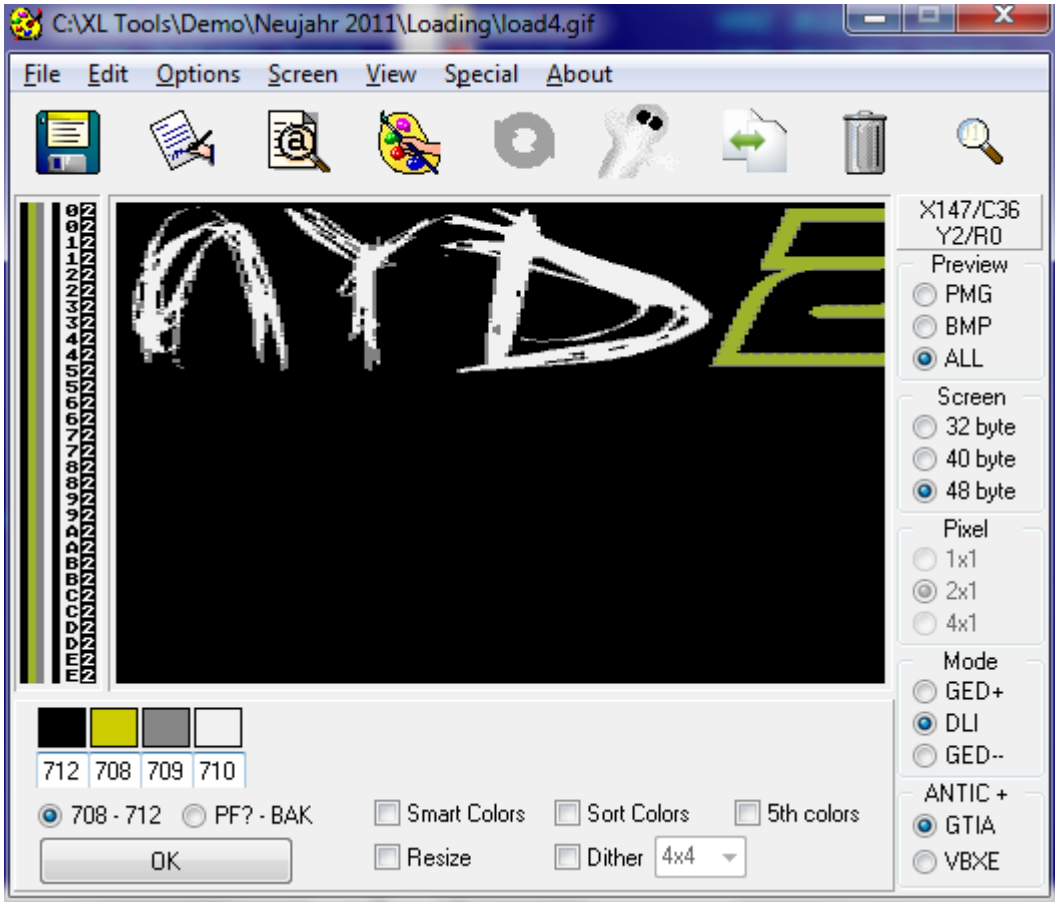
First of all, we need a little thing, to convert. Actual version of g2f allows you to choose between original ATARI formats (.MIC of Micro Illustrator or .PIC of ATARI Artise etc.) as well as jpg, gif or png files from your actual PC. But beware that the ATARI has 'limited' resolution!

For this little tutorial I show you how I did a logo for loading screen in a NYD 2011 production. I created a 4 color 320/240 picture with PSPro and saved it as gif file (without transparency!). Even 5 colors maybe allright.

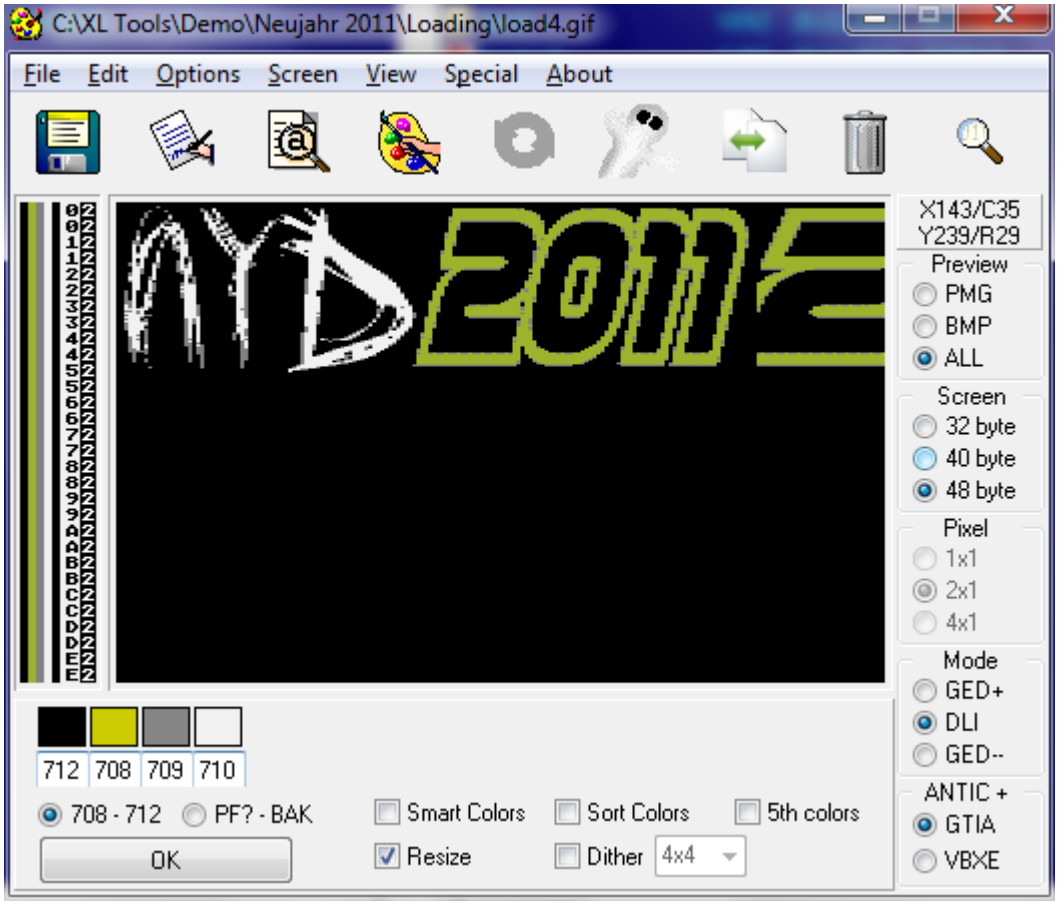

*picture 1: load4.gif*

I simply open this (via right click menu) with graph2font and we will get this screen:



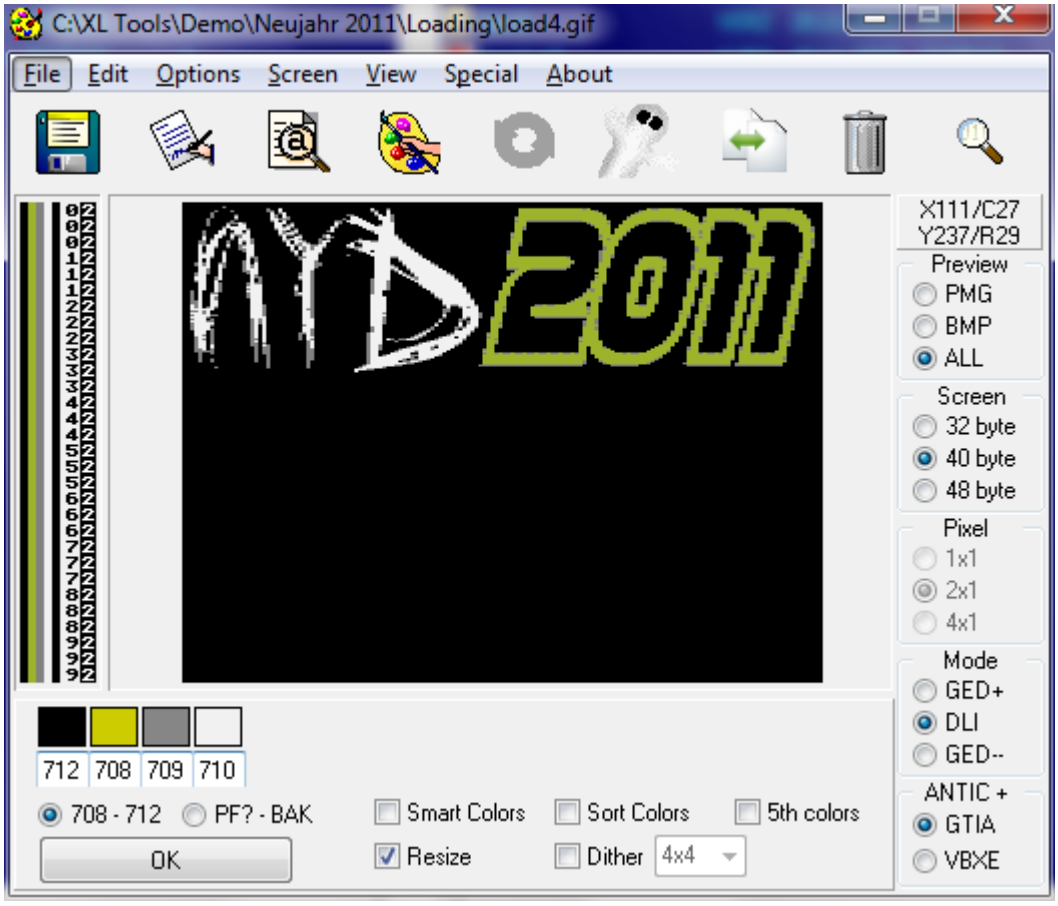

*picture 2: screen 1 – first load with g2f*

Ok looks nice, but we need to resize this:



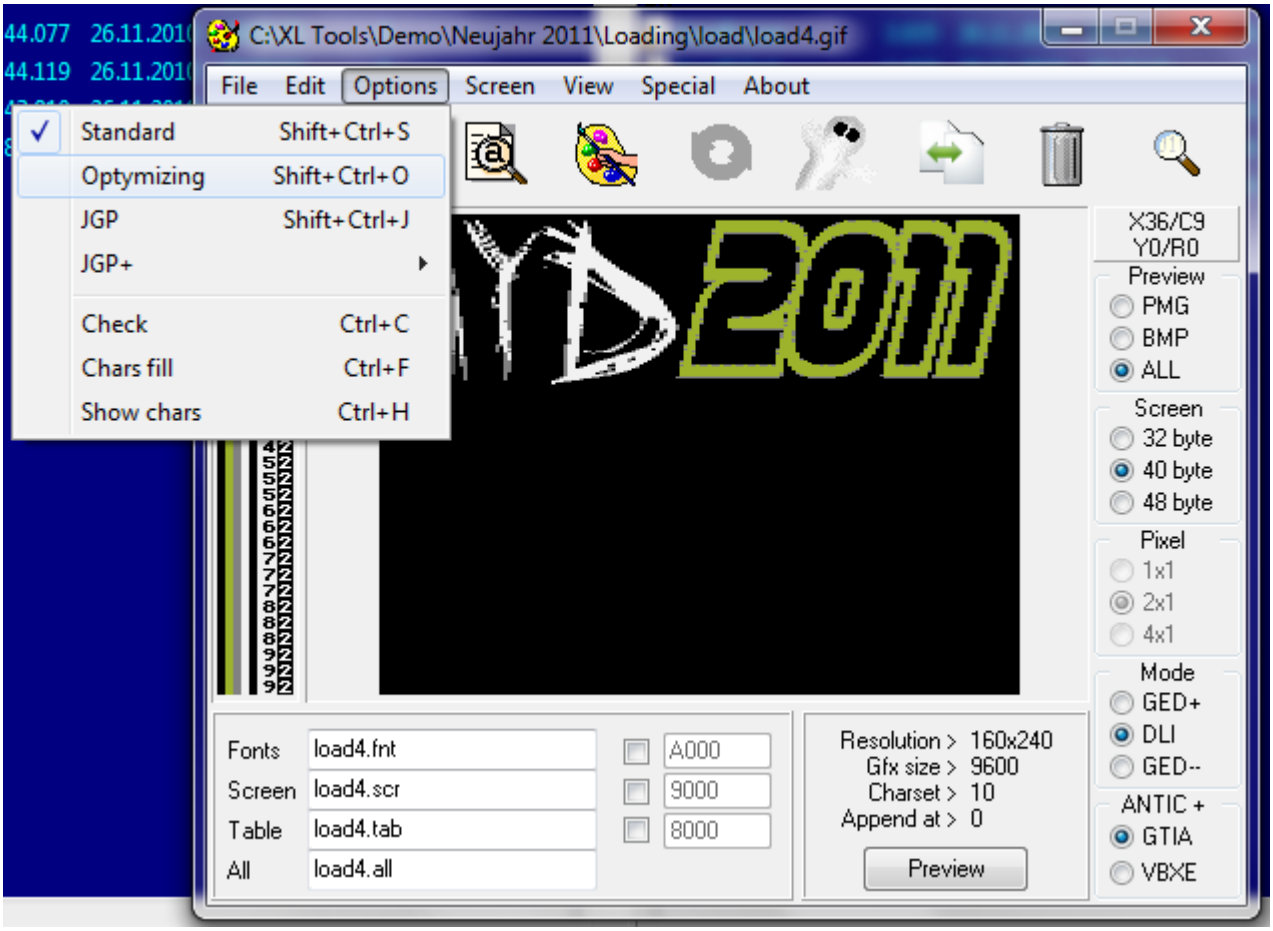

*picture 3: screen 2 – something's wrong here*

Hmm, there is still something wrong. Oh yes, I choosed to do 'only' a 320 pixel wide logo, so we have to change screen size from 48 bytes to 40 bytes (that´s the normal width of ATARI graphics → 320 pixels/8=40 chars per line):




*picture 4: screen 3 – the logo*

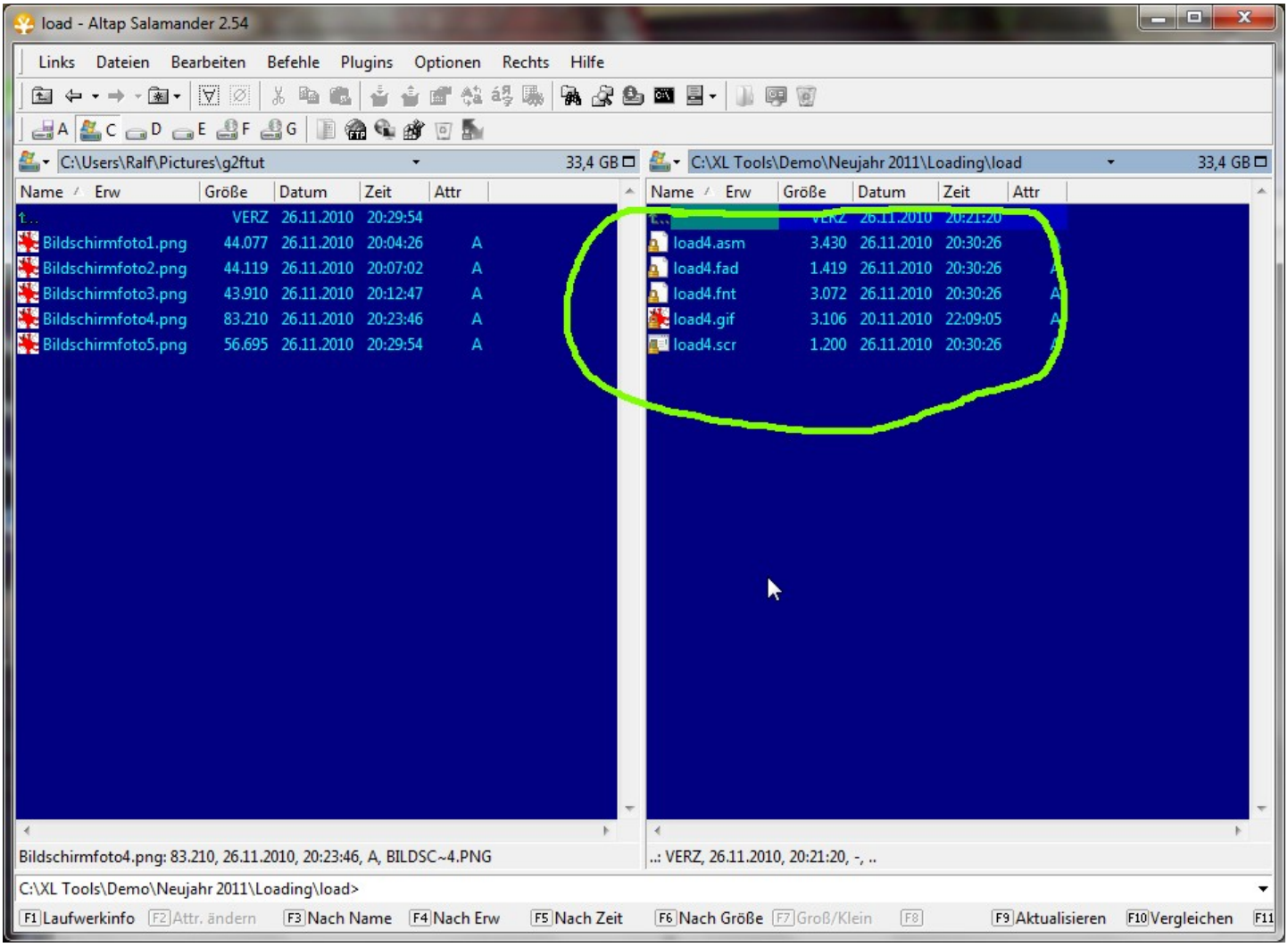
Now we have, what we want, so a click on the OK button is nice. I don´t want to edit the graphics now, cause the logo is still ok. But it might be useful to optimize the usage of chars and fonts to decrease the usage of memory. So click into Options / Optymizing:



*picture 5: screen 4 - optymizing*

Now a simple click on the save as asm will give us all, we need for the logo to use it at our own code.

The green marked files are the results:



*picture 6: screen 5 - resulting files*

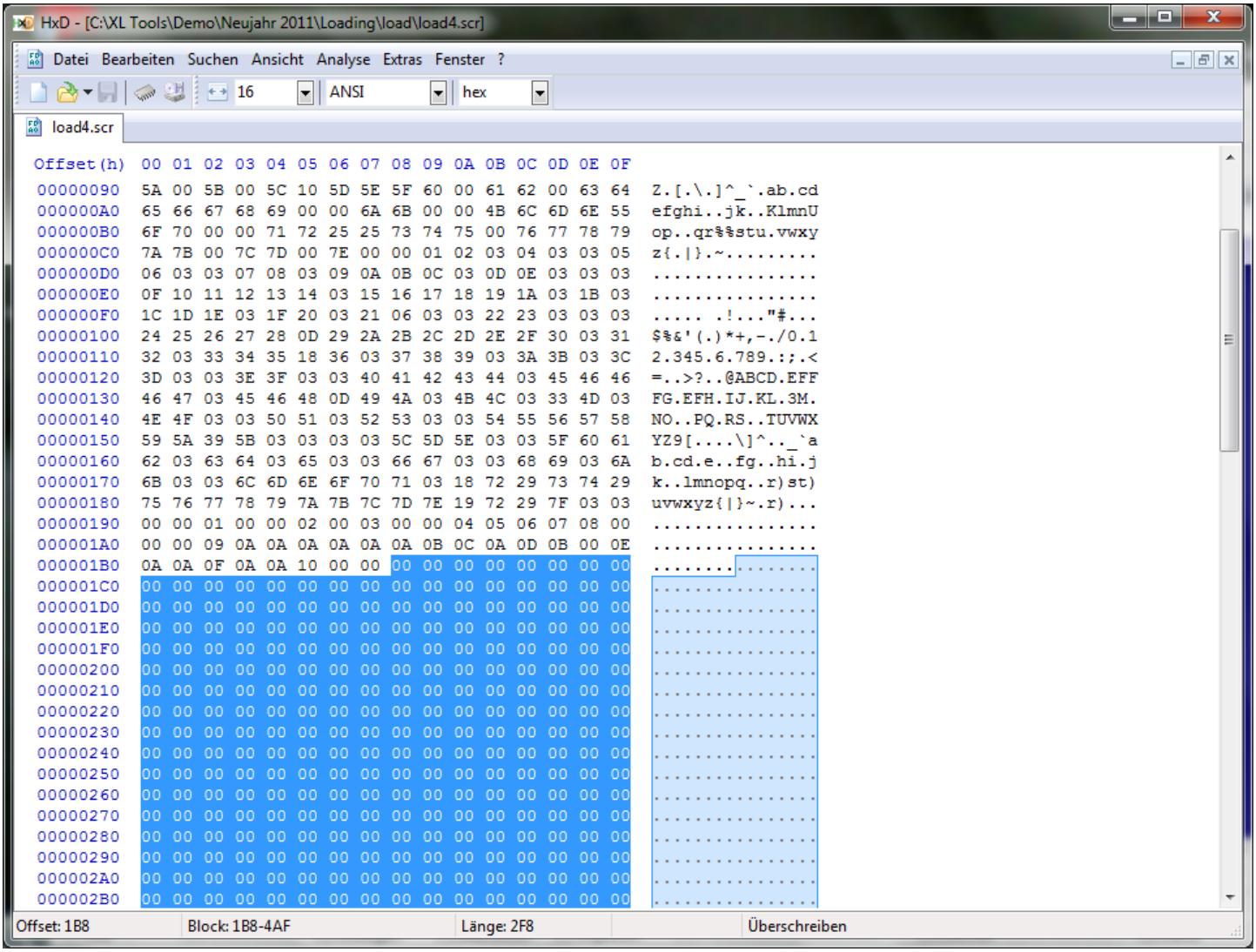
Now we have a asm file – simple use mads and you get a running program with your picture.
But we want to use the files for our own.

For that we need both, the fnt and the scr file.

The fnt contains all the needed fonts of our logo.

The scr has all the screen datas, we need → This must be on screen, to show the logo in right order. Here we got a file with more lines, than our logo finally has – as g2f saved all of the screen, so something must be cut off.

I use HxD (portable version) for that cuttings. I used 11 lines for my logo, so I only need the first 440 bytes of the scr file ($1b8 for HxD). The rest is filled with 0, and must not be on the file.




*picture 7: screen 6 - cutting*

The scr file is now 440 bytes long. Now we can start with coding.

I use mads for that, but even basic with little help of the right dli might be used.

```
        org $a000

ant     dta $F0,$70,$70,$70,$70,$70,$70,$44,a(scr)
        dta $04,$04,$04,$84,$04,$04,$04,$04,$84,$04
        dta $41,a(ant)

scr     ins 'load4.scr'

        .ALIGN $0400
fnt     ins 'load4.fnt'
```

Our program will start at $a000, and we put the dl there first, then we put in our screen data and at next usable location for charset, we put in the fonts.

The code for the DL is simple. First of all, we need dli, to set the charset and the colors. The logo should be in the middle of the screen, so we can use some empty lines. Then we have the first lma and antic 4 (graphics 12), then we give antic the address of the scr → so we are filling the screen with the needed chars.
The next lines, we can simply cut off of the created asm of g2f. We have the excact dli calls then. After 11 lines the logo is ready, so we end up the dl.

```
mainload


        mwa #ant 560        ; DL
        mwa #dli 512        ; DLI

        mva #$c0 $d40e      ; switch it on

l       jmp l               ; cut off for loader, just an endless

        rts                 ; not needed here, but for my loader
```

Now to the 'program'.

First we have to set the DL and DLI and then we switch them on. Nothing more to do here, so, if you want a loader, simply do the rts. To show you, the screen, as we do not load anything here, I set the endless jmp l here.

The DLIs and the init address setting:

```
dli     pha                 ; save a,y,x
        tya
        pha
        txa
        pha

        lda >fnt+$400*$00   ; set font
        sta chbase
c0      lda #$00            ; set colors
        sta colbak
c1      lda #$EA
        sta color0
c2      lda #$06
        sta color1
c3      lda #$0E
        sta color2
c4      lda #$00
        sta color3
        lda #$04
        sta gtictl

        mwa #dli1 512
        pla
        tax
        pla
        tay
        pla
        rti

dli1    pha
        tya
        pha
        txa
        pha

        lda >fnt+$400*$01   ; set font
```

```
        sta wsync
        sta chbase

        mwa #dli2 512       ; set next dli
        pla                 ; restore x,y,a
        tax
        pla
        tay
        pla
        rti

dli2    pha
        tya
        pha
        txa
        pha

        lda >fnt+$400*$02   ; set font
        sta wsync
        sta chbase

        mwa #dli 512
        pla
        tax
        pla
        tay
        pla
        rti


        ini mainload        ; use run for standalone program
```

Nothing more to do, than some register savings / restoring (to have a clean code, as documented by ATARI ;) ) and setting colors and chars.

The ini at the end is here, as I did a loading screen, normally a binary file sets it's running address, so use run instead ini then.

Hope, this little tutorial will help you using g2f screens in own productions.

Ralf Patschke, aká PP´s

---

all files included in this packages ;)

Many thanks to the authors of these nice tools – especially the ATARI related ones, I use very often!

Berlin, Germany, 2010-11-26