

HiSoft Software

*for CP/M-80 Computers
and compatibles*

More quality products from

HiSoft

High Quality Software

INFORMATION SHEET

All CP/M

HiSoft Pascal80

£49.95

HiSoft Pascal for CP/M systems (including Amstrad CPC and PCW computers) has been developed to provide an almost full implementation of Standard Pascal as detailed in the Pascal User Manual and Report by Kathleen Jensen and Niklaus Wirth (the originator of Pascal). We have been careful to adhere to this definition of Pascal while providing extra facilities to exploit your computer.

The compiler is written in Z80 and produces Z80 object code directly to an executable COM file (no P-codes or linkage); this means that the compiler package, occupies very little space in memory (less than 19K) and produces object code that runs very quickly indeed. The minimum run-time overhead is about 2K.

Despite its compact size HiSoft Pascal is an extensive implementation of the Pascal language viz:

Predefined Identifiers

CONST

MAXINT = 32767;

TYPE

BOOLEAN = (FALSE, TRUE);

CHAR (*The expanded ASCII character set of 256 elements*);

INTEGER = -MAXINT..MAXINT;

REAL (*A subset of the real numbers with 7 digit accuracy*);

TEXT = FILE OF CHAR;

VAR

INPUT, OUTPUT, TEXT;

PROCEDURE

READ; READLN; WRITE; WRITELN; PAGE; HALT; USER; POKE;

INLINE; NEW; MARK; RELEASE; OUT; PRON; PROFF; RESET;

REWRITE; GET; PUT; RANSEED; CHAIN; DISPOSE;

FUNCTION

ABS; SQR SQRT; ODD; ENTIER; ROUND; TRUNC;

FRAC; ORD; CHR; SUCC; PRED; EOLN; EOF;

INCH; INP; PEEK; ADDR; CPM; RANDOM; SIN;

COS; TAN; ARCTAN; EXP; LN; TRUNC; SIZE;

RECAST; MEMAVAIL;

Reserved Words

AND	ARRAY	BEGIN	CASE	CONST	DIV	DO
DOWNTO	ELSE	END	FILE	FOR	FORWARD	FUNCTION
GOTO	IF	IN	LABEL	MOD	NIL	NOT
OF	OR	PACKED	PROCEDURE	PROGRAM	RECORD	REPEAT
SET	THEN	TO	TYPE	UNTIL	VAR	WHILE
WITH						

All CP/M

HiSoft Pascal80

£49.95

The compiler supports all the data structures of Standard Pascal including Arrays, Sets, Records, Pointers, Text Files and user-enumerated types. Strings are supported as ARRAY[1..N] OF CHAR. Assignment of structured types, such as array-to-array assignment, is allowed and sets, strings and pointers may be compared. The CASE statement may optionally be terminated with ELSE. Procedures and functions may be fully recursive and procedures can take value or variable parameters. Records of any type may be created and accessed by field, optionally using the WITH construct for added convenience. Dynamic variables are created using the procedure NEW and accessed using pointers - this enables linked-list structures to be created. Further, the procedures MARK, RELEASE and DISPOSE allow control over the dynamic variable heap.

I/O is provided through the standard procedures READ(LN) and WRITE(LN), GET and PUT. Standard CP/M text files, binary files or the console may be used as the input/output devices. Sequential files of any type are supported whilst random access can be achieved using Pascal source libraries supplied. Formatting of output is supported through the use of write parameters; reals may be output in scientific format (Mantissa, exponent) or fixed-point format.

Interfacing with Z80 machine code is made easy through USER, PEEK, INLINE, SIZE and ADDR together with register variables. PEEK and POKE may take any tuple as their arguments, except sets and files. INLINE allows machine code to be included in the program at runtime, and ADDR returns the address of a Pascal program variable. Interfacing to the Z80 ports is provided through the function INP(P) which reads port P and the procedure OUT (P,C) which outputs the value C to port P. The operating system may be accessed using CPM (de,c).

Pascal80 programs can CHAIN another Pascal program, or any other CP/M program, preserving the global variable area. This, coupled with automatic saving of top-level procedures/functions during compilation, allows very large programs to be developed.

Many compiler options are available allowing, *inter alia*, output to be directed to a printer and runtime checks to be switched off for extra speed of execution. The 'F' option allows inclusion of source text from disk - this allows large programs to be compiled as well as providing the flexibility of storing program segments on disk.

Amstrad CP/M Plus (CPC6128, PCW8256/8512/9512) versions of Pascal80 are supplied with full source libraries to allow access to the machine's GSX graphics interface. This allows lines to be drawn, points to be plotted and areas to be filled etc. Full documentation is supplied along with example programs and a new, enhanced screen driver for the PCW machines.

The compiler comes with an extensive ring-bound manual with example programs and complete Pascal programs on the disk including an intelligent disassembler.

Also included in the package is our comprehensive screen editor, ED80 (see Devpac80 details for specification), which is fully integrated to allow a complete and easy development cycle. On a compilation or runtime error a helpful message is displayed, simply press a key and you will be returned into the editor with the cursor positioned near the offending symbol and with the message still displayed, ready to correct and re-compile.

The code produced by Pascal80 is compact (minimum size 2K), optimised and very, very fast.

Please study the above specification thoroughly; HiSoft Pascal, we believe, is the most powerful and the fastest Pascal compiler available for microcomputers at such a low price.

INFORMATION SHEET

All CP/M

HiSoft C

£49.95

A Fast, Interactive Version of the Popular C Language

HiSoft C runs on all CP/M computers including the Amstrad CPC and PCW machines. On the Amstrad it is supplied either on cassette or disk. The cassette version runs on the CPC computers, works with the firmware and includes extra libraries to call the operating system; HiSoft C for the Amstrad on disk includes versions to work under the native operating system, AMSDOS, and CP/M. A GSX graphics library is supplied for the Amstrad CP/M Plus computers.

HiSoft C performs extremely quickly both in terms of compilation and the overall development cycle; on detecting a compilation error you can enter the editor on the rogue line, correct the error and resume compilation automatically. The CP/M version is supplied with a special, interactive version of our screen editor ED80.

HiSoft C has sophisticated UNIX standard I/O features like random-access files and both binary and text access modes and also provides command line handling, including I/O redirection (for <input >output and >> appending).

statements;

```
do statement while (expression)
while (expression) statement
if (expression) statement else statement
switch (expression) statement
case expression: default:
for (initial_expression; test_expression;
loop_expression) statement
break; continue; return expression;
label: goto label; inline (machine_code);
(compound_statement)
```

data types:

```
char (8 bits)
int (16 bits) unsigned (16 bits)
short, long are accepted as int
```

C is a very flexible language which combines high-level structured features for both algorithms and data together with low-level access that allows direct control over the machine. HiSoft C supports both of these important abilities.

This means that programs can take full advantage of the computer's environment (thus enabling fast and compact execution) while being easy to write and debug.

HiSoft C comes with a substantial manual in a quality ring binder. There are numerous examples of every part of the language and complete programs to try out. For the expert, there is a detailed point-by-point comparison with UNIX C. The manual contains a full guide to C so there is no necessity for a separate textbook. Each element of the language is listed in the comprehensive contents, as are each of the library functions. When the compiler finds an error it displays an informative English message; and there is more help in the manual about why the message occurred and how to fix it. A checklist of common C mistakes is also included.

HiSoft C supports:

```
float, double are not available
pointers, functions, multi-dimensional arrays,
structures, unions
typedef, extern, static, auto
combinations of the above, as in K&R, with full checks,
initializers for global and static variables
```

expressions:

```
= += -= *= /= %= >> << &= ^= |=
?: || && ^ &
== != <= >= < >
<< >> + - * / %
* & - ! ~ ++ __ sizeof type_cast
( ) [ ] ( ) . ->
```

strings, characters, decimal, hex, octal constants, compile-time evaluation of constant expressions.

All CP/M

HiSoft C

£49.95

Selective library search is one feature of the pre-processor. It allows a source file to be scanned and just those functions which are actually needed are compiled into the program. Other preprocessor/compiler control features include:

```
# define macros    (with no parameters)
# include file
# include ?file?  (gives library search)
# translate       (to change the name of the
                  compiled program)
# list            (to turn listing off & on)
# data            (sets top of memory for
                  global data, CP/M only)
# error           (removes explanatory
                  error messages to give
                  more room for your
                  program)
```

input/output

```
getchar  putchar  keyhit  fopen getc  ungetc  putc  fclose  rawin  rawout  gets
puts    getw*   putw*   printf sprintf fprint  scanf  sscanf  fscanf  fgets  fputs
fread*  fwrite*  fname*  seek*  tell*   read*  write*  fsetbuf* fflush* freopen
```

strings

```
strcat   strncat  strcmp  strncmp  strcpy  strncpy
strlen  strchr  strrchr  strpbrk  strspn  stropsn
```

characters

```
isalnum  isalpha  isascii  iscntrl  isdigit  isgraph  islower  isprint
ispunct  isspace  isupper  isxdigit  tolower  toupper  toascii
```

arithmetic

```
min max abs sign srand rand
long_multiply(32 bit) long_add long_init
long_set long_copy
```

miscellaneous

```
peek poke inp out qsort calloc free sbrk
atoi _exit exit blt swap (memory)
```

Special CP/M functions*

```
cpm_bdos cpm2_bios cpm3_bios cpm_dir
read_file write_file instr itob strolower
strupper
```

```
# direct (for immediate execution
          or building up a program,
          AMSDOS only)
```

The compiler is supplied with a library of standard functions based on those found in a Unix environment, and other libraries designed to make maximum use of the Amstrad computers in a way that will be familiar to all users.

Some basic functions are built-in for efficiency whilst the rest are provided as C source code. This has three advantages - functions that aren't needed don't take valuable memory, the library provides examples of C programs doing a variety of jobs, and they can be modified to suit your exact needs.

There are CP/M, firmware and CPC BASIC libraries containing the following functions:

Special non-CP/M functions †

```
after beep every add_ticker
init_event cass_speed border catalog
cls draw event_disable event_enable
flash_speed ink inkey instr itob
joykey_functionkey_speed key_translation
line paper play plot
time read_file setup_sound
sound_check strolower strupper symbol
symbol_after
```

plus 30 of the most useful sound and graphic firmware calls.

* available on CP/M version only † available on non-CP/M version only

All CP/M

FTL Modula-2

£54.95

Your next Language from HiSoft

Face it, programming can be a chore or a joy. Some computer languages bog you down in petty details, some don't help you enough. Is there any one which is balanced?

YES, FTL Modula-2 makes programming fun yet fast. It's easy to learn, fast to use, yet very powerful. It makes small, fast, native code files.

Why Modula-2?

It was invented by Niklaus Wirth to perfect the flaws in Pascal. *And it does!*

Why FTL?

A quick answer: speed, convenience and versatility. It compiles over 500 lines a minute, handles complicated real-world programs, knows how to search your disks for its files. This is a professional product, not a toy. It's understandable, not daunting. FTL lets you think about the program, not the compiler!

Complete compiler

The full language, not a subset! Very efficient native code, too - up to four times more compact than competing compilers.

Great editor

Edit up to three files and switch between them anytime. Run the compiler, jump right back to errors. Define and execute macros. Edits any ASCII file, too!

200 page manual

A comprehensive manual covering everything from compiler use to editing comes with every copy of FTL.

CP/M access

Get a directory, do a BIOS call, read an I/O port.

Assembler

For really low-level operations, write an assembly module, link it in your own programs (The assembler isn't necessary for writing Modula-2).

Excellent linker

Intelligent, disk-based, fast. Automatically includes modules you need. Tells you if something needs recompiling.

Library Source Code included

Many other companies don't even offer their libraries, or do so at high cost. Ours is free.

Real number support, standard

Over 15 digits in accuracy, with accurate maths and transcendentals.

Supported and improved

Enhancements are constantly added to FTL - be it CP/M, 68000 or PC. Updates are reasonable so you won't go broke staying current.

Easy to learn, speed to burn

Can a beginner learn it? Yes! While FTL isn't a tutorial, there are plenty of examples. The manual is written in English, not some obscure dialect of Urdu. The section in the manual on errors is large and helpful too; not just *fix and try again*.

FTL has its own editor, written in Modula-2. It will edit up to three files at once - a tremendous plus. FTL encourages programs to be written in small pieces or modules - then compiled separately. No more finding one lousy error in line 3,002 and starting over. Once compiled, a module can be used forever. Modules isolate bugs, too.

If you know Pascal, it's not hard to learn Modula-2. Niklaus Wirth put the years of experience since he wrote Pascal into Modula-2, fixing most objections to Pascal. There are over a dozen books on programming in Modula-2, meaning lots of learning material.

INFORMATION SHEET

All CP/M**FTL Modula-2****£54.95**

Modula-2's strengths are manifold. It will talk to hardware, but it's not hard to learn. It uses English keywords instead of mathematical symbols. Because it's not a large language, the compiler is moved easily to different computers. Programs written in Modula-2 can quickly move to new machines, without change.

One price, one great product

At last, all you need to start programming is right in one place. With FTL and a textbook, it's a breeze to learn Modula-2. Once you do, your programs will be portable from CP/M up to VAX. Move your programs to new computers with great ease, too, since Modula-2 is a standard language.

Is switching to Modula-2 worth it?

Yes! Here's why:

Cuts the development cycle

Instead of re-inventing, re-use already written code. Write a module, compile it once, use it forever. Or use someone else's library.

Less recompilation

Separate compilation is built into Modula-2, not "bolted on". FTL checks variables between modules, too.

Fewer mysterious bugs

Many languages, like C, don't check types between compiled files. This leads to many mysterious and confusing errors, often taking days to find. Modula-2 always checks.

Support

We'll be here. FTL Modula-2 has been constantly updated since its first release three years ago.

Standard language

Unlike Pascal, standard Modula-2 is powerful without extensions. The same program runs from Z80s through VAX. Spend time writing programs, not converting them.

Makes team projects easier

Modula-2 encourages each team member to write an interface before any internal code. If it's not in the interface, it can't be changed by other modules.

Available for CP/M, 68000 & MSDOS

Write programs for Z80-based controllers and ROM-based systems or for general MSDOS use. Control interrupts and low-level I/O from Modula-2 or write an assembly module.

FTL Modula-2 is a complete, one-pass compiler. No language features have been left out. Some of the highlights:

Tight code

FTL Modula-2 crafts fast, small native code. Real number code is included only if needed. Minimum program size is under 2K for CP/M.

Accurate real numbers

Reals are accurate to 15 places and so is the maths. Many Pascals have only 7 digit accuracy. 15 places means great accuracy for business and scientific programming. Real number support is standard, too: nothing else to buy!

An outstanding manual

Written with knowledge and wit, its 200 pages cover language features, the editor, the linker, the assembler and machine considerations. There are 15 pages alone on compilation errors and what they mean.

INFORMATION SHEET

All CP/M

FTL Modula-2

£54.95

Large sets

Sets in FTL can have 1,024 elements, not the usual sixteen.

Extra-long identifiers

Variable names can be up to 32 characters long.

Incredible variety of source code

Source to the standard modules is provided so you can modify it yourself. It's also a pool of example code for learning the language.

Extra editor features in MS-DOS

Change directories, run programs, syntax-check programs, link, recover from disk errors, get a directory - without leaving.

Why Modula-2 is easier to write and maintain

Modula-2 syntax tames most of Pascal's hangups. IF statements look like this:

Modula-2 IF	Pascal IF
IF Name = ID THEN	IF Name=ID THEN
	BEGIN
Error (ID);	Error (ID);
EXIT;	GOTO 999;
	END
	ELSE
ELSIF Name<ID THEN	IF Name<ID THEN
CheckRLink;	CheckRLink;
END (* IF*);	

Immediately you see the difference: no more inserting a BEGIN when adding statements to an IF. IFs are easier to read and write. And the CASE statement is much better as you will see from the example in the next column:

Modula-2 CASE	Pascal CASE
CASE j OF	CASE j OF
0..9:	0,1,2,3,4,5,6,
	7,8,9: BEGIN
DEC(i, 10);	i:= i-10;
QT:= FALSE;	QT:= FALSE;
	END
10,11:	10,11:
Stop:= TRUE	Stop:=TRUE;
200: TuneBackEnd	200: TuneBackEnd
52:	52: BEGIN
DecipherCase;	DecipherCase;
QT := TRUE	QT:= TRUE;
	END;
ELSE HALT	(* Can't Do! *)
END (* CASE*);	END (* CASE *);

Notice how each CASE is easily extended: just add code, without BEGINS and ENDS! Each CASE is separated by the vertical bar character (|), and more can be added anywhere. Modula-2 adds an ELSE to the CASE, making it much more powerful.

Why Switch?

Separate compilation

In standard Pascal, an entire program must be recompiled after every change. In Modula-2, you only compile what you've changed. FTL Modula-2 also warns if something needs recompilation; if, for instance, you've declared a new variable.

Fast compilation

FTL is no slouch in speed, either. It's faster to disk than Turbo Pascal 3.0.

All CP/M

FTL Modula-2

£54.95

Standard means portable

All Pascals add features in a haphazard way. Modula-2 is standard so code can move to new machines.

The CARDINAL Type

How many times have you wanted to count past 32,000 and had to use a real number? Modula-2's CARDINAL type goes from 0 to 65,535 and is standard.

LONG Types

MSDOS FTL has LONGCARD and LONGINT types which count up to 4,294,967,296. Standard, not an extra.

Better FOR loops

Modula-2's count by any increment, not just one.

Opaque types

Write a file handler whose data can be used but not modified by other modules.

System access, standard

Get at DOS without assembly or special tricks. Or use FTL's own assembler if you like.

Technical improvements

Here's some more advanced reasons to buy FTL Modula-2:

The compiler is one-pass

directly generating native code. It's faster. The compiler is of unique design, not based on any older, slower compiler.

The disk-based linker

will make large .COM files independent of available memory.

Use assembly just like Modula-2

Assembly code is isolated in modules, with a definition, exactly as if you'd written in Modula-2. Call assembly routines just like Modula-2.

ROMable code support

Move data and code areas (starting addresses) independently to anywhere in memory. Set code exit location. Programs can easily be written for variable size TPAs (CP/M).

Procedures can be passed as parameters

Use FTL's sort routine on ANY data without writing your own. Write routines which work on any data type without rewriting!

Open array parameters

In Pascal, an array passed to a procedure must be fixed in length; FTL can pass any size array and its length. Open parameters can be VAR or constant.

Access to interrupts

Use Z80 mode 2 interrupts.

Assembler included

Assembly code may be linked into your programs. Assembly is not necessary for writing Modula-2.

Very accurate REAL numbers

Real numbers are accurate to over 15 places and so is FTL's maths.

Fast, small integer maths

If you need top speed then the integer maths comes into its own. See the PCW Benchmarks in this catalogue.

Requirements

FTL Modula-2 runs on all CP/M and CP/M Plus computers with at least 52K of tpa and 200K disk space.

INFORMATION SHEET

All CP/M

FTL Modula-2

£54.95

FTL Modula-2 Optional Extras

FTL Editor/ToolKit

£39.95 inclusive

The Editor/ToolKit contains full source to the editor in FTL Modula-2; the editor is written in Modula-2. It's a fast, macro editor which will edit up to three files at once, remembers where files were last edited, calls the compiler and returns.

The modules on this disk are valuable Modula-2 tools. Simply compile them and use them. Programs you write using the Editor/ToolKit are yours. The editor source code itself is copyrighted and may not be sold.

Disk contents include: comment file on the Editor/toolkit disk, contents, and files; CAT, CATenate program: displays files to screen or disk file; COMMAND, processes CP/M command line; COMPARE, compares text differences; COMPDIR, compares entire directories of files; DOMENU and control, display, and status modules; KEYBOARD, MACROS, keyboard I/O and keyboard macro control; MLU, library mangement utility; PATTERN, pattern-matching routine; SCREENIO, does screen I/O, GOTOXY, highlighting, line drawing, etc.

FTL Advanced Programmer's Toolkit

£39.95 inclusive

The FTL Modula-2 Advanced Programmer's Toolkit is a must for serious users of Z80 FTL Modula-2. It includes:

- Trimmer. Removes the unused procedures from your programs. Can dramatically reduce program size.
- Overlaying system. Lets you break the Z80 64K limit for your code. Ideal for porting large Modula-2 programs to CP/M. Source code provided so you can even make overlays work in a ROM-based system!
- Source code to the low level SYSTEM and REAL\$ modules. Lets you change the effect of run-time errors to, say, give an error message on an LCD.
- Multi-tasking kernel in source code. Can be used for real time applications. Will also let you write multi-tasking programs that run under CP/M if your system can provide a clock interrupt that can be re-vectored. This sadly does not include the Amstrad PCW machines.

Together with the standard FTL Modula-2 facilities for generating ROMable code we believe the Advanced Programmer's Toolkit gives the most cost-effective Z80 ROM development system available. If you wish to cross-develop from a PC, buying a fast Z80 card and the FTL Z80 system is faster and cheaper than a cross compiler !

In addition to CP/M FTL Modula-2 we have PC (PC1512, PC1640 & compatibles) and Atari ST versions of the package so that portable development is now a reality.

All you need is FTL Modula-2

INFORMATION SHEET

All CP/M

Devpac80

£49.95

Devpac80 The New Standard

Devpac80™ Version 2 is the latest assembly language development tool from HiSoft. It comprises a suite of three major programs running under CP/M.

The Editor

ED80 is a powerful, WordStar-compatible, configurable program editor. An essential feature of ED80 is its help screens; although the user can configure all the commands, a single key-press will bring down help screens. These are kept in coded form on the disk so they don't encroach on precious memory space.

ED80 features cut-and-paste editing with easy restoration of deleted text. The dynamic block-handling includes delete, move, copy, restore, plus a flexible read and write that can address either the disk, the printer (write only), or even another machine running ED80.

There is an advanced "find/substitute" facility with wild-cards and the screen-handling features intelligent side-scrolling with edit-time tab handling. Included in ED80 is the extremely useful "goto <line>" function enabling easy location of assembly-time errors.

ED80 also knows all the errors that were produced by the previous assembly and you can step through these one-by-one correcting them with just one keypress.

ED80 is only 9K in length and comes with a tutorial and an easy-to-use installation program for tailoring the program to the computer and to the user's preference.

For example, it is a simple matter to use the full 90 columns and 31/32 rows of the Amstrad PCW range.

The Assembler

GEN80 is a fast, high-specification 2-pass macro assembler. Handling over 3000 lines of source code per minute, GEN80 features standard Zilog mnemonics, macros, disk inclusion, conditional assembly, production of .REL or .COM files and full maths; Boolean operators, exponents and bit manipulations.

The many assembly options include:- generate no code file, generate a symbol table listing to disk/screen/printer, list to the printer, list to the disk, inhibit list, define label length, type in direct from the keyboard at the head of a file and many others for easy control over assembly. Any of the options can appear either on the command line or in the file.

Conditional assembly controls object code generation at assembly-time and as an example allows the user to develop a piece of software for many different machines and then by use of a single label generate each version of the object code from the same source. The feature may also be used during development of a program, where a label may act as a switch controlling the inclusion of sections of debugging code. Nesting of conditionals is supported.

Disk inclusion is an extremely powerful feature that allows the assembly of very large files (e.g. 200K of source). The presence in the source-file of the line; *I <filename> will cause GEN80 to search the disk for the named file and then to assemble it as though it were explicitly in the source. This system encourages the modular approach to programming as one may develop and save modules individually (perhaps build up one or more macro library files) and finally assemble a file consisting solely of *I statements. Includes may be nested.

All CP/M

HiSoft Devpac80

£49.95

Relocatable output (.REL files) allows large programs to be developed in stages and then linked together to produce the final, executable file. The intermediate REL files may also be linked with REL files produced by other languages; e.g. ProPascal™ and ProFORTRAN™, all MicroSoft™ languages, DR CBASIC™ and many more. REL files are considered the standard intermediate format under CP/M, you use the linker LINK.COM supplied with CP/M Plus to combine them together. The ASEG, CSEG and DSEG directives are supported to allow programs to have separate absolute, code and data segments allowing ROMable code to be developed easily. PUBLIC and EXTERNAL symbols allow the passing of labels between programs and the symbol table may be dumped at assembly time or link time for use with the symbolic debugger. External plus offset is supported.

Macros are handled as detailed in the Zilog specification. Parameters to macros may be either numeric or textual, macro calls may be nested giving a sophisticated recursive capability. Local labels may be declared and used.

Output can be directed to a COM file (for direct execution under CP/M) or a REL file (for linking) and this code generation can be turned off for a quicker, syntax-check assembly.

The Debugger

PMON is a comprehensive, highly advanced monitor and debugger. Of course PMON contains all the normal features such as single-step, dynamic multiple breakpoints and full register and memory display/modify but the sophisticated interpretative execution of PMON enables features such as conditional breakpoints (e.g. execute until register HL is 0, DE is -1 and the program counter is less than

16384), single-step in ROM, switch memory banks, interruptable code execution (if you suspect your code is crashing at a certain point, you can just let it execute and then interrupt at will), simple profiling with user-defined loop watchpoints (e.g. count how many times you've been through LOOP, optionally with a scale factor). All breakpoint commands handle your program symbols loaded from disk so, for example, you can set a breakpoint at LOOP to check the condition $(de=COUNT/2).or.(LINUM)=1000$.

If true, then a break is made displaying your program state. Very powerful.

The search facilities allow you to find sequences of bytes, strings or even parts of assembler mnemonics. For example you may find the next RLCA instruction or the next instruction involving the BC register pair.

The powerful disassembler allows any number of data areas to be specified and will produce a disk file of any size that can then be assembled using GEN80 or edited using ED80. Symbols from an assembled or linked file may be included for symbolic debugging and these symbols may be modified whilst debugging. You can, of course, symbolically debug any file produced by DR LINK™, MicroSoft L80™ or GEN80 because Devpac80™ supports the standard CP/M linker format.

PMON comes in two versions, the small version is only 7K allowing very large programs to be debugged whilst the extended version includes the conditional breakpoint code and full symbolic expression handling giving maximum power (it's still less than 13K long!).

Devpac80™ comes with an extensive, ring-bound manual containing a full tutorial to get you started with assembly-language programming.

INFORMATION SHEET

All CP/M

Nevada COBOL

£49.95

COBOL - The Business Language

COBOL, the COmmon Business Oriented Language is a mainstay of commercial programming and NEVADA COBOL is a perfect introduction to the language, ideal for use by students as part of their college course and by anybody interested in learning the language.

The file and record structures of COBOL are designed to facilitate report production and the number of programs written over the past 20 years make COBOL the most popular commercial programming language.

Nevada COBOL is based on the ANSI-74 standard and therefore can be used by beginners and seasoned programmers alike; it features both fast compile and execution times with small object modules together with the following features:

- ✓ Random and sequential files
- ✓ Picture characters AX9BO/VZ,*,-\$# CR DB
- ✓ Record description elements:
PICTURE, PIC, REDEFINES, VALUE,
FILLER, SYNC, USAGE, OCCURS
(1 level),
BLANK WHEN ZERO, JUSTIFIED
RIGHT
- ✓ COPY statement for library handling

- ✓ VALUE OF FILE-ID IS data-name allowing external filenames at runtime
- ✓ CALL..USING..CANCEL..EXIT..PROGRAM, LINKAGE SECTION with dynamic loading of up to 5 active called programs (COBOL, assembler etc) at any one time and chaining to new main programs
- ✓ PERFORM ... THRU ... TIMES ... UNTIL
- ✓ IF ... NEXT SENTENCE ... ELSE ... NEXT SENTENCE
- ✓ Compound conditionals AND/OR <=>
- ✓ GOTO ... DEPENDING ON ...
- ✓ Interactive ACCEPT/DISPLAY series WITH NO ADVANCING..UNIT
- ✓ INSPECT ... TALLYING ... FOR ... REPLACING ... BY ... ALL ... LEADING/ FIRST CHARACTERS ... BEFORE/AFTER INITIAL..
- ✓ MOVE ... TO ... SERIES
- ✓ ADD, SUBTRACT, MULTIPLY and DIVIDE ROUNDED GIVING ON SIZE
- ✓ easily understood error messages sent to a file

Nevada COBOL comes complete with HiSoft's full-screen editor, ED80, example programs, a 165-page, professional, ring-bound manual and is fully configured for CP/M systems.

CP/M Utilities

All our CP/M packages are supplied with three very useful utilities: **WP** (for copying files), **WD** (for deleting files) and **SD** (an extended DIRectory utility). Each is easy-to-use and takes up less than 1K disk space.

INFORMATION SHEET

Ams CP/M+

Knife Plus

£19.95

Complete Disk Recovery

Knife Plus is a full-featured disk recovery program for the Amstrad PCW range & the CPC 6128. Running under CP/M Plus, it allows you to copy and repair damaged disks, whether they are CP/M or Locoscript disks.

Included in the package is:

- a sector copier to copy your disks ignoring any bad data.
- full range of sector-edit commands.
- a build-file option allowing a corrupted file to be created sector-by-sector or block-by-block.
- an informative help screen.
- a host of essential utilities.

Knife Plus allows you to search through sectors, blocks or files with easy screen-editing of the data on the disk. Any updated information is only written back to the disk when you say so. All the features are available from simple, one-key commands and there is a help screen to assist you.

The sector copier supplied is essential for quick disk recovery and this, coupled with the power of the disk-editing available makes **Knife Plus** second-to-none for disk recovery.

Knife Plus is supplied with a number of extremely useful CP/M file-handling utilities:

WD makes file deletion a much simpler process, presenting each name in turn and asking if you want to delete it.

SD produces a much more detailed listing than DIR, showing each file's length and the number of records it occupies, along with the amount of disk space free. All in less than 1K.

UNERA is just the tool you wish you had when you've deleted a file by mistake; in almost all cases Unera will be able to recover the file as if it had never gone away.

WP is a really handy program for backing up disks or copying groups of files, optionally prompting you for each copy.

PAGE allows quick-and-easy printing of multiple text files with page numbers and form-feeds to produce a neat, paginated listing.

GENSUB generates submit files for you.

Knife Plus and all these other programs are supplied on a 3" disk with a complete, informative manual, packed full of advice and explanation on the art of disk recovery.

Knife Plus will recover both Locoscript documents and CP/M files.

Common disk error reports

The following are common error messages that are reported when trouble is experienced with disks on the Amstrad CP/M Plus computers. In most cases, **Knife Plus** will recover the disk for you:

track t seek fail - Retry, Ignore, Cancel?

track t, sector s data error - Retry, Ignore, Cancel?

track t, sector s no data - Retry, Ignore, Cancel?

track t, sector s missing address mark - Retry, Ignore, Cancel?

INFORMATION SHEET

Ams CP/M+

HiSoft FORTH

£19.95

**A Compiler with GSX Graphics
for your Amstrad CPC6128 & PCW8256/8512/9512**

HiSoft FORTH is a fast, compact language which produces programs that run many times faster than equivalent BASIC programs.

It is interactive and flexible in nature and reduces program development time compared with other compiled languages.

HiSoft FORTH has the useful ability to write and compile sections of code and then test them with the compiler already resident in memory.

Only when the program is fully debugged and tested need it be compiled into a stand-alone program which may be run as a normal CP/M program.

HiSoft FORTH is based on Fig-FORTH, has been specifically written to interface with the CP/M Plus environment and as such includes many features not usually found in FORTH compilers.

- CP/M Plus operating environment and file handling.
- Interactive screen editor.
- Fully structured assembler allowing assembly code definitions to be freely intermixed with high level FORTH code.
- Full support for CP/M GSX (graphics system extension).
- Low level CP/M BDOS interface.
- Large & expandable utility library.
- FORTH programs may be compiled to provide a stand-alone COM file which may be run from CP/M.
- Support for the PCW8256/8512/9512 RAM disk.
- 90-page manual with complete HiSoft FORTH reference plus useful tutorial on the FORTH language.

HiSoft FORTH comes on one 3" disk with an informative manual, suitable for the CPC6128, PCW8256 and PCW8512/9512 machines.

INFORMATION SHEET

Benchmarks

The following is a collection of benchmark timings for some of the programs listed in this catalogue. It is not meant to be complete but should give an indication of the performance of the packages.

The benchmarks are based on the Personal Computer World tests (for Pascal, FTL Modula-2 and C).

Package	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Pascal80	0.2	3.8	6.2	5.3	4.8	5.1	25.3	20.3	11.6	8.5	4.3	8.5	5.3	5.2	8.6
HiSoft C	1.3	15.5	17.0	15.5	17.1	17.8	***	***	99.1	24.5	6.0	24.8	6.7	6.7	***
Modula-2	.03	4.2	6.7	6.0	5.2	5.4	192	164	10.5	9.3	1.02	9.2	3.0	3.1	265

Timings in seconds, HiSoft C is integer only and thus has no timings for the float benchmarks (7,8 & 15). FTL Modula-2 is fairly slow on floats but *very* accurate.

What you need

HiSoft CP/M software runs on all CP/M 2.2 and CP/M 3 (CP/M Plus) computers unless otherwise specified. **Ams CP/M** means the Amstrad CPC and PCW range of computers whereas **Ams CP/M+** refers to the Amstrad CPC6128 and Amstrad PCW machines running under CP/M Plus.

Special versions of HiSoft Pascal80, Devpac80, C and Nevada COBOL are available for the Atari ST (using the CP/M emulator supplied), MSX disk and the Tatung Einstein computers. The MSX titles include MSXDOS, the CP/M compatible dos for MSX.

In general, one disk drive and at least 36K of tpa (transient program area) is required to run our CP/M software although FTL Modula-2 needs at least 54K of tpa and will certainly benefit from more available disk space.

We can supply in most common disk formats, including IBM 8". Please specify your format carefully.

Ordering

HiSoft Software is available from all good computer shops or, in case of difficulty, directly from HiSoft at The Old School, Greenfield, Bedford MK45 5DE UK. Tel: (0525) 718181, Fax: (0525) 713716.

You can pay by Access (Eurocard, Mastercard), Visa (Barclaycard, Carte Blanche), Girocheque or a cheque drawn on a UK bank in pounds sterling.

Prices include 15% VAT and postage within the United Kingdom.

Please phone or fax for our export prices. The export prices will include postage & packing but exclude any local tax and import duty.