

# Grafikus fájlok átkonvertálása, átdolgozott

PCLinuxOS Magazine – 2023. március

Írta: Paul Arnote (parnote)

2022 novemberében a PCLinuxOS fórumán tárgyaltak egy [témát](#), amit GOTHBITES indított megkérdezve, hogyan lehet a JXL-fájlokat (JPEG XL) megnézni. 2022 áprilisában egy [cikkb](#), a WEBP grafikákkal foglalkozva beszéltünk a JXL-ről. A PCLinuxOS felhasználók ... azaz a többségük, az operációs rendszertől függetlenül ... eléggé magukra vannak hagyva, ha JXL-fájlt kell megnézni, vagy használni. Kétlem, hogy a JXL-formátum teret nyerne, úgy fog járni, mint a JPEG Csoport összes eddigi próbálkozásával az eredeti JPG-formátum módosítására. Egyik sem érdemel említést széles körben, kivéve az irányító testület vezetőit és rajongóikat.

Az előbb említett fórumcsevegés második oldalán a társalgás az „újdonsült” WEBP” grafikus formátumra váltott át (a Google 2010 szeptemberében jelentette be nyílt forráskódúként, de a könyvtárának első stabil verziója 2018 áprilisáig nem jelent meg, a Wikipédia [szerint](#)), ami egyre fontosabbnak tűnik a széles körű elfogadása miatt, legalábbis a weben. Még arról is olvastam beszámolókat, hogy a Google Chrome böngésző a képeket WEBP formátumban menti, amikor és ha a „Kép mentése mint ...”-et választod weboldalás kép esetén. Ugyanakkor, a linuxosok által ismert és használt és elterjedt szöveg-, illetve kiadványszerkesztők (így a LibreOffice és a Scribus) egyike sem ismeri fel a WEBP grafikus formátumot, miközben a nagyobb böngészők és a GIMP támogatja a formátumot.

Ennek eredményeképp, a felhasználók folyamatosan egyre többször találkoznak vele. A WEBP grafika hatalmas előnye röviden annyi, hogy kisebb méreteés kiváló képminősége miatt a webfejlesztők kisebb fájlokkal dolgozhatnak, ami a weblapok gyorsabb betöltését és megjelenítését segíti. Bármikor, ha a felhasználók WEBP grafikába botlanak és szeretnék azt a LibreOffice-os művükben, vagy kiadványszerkesztőben felhasználni, a WEBP-képet át kell konvertálniuk PNG-be, vagy JPG-be (többnyire), amit azután a programjuk be tud olvasni.

Most tekerjünk vissza 10 évet, 2013 márciusához. Akkor, az „Xfce haladó felhasználók tippjei, trükkjei és beállításai: munka grafikus fájlokkal” c. [cikkb](#)

közzé tettem egy Zenity-vel felturbózott saját bash-szkriptet. Azt azóta is SOKAT használok, bár egy kicsit fejlesztettem és hozzá raktam a felhasználói menühöz. A fórumon tűz alá vettek, hogy a 10 éve készített szkriptemet fejlesszem tovább.

Nos, a következőkben találjátok a szkript továbbfejlesztett változatát. Az szkriptet kép formátumban mutatom meg, hogy a sorok megfelelő sorrendjét megtartsam. A szkript a Magazine weblapjáról [letölthető](#). Ha ezt megteszed, a szkriptet mentsoda, ahol a bash-fájlokat tárolni szoktad (célszerűen az elérési útvonaladban valahol), a .txt kiterjesztést a letöltött fájlról távolítsd el és adj a fájlnak futtatási jogot.

```
1. #! /bin/sh
2.
3. n=1
4.
5. EXT=$(zenity --list --column="Select One" --title="Convert Image" --width=250
   --height=250 --text="Select the image format\nto convert to:" jpg png webp
   avif tiff bmp gif)
6. if [ $? == 1 ]; then
7.     exit
8. fi
9. declare -l EXT
10.  EXT=$EXT
11.
12.  QUAL=`zenity --title="Convert Image" --width=250 --height=250 --entry --
   text="Enter the quality level:\n\n100 = Full Quality\n75 = 75%
   Quality\n50 = 50% Quality\n\nAny number between\n1 and 100 accepted."`
13. if [ $? == 1 ]; then
14.     exit
15. fi
16. if [[ ( $(($QUAL)) -lt 1 ) || ( $(($QUAL)) -gt 100 ) ]]; then
17.     zenity --title="Convert Image" --width=200 --height 100 --error --
   text="Exiting.\n\nNumber entry\nout of range."
18.     exit
19. fi
20.
21.if [ $EXT == "png" ];
22. then
23.     BACKGROUND="-background none"
24.     QUALITY=$((($QUAL / 10))
25. else
```

```
26. BACKGROUND="-background white"
27. QUALITY="$QUAL"
28.fi
29.
30.NEW_DIR=`zenity --title="Convert Image" --entry --text="Please enter the
destination directory.\n\nTo use the current directory,\nuse ./ as the
destination."`
31. if [ $? == 1 ]; then
32.     exit
33. fi
34.NEW_DIR=$NEW_DIR
35.mkdir $NEW_DIR
36.sleep 1
37.
38.for file in $@; do
39.     if [ ! -e $file ]; then
40.         continue
41.     fi
42.     name=$( echo $file | cut -f1 -d.)
43.     convert -quality $QUALITY $BACKGROUND $file $NEW_DIR/${name}-
$QUALITY.$EXT
44.     echo "$(($n * 100 / $#))"
45.     echo "# Processing file: $file"
46.     let "n = n+1"
47.
48.
49.done | (zenity --progress --title "Converting to "$EXT"..." --percentage=0 --
auto-close --auto-kill)
50.
51.exit 0
```

A szkript első 10 sora hozza létre a Zenity párbeszédablakot, (pontosabban az 5. sor), kérve a felhasználótól az eredeti, átalakítandó grafikus formátumú fájl kiválasztását. Módosítottam az eredeti szkriptet, hogy megakadályozza az eredeti fájl felülírását. Az előző változatban, ha ugyanazt a kiterjesztést adtad a kimeneti fájlnak, mint ami a bemenetié volt és azt az aktuális könyvtárba mentetted (./), kockáztattad az eredeti fájl felülírását. Az eredeti grafikus fájl felülírása potenciálisankáros. Természetesen, ha a képfájl már a kívánt formátumú, egyébként sincs túl sok okod a szkript használatára. De hibák bármikor előfordulhatnak, ezért gondoskodtam róla, hogy elkerüljük az ilyen katasztrófális végkimenetelt. Másik dolog, amit megtehetsz, hogy megadsz egy új könyvtárat (a szkript harmadik párbeszédablakában), ahová elhelyezed a kimeneti fájljaidat. Ám az én megoldásom teljesen lehetetlenné teszi, hogy az eredeti fájljaid akármikor felül legyenek írva.

Kimeneti fájlként választhatsz JPG, PNG, WEBP, AVIF, TIFF, BMP vagy GIF fájlformátumok közül. Az utóbbi kettőt csak a élvezet kedvéért adtam hozzá, nem túl sokan akarják használni bármelyiket is rendszeresen. Egyértelmű, hogy

miként lehet fájlformátumokat hozzáadni (vagy eltávolítani) az 5 sorban lévő parancshoz (tipp: a parancs vége felé van a felsorolás). Ha a szkripthez további formátum hozzáadása mellett döntesz, arra figyelj, hogy az ImageMagic ismerje (mivel a munka nehezét az végzi a szkriptben).

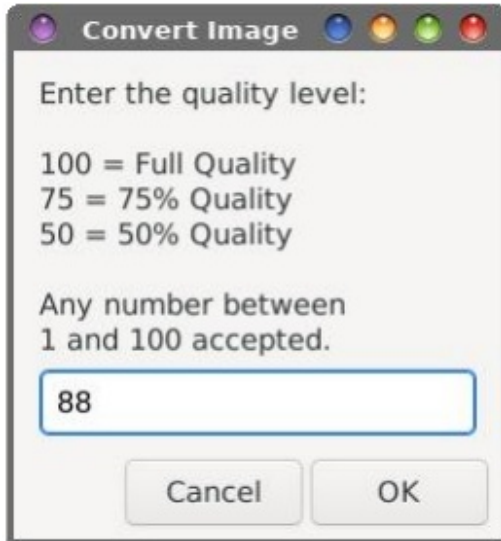
A 9. és a 10. sor az eredeti szkriptből maradtak, a felhasználó itt adja meg a fájlformátumot, arra ügyelve, hogy a fájlkiterjesztés kisbetűs szöveg legyen.



A következő párbeszéd (a 12-15. sorig a szkriptben), a kimeneti fájl minőségének meghatározására való. Bármilyen szám megadható 1 és 100 között. Vedd figyelembe, hogy a nagyobb szám kisebb tömörítést jelent (így a kimeneti fájl mérete nagyobb lesz és a képminőség jobb lesz), miközben a kisebb érték



nagyobb tömörítést jelent (így kisebb a kimeneti fájl méret és képminősége gyengébb).

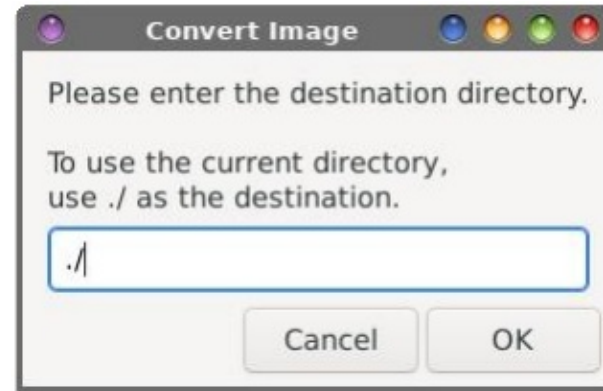


A szkript ezen verziójához adtam hozzá „hibaellenőrzést” is, a 16-19 sorban. A beírandó számnak 1 és 100 között kell lennie. Ha nem, akkor a Zenity hibaüzenete jelenik meg, ahogy az a következő képen látható és a szkript kilép. A szkript akkor is hibaüzenetet ad és kilép, ha „Enter” gombot nyomsz bármilyen bemeneti adat nélkül, illetve betűket, vagy szám-betű keverékét írsz számok helyett.



A harmadik (fő) párbeszédablak (a 30-36. sor jeleníti meg) az, ahol a kimeneti fájl könyvtárát meghatározod. Írhatod „./”-t kimenetnek (a bemeneti fájljal azonos

könyvtárat jelent). Vagy meghatározhatod más könyvtárat is. Ha a könyvtár, amit megadsz nem létezik, létrehozza neked.



A 38-47. sor állítja be a feldolgozási folyamat ciklusát (igen, több fájlt is beírhatod, vagy használhatod a dmsókat) az ImageMagick konvertáló parancsa számára (44. sor). A szkript ezen sora egészíti ki a bemeneti fájl nevét az általad meghatározott minőségi szint értékével, elkerülendő az eredeti fájl(ok) felülírását.



Végül, a 49. sor jeleníti meg az előrehaladástmutató párbeszédablakot, hogy lásd, a szkript a képfájlok feldolgozásának folyamatában hol jár, mielőtt a feladatok végrehajtását követően rendesen kilépne (51. sor).

Ahogy a legtöbb szkriptem, ez is futtatható önállóan a parancssorból bash-szkriptként, vagy (ahogy és a többi Xfce-felhasználó szokta) a Thunar-en belül Felhasználói műveletként. Ha „új” vagy az Xfce-ben és segítségre van szükséged a Thunar Felhasználói művelet beállításában, nézd meg a 2013. márciusi [cikket](#). Ott lépésről lépésre megmutatom, hogyan készíts Felhasználói műveletet ehhez a szkriphoz. Mate- és KDE-felhasználóknak sem jelenthet túl nagy gondot hasonló módon a Caja, vagy a Dolphin fájlkezelőhöz kötni, de én nem fogom a módját leírni, mivel sosem használom azokat az ablakkezelőket. Elnézést kérek az LXQt-sektől, de a PCFileMan egy teljesen más állatfaj.

### Összegzés

Ez a cikk nem foglalozik részleteiben a WEBP grafikus formátummal. Ahhoz lapozz vissza a 11 hónappal ezelőtt írt cikkemhez, ott mélyebben beleástam magam. Jó kimeneti formátum, ami nagyon jó képminőséget ad jelentős fájlméret-megtakarítás mellett.

Ugyanakkor, a LibreOffice és Scribus fejlesztőinek időbe telik ... és még másoknak is... hogy felzárkózzanak. Például, a kedvenc képnézegetőm a GpicView, mivel kicsi és gyors. Nem képes (jelenleg) a WEBP-képekbe belenézni.

A szkriptet nem próbáltam ki JXL-fájlokkal (miért kellene ezekkel a fájlokkal teleszemetelni a helyet, amikor olyan kis támogatás van hozzá?) de működni kell a konvertálásnak erről a formátumról is. Alapvetően, ha az ImageMagick „megérti” a bementi fájl formátumát, a szkriptnek képesnek kell lennie azt egy sokkal elterjedtebb képformátumra konvertálni, amit majd használni tudsz.

Azokra az esetekre, amikor grafikus formátumokat kell átkonvertálnod, kapod ezt a bash-fájlt, hogy sokkal könnyebben végrehajthasd a konverziót WEBP grafikus formátumról olyanra, amit azokban a programokban használni tudsz. Van más módja is a „feladat” végrehajtásának? Természetesen, de a szkript egyszerű, könnyű és ráadásul megbízható.

**The PCLinuxOS Magazine  
Special Editions!**

Windows Migration Guide  
September 2013

Enlightenment  
Special Edition  
June 2014

**THE NEW PCLinuxOS Magazine**  
Fall 2010

**The KDE 4 SC  
Special Edition**

**THE NEW PCLinuxOS Magazine**  
GTK Lightweight Desktops:  
Xfce & LXDE Special Edition  
November 2011

**The PCLinuxOS Magazine**  
Openbox  
Special Edition  
March, 2012

**The NEW PCLinuxOS Magazine**  
Command Line Interface Intro  
Special Edition  
December, 2010

**Get Your Free Copies Today!**