

# PDF-fájlok konvertálása könnyen képfájlokká

PClinuxOS Magazine – 2017. január

Írta: Paul Arnote (parnote)

A PDF-ek olyan általánossá váltak, mint a számítástechnika maga. Az 1990-ben kifejlesztett Adobe Portable Document Format megnyerte a digitális dokumentumokért folytatott harcot a DjVu, Envoy, Common Ground Digital Paper, Farallon Replica és Adobe saját PostScript formátumával szemben.

Az Adobe 1993-ban tette szabadon elérhetővé a PDF-specifikációját, de továbbra is az Adobe által ellenőrzött jogvédett formátum maradt egészen 1998. július 1-jéig, amikor hivatalosan is nyílt szabvánnyá nyilvánította. Az Adobe továbbra is rendelkezik a PDF-formátum szabadalmával, de szabadon hozzáférhetővé tett minden olyan Adobe szabadalmat, ami PDF-képes alkalmazások készítéséhez, eladásához és terjesztéséhez kell. <sup>1</sup>.

Gyakorlatilag bárki, aki már használt számítógépet, használt PDF-fájlt is. Tényleg ENNYIRE elterjedt. Sok, sok kormányzati űrlapot PDF-fájlként terjesztenek. Ám az Adobe továbbra is használ néhány nem szabványos szabadalmazott technológiát, így a felhasználók kompatibilitási problémákba futhatnak bele, amikor ilyen technológiával készült PDF-fájlokat néznek nem-Adobe szoftverrel. Még ha az Adobe publikálja is ezen technológiák specifikációját a weboldalain, azokra, vagy használatukra vonatkozóan nincsen szabványosítás.

Ha sok PDF-fel van dolgod (ahogy nekünk a PCLinuxOS Magazine-nál), előfordulhat, hogy egyszerűbbnek találod valamilyen PDF oldal (vagy lapok) megjelenítését grafikus fájl (vagy fájlok) formájában. Valójában, minden hónapban ezt csinálom a címlappal. Megtervezem/zük a magazin címlapját Scribus-szal és PDF-be exportáljuk. A

címlapot minden hónapban grafikus formában jelenítjük meg a magazin honlapján csakúgy, mint más helyeken (sajtóközlemény, Google+, PCLinuxOS-fórum stb.).

Mint ilyen, mindig keresem a módját a minden egyes hónapban megismétlődő folyamat modernizálására. A Scribus rendelkezik olyan lehetőséggel, hogy az adott oldalt képként mentse, de a legutóbbi néhány verzióban ez a tulajdonság megsérült. Ehhez vallásosan ragaszkodtam egészen addig, amíg meg nem sérült. A „biztonsági tartalékokat” használva, meg tudom nyitni a PDF-fájlt GIMP-ben és exportálhatom a kiválasztott lapokat képfájlba. Ezt az eljárást használtam a címlapok képfájljainak készítésére egészen a 2016. decemberi számig. Ám ez a folyamat is terhessé vált és több lépéses. Most találtam egy jobb módszert, ami sokkal de sokkal egyszerűbb.

## És most a csúcs ... ImageMagick

Soha nem múlóan csodálom az ImageMagick eszköz képességeit. Az ImageMagick nem pusztán egy eszköz – sokoldalú grafikus eszközök tárháza. Kevés olyasmi van, amit az ImageMagick parancscsomag nem tud megtenni, legalábbis a képfájlokat illetően.

Íme a parancs, amire szükséged van (terminál parancssorából), a legalapvetőbb formájában:

### convert be\_fájl.pdf ki\_fájl.jpg

Tudd, ha ezt többlapos PDF-fájlon próbálsz ki, minden egyes lapról külön képfájlt fogsz kapni. Szóval, én a helyedben nem próbálnám ezt ki a Háború és békén. A végeredmény egy nagy rakás fájl lenne ki\_fájl-0.jpg, ki\_fájl-1.jpg, ki\_fájl-2.jpg stb. névvel.

Nem kell JPG fájlokat sem használni. Az átalakító (convert) parancs elfogad más, használni kívánt képfájl-kiterjesztéseket is, mint a „png”, vagy akár a „bmp” (de használja bárki is még ez utóbbi nem tömörített grafikus fájl?). A convert megérti, hogyan kell kiírni bármilyen, általad meghatározott képfarmátumban és a kimenetet annak megfelelően állítja be.

Megváltoztathatjuk ezt az alapbeállítást a parancs alábbi módú kisebb módosításával:

### convert be\_fájl.pdf[0] ki\_fájl.jpg

A convert parancs ezt úgy értelmezi, hogy a megadott PDF-fájl „1. oldal”-ából készíteni akarsz egy JPG-fájlt. Mivel a convert parancs 0-tól kezdi a számolást, a PDF 1. oldalát úgy tekinti, mint 0. oldal, a 2. oldalt, mint 1. oldal, a 3. oldalt, mint 2. oldal stb.

Hasonlóképpen, megadhatunk oldaltartományt is, valahogy így:

### convert be\_fájl.pdf[12-15] ki\_fájl.jpg

A parancs veszi a PDF-fájlból a 13., 14., 15. és 16. oldalt, majd készít belőlük JPG-fájlokat ki\_fájl-12.jpg, ki\_fájl-13.jpg, ki\_fájl-14.jpg és ki\_fájl-15.jpg néven. Amikor meghatározod, hogy egy PDF-fájl egy tartományát konvertálja képpé, ne felejtsd el kivonni egyet a PDF oldalszámolásából, hogy ténylegesen a kívánt oldalakat konvertálja.

Természetesen használhatjuk az ImageMagick bármely másik utasítását, amit csak a convert parancs megért. Például:

### convert -resize 498 -bordercolor black -border 1x1 be\_fájl.pdf[0] ki\_fájl.jpg



Ez a parancs készít egy 500 pixeles képet, 1pixeles fekete kerettel a meghatározott PDF-fájlból, és elkészíti a megadott kimeneti fájlt. Ne csüggedj! A képszélességet nem adtam meg hibásan 498 pixelként. Amikor a fekete keretet hozzáadjuk a végső képhez, akkor a képszélesség összesen 500 pixel lesz.

### „Automagic”-ká alakítani

Mivel Xfce az elsődleges asztali környezetem, egy kicsit automatizáltam a folyamatot. Először készítettem egy felhasználói bash-fájlt, amit mag-resize.sh-nak neveztem el. A név nem fontos, amíg tudom, mit csinál. Hívhattam volna bettyboop.sh-nak is, ha akartam volna. Íme itt van:

```
#!/bin/bash
```

```
n=1
```

```
for file in $@; do
    if [ ! -e $file ] ; then
        continue
    fi

    name=$( echo $file | cut -f1 -d. )

    convert -quality 10 -background
white $file[ 0 ] ./${name}. png
    convert -resize 790 -quality 85
- background white - bordercolor black
- border 1x1 $file[ 0 ] ./${name}- 792. j pg
    convert -resize 450 -quality 80
- background white - bordercolor black
- border 1x1 $file[ 0 ] ./${name}- 452. j pg
    convert -resize 598 -quality 80
- background white - bordercolor black
- border 1x1 $file[ 0 ] ./${name}- 600. j pg
    convert -resize 254 -quality 80
- background white - bordercolor black
- border 1x1 $file[ 0 ] ./${name}- 256. j pg
    convert -resize 168 -quality 80
- background white - bordercolor black
- border 1x1 $file[ 0 ] ./${name}- 170. j pg
    convert $file[ 0 ] - crop 792x120+0+0
```



```
quality 90 banner. j pg
echo $( ( $n * 100 / $# ) )
echo " # Processing file: $file"
let " n = n+1"

done | ( zenity -- progress -- title
" Resizing. . ." -- percentage=0 -- auto- close
-- auto- kill)

exit 0
```

Ezután, hogy ne kelljen a parancssori prompt-ot előhívnom a bash-szkriptem használatához, csináltam egy Thunar felhasználói műveletet. Most csak annyit kell tennem, hogy Thunar-ben a fájlon jobb gombbal kattintok és a felhasználói menüből kiválasztom és hátra dőlök. A „munkát” automatikusan elvégzi, és a magazin címlapjának összes méret, valamint formátumváltozata varázslatosan megjelenik a PDF-fájllal azonos könyvtárban.

A fenti a példa kimenete, amit a Thunar felhasználói menüjéből kiválasztva a bash-szkriptet futtató művelet hozta létre a fenti képet. A szkript egy „teljes méretű” 792x612 pixeles, jó minőségű PNG fájlt készít el csakúgy, mint az 1 pixel fekete keretű 792, 600, 452, 256 és 170 pixel széles JPG változatokat. Az utolsó lépés, hogy a címlapból készít egy vágott változatot, amit minden hónapban a magazin HTML-verziói weblapján használunk fel.

### Összegzés

Minél jobban megismered az ImageMagick-ot és a rakat parancssori eszközét, annál jobban ráébredsz, hogy megkönnyítheti az életedet. Egészen mostanáig nem voltam tisztában azzal, hogy a convert parancs olvassa a PDF-fájlokat. Most már tudom, hogy tudja és felfedett valamennyit a nagy erejéből, átvállalva a korábban időt rabló feladatot és olyan egyszerűvé téve, hogy a hároméves Ryan fiam is képes a feladatot végrehajtani.

1. Forrás: [https://en.wikipedia.org/wiki/Portable\\_Document\\_Format](https://en.wikipedia.org/wiki/Portable_Document_Format)



**PCLOS-Talk**  
Instant Messaging Server

Sign up TODAY! <http://pclostalk.pclosusers.com>

## Screenshot Showcase



Posted by footstep11, on December 15, running KDE.



**DOS GAMES ARCHIVE**  
[WWW.DOSGAMESARCHIVE.COM](http://WWW.DOSGAMESARCHIVE.COM)



**Donate NOW**

**PCLinuxOS**

gofundme