

# Raw Frame for Generic 16 KB cart images

```
; raw frame for generic 16 KB cartridge images
; v 1.0 enthusi 04/2012
; this 16 KB Cartridge framework was written for http://www.rgcd.co.uk
; feel free to use/change this code and give credits :)
; you will find this document also at http://codebase64.org
; this is a VERY simple but efficient approach, you can make more
; sophisticated usage of ROM using an own depacker routine etc....
; sources are in XA format but no special features are used
; I strongly recommend the usage of cartconv which comes with vice
; you can as well set up your own crt-header, which will look
; more or less like this:
```

```
;.asc "C64 CARTRIDGE  "
;.byte $00,$00    ;header length
;.byte $00,$40    ;header length
;.word $0001     ;version
;.word $0000     ;crt type
;.byte $00       ;extrom line
;.byte $00       ;game line
;.byte $00,$00,$00,$00,$00,$00 ;unused
;.asc "MY NAME"
;name
;.dsb ($0040-name),0
;;chip packets
;.asc "CHIP"
;.byte $00,$00,$40,$10 ;chip length?
;.byte $00,$00    ;chip type
;.byte $00,$00    ;bank
;.byte $80,$00    ;adress
;.byte $40,$00 ;length
;ROM part follows...
```

The following is for creating a .bin file of 16384 Bytes length as you would burn it onto EPROM, or to use as input for cartconv and alike.

```
;-----
; example usage
; xa frame.asm -o frame.bin
; cartconv -t normal -i frame.bin -n 'my cart' -o frame.crt
; x64 -cartcrt frame.crt
;-----

;no load-adress for bin-file, so no header here

*=$8000
.word launcher ;cold start
.word launcher ;warm start
```

```
.byte $c3 ;c
.byte $c2 ;b
.byte $cd ;m
.byte $38 ;8
.byte $30 ;0

launcher
    stx $d016
    jsr $fda3 ;prepare irq
    jsr $fd50 ;init memory
    jsr $fd15 ;init i/o
    jsr $ff5b ;init video
                    ;make sure this sets up everything you need,
                    ;the calls above are probably sufficient

    ldx #$fb
    txs

;set up starting code outside of cartridge-area
move_starter
    ldx #(starter_end-starter_start)
loop1
    lda starter_start,x
    sta $100,x
    dex
    bpl loop1
    jmp $100
;-----
starter_start
    ldx #$40 ;64 pages = 256 * 64 = 16384 Bytes
    ldy #0
loop
src
    lda exomized_data,y
dst
    sta $801,y
    iny
    bne loop
    inc src+2-starter_start+$100
    inc dst+2-starter_start+$100
    dex
    bpl loop

;make sure settings for $01 and IRQ etc are correct for your code
;remember THIS table from AAY64:

;      Bit+-----+-----+-----+
;      210| $8000-$BFFF |$D000-$DFFF|$E000-$FFFF |
; +---+---+-----+-----+-----+
; | 7 |111| Cart.+Basic |   I/O   | Kernal ROM |
; +---+---+-----+-----+-----+
```

```

; | 6 |110|      RAM      |      I/O      | Kernal ROM |
; +---+---+-----+-----+-----+-----+
; | 5 |101|      RAM      |      I/O      |      RAM      |
; +---+---+-----+-----+-----+-----+
; | 4 |100|      RAM      |      RAM      |      RAM      |
; +---+---+-----+-----+-----+-----+
; | 3 |011| Cart.+Basic | Char. ROM | Kernal ROM |
; +---+---+-----+-----+-----+-----+
; | 2 |010|      RAM      | Char. ROM | Kernal ROM |
; +---+---+-----+-----+-----+-----+
; | 1 |001|      RAM      | Char. ROM |      RAM      |
; +---+---+-----+-----+-----+-----+
; | 0 |000|      RAM      |      RAM      |      RAM      |
; +---+---+-----+-----+-----+-----+

```

lda #\$35 ;cart is always on instead of BASIC unless it can be switched off via software

sta \$01

jmp \$80d ;for exomizer, i.e.

starter\_end

-----

exomized\_data

.bin 2,0,"data.exo"

;syntax for exomizer 2.0.1:

;exomizer sfx sys game.prg -o data.exo

main\_file\_end

;fill up full \$4000 bytes for bin file (\$c000-\$8000=\$4000)

.dsb (\$c000-main\_file\_end),0

From:

<https://codebase64.org/> - **Codebase 64 wiki**

Permanent link:

[https://codebase64.org/doku.php?id=base:code\\_frame\\_for\\_16\\_kb\\_crt-images](https://codebase64.org/doku.php?id=base:code_frame_for_16_kb_crt-images)

Last update: **2015-04-17 04:30**

