std::fopen

Defined in header <cstdio>

```
std::FILE* fopen( const char* filename, const char* mode );
```

Opens a file indicated by filename and returns a file stream associated with that file. mode is used to determine the file access mode.

Parameters

filename - file name to associate the file stream to

mode - null-terminated character string determining file access mode

File access mode string	Meaning	Explanation	Action if file already exists	Action if file does not exist
"r"	read	Open a file for reading	read from start	failure to open
"w"	write	Create a file for writing	destroy contents	create new
"a"	append	Append to a file	write to end	create new
"r+"	read extended	Open a file for read/write	read from start	error
"w+"	write extended	Create a file for read/write	destroy contents	create new
"a+"	append extended	Open a file for read/write	write to end	create new
File access mode flag "b" can optionally be specified to open a file in binary mode. This flag has no effect on POSIX systems, but on Windows, for example, it disables special handling of `'\n'` and `'\x1A'`. On the append file access modes, data is written to the end of the file regardless of the current position of the file position indicator.				
File access mode flag "x" can optionally be appended to "w" or "w+" specifiers. This flag forces the function to fail if the file exists, instead of overwriting it. (C++17)				
The behavior is undefined if the mode is not one of the strings listed above. Some implementations define additional supported modes (e.g. Windows (https://msdn.microsoft.com/en-us/library/yeby3zcb.aspx)).				

Return value

If successful, returns a pointer to the object that controls the opened file stream, with both eof and error bits cleared. The stream is fully buffered unless *filename* refers to an interactive device.

On error, returns a null pointer. POSIX requires (http://pubs.opengroup.org/onlinepubs/9699919799/functions/fopen.html) that errno is set in this case.

Notes

The format of filename is implementation-defined, and does not necessarily refer to a file (e.g. it may be the console or another device accessible through filesystem API). On platforms that support them, filename may include absolute or relative filesystem path.

For portable directory and file naming, see C++ filesystem library or boost::filesystem (http://www.boost.org/doc/libs/release/libs/filesystem/doc/index.htm) .

Example

Run this code

```
#include <cstdio>
#include <cstdlib>

int main()
{
    int is_ok = EXIT_FAILURE;
    FILE* fp = std::fopen("/tmp/test.txt", "w+");
    if(!fp) {
        std::perror("File opening failed");
        return is_ok;
    }
```

```
    int c; // note: int, not char, required to handle EOF
    while ((c = std::fgetc(fp)) != EOF) { // standard C I/O file reading loop
       std::putchar(c);
    }

    if (std::ferror(fp)) {
        std::puts("I/O error when reading");
    } else if (std::feof(fp)) {
        std::puts("End of file reached successfully");
        is_ok = EXIT_SUCCESS;
    }

    std::fclose(fp);
    return is_ok;
}
```

Output:

```
End of file reached successfully
```

See also

fclose	closes a file (function)
fflush	synchronizes an output stream with the actual file (function)
freopen	open an existing stream with a different name (function)
C documentation for **fopen**	