manager does.

[~/.local/bin/vlc.sh](#)

```
#!/bin/bash  
  
shopt -s nocaseglob  
shopt -s extglob  
extensions='@(avi|mp4|mkv|m4v|mov|mpg|mpeg|wmv|ogg|flac|m4a|mp3|wav|web  
m|rmvb|asf|flv)' # list of extensions for searching in current  
directory  
  
# kill other instances of vlc to keep playlist clean (one-instance  
mode)  
killall vlc; sleep 0.1
```

```

# launch empty vlc if no argument provided
if [ -z "$1" ]; then
    vlc; exit
fi

# parse argument
filename=$(realpath -- "$1")
dirname=$(dirname "$filename")
basename=$(basename "$filename")

# count files with matching extension, and get position of filename in
current directory
n=$(ls "${dirname}"/*.{extensions} -1 2>/dev/null | sort -f -V | wc -
l)
pos=$(ls "${dirname}"/*.{extensions} -1 2>/dev/null | sort -f -V |
grep -n -F -- "${basename}" | cut -d: -f1)

# if the filename does not have one of the extension above, launch vlc
with provided filename
if [ -z "$pos" ]; then
    vlc -- "${filename}"
    exit
fi

# change positions in playlist such as the first element is the opened
file
ls "${dirname}"/*.{extensions} -1 | sort -f -V | tail -n$((n-$pos+1))
> /tmp/vlc.m3u
ls "${dirname}"/*.{extensions} -1 | sort -f -V | head -n$((pos-1))
>> /tmp/vlc.m3u

# launch playlist
IFS=$'\n'; read -d '' -r -a files < /tmp/vlc.m3u; vlc --norandom
"${files[@]}"

```

```

chmod 755 ~/.local/bin/vlc.sh
sed "s#Exec=/usr/bin/vlc --started-from-file
%U#Exec=/home/$USER/.local/bin/vlc.sh#g" /usr/share/applications/vlc.desktop
> ~/.local/share/applications/vlc.desktop

```

## Windows

Copy vlc.ps1 and vlc.bat in your home directory, then you can open your video/audio file with vlc.bat like this :

vlc.bat video.mp4.

You may want to run Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy UNRESTRICTED once in

a powershell terminal to allow execution of powershell scripts.

You can check if the variable \$vlcPath is correct in vlc.ps1, and modify known extensions with the variable \$Extensions.

## vlc.bat

```
@echo off
powershell -File %USERPROFILE%\vlc.ps1 %1
```

## vlc.ps1

```
Param([string]$filename)

# kill other instances of vlc to keep playlist clean (one-instance
mode)
Stop-Process -Name vlc -ErrorAction SilentlyContinue

$vlcPath = "C:\Program Files (x86)\VideoLAN\VLC\vlc.exe"
$Extensions='(\.avi$|\.mp4$|\.mkv$|\.m4v$|\.mov$|\.mpg$|\.mpeg$|\.wmv$|
\.ogg$|\.flac$|\.m4a$|\.mp3$|\.wav$) '

if ([string]::IsNullOrEmpty($filename)) {& $vlcPath; exit} # launch
empty vlc if no argument provided

# parse argument
$filename = Resolve-Path -LiteralPath $filename
$dirname = Split-Path -LiteralPath $filename
$basename = Get-ChildItem -LiteralPath $filename | ForEach-Object {
$_ .Name }

# count files with matching extension, and get position of filename in
current directory
$filelist = Get-ChildItem -File -LiteralPath $dirname | ForEach-Object
{ $_.Name } | Select-String -Pattern $Extensions | ForEach-Object {
$_ .Line }
$count = $($filelist | Measure-Object -Line).Lines
$position = $filelist | Select-String -Pattern $basename | ForEach-
Object { $_.LineNumber }

# if the filename does not have one of the extension above, launch vlc
with provided filename
if ([string]::IsNullOrEmpty($position)) {& $vlcPath $filename; exit}

# change positions in playlist such as the first element is the opened
file
$filelist | Select-Object -last ($count-$position+1) | ForEach-Object {
"$dirname\" + $_ } > $Env:TMP\vlc.m3u
$filelist | Select-Object -first ($position-1) | ForEach-Object {
"$dirname\" + $_ } >> $Env:TMP\vlc.m3u
```

```
# launch playlist
& $vlcPath $(Get-Content $Env:TMP\vlc.m3u)
```

## VLC Delete file option

So put the file in ~/.local/share/vlc/lua/extensions. I call it DeleteFile.lua.

When you start VLC you get a new menu item: View > Detetefile. It is this last step I was missing.

It now searches for a dir called .waste in the dir of the file and any (grand\*)parent dirs, and moves the file there. So you need to create a dir called .waste.

When you start VLC you get a new menu item: View > Move currently playing file into wastebasket.

### WasteBasket.lua

```
--[[
INSTALLATION (create directories if they do not exist):
- put the file in the VLC subdir /lua/extensions, by default:
* Windows (all users): %ProgramFiles%\VideoLAN\VLC\lua\extensions\
* Windows (current user): %APPDATA%\VLC\lua\extensions\
* Linux (all users): /usr/share/vlc/lua/extensions/
* Linux (current user): ~/.local/share/vlc/lua/extensions/
* Mac OS X (all users):
/Applications/VLC.app/Contents/MacOS/share/lua/extensions/
- Restart VLC.
- The extension can then be found in the menu:
  View > Move current playing file into wastebasket
]]--

--[[ Extension description ]]

function descriptor()
  return { title = "Wastebasket" ;
    version = "0.10" ;
    author = "Mark Morschhäuser/Ole Tange/Wulf Rajek" ;
    shortdesc = "Move current playing file into wastebasket";
    description = "<h1>Wastebasket</h1>"
      .. "When you're playing a file, use Wastebasket to "
      .. "easily move this file to a .waste-dir with one click. "
      .. "<br>This will NOT change your playlist, it will move the file "
      .. "itself. "
      .. "<br>Wastebasket will search for a dir called .waste "
      .. "in the dir of the file and all parent dirs of that.";
    url =
      "https://gitlab.com/ole.tange/tangetools/tree/master/wastebasket"
```

```
}
end

--[[ Hooks ]]

-- Activation hook
function activate()
    local filename,dst,wdir = filename_dst_wastedir()
    -- create waste directory if it doesn't exist
    if(not directory_exists(wdir)) then
        -- escaping single quote while surrounding dir in single quotes for
        -- dirs with spaces
        os.execute("mkdir '" .. string.gsub(dirname(filename) .. wdir,
        "\\'", "\\\\" ) .. "'")
        wdir = dirname(filename) .. wdir
    end
    if(directory_exists(wdir)) then
        d = vlc.dialog("Wastebasket")
        d:add_label("Move <b>".. filename .. "</b> to <b>" .. wdir ..
        "</b>?")
        d:add_button("Move", delete)
        d:add_button("Cancel", close)
        d:show()
    else
        d = vlc.dialog("Wastebasket - no dir found")
        d:add_label(".waste is not found anywhere in parent dirs")
        d:add_button("Cancel", close)
        d:show()
    end
    vlc.msg.dbg("[Wastebasket] Activated")
end

function filename_dst_wastedir()
    -- get the current playing file
    local item = vlc.input.item()
    -- extract its URI
    local uri = item.uri()
    -- decode %foo stuff from the URI
    local filename = vlc.strings.decode_uri(uri)
    -- remove 'file://' prefix which is 7 chars long
    filename = string.sub(filename,8)

    -- find .waste in parent dirs
    local wdir = wastedir(dirname(filename))
    return filename,wdir .. "/" .. basename(filename),wdir
end

function wastedir(dir)
    -- recursively search for .waste in parent dir
```

```

vlc.msg.dbg("[Wastebasket/wastedir] Looking at " .. dir)
local wdir = dir .. "/" .. ".waste"
if directory_exists(wdir) then
    vlc.msg.dbg("[Wastebasket/wastedir] Found wastedir: " .. wdir)
    return wdir
end
-- try the parent dir
local parent = dirname(dir)
if(parent == dir) then
    -- we have reached root (/)
    -- return wdir (which does not exist)
    return wdir
end
vlc.msg.dbg("[Wastebasket/wastedir] parent " .. parent)
if directory_exists(parent) then
    return wastedir(parent)
else
    return parent
end
end

function directory_exists(dir)
    -- Simple checker if dir exists
    -- shell quote the dirname
    dir, _ = dir:gsub("([\002-\009\011-\026\\#?`(){}%[%]^*<=>~|;
\\!$&'\130-\255])", "\\%1")
    dir, _ = dir:gsub("\n", "'\n'")
    return os.execute("cd " .. dir)
end

function deactivate()
    -- Deactivation hook
    vlc.msg.dbg("[Wastebasket] Deactivated")
    vlc.deactivate()
end

function close()
    deactivate()
end

--- Function equivalent to basename in POSIX systems
--@param str the path string
function basename(str)
    local name = string.gsub(str, "(.*)/.*", "%2")
    return name
end

function dirname(str)
    local name = string.gsub(str, "(.*)/.*", "%1")
    return name
end

```

```
function delete()
  local filename,dst,wdir = filename_dst_wastedir()
  if(directory_exists(wdir)) then
    vlc.msg.dbg("[Wastebasket]: Move to " .. dst)
    local retval, err = os.rename(filename,dst)
    if(retval == nil) then
      -- error handling; if moving failed, print why
      vlc.msg.dbg("[Wastebasket] error: " .. err)
    end
  else
    d = vlc.dialog("Wastebasket - no dir found")
    d:add_label(".waste is not found anywhere in parent dirs")
    d:add_button("Cancel", close)
    d:show()
  end
  close()
end

-- This empty function is there, because vlc pested me otherwise
function meta_changed()
end
```

<https://github.com/surrim/vlc-delete/>

[delete.lua](#)

```
--[[
  Copyright 2015-2019 surrim

  This program is free software: you can redistribute it and/or
  modify
  it under the terms of the GNU General Public License as published
  by
  the Free Software Foundation, either version 3 of the License, or
  (at your option) any later version.

  This program is distributed in the hope that it will be useful,
  but WITHOUT ANY WARRANTY; without even the implied warranty of
  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
  GNU General Public License for more details.

  You should have received a copy of the GNU General Public License
  along with this program. If not, see
  <http://www.gnu.org/licenses/>.
]]--

function descriptor()
  return {
```

```

    title = "VLC Delete";
    version = "0.1";
    author = "surrin";
    url = "https://github.com/surrin/vlc-delete/";
    shortdesc = "&Remove current file from playlist and disk";
    description = [[
<h1>vlc-delete</h1>
When you're playing a file, use VLC Delete to
delete the current file from your playlist and <b>disk</b> with one
click.<br>
This extension has been tested on GNU Linux with VLC 2.1.5.<br>
The author is not responsible for damage caused by this extension.
    ]];
}
end

function fileExists(file)
    return io.popen("if exist " .. file .. " (echo 1)") : read "*l" ==
"1"
end

function sleep(seconds)
    local t0 = os.clock()
    local tOriginal = t0
    while os.clock() - t0 <= seconds and os.clock() >= tOriginal do end
end

function windowsDelete(file, trys, pause)
    if not fileExists("\\" .. file .. "\\") then return nil, "File does
not exist" end
    for i = trys, 1, -1
    do
        retval, err = os.remove(file)
        --retval, err = os.execute("del " .. file )
        if retval == true then
            return true
        end
        sleep(pause)
    end
    return {nil, "Unable to delete file"}
end

function removeItem()
    local id = vlc.playlist.current()
    vlc.playlist.delete(id)
    vlc.playlist.gotoitem(id + 1)
    vlc.deactivate()
end

function activate()
    local item = vlc.input.item()

```

```
local uri = item:uri()
uri = string.gsub(uri, "^file:///", "")
uri = vlc.strings.decode_uri(uri)
vlc.msg.info("[vlc-delete] removing: " .. uri)

if (package.config:sub(1, 1) == "/") then -- not windows
    retval, err = os.execute("trash-put --help > /dev/null")
    if (retval ~= nil) then
        uri = "/" .. uri
        retval, err = os.execute("trash-put \"" .. uri .. "\"")
    else
        retval, err = os.execute("rm --help > /dev/null")
        if (retval ~= nil) then
            uri = "/" .. uri
            retval, err = os.execute("rm \"" .. uri .. "\"")
        end
    end
end
if (retval ~= nil) then removeItem() end
else -- windows
    removeItem() -- remove from playlist first so the file isnt
locked by vlc
    uri = string.gsub(uri, "/", "\\")
    retval, err = windowsDelete(uri, 3, 1)
end

if (retval == nil) then
    vlc.msg.info("[vlc-delete] error: " .. err)
    d = vlc.dialog("VLC Delete")
    d:add_label("Could not remove \"" .. uri .. "\"", 1, 1, 1, 1)
    d:add_label(err, 1, 2, 1, 1)
    d:add_button("OK", click_ok, 1, 3, 1, 1)
    d:show()
end
end

function click_ok()
    d:delete()
    vlc.deactivate()
end

function deactivate()
    vlc.deactivate()
end

function close()
    deactivate()
end

function meta_changed()
```

end

<https://superuser.com/questions/721112/vlc-how-to-delete-file-on-disk>

From:

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

Permanent link:

<http://wuff.dyndns.org/doku.php?id=howto:vlc-playlist-from-file&rev=1678729006>

Last update: **2023/05/29 11:53**

