

Source code conversion

Do you have old turbo assembler sources lying around with PETSCII chars in them that look wrong when uploaded here? Do you have that bad habit of mixing spaces and tabs when you indent your code, so the code looks messy when uploaded here? Then you might want to know a bit about the possibilities for source code conversion that is out there.

Turbo Assembler source to ASCII conversion

There is a tool called [TMPview 1.3](#), which converts the binary format of turbo assembler source files to ASCII text, without forcing the user to go through the intermediate step of first exporting the source files in Turbo Assembler on C64 to SEQ format and then convert the SEQ files (PETSCII) to ASCII files.

VisAss & AssBlaster to ASCII conversion

Use [vis2ascii](#).

Hypra-Ass to ASCII conversion

The following bash script converts the encoded file to PETSCII using [cbmbasic](#):

```
printf "LOAD \"hypra_source.prg\"\nLIST\n" | cbmbasic | tail -n +11 | sed '$d' | LANG=C cut -d " " -f 2- > source.s
```

You can do this manually, by typing this into the cbmbasic shell:

```
LOAD"hypra source.prg"  
LIST
```

And redirecting or copy/pasting the output.

As a second step, you need to convert the PETSCII output to ASCII (see below).

And as a final step, use the following Python script to fix up the Hypra-Ass-specific indentation:

```
lines = [line.rstrip('\n') for line in open(sys.argv[1])]

for line in lines:
    if line[0] == ';':
        print line
    else:
        index = line.find(' ')
        if index == 0:
```

```
        print '\t' + line[1:4] + ' ' + line[4:]
    elif index == -1:
        print line
    else:
        print line[:index] + '\t' + line[index+1:index+4] + ' ' +
line[index+4:]
```

PETSCII to ASCII conversion

...but if you can't/don't want to use TMPview for some reason, then the following applies:

The first step in converting a turbo assembler source file to something that can be compiled on a crossassembler or posted on this site is to save your source as a SEQ file in Turbo Assembler (← W instead of ← S). Then convert the file from PETSCII to ASCII on your PC/Mac/Amiga

With Win32 ASCII <-> PETSCII converter

- [Win32 ASCII <-> PETSCII converter](#)

With petcom

...or you could use “petcom” by Craig Bruce, and it can actually also do the opposite (convert from ASCII to PETSCII).

- [petcom.c](#) - Zipped copy of petcom.c by Craig Bruce
- [petcom-bin-macosx.zip](#) - Executable binary for MacOSX
- [petcom.lha](#) - Executable for Amiga OS4.0 (compiled by Spot/UP)
- Please add more executable binaries for other platforms here...

```
USAGE 1: petcom petscii_infile.s > ascii_outfile.s
USAGE 2: petcom - ascii_infile.s > petscii_outfile.s
```

The file petcom.c is quite plain and short, so it should be easy to compile with a bunch of different compilers on different systems. For example, if you have the gcc compiler, you could use the command below to compile an executable binary:

```
gcc -o petcom petcom.c
```

In case you compile binaries that are not added on this page already, consider uploading them here for the benefit of others.

Source code indentation

Okay... So now you have an ASCII version of your code. If you want to (re)indent your code you can use a perl script named 6510i.pl that I wrote. It requires you to have perl installed on your system.

The script places labels at column 0 (you can also force addition of ":" at end of labels), and it places code at a column specified by the "-t NUMBER" parameter (default is 2 tabs), and tries to find out whether comments should end up in column 0, the code column, or a "third" comment column (specified by the "-c NUMBER" parameter). It can also remove trailing spaces/tabs at end of lines (using the "-x" parameter) and you can specify what the comment marker that is used in the source (if it is something else than the standard ";" character). In case you want the indentation to use space chars instead of tabs, you can tell this to the script by using the parameter "-n", for (n)o tabs. To set the number of spaces corresponding to a tab character in your text editor, use the "-s NUMBER" option.

DO NOT overwrite the infile with the output of this script, as it *might* mess up your source up. Better be careful...

- [6510i-v0.01.zip](#) - 6502/6510 asm indentation perl script by FTC.

```

USAGE: 6510i.pl [options [arguments]] INFILE
USAGE: 6510i.pl [options [arguments]] INFILE > OUTFILE
USAGE: cat INFILE | 6510i.pl [options [arguments]]

-t N      Tab column for code (default 2).
-s N      Number of spaces corresponding to one tab (default 4).
-c N      Tab column for comments after code. Must be greater than tab
column
          for code. (default codetabs+4).
-m STR    Specify comment marker (most use ";" but KickAssembler uses
"//").
-n        No tabs. Convert all tab characters to spaces.
-C        Force addition of colon to labels.
-x        Remove trailing spaces at end of lines.
-h        Print this help. Equivalent with --help and -?.

```

For example, to convert a turbo assembler source from PETSCII to ASCII, and indent the code to three tabs, you could do:

```
petcom petscii_src.s | 6510i.pl -t 3 > indented_ascii_src.s
```

Conversion from one assembler to the other

In case YOU wrote a script like acme2ca65.pl, or dreamass2kickassembler.pl and so on, it would be nice of you to add it here...

Noice Tasm to Kick Assembler converter

The noice online Turbo Assembler to Kick Assembler converter is available at <http://tasmtokickass.insoft.se/>. It is mainly used for converting old c64 Turbo Assembler source files in text format (import with [TMPview](#)) to Kick Assembler format. Most stuff used on real c64 turbo assembler should work. Newer stuff like macros does not work for now.

From:

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

Permanent link:

https://codebase64.org/doku.php?id=source_conversion

Last update: **2016-08-24 03:30**

