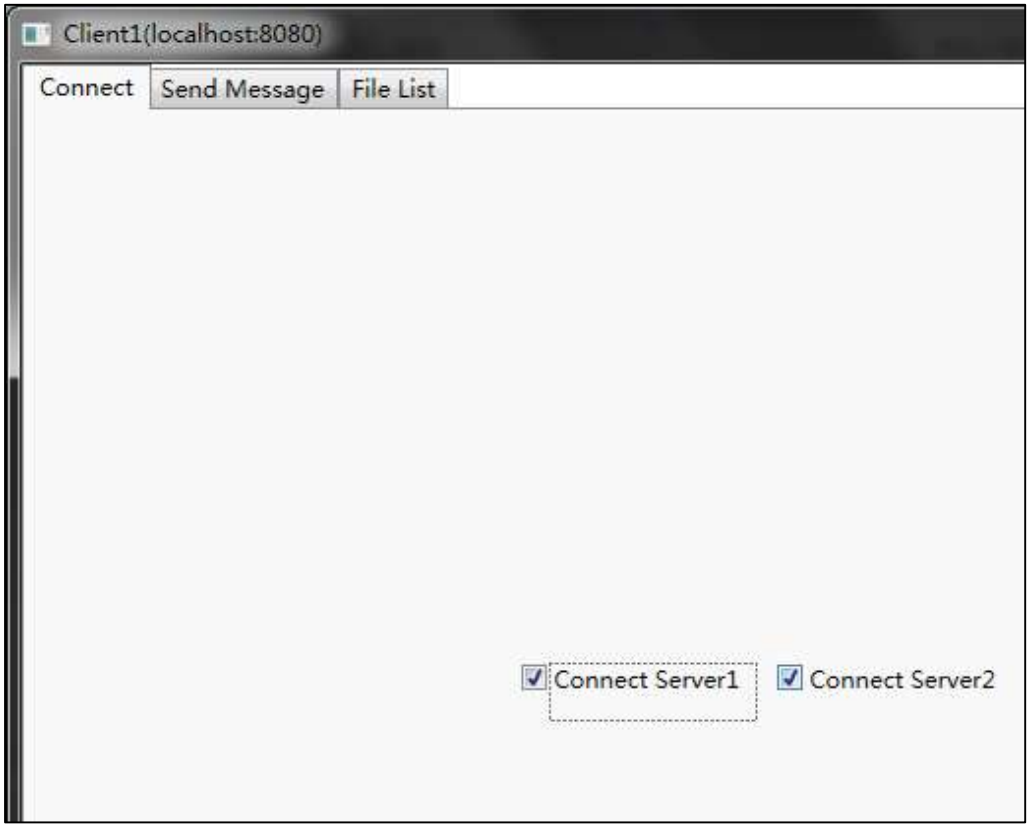
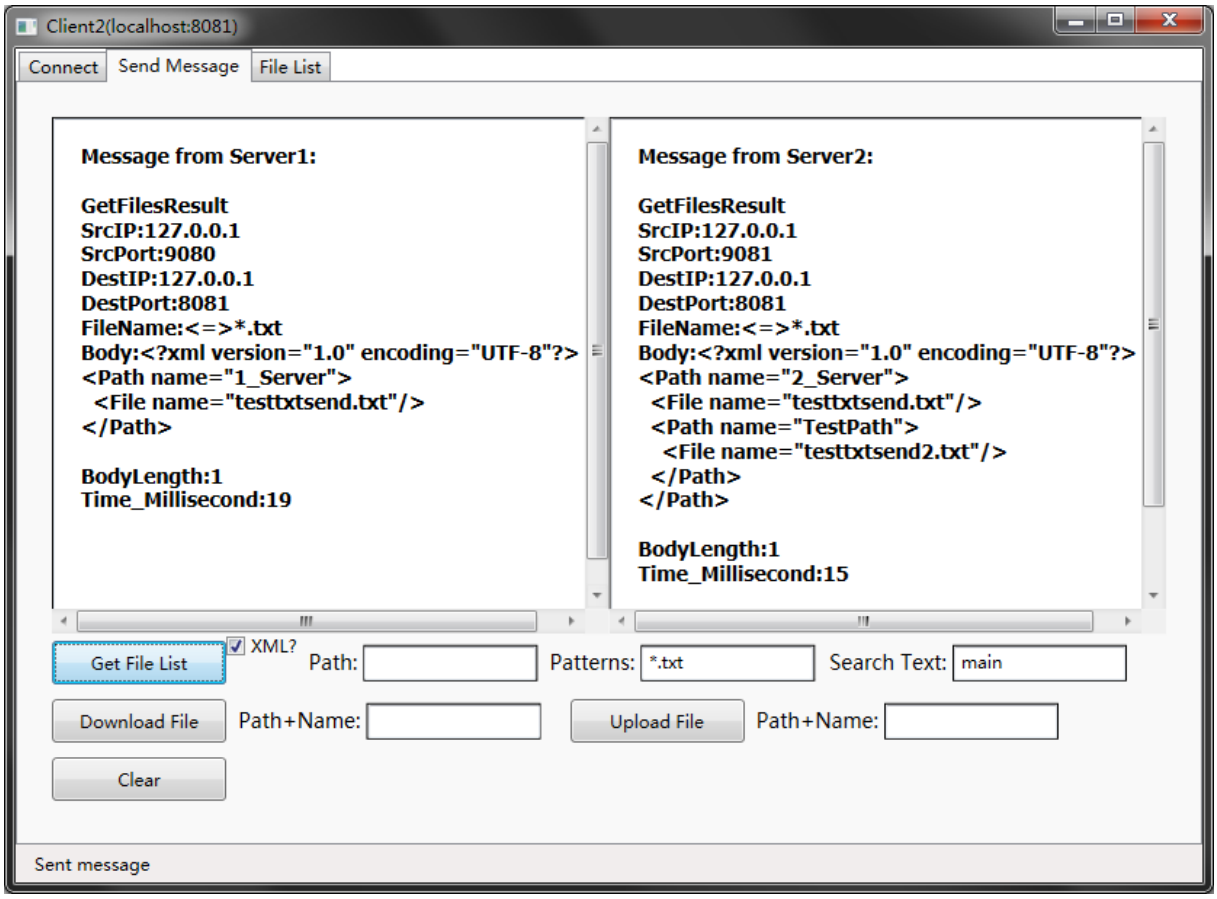
Abstract

- Client 1 program's root path was in '**CppCli-WPF-App**' file; Client 2 in '**2_client_WPF**'; Server 1 in '**1_Server**'; server 2 in '**2_Server**'
- Port#: client1=8080; client2=8081; server1=9080; server2=9081
- In order to test the program, please specify the server to connect in Connect Tab. You can connect to **1-2 servers at the same time** during a file search or text search; but you can **only connect to 1 server** during file upload/download.
- 'Get File List' button in send message tab was used for file search and text search. Input **nothing** in the following text box and press the button would give you all files rooted at server path.
- 'Download/Upload File' button in require 'Path+Name' to specify the file to transfer. Since the path was set to the client root path by default, you could put the file in the client/server root path, input the **file name only**, and perform the action.
- However the download action would give you an 'uploadsuccess/uploadfail' command prompt, it is indeed doing file download.
- 'File list' tab was built in the original demo was preserved for your reference.
- You could use 'testsendbin.pdf' and 'testtxtsend.txt' for testing file upload and download.

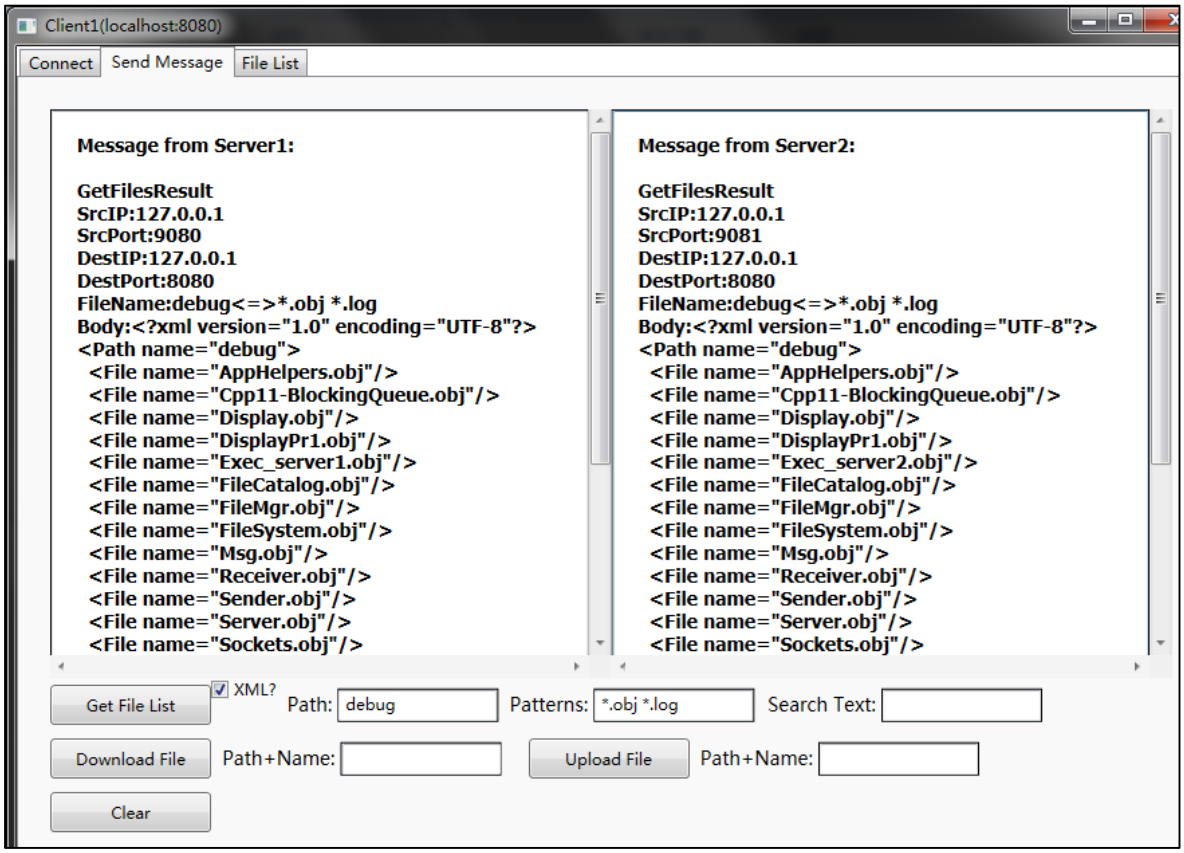
Sample Input/Output

1. Search text "main" in server1 and server2 with file pattern '*.txt', get XML string result.

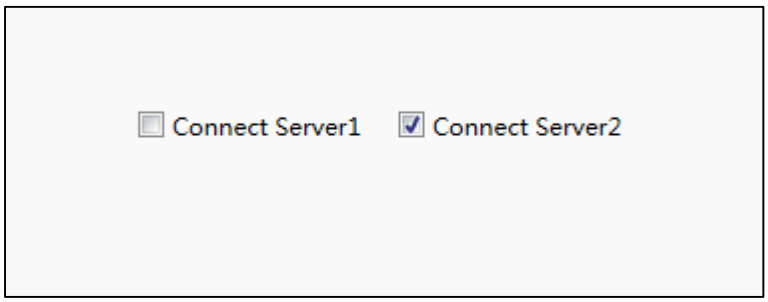


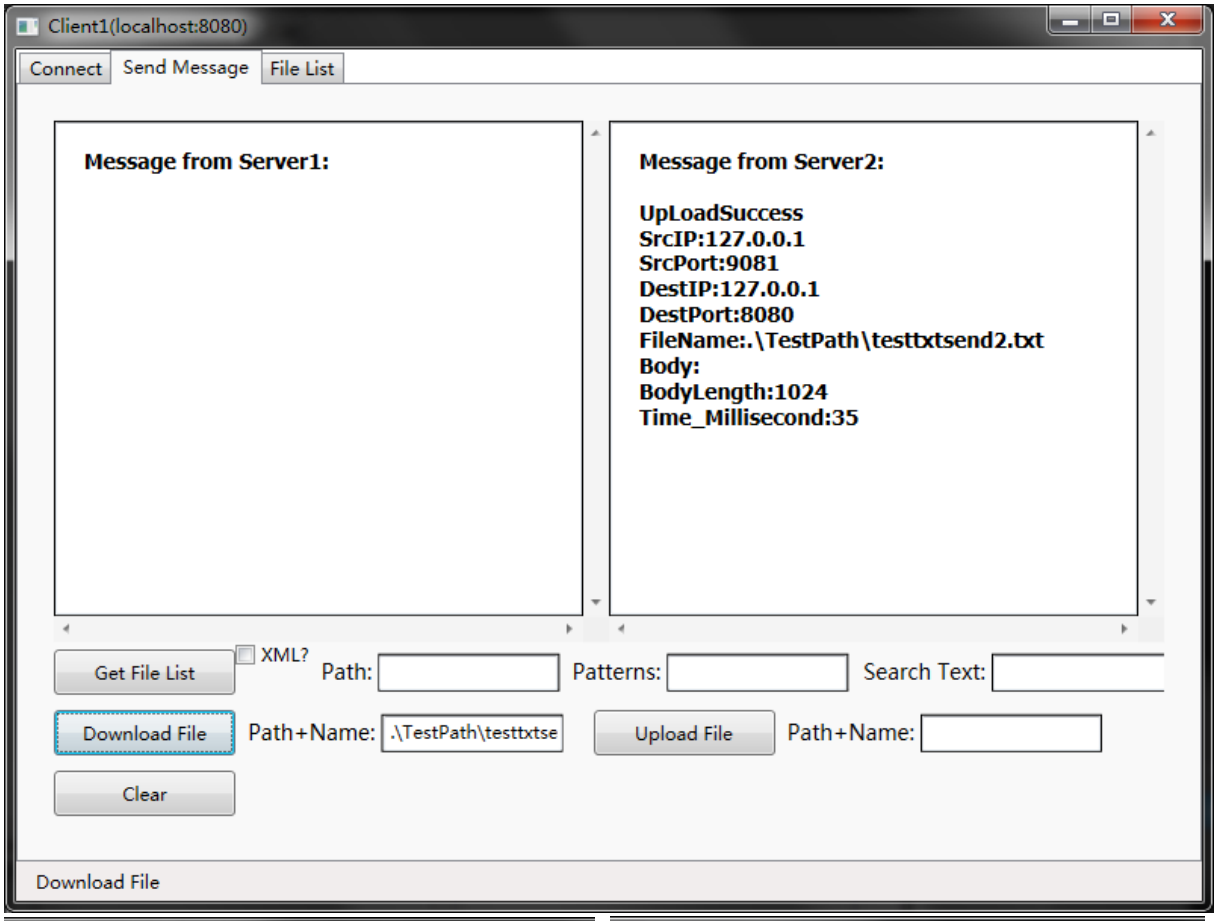


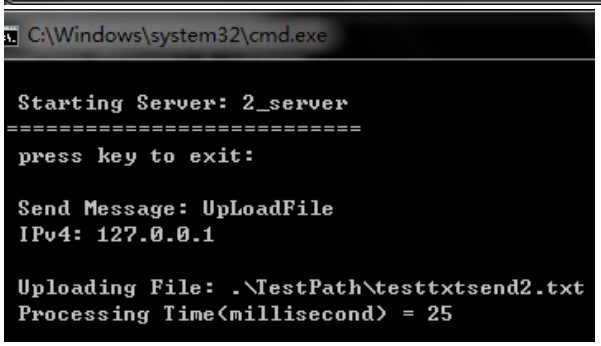
2. Get file list form server1 and 2 in path ‘debug’, and file patterns *.obj, *.log

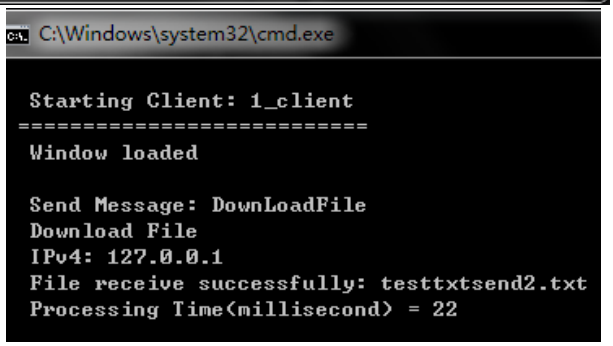


3. Download file ‘testtxtsend2.txt’ form server2 (.\2_Server\TestPath\) to client1(.\CppCli-WPF-App\)

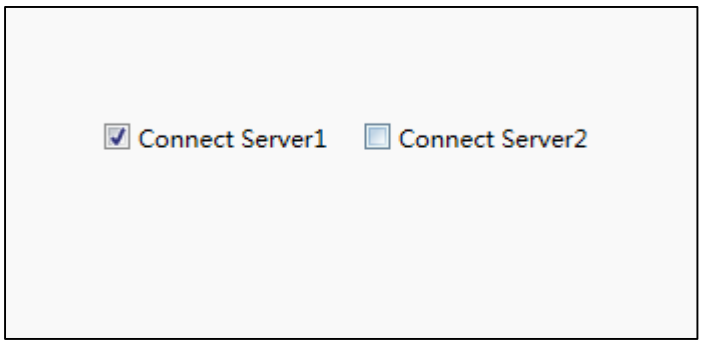


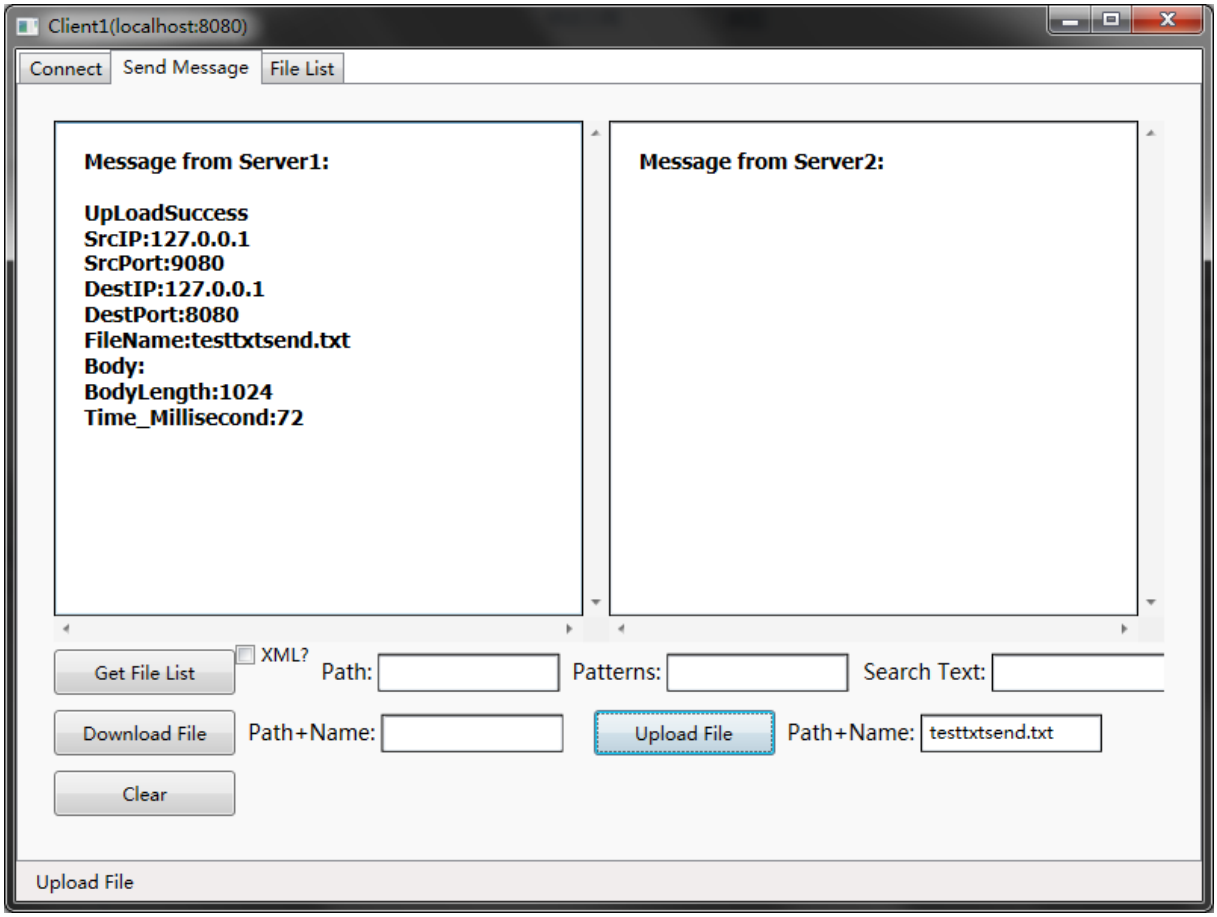


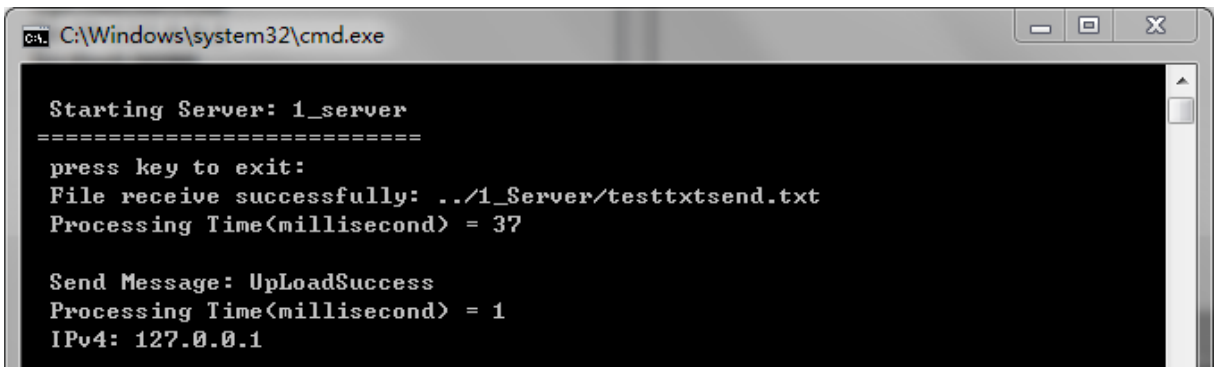




4. Upload file ‘testtxtsend.txt’ form client1 to server1







Detailed Explanation

Requriement3
Please use Download/Upload File button in ‘Send Message’ Tab for corresponding requirement; please use Get File List Button in ‘Send Message’ Tab for exploring one specific category and performing file searches.

Requriement4
Client process is obviously using Windows Presentation Foundation GUI and could support file transfer and processing requests and displaying results. Performance information was displayed in message box (Time_Millisecond) after every message.

Requriement5
Please connect to both server 1 and 2 to perform multiple end-points text search.

Requriement6
In response to a text search command, return a list of files that contain the text and the paths and machines on which they reside. Since each server message would go to the corresponding message box, it’s easy to tell which machine the files reside. I designed it this way for a clear demo. You could also tell the machine by the ‘SrcIP’ and ’SrcPort’ in the returned message.

Requriement7
Please test that press ‘Get File List’ button (in the send message Tab) and set path/patterns in its following text box.

Requriement8
Please check the ‘XML?’ check box before inquiring files and you shall see the returned XML string. To test whether it’s a valid XML string, please copy (Ctrl+C) the string into and .xml file and see if it could be properly opened and displayed.

Requriement9
I didn’t do that.

Requriement10. The time required to carry out a processing request is displayed on the server console; and the end-to-end time is displayed in the message box. This used std::chrono::high_resolution_clock, and please check the code for that.