Exomizer Memory Crunching / Decrunching for Beginners

Exomizer comes with a manual on how to compress data. However there may be people who are quite new to using the Exomizer tool, and the decruncher source, provided with Exomizer. I find Exomizer to be VERY useful for level packing single or multiload files. Some times decrunching binary/source data can be a right pain in the neck. It is also very flexible with decrunching memory.

Depending on the path used the command used (for example compressing a koala paint picture) is :

```
exomizer mem -l $2000 mypic.prg,$6000 -o mypackedpic.prg
```

This will load a koalapaint picture to $6000, then pack the memory to $2000. Exomizer then displays the pack result at the end of the crunching phase. If not using a crossassembler You must note down the END address in which the pack address lies. However if you are using a cross assembler, then you don't need to note the memory. Since you can use labels to indicate the end of the current load address.

Using Exomizer's Decruncher on a Cross Assembler

An example program (in ACME/C64Studio):

```
!to "koalapaintdisplayer.prg",cbm

  *=$1000
  sei
  ldx #<PicDepackEndAddress
  ldy #>PicDepackEndAddress
  stx $c010 ;Exomizer depack address low
  sty $c011 ;Exomizer depack address hi
  jsr $c000 ;Exomizer decrunch source

  ;Rest of pic display code....
  ... Type in your own pic display code
  *=$2000
  PicDepackStartAddress
  !binary "mypackedpic.prg",,2
  PicDepackEndAddress
```

Using Exomizer's Decruncher on a Native C64

What about if you wanted to use Exomizer when using a native C64 instead of a cross assembler. It is quite easy. Exomizer's level decrunch is very flexible when it comes to areas to place the crunched data (Providing the crunched data is NOT loaded on to memory that is being used for something else. A machine code monitor, such as the Action / Retro Replay fastload is helpful. Where there is spare memory in your program, you load the crunched data to a chosen address. Take note of the END address where the program lies. Then use the end address as a LOW/HI BYTE target. Then call the Exomizer to decrunch from that address. You can also load in the next crunched file one byte AFTER

the END address of the previous file. Note down the end address and so on.