ZXText2P v1.0

What is ZXText2P?

It's an open-source command line utility that takes a text file containing a ZX81 BASIC program, and turns it into a ".P" file suitable for loading into your favourite ZX81 or TS1000 emulator.

Download

zxtext2p.zip - This zip file contains the C source code, a readme file, and versions of zxtext2p for **Windows** and **Linux**, as well as full documentation in HTML format.
zxt2dos.zip - Contains a 16 bit DOS binary for use with a 286 or better processor running MS-DOS. Use this version only if you have an operating system OLDER than Windows 95 (e.g. Windows 3 or MS-DOS 5).

How do I use it?

From a command prompt, type "zxtext2p mytextfile.txt". Assuming the text file contains a valid ZX81 program, you will get a file named "out.p" that you can use with an emulator (or transfer to tape and use with a real ZX81, using the appropriate tools).

There are a number of options that you can specify on the command line. They are as follows:-

Option	Meaning
-h	Show usage help.
-o output-file	Write the output to the named file (default="out.p").
-l	Use labels mode (see the section on labels, below).
-s line	In labels mode, sets the starting line number (default=10).
-i	In labels mode, set the line number increment (default=2).

For example, **`zxtext2p -l -s 100 -i 5 -o mygame.p gamesrc.txt`** will read the file *gamesrc.txt*, which must contain labelled code rather than code with line numbers, and will produce a .P file called *mygame.p* The first line will be numbered 100, the next 105, and so on.

Labelled Code

Using the "-l" command line option allows you to write labelled code. zxtext2p will produce the appropriate line-numbered output. Program labels must always start with an @ character, and must terminated with a colon (':'). Here is an example of labelled ZX81 code:-

```
   CLS
   PRINT "HELLO WORLD."
@loop1:
   PRINT "THIS IS A LOOP"
   GOTO @loop1
```

Escape codes, Graphics, and Formatting

Formatting
The text in the input file can be in any combination of upper or lower case. For example, this is perfectly valid input:-

```
10 scroll
20 PRINT "What is your name?"
30 Input n$
40 pRiNt "hEllo ";N$
50 StoP
```

Source Comments and Blank Lines
Any blank lines in the input file will be ignored by zxtext2p, as will any lines beginning with a '#' character. This means that you can include source-code comments in the input file and not have them take up any space in the resultant .P file. Any leading or trailing spaces on a line will also be ignored. For example:-

```
# MyGame - version 0.01
#
# By Fred Bloggs

   10 REM This is the first line that will appear in the output
   20 CLS
   30 PRINT "Welcome to My Game."
```

You can also split long lines by placing a single backslash ("\") character at the end of the line you wish to wrap:-

```
   1120 IF lives>3 AND score>20000 THEN \
        GOSUB 4200
```

Inverse Video
Inverse video (white on black) text can be included in your program by using the percent symbol ('%') as a character prefix. For example:-

```
   10 PRINT "%I%n%v%e%r%s%e% %V%i%d%e%o Normal Video"
```

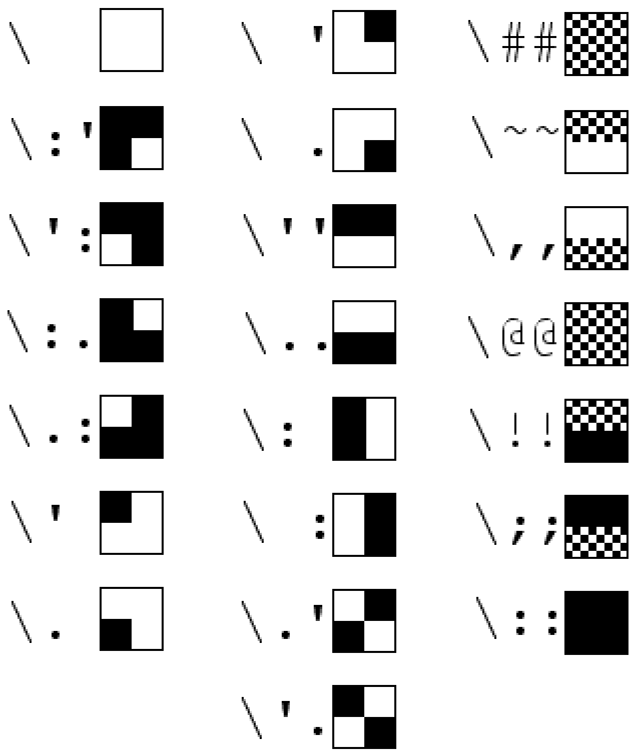
Escape Codes and Block Graphics
In BASIC, the quote symbol (") is used to delimit strings of text. If you want a quote symbol to appear in the middle of a string, you must therefore prefix it with a backslash ("\") escape character. For example:-

```
   10 PRINT "He said, \"Hello.\""
```

The full range of ZX81 block graphics characters can easiliy be inserted into a program by using the backslash ("\") escape character followed by a 2 character "ASCII-art" style rendition of the desired graphics character. For example:-

```
   10 PRINT "\:' \':"
```

The availble graphics characters, and their escape codes, are shown below. Notice how the pattern suggested by the dots/commas/apostrophes is used to form the appropriate graphics symbol.



Author: Chris Cowley
ZXText2P uses portions of code from zmakebas by Russell Marks