XTRICATOR

version 1.10

**** XTRICATOR ZX81 EMULATOR -- Officially Released Version 1.10 ****

by Carlo Delhez, August 1992

*** NOTE *** : A quick reference guide for getting to know XTricator in a short time can be found in the file 'Quick_txt'; first-time users may find it useful to print that file & follow the instructions there.

CONTENTS

INTRODUCTION

The ZX81 is the second computer marketed by Sir Clive Sinclair in the early eighties. In 1980, he introduced his first computer and called it 'ZX80'. In those days, computers were far from the 'common good' they are nowadays and the ZX80 was a relatively cheap machine. It was offered for around 100 pounds (and was marketed as 'the worlds first complete personal computer for under 100 pounds'), while its closest competitor (the VIC20 from Commodore) was about three times as expensive (not to mention the Apple machines of those days and the first-ever IBM MS-DOS PC with 64k RAM and 180k diskdrive which sold for about 3000 pounds). Although the ZX80 was quite a success (50,000 sold), it had some shortcomings, e.g. it could only do integer calculations (no floating point), the screen flashed at every keypress and went blank during execution of programs (this did increase speed but was not very 'ergonomic'). An upgraded version of the ZX80 appeared within a year. It had an 8k ROM (instead of the 4k of the ZX80), could do floating point mathematics and offered a choice between FAST operation with black screen (as on the ZX80) or SLOWer operation with normal screen. This updated version was called ZX81. It had a Z80A processor operating at 3.2 MHz, 1k RAM (yes! only 1024 bytes, but expandable to 48k), 32x24 text screen and 64x48 graphics resolution in black and white. Programs could be stored on tape and be retrieved at 250 baud. The first versions sold for 70 pounds (ready built) and 50 pounds (DIY kit); 50,000 ZX81's were purchased in the first 3 months after its release. In later years the ZX SPECTRUM was marketed (colour, sound, high resolution, a lot faster thanks to screen-hardware instead of -software, ASCII based character set and more expansion possibilities). Obviously, the SPECTRUM was a very good upgrade for the ZX81 and unfortunately most people sold their ZX81 to buy a SPECTRUM. Still some years later, the last computer from

Sinclair appeared, the QL (Quantum Leap). It was introduced as a serious business machine and did not add to the line of ZX products (mainly because the processor is an 68000 and not a Z80). Again unfortunately, most people stuck to their SPECTRUM and the QL was not a great success, despite its very advanced features like multitasking and its excellent and versatile operating system QDOS. The younger and older brother of the SPECTRUM are now united thanks to this ZX81 emulator for the QL: XTricator!

PROGRAM DESCRIPTION

XTricator is - stated very accurately - a Z80 Emulator in a Sinclair ZX81 environment, written for the Sinclair QL and its clones. The Z80 Emulator is able to execute programs written in the Z80 machine code language. The ZX81 environment includes the original ZX81 ROM and additionally caters for "hardware compatibility" (such as keyboard reading, screen output control and redirecting of signals originally heading for devices such as tape and lineprinter). The Z80 Emulator and the ZX81 environment are joined in a single block of 68000 machine code, which - as a whole - shall be referred to as ZX81 Emulator, and is named XTricator. XTricator supports multitasking and is designed to be compatible with all versions of the QL ROM, including Minerva.

NOTE: XTricator contains the original ZX81 ROM code. This code is copyright software. Therefore, XTricator may ONLY be used by you if you own a ZX81 YOURSELF. In that case, you are IN PRINCIPLE able to LEGALLY transport the ROM code from your ZX81 to your QL.
Any usage of XTricator by people NOT owning a ZX81 is in conflict with copyright laws.

REGISTRATION

XTricator is distributed as ShareWare. You are allowed to make copies for other QL users, provided you copy ALL the files and do NOT make changes in any one of these files.

If you like to use XTricator, please express your appreciation and support my efforts by becoming a REGISTERED user. You will then receive a laser-printed copy of this manual, you will be informed about new releases, and can update to a more recent version for a small fee just to cover my costs. Registration also includes conversion of your programs from tape to disk, free technical support by the author and the possibility to obtain a custom-made version of XTricator to suit your specific hardware configuration. Additionaly, you get access to a huge ZX81 software library (over 300 programs available already). Registered users will receive the first two disks out of the series (containing 160 original ZX81 programs in total!) and can obtain successive disks against just the cost-price. All programs on the disks are tested on XTricator and documented in separate text files.

To become a registered XTricator user, send a EuroCheque or money order worth Hfl 50 (Dutch Guilders) made payable to J.L. Delhez, plus a note stating your name, address and QL configuration (or compatible) to me (address at the bottom of this document). The most recent version of XTricator on a 3.5" DS/DD floppy (3M) and a printed copy of this manual and the library disks

will be dispatched to you. Please do not send cheques in a currency other than Dutch Guilders. If you wish, you may also send cash in any currency, but please add 15% for exchange costs.

COMPATIBILITY

Since the original ZX81 ROM in Z80 code is included (with only a few minor changes, of no interest for ordinary use) almost complete software compatibility is guaranteed. Almost, since only some VERY specific software can obviously NOT be emulated: programs using either standard or non-standard input/output ports (e.g. the ear/mic-lines and lines connected to external hardware devices not available on the QL such as sound-generators or EPROM programmers; note however that the keyboard and printer input-lines are emulated correctly), and programs taking over the interrupt vector (e.g. ZX81 sprite programs, although the characterset can be redefined in the usual way and also Hi-Res is available) are NOT supported by XTricator.

Apart from these exceptional applications, ANY program running on the ZX81 should ALSO run on XTricator.

COMPARISON

XTricator has some advantages and disadvantages with respect to the ZX81. To name just a few advantages: safe housing in a well-stabilized computer (none of the tricky business with wobbling RAM packs), full 64k RAM available (including POKEable ROM and 8-16k area), reliable and fast data-storage (no more tapes!), the ZX81 display file can be moved to any part of memory without a crash, machine code can also be executed in the top 32k of memory (on the ZX81 only the bottom 32k could be used), easy screen invert option, quick reset etc. etc. Additionally, XTricator happily MultiTasks, so you can run several copies of XTricator at a time, e.g. one copy to assemble a program, another copy to play a game meanwhile (provided you have enough memory in your QL).

If you have a standard 8 MHz 68008 QL, a main disadvantage is SPEED! A common property of all emulators known to man is the general lack of speed. Since single Z80 instructions are being translated in at least six and sometimes as much as tens of 68000 instuctions, XTricator is bound to be slow. Luckily, the ZX81 usually runs in SLOW mode, in which a giant portion of CPU time (about 75%) is consumed by the screen-interrupt, making the Z80 run effectively about four times slower. Thanks to this SLOW mode, XTricator is capable of performing at about 30% of the ZX81 in SLOW mode. Obviously, in FAST mode (which neglects interrupts), this decreases to one fourth, i.e. 8% (XTricator has no real FAST mode; QL I/O and interrupts are not that expensive on processor overheads). Although 30% may seem terribly slow, this speed is sufficient for running most (machine code) games and utilities. In fact, the ZX81 is quite a fast computer and nearly all machine-code games use fixed delay-loops for proper human operation. Taking out these delay loops (or even better: making them processor speed independent) will make most games as fast and exciting as they were on the ZX81. Experimenting a little, I found that professional machine-code utilities (like TFO DataBase, Artic Assembler, Campbell DisAssembler and Coral Basic Interpreter) hardly suffer from the

decrease of speed, since most of the time you are either typing something or reading output produced by the programs.
Additionally, BASIC programs can be compiled with an existing ZX81 compiler (e.g. MCODER-2), often resulting in a giant speed gain.

If you are so lucky to own a Gold Card with a 68000 processor running at 16 MHz, speed is no longer a problem and XTricator becomes a real THRILL. The emulator then runs at a speed somewhere between 100% and 200% of the original ZX81 (depending on priority settings). The same should apply if you own a QL Emulator for the Atari ST. More details on the effective clock frequency are given later on.

STARTING XTRICATOR

For running XTricator, the following hardware is required:

* A QL (or clone) with at least 256k of RAM memory to store the program and for allocation of the required work space.

The following software is optional, but highly recommended:

* The well known Window Manager (and Pointer Interface) environment to have maximum flexibility in MultiTasking.

* Toolkit-II by Tony Tebby, mainly for Job Management.

Compatible hard- or software is ofcourse allowed as well. Obviously, for reasons of copyright, the Window Manager and Toolkit are not included in the XTricator package.

If you do not use the Window Manager environment, many of the possibilities discussed below do not apply for you. You should use EW (or EXEC_W) instead of EX, but then you cannot use CTRL-C or change priorities. However, XTricator runs a little bit faster!

Before starting XTricator, make sure Toolkit-II and the Window Manager are present (or create a simple boot-program to do this; you could, for example, edit the file 'XTricator_Xample_Boot'). Next, install the QZdd (QZ device driver) by typing:

```
LRESPR FLP1_QZDRIVER_BIN
```

and then:

```
EX FLP1_XTRICATOR_EXE
```

(or use an equivalent hotkey/button-command when you are using the QJUMP extended environment.)
This will start up XTricator, showing a dark-red screen with a black square in the middle. You should also hear a modest beep. The black square is the main ZX81 screen. This black block remains black for a short while. In this period the ZX81 memory is cleared, the System Variables are initialized and the ZX81 Display File is expanded to full size. A pause like this also happens when you switch on an ordinary ZX81 with more than 3.25k of RAM. Finally, the famous inverse-K-cursor appears and XTricator is ready for use.

Before typing anything, return to SuperBasic by pressing CTRL-C. Now type the command JOBS. You will see that XTricator consists of three jobs, viz.:

* XTricator : This is the main part of the emulator, performing the actual translation from Z80 to 68000 code.

* XTr_IO : This job is concerned with all input/output activities, e.g. reading of the keyboard and displaying the screen.

* XTr_Vita : The Spirit of XTricator; as soon as this job is removed, XTricator kills itself (more about that later).

The first job is at priority 64. The priority of 'XTr_IO' is either 4 or 16. When XTricator detects a total system memory of 1920k, it assumes you are using the fast Gold Card and a I/O priority of 16 is used. In all other cases, the value 4 is used. Please note that issuing (for example) 'RES_SIZE 896' on a Gold Card before starting XTricator will lead to priority 4 since XTricator sees a system memory of 'only' 896k. Ofcourse, the priorities can still be changed to suit a required computing/display ratio.
The priority of 'XTr_Vita' should always be 0. Just for fun, try:

```
SPJOB XTR_VITA,127 : JOBS
```

Besides the above three jobs, also a 50 Hz interrupt has been linked to keep track of the ZX81 internal clock (Frames, address $4034). Because of this interrupt, keep in mind NEVER to delete either the 'XTricator' or the 'XTr_IO' job. The ONLY proper way to stop XTricator is

```
RJOB XTR_VITA       ( or equivalent )
```

As soon as you switch to XTricator by means of CTRL-C, XTricator will kill itself, closing all channels, releasing its work-space and also unlinking the 50 Hz interrupt. Any other way of removing XTricator will most likely crash your QL.

In Pointer Interface terms, Z80 emulation (XTricator job) is completely "unlocked" as it is not connected to any window or keyboard queue. So, when XTricator is temporarily buried by another job, the Z80 happily keeps on running, unless it needs to wait for a keypress which it requests from the (suspended) XTr_IO job. Please note that the ZX81 ROM prints to the ZX81 Display File while XTr_IO caters for the conversion from ZX81 DF to QL screen!

Having read all this, switch back to the ZX81 screen (use CTRL-C) and start having fun!

SCREEN OUTPUT

The screen output frequency is determined by the priority of XTr_IO. The wobbling 'XTricator' at the bottom righthand corner of the screen helps you determine this frequency. If this text makes one oscillation in one second, the screen output frequency is 12 Hz. On a 68008 based QL, the frequency

defaults to about 2.5 Hz (1 oscillation in 5 seconds) and should not be increased too much as that slows down Z80 emulation considerably.

On a Gold Card, you can easily play around with the setting of the XTr_IO job. Making the value lower than 16 will speed up Z80 emulation but you get less screen updates per second, so movements become more chunky. A value higher than 16 will reduce Z80 speed performance but screen changes are seen more smoothly. Some examples:

```
SPJOB XTR_IO,4    : slower screen         : speed approx. 170%
SPJOB XTR_IO,16   : the default          : speed approx. 130%
SPJOB XTR_IO,127  : extremely fast screen : speed approx.  75%
```

KEYBOARD FACILTITIES

A problem for many users may be the keyboard: the ZX81 works with single keypress entries (e.g. pressing 'P' gives 'PRINT'). However, it is not really necessary to have a ZX81 or ZX81-manual near: just press CTRL-F1 to see the full ZX81 keyboard lay-out with all characters, keywords, functions and graphics. Press F1 (no CTRL needed this time) to return to the 'normal' ZX81 screen. The help screen is a scanned image of the original ZX81 keyboard.

As an example of keyboard operation, take a look at the A-key on the help screen. It contains the following symbols:

- the letter A itself
- the command NEW
- the command STOP
- the function ARCSIN (abbreviated ASN)
- a gray block

Pressing the A-key will produce one of these symbols, depending on the current setting of the cursor:

cursor	keypress	effect
K	A	NEW
K or L	Shift-A	STOP
L	A	A
F	A or Shift-A	ARCSIN
G	A	inverse letter A
G	Shift-A	gray block

The K-cursor (Keyword) always appears at the beginning of a command line and after the THEN keyword; the L-cursor (Letter) appears elsewhere. The F- and G-cursors (Function and Graphics) can be activated by the user by pressing Shift-Enter and Shift-9, respectively.

The default XTricator screen color is white characters on black background (on the ZX81 you have the inverse). For text handling, I find this quite comfortable for the eyes. For games with a lot of graphics, the inverse may look better. Press CTRL-F2 to change the display from white on black to black on white and vice versa.

Another non-standard keyboard function is CTRL-ESC, which is in fact a 'super-break' facility. Pressing CTRL-ESC will stop the current program (BASIC or machine code) and gives error message D after the ESC-key has been released. Avoid using CTRL-ESC in the editor-mode or while using the Coral Basic TRACE option. In general, it is best to use CTRL-ESC only if there is no other way of stopping a program.
CTRL-ESC may cause a ZX81 crash when used from within a machine code program. Although this rarely happens, please press CTRL-ALT-ESC (see below) to cure the crash.

NOTE: pressing CTRL-ESC in the regular ZX81 editor will cause a 'keyboard lock'; please press CTRL-ESC once more to re-activate the keyboard. In the CBI editor, CTRL-ESC merely produces a Space.

A full ZX81 system reset (RST 0) can be obtained by pressing CTRL-ALT-ESC. This refreshes the ZX81 ROM area (i.e.: the default contents of the 0-8k area are restored), and a RST 0 is executed (i.e.: RAMtop is set to 32768 and the bottom 16k of RAM is cleared). This is the safest way for recovering after some crash or for just restarting the ZX81. Note that the 8-16k area and the top 32k of RAM are not altered in any way.

If you are working with a Hi-Res program, there is an easy to way to switch from the Hi-Res screen to the normal ZX81 screen (e.g. to type a command or to view an error message). Press CTRL-F3 to toggle between both screens. If you wish to quit Hi-Res altogether, it is advised to use CTRL-ALT-F3 instead. This will tell XTricator that the Hi-Res is not needed anymore. Please refer to the section "High Resolution" for more details.

The CTRL-keypresses should always work IMMEDIATELY, whatever the ZX81 is doing at that moment (yet not during QZ file command handling). A modest beep will be generated when the keypress is recognised by XTricator; you can then safely release the key(s) and await the desired the result.

ConTRoL Summary:

CTRL-ESC	Refresh ZX81 ROM, set default character set and abort current ZX81 program (BASIC or M/C) with error 'D';
CTRL-ALT-ESC	Refresh ZX81 ROM and reset ZX81;
CTRL-F1	Display graphical keyboard help screen;
CTRL-F2	Invert screen from b/w to w/b and v.v.;
CTRL-F3	Switch from Hi-Res to Lo-Res and v.v.;
CTRL-ALT-F3	End Hi-Res session, resume in Lo-Res and set default character set.

Note that some PC keyboard interfaces for the QL (like QL-Keyboard-90 from Jurgen Falkenberg) have CTRL-F1 and CTRL-F2 mapped on F11 and F12 of the PC keyboard. These keypresses work just fine with XTricator.

KEYBOARD COMPATIBILITY

A lot of effort has been put in interfacing between the QL keyboard and the ZX81 keyboard. This is not quite as easy as it may seem. As an example, take

the comma-character (','). On the QL you have a dedicated comma-key, but on the ZX81 you have to press SHIFTed full-stop ('.'). Now suppose you press SHIFT-'.' on the QL. What is intended: a comma (as on the ZX81) or '>' (as on the QL)? The meaning may well depend on the program currently running. In text-handling programs, a '>' seems most useful, but for games (which allow you to press many keys at once for complicated steering mechanisms) you probably want ',' as a reply.
Another problem is (for example) SHIFT-'9'. On the ZX81 this is used to toggle Graphics Mode, but on the QL keyboard it gives ')'. So, another key should be used for Graphics, e.g. SHIFT-'G', but on the ZX81 this gives 'LLIST'....
Confused?

To overcome all these difficulties, TWO Keyboard Emulator Modes are available. These are distinguished by pressing or not pressing the ALT-key, and the default emulator (i.e. the one NOT using ALT) can be toggled by CAPSLOCK (see little 'LED' at the top of the screen). The two emulator modes have many keys in common:

BOTH KEY EMULATOR MODES
---

* All unSHIFTed keys will appear as they are shown on the QL keyboard.
* ENTER, the Cursor Keys and Space Bar respond as you expect.
* CTRL-Left functions as RubOut (ZX81 SHIFT-'0').
* Special keys : '[' gives '<'
  ']' gives '>'
  ''' gives '"'

KEY EMULATOR MODE 1 : if CAPSLOCK OFF : without ALT
--- if CAPSLOCK ON : with ALT

All SHIFTed keys will appear as on the ZX81 keyboard.

Examples: SHIFT-'B' gives '*'
SHIFT-'.' gives ','
SHIFT-';' gives SHIFT-SHIFT-'X', i.e. just ';'

KEY EMULATOR MODE 2 : if CAPSLOCK OFF : with ALT
--- if CAPSLOCK ON : without ALT

All shifted keys will appear as on the QL keyboard. Consequently, specific ZX81 keypresses (i.e. SHIFT-'9' for Graphics) do not work.

Examples: SHIFT-'8' gives '*'
SHIFT-'.' gives '>'
SHIFT-';' gives ':'

In this mode, some special keypresses are available:

SHIFT-'6' gives '**' (instead of '^')
SHIFT-'7' gives 'AND' (instead of '&')
SHIFT-'\' gives 'OR' (instead of '|')
SHIFT-'[' gives '<=' (instead of '{')
SHIFT-']' gives '>=' (instead of '}')
SHIFT-''' gives '""' (instead of '"')

Note how the QL single quote (''') is converted to a ZX81 double quote ('"') whereas the QL double quote is converted to a ZX81 quadruple quote ('""'). XTricator effectively doubles the number of quotes!

It is advised to experiment a bit to see which keys do what. In general, CTRL-F1 together with Key Emulation Mode 1 should cure most of your keyboard problems!

FILE MANAGEMENT

Originally, the ZX81 did not have a Disk Operating System. Programs could be stored on tape only (at a rather dull rate of 300 baud, taking more than 7 minutes to save or load a 16k program). As an alternative, several diskdrive systems were introduced by various companies but none of these were very successful. People preferred to keep using tapes, and software was written to speed-up tape data communication to, say, 6,000 baud (20 seconds for 16k). This was quite acceptable and a single tape of 60 minutes could contain up to 2.5 M information. However, tapes are not a very reliable medium and the sequential storage of tapes obstructs quick access. As a solution, I used an RS232 interface to connect my ZX81 to my QL, offering more advanced storage possibilities (diskdrives) and being very well suited to act as a fileserver thanks to its multitasking capabilities. I wrote this fileserver myself, as well as the software needed on the ZX81 to do the communication with the QL. Data could now be exchanged at a rate of 9600 baud (13 seconds for 16k) and could be accessed in a quick and simple way. I called this serial communication system "QZ", which served as an example for the "Disk Operating System" offered by XTricator.

QZ was first launched December 1989, version 2.0 introduced August 1991. A detailed description regarding the possibilities of QZ and its device driver is omitted here. If you are interested in QZ, please contact me in writing (address at the bottom of this document). A document on the QZ device driver is included as the file 'QZdriver_txt'.

The default file storage and retrieval device is FLP1; this is automatically 'QZ enabled' (i.e. XTricator performs a 'QZ_ON flp1_'). For using another device, first issue a 'USE:' command (see below).

To SAVE or LOAD a program, just type

```
SAVE "<name>"
LOAD "<name>"
```

which are the equivalents of the QZ commands

```
SAVE:P:<name>:
LOAD:P:<name>:
```

LOADing a non-existing program results in Error F. If you SAVE a program which already existed, it will simply be overwritten without any user confirmation. Loading or saving a file with a NAME of length zero also results in error F.

Since QZ filenames are at most 12 characters long, only the first 12 characters of the SAVE/LOAD name will be used. Do NOT include INVERSE characters in the name, since these are used internally by the ZX81 to indicate the end of the filename.

Remember that XTricator does not really have a distinct FAST or SLOW mode; programs are always SAVEd in such a way that they are in SLOW mode when they are reLOADed on the original ZX81. So, take care when a program requires FAST mode for proper operation on the ZX81!
For safety reasons, the CBI interpreter (when active) is reset after loading a program. Again, take care if the loaded program requires (for example) 'NOSTALGIC OFF' mode.

To see the QZ files on FLP1, you must type the QZ command DIR. You can do this either in a LOAD or in a SAVE command, but the DIR-command MUST be preceeded by an asterisk ('*'). So:

```
          SAVE "*DIR:"
and
          LOAD "*DIR:"
```

will both show a (paginated) directory of the QZ files on current device.
The '*' indicates that a command is given instead of a filename.
Consequently, a filename cannot begin with '*'!

The commands currently available are discussed below. For a more detailed description, refer to the most recent QZ manual.

```
--- BACK: ; Go one step back in directory tree structure.
          ; Example : LOAD "*BACK:"

--- CD:   ; Change from current directory to indicated subdirectory.
          ; Example : LOAD "*CD:GAMES:"

--- DEL:  ; Delete a file; no error is reported when the file does not exist.
          ; Example : LOAD "*DEL:P:NONSENSE:"

--- DIR:  ; Show (paginated) overview of files in current directory; at the
          ; end of each page but the last, press 'Q' to stop or any other
          ; key to view the next page.
          ; Example : LOAD "*DIR:"

--- KILL-XTR:  ( not really a QZ command )
          ; End the current XTricator job; see section 'Stopping' for
            more details.
          ; Example : LOAD "*KILL-XTR:"

--- LOAD: ; Used for LOADing special files (NOT 'P'-files). All the special
            filetypes are supported: 'B' (for 'Basic'), 'C' (for 'Code'),
            'L' (for 'Lines'), 'S' (for 'Screen') and 'V' (for 'Variables').
          ; Examples:
```

```
                (1) LOAD "*LOAD:B:VARTEST:"
                     deletes all the lines of the current BASIC programs and
                     then merges the program in the file 'VARTEST.B'; note that
                     the BASIC variables are not influenced! The LOAD:B command
                     always generates Error I (data Inserted).

                (2) LOAD "*LOAD:C:NAMES:40000:"
                     loads the file 'NAMES.C' to address 40000;
                     during LOAD, address 65535 is never exceeded, so the
                     maximum number of bytes that can be loaded is (in this case)
                     65536 - 40000 = 25536, and similarly for other addresses.

                (3) LOAD "*LOAD:L:SUBROUTINE:"
                     merges the lines in the file 'SUBROUTINE.L' with the current
                     BASIC program; if a line already exists, it will be replaced
                     by the new line; note that the BASIC variables are not
                     influenced. The LOAD:L command always generates Error I
                     (data Inserted).

                (4) LOAD "*LOAD:S:READ-ME:"
                     loads the file 'READ-ME.S' into the display file; the
                     display file must be fully expanded; this is not checked!
                     An 'S' file is always 793 bytes long.

                (5) LOAD "*LOAD:V:DATABASE:"
                     clears the entire variables area and LOADs the variables of
                     the file 'DATABASE.V'; this is NOT a MERGE facility; the
                     LOAD:V always generates Error I (data Inserted).

            ; Please note the significant difference between LOAD:B and LOAD:L !
            ; LOADing filetypes 'B', 'L', 'P' or 'V' may cause a ZX81 crash if
              there is insufficient memory left below RAMtop to insert the file;
              please check available memory before attempting to LOAD; in case
              a crash occurs, recover with CTRL-ALT-ESC.

--- MAIN: ; Go to main (root) directory of current device.
          ; Example : LOAD "*MAIN:"

--- MD:   ; Make (create) indicated subdirectory on current directory.
          ; Example : LOAD "*MD:UTILITIES:"

--- RD:   ; Remove indicated subdirectory from current directory.
          ; Example : LOAD "*RD:TEMPORARY:"

--- REMOVE:
          ; Equivalent to 'DEL:'; See there!
          ; Example : LOAD "*REMOVE:P:TEMPORARY:"

--- ROOT: ; Equivalent to 'MAIN:'; See there!
          ; Example : LOAD "*ROOT:"

--- SAVE: ; Used for SAVEing special files (NOT 'P'-files). All special file
            types are supported: 'B' (for 'Basic'), 'C' (for 'Code'), 'L' (for
            'Lines'), 'S' (for 'Screen') and 'V' (for 'Variables'). The
```

```
            'SAVE:' command overwrites an existing file with the same name
            (if any) without user confirmation.
          ; Examples:
            (1) LOAD "*SAVE:B:PLEASE:"
                 Saves all the BASIC lines currently in memory to the file
                 'PLEASE.B'; note that system variables, BASIC variables and
                 screen are NOT written to file!

            (2) LOAD "*SAVE:C:BINARY:20000:8192:"
                 Saves 8192 bytes of memory, starting at address 20000, to the
                 file 'NAMES.C';
                 The last address of the SAVEd code block may not exceed
                 address 65535; in that case an error message is given.

            (3) LOAD "*SAVE:L:SECTION:450:780:"
                 Saves the lines of the current program with line numbers
                 between 450 and 780 (both inclusive) to the file 'SECTION.L';
                 if no lines exist is this region, an empty file is created.
                 Note that the first line number must be less than or equal to
                 the last line number; the last line number is smaller than
                 10000; the line numbers itself need not exist.

            (4) LOAD "*SAVE:S:PICTURE:"
                 Save the current display file to the file 'PICTURE.S';
                 an 'S' file is always 793 bytes long, i.e. it is assumed
                 that the display file is fully expanded

            (5) LOAD "*SAVE:V:DISCOGRAPHY:"
                 Save the contents of the variables area (including the CHR$
                 128 at the end) to the file 'DISCOGRAPHY.V'.

--- SHOW: ; Shows information about the current version of XTricator and
            presents a list of the available QZ commands.
          ; Example : LOAD "*SHOW:"

--- USE:  ; Set default device for LOAD, SAVE and QZ file-handling commands.
          ; The device name is four characters long (no underscore), and can
          ; be FLP1 through FLP8 or RAM1 through RAM8.
          ; The previous device is 'QZ disabled' (QZ_OFF), the new device is
          ; automatically 'QZ enabled' (QZ_ON).
          ; Example : LOAD "*USE:RAM1:"

It is allowed to use more than one command in a single LOAD or SAVE. Just use
a semi-colon (';') in-between two successive commands, e.g.

          LOAD "*USE:RAM2:;CD:UTILS:;MD:SOURCES:;DIR:"
or
          SAVE "*SAVE:S:SNAPSHOT:;LOAD:C:MACHINE:31000:"

Some other QZ commands are not yet available (e.g. BUFFER, CUT and RENAME).
Any non-supported command after the '*' will result in Error C (Syntax Error).
If a standard QDOS TRAP used by the QZ interpreter returns with an error, the
ZX81 screen is cleared, the erroneous command is shown and you are told which
QDOS error occured.
```

If you like, you also issue the directory management commands from QDOS instead of from XTricator. E.g.: if the current QZ device of XTricator is FLP1, then typing 'QZROOT flp1_' in QDOS has the same effect as typing 'LOAD "*ROOT:"' in XTricator (both commands are guided to the QZdd!).
You can even let the ZX81 (via the QZ FileServer) and XTricator share the same device, so that programs just SAVEd by XTricator can be LOADed directly into the ZX81 and vice versa without removing any job.
Finally note that string expressions after SAVE or LOAD instead of a simple quoted text are also allowed.

WARNING 1: It is advised not to stuff your disks with files, but to leave at least 4k (9 sectors) free for device driver work space. For example, the DIRectory command creates a temporary file, so it fails if insufficient disk space is available.

WARNING 2: Do not try to use THIS version of XTricator with disks created by QZ fileserver version 1.3 or lower or by XTricator version 0.79 or lower; they are completely INCOMPATIBLE! Please use CONVERT2 first.

WARNING 3: The file management system writes all information (e.g. directory and error messages) directly to the ZX81 screen, NOT to the QL screen. For this, it expects a fully expanded display file.
If the display file is not entirely expanded (e.g. after a SCROLL statement or with RAMtop below address 19712), you will hear a modest beep and error 4 is displayed, indicating that there is not enough screen space for the required information!

PRINTER OUTPUT

A simple printer driver is available to emulate the ZX81 printer output. The printer is assumed to be connected to SER1 (no optional parameters). The required baud rate can be set in SuperBasic (before starting XTricator or just before printing). The produced output is in ASCII only (no printer control codes) and a simple LF (ASCII 10) is given at the end of each line (i.e. no CR, ASCII 13).

Inverse-video ZX81 characters are converted to non-inverted ones and ZX81 graphics symbols are printed as spaces (so you can draw them yourself).

The ZX81 commands COPY, LPRINT and LLIST work as expected. Also, programs using the appropriate ROM routines for printing (not Hi-Res!) or programs setting bit 1 of address $4001 before using RST $10 function perfectly.
It cannot be avoided that listings are printed in the awkward 32-column format. However, each line is preceeded by 16 spaces for a better page lay-out.

If you send data to the printer whilst no printer is connected, XTricator will persistently try to send data anyway. Press <ESC> to tell XTricator that no printer is connected. Note that this means that no printer output is generated by XTricator anymore for as long as the current job lasts.

HIGH RESOLUTION

Sir Clive Sinclair himself said that it could not be done...

In 1983, a first attempt to High Resolution on the ZX81 was presented when the High Resolution Toolkit by Richard Taylor was marketed. Later, companies like Software Farm produced several games in High Resolution. Taking a closer look at this 'High Resolution', one can easily see that it is not real Hi-Res: for the 256 bit patterns which are possible for 8 successive pixels, at most 128 are available, i.e. more than half of the patterns cannot be used. The result of this Semi Hi-Res may however still be quite satisfactory.

Hi-Res is detected by XTricator by trapping an immediate change in value of interrupt register IX. The term 'immediate' means that only the instruction 'LD IX,nn' ($DD21) is trapped. You can avoid the trap by using a sequence like 'LD HL,nn / PUSH HL / POP IX'. However, all Hi-Res programs tested use the (more obvious) immediate method.

After the detection, XTricator first tries to emulate the Semi Hi-Res by looking for the characteristically shaped 6k display file in the bottom 16k of RAM and printing it to the QL screen appropriately.

If XTricator is unable to locate this display file, it will simply assume that you are going to use the (real!) Hi-Res driver supported by Extended Basic Level 3 (author: Frits Beniest) and will set the display file at address $2800 (10240). This Hi-Res gives access to a full 256x192 resolution in black and white.

Non-Hi-Res interrupt servers (like an on-screen real-time clock and sprites) will obviously cause this Hi-Res display to be switched on. Pressing CTRL-ALT-F3 will solve the problem. Please note once more that such deviating interrupt servers are NOT supported by XTricator.

When XTricator is in any of the above Hi-Res modes, the normal ZX81 display cannot be seen. Press CTRL-F3 to toggle to the normal screen and back! Alternatively, Hi-Res can be stopped completely by pressing CTRL-ALT-F3. This will switch you back to normal Lo-Res and will also set the default character set (0 in Extended Basic Level 3).

Some notes for Extended Basic Level 3 (EB3) users are given below:

* The Hi-Res screen is emulated correctly, but some of the EB3 commands cannot be used. These are:

    --- SDP / LDP (Save and Load DisPlay); Use QZ commands instead:
        SDP ... SAVE:C:<name>:10240:6144:
        LDP ... LOAD:C:<name>:10240:

    --- SKS / LKS (Save and Load Kharacter Set); Use QZ commands instead:
        SKS ... SAVE:C:<name>:8192:2048:
        LKS ... LOAD:C:<name>:8192:

    --- SPEED ; used for 'flashing characters', not emulated by XTricator.

--- IN, OUT, HR/COPY ; neglected by XTricator.

* In his manual, Frits Beniest says that pressing SHIFT during 6 seconds will switch you back to Lo-Res. On XTricator, just use CTRL-(ALT)-F3 (see above).

As fas as I know, only HRWP (Hi-Res Word Processor) does not function properly on XTricator, but I'm working on that!

INCOMPATIBILITIES

Some programs behave strangely when you RUN them under XTricator. Generally, these programs are to blame, NOT XTricator! Known 'bugs' are listed below:

*** FIGFORTH (Version 1 by Artic) does not seem to work at all. The reason is a badly programmed memory test. FIGFORTH checks memory from bottom to top. Two conditions should stop the memory check: (1) Non-RAM memory or (2) reaching the Stack Pointer. In fact, only (1) is checked, yet in a very mysterious way this seems to work properly on a ZX81, probably because interrupts change the stack once in a while, which is interpreted as non-RAM memory (the memory test itself does not alter the stack in any way). Tricky business!
On XTricator, interrupts don't change the stack and non-RAM memory doesn't exist, so the memory check goes on until it overwrites its own code...
To get a working copy of FIGFORTH, POKE the following numbers at the addresses 17247 onwards: 0, 0, 57, 24, 11 (all decimal).
Then SAVE, type NEW, LOAD again and start by GOTO 20.
Note that issueing a NEW is essential before LOADing the program!

*** ZX-CHESS-II by Artic causes the Hi-Res display to appear time and time again. The reason is that ZXCHESS makes unofficial use of the IX register. Since ZXCHESS was designed to run in FAST mode, the IX register can be changed at will without a crash, but (in my opinion) only lazy programmers did this! There is not a simple solution available. Just press CTRL-F3 each time the Hi-Res screen pops up...
ZX-CHESS-II can be found on Library Disk #2.

*** DISASSEMBLER by Campbell has the same problem as ZX-CHESS-II: unofficial use of the IX register in FAST mode (this is also the reason why the program crashes in SLOW mode on the ZX81!). Again: no solution, use CTRL-F3 at the end of each disassembled page.
The Campbell Disassembler can be found on Library Disk #1.

*** FORTY NINER by Software Farm uses a processor-speed independent delay loop at several places in the program. It 'peeks' the low byte of Frames, decrements it with a certain value and waits until the low byte of Frames reaches this decremented value. As there is a large possibility that XTricator is temporarily suspended at the moment that the low byte of Frames takes this value, the program may 'miss' the value and will have to wait another 5 seconds before the low byte has the required value again - this obviously makes the program extremely slow.
As a solution, break the program by CTRL-ESC at the title page,

issue the following Pokes: POKE 24490,56 and POKE 26707,56
and re-save by RUN 100.
The program FORTYMIN.P on Library Disk #2 has been treated this way.
Note that screen objects may now appear all on top of each other...

ROM AND 8-16K AREA

The ROM and 8-16k area are in fact also part of RAM, i.e. they are not POKE-protected. This can be nice for making slight changes in the ZX81 ROM or for using the 8-16k area for machine code utilities that can be loaded from disk.

At startup, the 8-16k area contains a relocated version of the CORAL BASIC INTERPRETER (CBI version 7.0). You can start this program by typing

```
RAND USR 8192
```

This should give 0/0 in line 22 and a black cursor in line 23.
CBI may be useful for people with little ZX81 experience since it has both a full-entry decoder, full-screen editor and explains error messages on request. All usual ZX81 commands are available, as well as 46 new ones (including commands for PROCedures with two-way data exchange and structured loops). Naturally, CBI was written for the ZX81, but runs equally well on XTricator. For details on this program, contact me in writing. Two useful commands to enter when the black cursor appears are

```
ERR MSGS ON          (explains error messages)
NOSTALGIC OFF        (enable full screen editor)
```

Note that you have to type the letters one-by-one, the spaces are optional. The sytem variables of CBI are stored at address 65400 onwards (corresponding to address 32730 in the original version). The CBI program ends at address 13522, so the part of memory from address 13523 up to 16383 is free for your own use.

STOPPING

There are three possible ways to end your XTricator session. The first is by typing

```
RJOB XTR_VITA        ( or equivalent )
```

in SuperBasic and then returning to XTricator by CTRL-C. This will clean up the QL, removing all channels, jobs and interrupts owned by XTricator. Also, the QZ device you were using is QZ-disabled (QZ_OFF). The second way is to use a special QZ command from within XTricator; just type:

```
LOAD "*KILL-XTR:"
```

which is in fact equivalent to removing the XTR_VITA job. Since this requires a lot of typing, there is a third alternative:

```
LOAD " STOP "
```

with STOP typed as a keyword (Shift-'A') will also end the XTricator session. If you do not work in the Window Manager environment, you can obviously use the latter two methods only.

CLOCK FREQUENCY

The effective clock frequency can be determined in serveral ways. First of all, you can RUN a (long) BASIC program which does not have any INPUT, INKEY$, RND, PAUSE or similar commands (i.e. execution time must be fully determinate and may not contain processor-independent loops). You can then compare the execution time on XTricator to the time which this same program requires on a ZX81.
The ZX81 runs at 3.2 MHz in FAST mode and at 0.8 MHz in SLOW mode. Supposing a program on Xtricator takes three times longer to execute than on the ZX81 in SLOW, the effective clock frequency is (0.8 MHz)/3 = approx. 0.25 MHz.

Another way to find the clock frequency is to use a machine code test loop, embracing a single instruction which is executed many (e.g. 65536) times. If you then count the number of frames needed (and subtract the number of frames needed for just the empty loop) you can get an idea of the clock frequency of that specific instruction, provided you know the number of T-cycles the Z80 would normally need for that same instruction.

On a normal 68008-based QL, I found from the latter method specific instruction clock frequencies varying between 0.1 and 0.4 MHz. This seems to agree with the average (effective) value of 0.25 MHz. (Due to the emulation process the specific clock frequency varies per instruction!)
With a 16 MHz Gold Card, this number can be increased up to 1.5 MHz.

Z80 EMULATION

As far as I have been able to test, all Z80 instructions are emulated correctly: since the ZX81 ROM is the basis for XTricator to function, the latter statement must be true. Also, hundreds of ZX81 programs have been tested on XTricator and all of them turned out to work without any problem! (These programs include (commercially marketed) games and utilities in BASIC and in machine code!)

Some brief remarks for Z80 freaks:

* The Half Carry and Subtract flag of the Z80 flag-register (f) are NOT emulated. Nearly all Z80 instructions alter these flags, but there is only one instruction which uses the result, viz. DAA. Since DAA is used only once in the ZX81 ROM (where it has been replaced by an equivalent non-standard Z80 instruction) and since proper emulation of the Half Carry greatly reduces operation speed of XTricator, I think this exclusion is a sensible decision. Note that, when XTricator encounters a DAA instruction, a warning message is given at the bottom of the screen. Press any key to proceed (but keep your fingers crossed!).

```
Programs using DAA : --- Reversi (only the machine coded version)
                     --- MCTT (Machine Code Testing Tool)
                     --- Frogger;
                            cure by: POKE 17250,0
                                     POKE 21725,0 (hexadecimal scoring!)
```

If you should come across other programs which display the DAA-warning, (but not as a result of some crash) please contact me.

* The IN instructions are all neglected, except for requests for port $FE, (the keyboard) and port $FB (the printer). Also, OUT instructions are neglected. Note, however, that the instructions IND, INDR, INI, INIR, OUTD, OTDR, OUTI and OTIR alter the registers BC and HL (and also (HL)'s) appropriately!

* HALT waits for an interrupt. Interrupts are not emulated and therefore HALT is made equal to NOP.

* Register I points to the ZX81 character set. Default value is I=$1E which means that the character set starts at address $1E00 (last half K of ROM). If you want to change the characterset, first POKE the new set somewhere in memory (at an address which is a multiple of 512 bytes) and then alter register I accordingly (so: only even values for I are allowed). This will instantaneously change the display.
  The reversed sequence of this procedure will not work the way you would expect! I advise not to change the character set present in ROM.

* Some instructions cannot be used for changing the ROM area (0-8k). These are:

```
LD (HL),A          LD (DE),A
LD (HL),D          LD (HL),E
LDIR
```

The reason for this is to protect the ROM against itself! The programmers of the ZX81 ROM have made use of the write-protection of the 0-8k area for simplifying their code (or is it a hitherto unknown bug of the ZX81 ROM ?). So, XTricator (which has no write-protection in the ROM area) would alter its own code, which is (in general!) a bad thing to do. Therefore, the above instructions are emulated such that they have no effect in the first 8k of memory.
Note that this also protects the ROM against a program like MCODER-2, which does the same kind of nasty things to the ZX81 ROM-area...

WRITING PROGRAMS

For the success of XTricator and perhaps for a revival of the ZX81, I recommend authors of new programs (either in machine code or in BASIC) to write their software such that it will run on both XTricator and on the ZX81. As an example, you could perform nice tricks with XTricator by POKEing its ROM, but that is not possible on the ordinary ZX81, i.e. INCOMPATIBLE! Please avoid such tricks or - if they are absolutely necessary - clearly include a message telling on which machine(s) the program runs (ZX81, XTricator, XTender

or the Atari Emulator). I also advise to use processor independent delay loops where necessary (mostly games and editors). Basically, these look like this:

```
REPEAT
  SET CLOCK
  BEGIN
    PERFORM ACTION
  END
  REPEAT
    READ CLOCK
  UNTIL CERTAIN TIMELIMIT EXCEEDED
FOREVER
```

Programs written this way are equally fast on every computer (unless the ACTION part takes more time than given by CERTAIN TIMELIMIT). In fact, several existing ZX81 programs were really written this way and are consequently not sped up by a fast emulator (only decelerated by a slow one...)!
For reading and setting the clock, make use of the system variable Frames. Frames is a two-byte number, decrementing at 50 Hz. Please note that

(1) Frames is decremented in SLOW mode ONLY on the ZX81;
(2) Frames is decremented in FAST AND SLOW mode on XTricator;
(3) Frames on XTricator may decrement too slow when several copies of XTricator are running in the QL or when other interrupts are present; this might give you the distinct impression that XTricator is speeding up (such disenchantment!).

Also in BASIC programs, use Frames (or PAUSE) instead of some empty FOR-NEXT loop.

PROGRAM TRANSPORTATION

Some trouble is caused by the fact that ZX81 programs were usually stored on tape instead of disk. Two solutions are available: firstly, you can purchase an RS232 interface for your ZX81, supplied with a comprehensive file-server, and send programs directly from ZX81 to QL (and vice versa). Secondly, you can send me your tapes (unprotected and of reasonable quality only!), and for a small fee per program I will send back a 3.5" DS/DD floppy with your programs on it, directly suited for use with XTricator.

The first solution is advantageous for two reasons: first of all, you have a physical connection between ZX81 and QL and could - for example - edit programs on XTricator and run them on the ZX81 (or, again, vice versa). Secondly, the file-server (called QZ) works both on the ZX81 (via the serial line) and on XTricator. On XTricator, you can use the ordinary ZX81 SAVE and LOAD commands, which write programs to disk and read programs from disk in a format directly compatible to the QZ file format. Additionally, you can pass commands and parameters which are equivalent to the set of commands supported by the current QZ version (e.g. DIR:).

For this solution you will need a serial interface plus cable for your ZX81. You have two possibilities:

(1) Build the interface yourself. A drawn scheme is available for free from me. Total cost of components and circuit board is about Hfl 15,- (or UK-pound sterling 5, or US $ 9).
(2) Buy the interface ready-made from the Dutch Homecomputer Club (HCC). Total cost is about Hfl 50,- (or UK-pound sterling 16, or US $ 30). The address of the club is: HCC, Postbus 149, 3990 DC Houten, Netherlands. Phone: (Holland) 3403 78788.

The Fileserver software is ShareWare and consists of a program on tape, and two 3.5" DS/DD disks, also containing the manual. This software can be obtained from me by sending me five 3.5" DS/DD brand disks (you will receive two of them back, the other three are for the tape and P&P).

OTHER ZX81 EMULATORS

The QL is not the only computer for which a ZX81 emulator has been written. An equivalent ZX81 emulator has also been written for the ATARI ST computer and a ZX81 emulator for MS-DOS machines is currently being developed.

I have no personal experience with the emulator written for the ATARI, but it seems that programs running on that emulator can be LOADed directly into XTricator and RUN without problems. Note that you will have to use a disk converter (e.g. XOVER, ATR driver, AtariDOS or MultiDISCover) to get the programs on QL disks. Note that the files with extension '81' on the ATARI emulator are compatible to the 'P' files of XTricator. The basic procedure for import is briefly explained below for two disk converters:

**** Using XOVER : Type the following commands (or equivalent):

```
QZ_OFF                          * disable QZ device driver
EX XOVER                        * insert ATARI disk in FLP1_
                                * select IMPORT option (DOS -> QDOS)
                                * DOS device:  FLP1_
                                * QDOS device: RAM1_
                                * now select the files to be imported
                                * and exit XOVER (return to main menu)
QZ_ON flp1_                     * enable QZdd for FLP1
WCOPY ram1__81 TO flp1__p       * copy all '81' files on RAM1
                                * to 'P' files on FLP1
```

**** Using the ATR driver : Type the following commands (or equivalent):

```
ATR_CONV 0                      * no file conversion whatsoever
WCOPY atr1_ TO ram1_            * copy ATARI files (in FLP1) to RAM1
QZ_ON flp1_                     * enable QZdd for FLP1
WCOPY ram1__81 TO flp1__p       * copy all '81' files on RAM1
                                * to 'P' files on FLP1
```

It seems that the ATARI emulator has problems to LOAD files not SAVEd by the emulator itself (again, I have no personal experience). As far as I know, XTricator 'P' files cannot be converted to ATARI '81' files because of this

'sensitivity'. It is ofcourse the virtue of XTricator that it DOES work the other way round!

The ZX81 emulator for MS-DOS (called 'XTender') is being developed by myself and takes XTricator as a reference, i.e. most possibilities of XTricator will also become available for XTender. Just as XTricator, XTender can Save and Load 'P'-files, so exchange should not give rise to any difficulties. Again, you will need a disk-converter to get the MS-DOS files on QL disks and vice versa. Preliminary releases of the ZX81 emulator for MS-DOS are available now (see bottom of this document).

WARRANTY

If you come across a ZX81 program which doesn't work on XTricator and also doesn't fall in any of the "incompatibility catagories" discussed in this manual, please send me a copy of the program on a 3.5" floppy. If XTricator is to blame, you will receive a new (repaired!) version of XTricator for free.

ABOUT THE AUTHOR

I was born in 1967 and bought my first computer (guess what, it was a ZX81!) in 1982. With no prior programming experience at all, I started writing simple Basic programs. After about one year, it appeared to me that Basic just couldn't solve the problems I devised. So, I turned to Z80 machine code (with obliging help of Toni Baker; that is, of her book). A whole new world of possibilities opened up before my very eyes. High-speed compact code and access to all system utilities at lowest level, what else does a programmer need? Many Z80 programs were born, and some of them were even published in international magazines. In 1987, I decided to buy a QL (during sales) and found this younger brother (sister?) just as interesting as the ZX81. It also gave me a good excuse for learning 68000 machine code. As I saw that less and less people were actually using the ZX81, I sought a way for elongating its existence. I thought that (amongst other things) a ZX81 emulator for the QL would be a good way to do so, and I started writing this program in the end of 1989. By now, in 1992, you can judge the result of my efforts. I sincerely hope that many (former) ZX81 users will take some time to leap back in history and (re)enjoy the overwhelming simplicity of the 64k, 8-bit, 3.2 MHz Z80, black & white computer with 64x44 graphics resolution, which introduced over a million people worldwide (including myself) to the wonderful world of computing. My present-day occupation as engineer of physics offers me the opportunity to work with the latest technology in computers, but at home I still prefer to use those absorbing machines made by Sinclair...

ZX-91

Andre Baune is the editor of a small, monthly magazine called ZX-91, intended for all the ZX81 users around the world. The magazine was first released in 1991, when the ZX81 celebrated its tenth aniversary.
A quote from issue #1:

> "... I want to lift the spirit of the users and the believers in this marvelous little computer. I also want to reach the far away users and/or programmers. In this newsletter you will get print-out, pictures, and articles ..."

No ZX81-user can consider not subscribing to this newsletter! Besides small programs and the latest news, it contains a large list of Sinclair/Timex supporters, users groups and dealers who can still provide you with ZX81 hardware and/or software. Over 250 copies of ZX-91 #1 have already been sent and more are being sent each month.
To obtain the latest issue, or more information, send a note together with a self-addressed envelope to:

> Andre Baune,
> 304 Scott,
> Chateauguay, Quebec,
> Canada J6J 4H5.

And do me a favour, will you? Please mention where you got the information about ZX-91 from!

XTRICATOR OFFICIAL RELEASE HISTORY

Please read the appropriate section(s) for more details about new features!
A program name between brackets indicates that this programs is now working properly as a result of the bug repair concerned.

Version	Date	Remarks
0.00	February 1989	- development of the program started
0.xx	02/89 - 12/91	- many preliminary versions are distributed
1.00	January 1992	- first official release after 3 years development
1.01	April 1992	- fatal error in '*LOAD:' commands - crahing the QL - has been corrected
1.02	May 1992	- output of '*SHOW:' command has been extended and now shows version number etc.
1.03	June 1992	- RR flag-bug has been fixed (discovered by a malfunctioning Spectrum program in 'Spectator') - XTricator can now be ended by the command LOAD " STOP " - the screen mode is no longer refreshed when XTricator is killed (looks better with QPAC2)
1.04	July 1992	- the manual contains a new section about the Canadian ZX-91 magazine - the sections 'Keyboard Facilities' and 'File Management' have been slightly extended - CTRL-ESC will now force the default ZX81 character set before printing error D - FAST and SLOW no longer affect the character set being used (Font Demo) - all Z80 registers are cleared by CTRL-ALT-ESC

```
1.10      August 1992       - the ConTRoL key handler has been completely
                              rewritten; it will now IMMEDIATELY recognize
                              and interpret any special CTRL-keypress (most
                              importantly: CTRL-ESC and CTRL-ALT-ESC); this
                              means that any program can be stopped at once,
                              even when it when it is crashed or when it
                              doesn't read the keyboard
                            - minor keyboard IN bug repaired (ZX Asteroids)
                            - minor PAUSE routine bug fixed (Bug Byte ZXAS)
                            - emulation of POP AF and PUSH AF has been
                              improved; the sequence POP AF / PUSH AF does
                              not change the stack contents anymore
                              (Bug Byte ZXAS)
                            - section 'Incompatibilities' in manual extended
                              with a note about Forty Niner
```

FINAL NOTES

Speed tests performed on a regular 8 MHz 68008 QL with 896k Trump Card and
on a QL fitted with a 16 MHz Gold Card, in both cases XTricator running as
single-tasking job at default priority.
Just to keep the lawyers happy: ZX, ZX81, QL and QDOS are registered
trademarks of Sinclair Research Limited.
The Gold Card is available from Miracle Systems Ltd, York, U.K.
Special thanks to Martin van der Zwan for thorough testing and a lot of
useful ideas!
For more information on XTricator, QZ, QZdd or Coral Basic Interpreter,
do not hesitate to contact me at subjoined address in writing:

Carlo Delhez,
Emmastraat 3,
4651 BV Steenbergen,
Netherlands.

via E-Mail on Internet:
tnndcarlo@cycl.phys.tue.nl   (until May 1994)

Thank you for reading the manual & for using XTricator !!

PLEASE READ THE FOLLOWING MESSAGE ONCE MORE !!!

XTricator contains the original ZX81 ROM code. This code is copyright
software. Therefore, XTricator may ONLY be used by you if you own a
ZX81 YOURSELF. In that case, you are IN PRINCIPLE able to LEGALLY
transport the ROM code from your ZX81 to your QL.
Any usage of XTricator by people NOT owning a ZX81 is in conflict
with copyright laws.
