Packers And Crunchers
by Richard/civitas/tnd

People been emailing me to ask for help on how to use a certain packer or cruncher. so i've writting a small and brief article on how to get started using a cruncher. for those who don't know how to use a cruncher program, don't call yourself thick, because you're not. this feature tells yo how you should use one and what the main purposes of cunchers or packers really is.

what is a packer?

when it comes to programming software on the good, old commodore64, in assembly language there comes a time, where you need to preserce disk space for other files. this is where a packer comes in handy. but not only programmers need to cope with them. think of a disknote that has just a few blocks too much to fit on your disk. now, checkout your cruncher and maybe it'll fit then .

basically, a packer is a program that compresses your programs and the rest of the data that have been written. a packer checks for wasted space, which is marked $00 - then it breaks the files down. once the packer scans and checks for wasted space, it squashes the program to the minimal size the packer can deal with.

packers are very useful for large programs. ml-programers have to take care of their files' sizes, but most packers can handle memory from $0400 to $ffff. two different types of packers follow:

the double disk checkin packer

this one works in 2 passes: it scans the program on your disk, it checks for op-codes in pass 1 (scanning) and then in pass 2 the packer starts packing your program by reading the disk again and checking the zeroes. an example would be the 'byg compactor+' and 'unipacker', which both do exactly this.

the quick 1-pass packer

there are also these types of packers, which only work in 1-pass mode. basically, the program loads from disk and then the packer quickly packs the program by using simple packing techniques. unfortunately, these packers are not as good as the previously mentioned versions. they're still useful. an example of this type of packer is 'sledgehammer'.

what is a cruncher?

a cruncher is similar to a packer, but a cruncher uses better compression rates. compared to packers, cruncher are slowerto use. a cruncher can only crunch a program, which has a maximum number of 242 blocks, else the cruncher won't work.

now, how does the cruncher work? basically, it reads the specified filename, which you have chosen and then crunches the program. the cruncher checks for any wasted space, which your program has left and then reduces the file more. a cruncher can take as long as five minutes to 3 hours, depending on the speed of the cruncher and the size of your file. there are two types of cruncher.

singlefile cruncher

a singlefile cruncher uses a 1-pass mode and crunches your file straight away.speeds may vary. examples for this kind of crunchers would be 'time cruncher','equal sequence' , 'power cruncher' and many others.

cruel cruncher

this one is named after its most prominent example. it uses a 2-pass mode instead of just one that is why it gives back better packing-results, but you would have to wait twice the time compared to a single-file-cruncher. a cruelcruncher scans the code first and then it crunches it down in the second passmode. examples might be 'cross crueller''crestcrueller' , 'the cruncher ab' or the 'byteboiler' ... and probably many others.

how to use packers/crunchers

now that we had the basics, this is the practically side most questions aim at. when you are using a packer or cruncher, you would need to know how to convert decimal into hexadecimal.
sounds easy, huh? well, it's no that easy, i'm afraid. here's a simple table on the conversion of decimal to hexadecimal:

0	$00	so when we're looking at
1	$01	this table beneath, we know
2	$02	that 10 is $0a and 16 is
3	$03	$10. but that's not all. 17
4	$04	would be $11 and so on. if
5	$05	you own an action replay
6	$06	cartridge or any ml-monitor
7	$07	then you're in luck as you in luck as you
8	$08	will be able to convert
9	$09	numbers with that tool. now,
10	$0a	what do we need this for?
11	$0b	all packers (with a few ex-
12	$0c	ception such as 'power
13	$0d	cruncher' which also takes
14	$0e	a wildcard for startaddy)
15	$0f	need to know
16	$10	address of the program we
17	$11	want to pack .