

# Random File I/O

This describes how to implement random access files with fixed length records. This is made possible by opening the file for RANDOM access then using the SEEK command to position the read/write pointer within the file.

For random access the file should be opened with the keyword RANDOM. For example:

```
OPEN "filename" FOR RANDOM AS #1
```

To seek to a record within the file you would use the SEEK command. For example, the fifth record in a file that uses 64 byte records starts at byte 256, so you would use the following to point to it:

```
SEEK #1, 256
```

When reading from a random access file the INPUT\$() function should be used as this will read a fixed number of bytes (ie, a complete record) from the file. For example, to read a record of 64 bytes you would use:

```
dat$ = INPUT$(64, #1)
```

When writing to the file a fixed record size should be used and this can be easily accomplished by adding sufficient padding characters (normally spaces) to the data to be written. For example:

```
PRINT #1, dat$ + SPACE$(64 - LEN(data$));
```

The SPACE\$() function is used to add enough spaces to ensure that the data written is an exact length (64 bytes in this example). The semicolon at the end of the print command suppresses the addition of the carriage return and line feed characters which would make the record longer than intended.

Two other functions can help when using random file access. The LOC() function will return the current position of the read/write pointer and the LOF() function will return the total length of the file in bytes.

The following program demonstrates random file access. Using it you can append to the file (to add some data in the first place) then read/write records using random record numbers. The first record in the file is record number 1, the second is 2, etc.

```
RecLen = 64
OPEN "test.dat" FOR RANDOM AS #1
DO
  abort: PRINT
  PRINT "Number of records in the file =" LOF(#1)/RecLen
  INPUT "Command (r = read, w = write, a = append, q = quit): ", cmd$
  IF cmd$ = "q" THEN CLOSE #1 : END
  IF cmd$ = "a" THEN
    SEEK #1, LOF(#1)
  ELSE
    INPUT "Record Number: ", nbr
    IF nbr < 1 or nbr > LOF(#1)/RecLen THEN PRINT "Invalid record" : GOTO abort
    SEEK #1, RecLen * (nbr - 1)
  ENDIF
  IF cmd$ = "r" THEN
    PRINT "The record = " INPUT$(RecLen, #1)
  ELSE
    LINE INPUT "Enter the data to be written: ", data$
    PRINT #1, data$ + SPACE$(RecLen - LEN(data$));
  ENDIF
LOOP
```

Random access can also be used on a normal text files. For example, this will print out a file backwards:

```
OPEN "file.txt" FOR RANDOM AS #1
FOR i = LOF(#1) - 1 TO 0 STEP -1
  SEEK #1, i
  PRINT INPUT$(1, #1);
NEXT i
CLOSE #1
```