APPENDIX 3 DATA REPRESENTATION AND STORAGE.

A 3.1 Data Representation.

The following discussion details how data is represented internally by Hisoft Pascal 4.

The information on the amount of storage required in each case should be of use to most programmers (the SIZE function may be used see Section 2.3.6.7); other details may be needed by those attempting to merge Pascal and machine code programs.

A 3.1.1 Integers.

Integers occupy 2 bytes of storage each, in 2's complement form.
Examples:

$$
\begin{array}{rcl}
1 & \equiv & £0001 \\
256 & \equiv & £0100 \\
-256 & \equiv & £FF00
\end{array}
$$

The standard Z80 register used by the compiler to hold integers is HL.

A 3.1.2 Characters, Booleans and other Scalars.

These occupy 1 byte of storage each, in pure, unsigned binary.

Characters: 8 bit, extended ASCII is used.

$$
\begin{array}{rcl}
\text{'E'} & \equiv & £45 \\
\text{'['} & \equiv & £5B
\end{array}
$$

Booleans:
$$ORD(TRUE) = 1 \quad \text{so TRUE is represented by 1.}$$
$$ORD(FALSE) = 0 \quad \text{so FALSE is represented by 0.}$$

The standard Z80 register used by the compiler for the above is A.

A 3.1.3 Reals.

The (mantissa, exponent) form is used similar to that used in standard scientific notation - only using binary instead of denary. Examples:

$$2 \equiv 2*10^0 \quad \text{or} \quad 1.0_2 * 2^1$$

$$1 \equiv 1*10^0 \quad \text{or} \quad 1.0_2 * 2^0$$