

## Error handling in File I/O

Sometimes during file operations, errors may occur. Let us take an example, a file being opened for reading might not exist. Or a file name used for a new file may already exist. Or an attempt could be made to read past the end-of-file. Or such an invalid operation may be performed.

To find the solution for such errors and to ensure smooth processing, C++ file streams inherit 'stream-state' members from the `ios` class that store the information on the status of a file that is being currently used. The current state of the I/O system is held in an integer, in which the following flags are encoded:

Name	Meaning
<code>eofbit</code>	1 when end-of-file is encountered, 0 otherwise.
<code>failbit</code>	1 when a non-fatal I/O error has occurred, 0 otherwise
<code>badbit</code>	1 when a fatal I/O error has occurred, 0 otherwise
<code>goodbit</code>	0 value

## C++ Error Handling Function

There are several error handling functions supported by class `ios` that help you read and process the status recorded in a file stream.

Following table lists these error handling functions and their meaning:

Function	Meaning
<code>int bad()</code>	Returns a non-zero value if an invalid operation is attempted or any unrecoverable error has occurred. However, if it is zero (false value), it may be possible to recover from any other error reported and continue operations.
<code>int eof()</code>	Returns non-zero (true value) if end-of-file is encountered while reading; otherwise returns zero (false value).
<code>int fail()</code>	Returns non-zero (true) when an input or output operation has failed.
<code>int good()</code>	Returns non-zero (true) if no error has occurred. This means, all the above functions are false. For example, if <code>fin.good()</code> is true, everything is okay with the stream named as <code>fin</code> and we can proceed to perform I/O operations. When it returns zero, no further operations can be carried out.
<code>clear()</code>	Resets the error state so that further operations can be attempted.

The above functions can be summarized as eof() returns true if eofbit is set; bad() returns true if badbit is set. The fail() function returns true if failbit is set; the good() returns true there are no errors. Otherwise, they return false.