



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
Error Handling in File I/O

- When trying to read a file beyond indicator.
- When trying to read a file that does not exist.
- When trying to use a file that has not been opened.
- When trying to use a file in an inappropriate mode i.e., writing data to a file that has been opened for reading.
- When writing to a file that is write-protected i.e., trying to write to a read-only file.

Failure to check for errors then the program may behave abnormally therefore an unchecked error may result in premature termination for the program or incorrect output.

Below are some [Error handling](#) functions during [file operations](#) in C/C++:

`ferror()`:

In C/C++, the library function [ferror\(\)](#) is used to check for the error in the stream. Its prototype is written as:

```
int ferror (FILE *stream);
```

The **ferror()** function checks for any error in the stream. It returns a value zero if no error has occurred and a non-zero value if there is an error. The error indication will last until the file is closed unless it is cleared by the **clearerr()** function.

Below is the program to implement the use of **ferror()**:

```
// C program to illustrate the
// use of ferror()
#include <stdio.h>
#include <stdlib.h>

// Driver Code
int main()
{
    FILE* fp;

    // If a file is opened which does
    // not exist, then it will be an
    // error and corresponding errno
    // value will be set
    char feedback[100];

    int i;
    fp = fopen("GeeksForGeeks.TXT", "w");

    if (fp == NULL) {
        printf("\n The file could "
            "not be opened");
        exit(1);
    }
}
```



SNS COLLEGE OF TECHNOLOGY, COIMBATORE –35
(An Autonomous Institution)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

```
printf("\n Provide feedback on "  
      "this article: ");  
fgets(feedback, 100, stdin);  
  
for (i = 0; i < feedback[i]; i++)  
    fputc(feedback[i], fp);  
  
// Error writing file  
if (ferror(fp)) {  
    printf("\n Error writing in file");  
    exit(1);  
}  
  
// Close the file pointer  
fclose(fp);  
}
```

Output:

```
Provide feedback on this article: Test feedback  
-----  
Process exited after 46.13 seconds with return value 0  
Press any key to continue . . .
```