

# C library function - freopen()

## Description

The C library function **FILE \*freopen(const char \*filename, const char \*mode, FILE \*stream)** associates a new **filename** with the given open stream and at the same time closes the old file in the stream.

## Declaration

Following is the declaration for freopen() function.

```
FILE *freopen(const char *filename, const char *mode, FILE *stream)
```

## Parameters

- **filename** – This is the C string containing the name of the file to be opened.
- **mode** – This is the C string containing a file access mode. It includes –

Sr.No.	Mode & Description
1	<b>"r"</b> Opens a file for reading. The file must exist.
2	<b>"w"</b> Creates an empty file for writing. If a file with the same name already exists then its content is erased and the file is considered as a new empty file.
3	<b>"a"</b> Appends to a file. Writing operations appends data at the end of the file. The file is created if it does not exist.
4	<b>"r+"</b> Opens a file to update both reading and writing. The file must exist.
5	<b>"w+"</b> Creates an empty file for both reading and writing.
6	<b>"a+"</b> Opens a file for reading and appending.

- **stream** – This is the pointer to a FILE object that identifies the stream to be re-opened.

## Return Value

If the file was re-opened successfully, the function returns a pointer to an object identifying the stream or else, null pointer is returned.

## Example

The following example shows the usage of freopen() function.

```
#include <stdio.h>
```

[Live Demo](#)

```
int main () {
    FILE *fp;

    printf("This text is redirected to stdout\n");

    fp = freopen("file.txt", "w+", stdout);

    printf("This text is redirected to file.txt\n");

    fclose(fp);

    return(0);
}
```

Let us compile and run the above program that will send the following line at STDOUT because initially we did not open stdout –

```
This text is redirected to stdout
```

After a call to **freopen()**, it associates STDOUT to file **file.txt**, so whatever we write at STDOUT that goes inside **file.txt**. So, the file **file.txt** will have the following content.

```
This text is redirected to file.txt
```

Now let's see the content of the above file using the following program –

```
#include <stdio.h>

int main () {
    FILE *fp;
    int c;

    fp = fopen("file.txt", "r");
    while(1) {
        c = fgetc(fp);
        if( feof(fp) ) {
            break ;
        }
        printf("%c", c);
    }
    fclose(fp);
    return(0);
}
```