

# Implement an HLS stream in C++ using two different sources based on .m3u8 files

In this example, the code reads the URLs from two different .m3u8 files `source1.m3u8` and `source2.m3u8`, combines and sorts them, and writes the combined list of URLs to a new .m3u8 file `stream.m3u8`. The output file can be used as the manifest for an HLS stream.

```
#include <iostream>
#include <fstream>
#include <vector>
#include <algorithm>

const std::string source1 = "source1.m3u8";
const std::string source2 = "source2.m3u8";
const std::string output = "stream.m3u8";

int main()
{
    std::vector<std::string> urls1;
    std::vector<std::string> urls2;

    // Read URLs from source1
    std::ifstream file1(source1);
    std::string line;
    while (std::getline(file1, line))
    {
        if (line[0] != '#')
        {
            urls1.push_back(line);
        }
    }

    // Read URLs from source2
    std::ifstream file2(source2);
    while (std::getline(file2, line))
    {
        if (line[0] != '#')
        {
            urls2.push_back(line);
        }
    }

    // Combine and sort URLs
    std::vector<std::string> urls(urls1);
    urls.insert(urls.end(), urls2.begin(), urls2.end());
    std::sort(urls.begin(), urls.end());

    // Write combined URLs to output
    std::ofstream stream(output);
    stream << "#EXTM3U\n";
    for (const auto& url : urls)
    {
        stream << url << "\n";
    }

    return 0;
}
```