The FILE STATUS clause is available for any type of file organization and allows the programmer to distinguish between the many different types of I/O error conditions. The concept was first introduced in Chapter 6 in connection with debugging (see page 158). The operating system automatically returns a two-position field known as the *I/O status* (or file status bytes) to the data name designated in the FILE STATUS clause. The value of the file status bytes may be interrogated by the programmer, who is thus able to more closely monitor the results of any I/O operation.

Table 18.1 lists the various file status codes and their meaning. The use of file status codes is illustrated in the ensuing program to create an indexed file.

Table 18.1 File Status Codes

00	A successful input/output operation is performed with no further information available.
04	A READ is successful, but the length of the record being processed does not conform to the fixed file attributes for that file.
05	An OPEN is successful, but the referenced optional file is not present at open time.
07	An input/output statement is successful; however, for a CLOSE with NO REWIND, REEL/UNIT, or FOR REMOVAL or for an OPEN with NO REWIND the referenced file is on a nonreel/unit medium.
10	A sequential READ is attempted and no next logical record exists because (1) the end of file has been reached; or (2) an optional input file is not present.
14	A sequential READ is attempted and the number of significant digits in the record number is larger than the size of the key data item described for the file.
15	A sequential READ statement is attempted for the first time on an optional file that is not present.
21	A sequence error exists for a sequentially accessed indexed file.
22	An attempt is made to write or rewrite a record that would create a duplicate prime record key or duplicate alternate record key without the DUPLICATES phrase.
23	An attempt is made to randomly access a record that does not exist in the file, or a START or random READ is attempted on an optional input file that is not present.
24	An attempt is made to write beyond the externally defined boundaries.
25	A START statement or a random READ statement has been attempted on an optional file that is not present.
30	A permanent error exists and no further information is available concerning the input/output operation.
34	A permanent error exists because of a boundary violation; an attempt is made to write beyond the externally defined boundaries.
35	A permanent error exists because an OPEN with the INPUT, I/O, or EXTEND phrase is attempted on a nonoptional file that is not present.
37	A permanent error exists because an OPEN is attempted on a file and that file will not support the open mode specified: (1) EXTEND or OUTPUT phrase specified but not supported by the file; (2) I/O phrase is specified, but input and output operations are not supported by the file; or (3) INPUT phrase is specified, but the file will not support READ operations.
38	A permanent error exists because an OPEN is attempted on a file previously closed with a lock.
39	The OPEN is unsuccessful because a conflict has been detected between the fixed file attributes and the ones specified for that file in the program.
41	An OPEN statement is attempted for a file in the open mode.
42	A CLOSE statement is attempted for a file not in the open mode.
43	In the sequential access mode, the last input/output statement executed for the file prior to the execution of a DELETE or REWRITE statement was not a successfully executed READ statement.
44	A boundary violation exists because of an attempt to: (1) write or rewrite a record that is larger than the largest or smaller than the smallest record allowed by the RECORD IS VARYING clause of the associated file-name, or (2) rewrite a record and the record is not the same size as the record being replaced.
46	A sequential READ is attempted on a file open in the input or I/O mode and no valid next record has been established because the preceding: (1) START was unsuccessful, (2) READ was unsuccessful but did not cause an at-end condition, or (3) READ caused an at-end condition.
47	The execution of a READ or START is attempted on a file not open in the input or I/O mode.
48	The execution of a WRITE is attempted on a file not open in the I/O, output, or extend mode.
49	The execution of a DELETE or REWRITE statement is attempted on a file not open in the I/O mode.