

SALE!

✕

GeeksforGeeks Courses Upto 25% Off Enroll Now!



Save 25% on Courses DSA Data Structures Algorithms Interview Preparation Data Science T

I/O Redirection in C++

Difficulty Level : Medium • Last Updated : 22 Jun, 2022

[Read](#) [Discuss](#) [Courses](#) [Practice](#) [Video](#)

In C, we could use the function [freopen\(\)](#) to redirect an existing FILE pointer to another stream. The prototype for `freopen()` is given as

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

For Example, to redirect the stdout to say a textfile, we could write :

```
freopen ("text_file.txt", "w", stdout);
```

While this method is still supported in C++, this article discusses another way to redirect I/O streams.

C++ being an object-oriented programming language, gives us the ability to not only define our own streams but also redirect standard streams. Thus, in C++, a stream is an object whose behavior is defined by a class. Thus, anything that behaves like a stream is also a stream.

Streams Objects in C++ are mainly of three types :

- **istream** : Stream object of this type can only perform input operations from the stream
- **ostream** : These objects can only be used for output operations.
- **iostream** : Can be used for both input and output operations

All these classes, as well as file stream classes, derived from the classes: `ios` and `streambuf`. Thus, `filestream` and `IO stream` objects behave similarly.

All stream objects also have an associated data member of class `streambuf`. Simply put, `streambuf` object is the buffer for the stream. When we read data from a stream, we don't read it directly from the source, but instead, we read it from the buffer which is linked to the

source. Similarly, output operations are first performed on the buffer, and then the buffer is flushed (written to the physical device) when needed.

C++ allows us to set the stream buffer for any stream, So the task of redirecting the stream simply reduces to changing the stream buffer associated with the stream. Thus, to redirect a Stream A to Stream B we need to do:-

AD

1. Get the stream buffer of A and store it somewhere
2. Set the stream buffer of A to the stream buffer of B
3. If needed to reset the stream buffer of A to its previous stream buffer

We can use the function [`ios::rdbuf\(\)`](#) to perform below two operations.

- 1) `stream_object.rdbuf()`: Returns pointer to the stream buffer of `stream_object`
- 2) `stream_object.rdbuf(streambuf * p)`: Sets the stream buffer to the object pointed by `p`

Here is an example program below to show the steps

CPP

```
// Cpp program to redirect cout to a file
#include <fstream>
#include <iostream>
#include <string>

using namespace std;

int main()
{
    fstream file;
    file.open("cout.txt", ios::out);
    string line;

    // Backup streambuffers of cout
    streambuf* stream_buffer_cout = cout.rdbuf();
    streambuf* stream_buffer_cin = cin.rdbuf();
```

```
// Get the streambuffer of the file
streambuf* stream_buffer_file = file.rdbuf();

// Redirect cout to file
cout.rdbuf(stream_buffer_file);

cout << "This line written to file" << endl;

// Redirect cout back to screen
cout.rdbuf(stream_buffer_cout);
cout << "This line is written to screen" << endl;

file.close();
return 0;
}
```

Output:

```
This line is written to screen
Contents of file cout.txt:
This line written to file
```

Time Complexity: $O(1)$

Space Complexity: $O(1)$

Note:

The above steps can be condensed into a single step

```
auto cout_buf = cout.rdbuf(file.rdbuf())

// sets couts streambuffer and returns the old
streambuffer back to cout_buf
```

References:

[C++ IOS](#)

138

Related Articles

1. C++ Error - Does not name a type
2. Execution Policy of STL Algorithms in Modern C++