

MULTI LAYO1 JOBS OP 1 A4

(English on the second half)

met handmatige Postscript edit

Als voorbereiding moeten de jobs voorberekt worden van Layo1 output *.Pxx naar *.ps of *.eps formaat met behulp van het Layo1 PS repair programma van mij. Fouten of bugs in de positie code zijn dan gelijk opgepoetst. Ook zijn dan de aangepaste grijs stappen omgezet in kleur codering. (setgray naar setrgbcolor)

Één job, de eerste links onderaan, moet de volle A4 (afdruk) parameters ingebakken krijgen (%%PageBoundingBox) , met eveneens A4 afdruk code in de header dus. Dat is: de bladspiegel wordt op A4 gezet. Alleen aan het echte eind van de latere totaal job staat dan als laatste "showpage". NIET meer op de huidige plek.

De extra job(s) moeten van het type *.EPS zijn. Zonder afdruk commando "showpage" en zonder pagina afdruk maat en code. Wel met de eigen (%%BoundingBox:) maat.

Het eerste oorspronkelijke PS bestand heeft zijn afsluiter "grestore" net voor de eerste "gsave" van de opvolgende EPS. De afzonderlijke EPS blokken staan tussen afscherm codes, na de eerste (A4) job, zo:

```
grestore
gsave
{volledige EPS bestand}
grestore
```

Aan het eind van het Postscript code bestand staat op de regel vòòr "showpage" dus de afsluit code "grestore" die hoort vanzelf bij de laatst ingevoegde EPS.

De allereerste "gsave" staat net voor de eerste regel met /macro code. één regel eerder kan misschien ook, maar dat is niet meer uitgebreid getest. En is ook NIET handig gebleken. Zoals het NU is, is het goed.

De allereerste "gsave" gaat zo:

```
{statusdict begin 595 841 0 1 setpageparams end} stopped pop
106 45 {dup mul exch dup mul add 1.0 exch sub} setscreen
gsave
/layo_units {0.05625 mul} def
```

dan bij het eind van de eerste PS en begin EPS ("showpage" dus weg):

```
0.0 g
grestore
```

gsave

%!PS-Adobe-2.0 EPSF-2.0

%%Creator: Layo1_4 6.1 Edit layerP12 by LINO5 - PE1ABR

voor alle volgende toegevoegde EPS files ziet het er hetzelfde uit.

maar nogmaals aan het eind van het totale bestand:

0.0 g

%%EOF

grestore

showpage

=====

Tot zover de postscript code aanpassingen.

=====

Echter.....

Alle jobs staan over elkaar heen, klem in het Postscript startpunt, dat is linksonder, zonder marge.

Alle afzonderlijke jobs moeten nog een positie correctie krijgen m.b.v. het "x-as y-as translate" commando. Het gebruikte maatsysteem is door de positie van gsave/grestore opdracht code overal in Layo1 punten. Het kan ook door elkaar in PS en Layo1, maar dat doen we niet. Zie voor sommetjes onderaan (PS <--> Layo1 <--> mm)

wanneer er al een mini oorsprong marge van bijv. 5 mm gedefinieerd is in de output driver zie je die terug als (in Layo punten):

```
1.0000 layo_units 1.0000 layo_units scale
```

```
252 252 translate
```

252 in Layo1 punten is dan 14 PS punten en 4,9 (= afgerond 5) mm

als je op de film afdruk iets wil schuiven in de positie, maak je er het liefst 5 cm van, geeft dus (alleen in de eerste job !!!)

```
1.0000 layo_units 1.0000 layo_units scale
```

```
2540 2540 translate
```

Het eerste getal is x-inspring van de linker kantlijn, het tweede getal is y-opschuiven in de hoogte

In ALLE volgende (EPS) jobs zoeken we deze "translate" op en maken de eerste inspring overal gelijk (bijv. 2540 = 5 cm), het tweede getal is het verplichte opschuiven in de hoogte. LET WEL in Layo1 punten systeem.

Aanwezige macro code uit het PS bestand:

```
/layo_units {0.05625 mul} def
```

dit geeft / betekent:

Layo1 punt x 0.05625 = Postscript punt

of

Postscript x 17,7777778 = Layo1 punt

en

1 Postscript punt = 25,4 / 72 = 0,352777778 millimeter

van millimeter naar Layo1 punten = x 50,8.

bovenstaande gebruikte marge inspring van 50 mm (afgerond) = 50 x 50.8 = 2540 Layo1

====

Op dezelfde manier kun je met het tweede getal in "translate" ook opschuiven in de x-richting. Ook weer in Layo1 punten tal-stelsel.

MULTI LAYO1 JOBS ON 1 A4, with manual Postscript edit

As preparation, the jobs must be pre-processed from Layo1 output *.Pxx to *.ps or *.eps format using the Layo1 PS repair program. Errors or bugs in the position code are then immediately cleaned up. The modified gray steps are also converted into color coding. (setgray to setrgbcolor)

One job, the first one at the bottom left, must have the full A4 (print) parameters embedded (%%PageBoundingBox), with A4 print code in the header as well. That is: the page layout is set to A4.

Only at the actual end of the later total job will "showpage" appear as the final element. NO LONGER in the current position.

The extra job(s) must be of type *.EPS. Without the print command "showpage" and without page print size and code. However, with its own (%%BoundingBox:) size. The first original PS file has its closing code "grestore" just before the first "gsave" of the subsequent EPS.

The individual EPS blocks are enclosed in masking codes, after the first (A4) job, like this:

```
grestore  
  
gsave  
  
{entire EPS file}  
  
grestore
```

At the end of the Postscript code file, on the line before "showpage", is the closing code "grestore", which naturally corresponds to the last inserted EPS.

The very first "gsave" is located just before the first line with /macro code. One line earlier might also be possible, but that has not been extensively tested. And it has also proven NOT to be practical. As it is NOW, it is fine.

The very first "gsave" goes like this:

```
{statusdict begin 595 841 0 1 setpageparams end} stopped pop  
106 45 {dup mul exch dup mul add 1.0 exch sub} setscreen  
  
gsave  
  
/layo_units {0.05625 mul} def
```

then at the end of the first PS and beginning EPS (so "showpage" is removed):

```
0.0 g  
  
grestore  
  
gsave  
  
%!PS-Adobe-2.0 EPSF-2.0  
  
%%Creator: Layo1_4 6.1 Edit layerP12 by LINO5 - PE1ABR
```

for all subsequent added EPS files it looks the same.

but once again at the end of the total file:

```
0.0 g
%%EOF
grestore
showpage
```

=====

That concludes the PostScript code adjustments.

=====

However.....

All jobs overlap, clamped at the Postscript starting point, which is bottom left, without margin.

All individual jobs still need to be positioned using the "x-axis y-axis translate" command. The measurement system used is Layo1 points throughout due to the position of the gsave/grestore command codes. It is also possible to mix them in PS and Layo1, but we are NOT doing that. See calculations below (PS <--> Layo1 <--> mm)

when a mini origin margin of e.g. 5 mm is already defined in the output driver, you will see it reflected as (in Layo points):

```
1.0000 layo_units 1.0000 layo_units scale
252 252 translate
```

252 in Layo1 points is then 14 PS points and 4.9 (= rounded to 5) mm

if you want to shift the position slightly on the film print, preferably make it 5 cm, so gives (only in the first job !!!)

```
1.0000 layo_units 1.0000 layo_units scale
2540 2540 translate
```

The first number is x-indentation of the left margin, the second number is y-shift in height

In ALL subsequent (EPS) jobs we look up this "translate" and make the first indentation equal everywhere (e.g. 2540 = 5 cm) the second number is the mandatory height shift. PLEASE NOTE in Layo1 point system.

macro code from a PS file:

```
/layo_units {0.05625 mul} def
```

this gives:

Layo1 point x 0.05625 = Postscript point

or

Postscript x 17.7777778 = Layo1 point

1 Postscript point = 25.4 / 72 = 0.352777778 millimeter

from millimeter to Layo1 points = x 50.8.

above used margin indentation of 50 mm (rounded) = 50 x 50.8 = 2540 Layo1

====

In the same way, you can also shift jobs in the x-direction using the second number in "translate".
Again in Layo1 point number system.