File Handling in QBasic - Notes

FILE HANDLING IN QBASIC

---

1. INTRODUCTION

In QBasic, file handling means storing and retrieving data from files on disk.

This allows programs to save data permanently (not just in memory) and read it back when needed.

There are three main types of files in QBasic:

a) Sequential Access Files

b) Random Access Files

c) Binary Files (less common in simple QBASIC programs)

---

2. TYPES OF FILES

A) SEQUENTIAL ACCESS FILES

- Data is stored and read in a sequence (one after another).
- Suitable for text files and lists.
- Reading always starts from the beginning.
- Common for simple data storage.

Statements used:

OPEN "filename" FOR OUTPUT AS #1 -> Create/write to file (overwrites existing data)

OPEN "filename" FOR APPEND AS #1 -> Add data to the end of file

OPEN "filename" FOR INPUT AS #1 -> Read data from file

PRINT #1, data -> Write to file

INPUT #1, variable -> Read from file

CLOSE #1 -> Close the file

File Handling in QBasic - Notes

Example (Writing):

```
OPEN "student.txt" FOR OUTPUT AS #1
PRINT #1, "Ram"
PRINT #1, "Shyam"
CLOSE #1
```

Example (Reading):

```
OPEN "student.txt" FOR INPUT AS #1
INPUT #1, name$
PRINT name$
CLOSE #1
```

B) RANDOM ACCESS FILES

- Data can be read/written directly at a specific position (record).
- Useful for databases where each record has a fixed size.
- Requires defining a record type with fixed-length fields.

Steps:

1. Define a TYPE structure for data fields.
2. Use OPEN ... FOR RANDOM.
3. Use GET and PUT statements for reading/writing records.

Example:

```
TYPE Student
  name AS STRING * 20
  age AS INTEGER
END TYPE

DIM rec AS Student
```

File Handling in QBasic - Notes

```
OPEN "student.dat" FOR RANDOM AS #1 LEN = LEN(rec)

rec.name = "Sita"
rec.age = 20
PUT #1, 1, rec   ' Store at record 1

CLOSE #1
```

C) BINARY FILES

- Store data in binary form.
- Used for special data like images or compiled programs.
- Requires precise control over bytes.

Example:

```
OPEN "file.bin" FOR BINARY AS #1
PUT #1, , data$
CLOSE #1
```

---

3. COMMON FILE HANDLING COMMANDS IN QBASIC

OPEN - Opens a file for reading, writing, or appending.

CLOSE - Closes an open file.

PRINT # - Writes data to a file.

INPUT # - Reads data from a file.

GET - Reads a record from a random access file.

PUT - Writes a record to a random access file.

EOF - Checks if the end of file is reached.

File Handling in QBasic - Notes

---

4. END OF FILE (EOF) FUNCTION

- Returns TRUE when the file pointer reaches the end of the file.
- Used in loops to read until the end.

Example:

```
OPEN "student.txt" FOR INPUT AS #1
WHILE NOT EOF(1)
  INPUT #1, name$
  PRINT name$
WEND
CLOSE #1
```

---

5. FILE HANDLING MODES SUMMARY

Mode	Purpose	Example
OUTPUT	Create new file / overwrite existing	OPEN "data.txt" FOR OUTPUT AS #1
APPEND	Add data to existing file	OPEN "data.txt" FOR APPEND AS #1
INPUT	Read from file	OPEN "data.txt" FOR INPUT AS #1
RANDOM	Access file by records	OPEN "data.dat" FOR RANDOM AS #1
BINARY	Access file byte-by-byte	OPEN "data.bin" FOR BINARY AS #1

---

6. ADVANTAGES OF FILE HANDLING

- Stores data permanently.

File Handling in QBasic - Notes

- Can read and process large amounts of data.
- Can create and maintain databases.

7. LIMITATIONS

- Sequential files require reading from the start for every search.
- Random access files need fixed-length records.
- File handling requires careful error checking.

---

8. SAMPLE PROGRAM: STORING AND READING NAMES

```
CLS
OPEN "names.txt" FOR OUTPUT AS #1
FOR i = 1 TO 3
  INPUT "Enter name: ", n$
  PRINT #1, n$
NEXT i
CLOSE #1

PRINT "Stored names:"
OPEN "names.txt" FOR INPUT AS #1
WHILE NOT EOF(1)
  INPUT #1, n$
  PRINT n$
WEND
CLOSE #1
```

END OF NOTES