Connect.

Doxygen 1.9.4

Chapter 1

1.1

.

Chapter 2

2.1

.

Chapter 3

3.1

.

Chapter 4

4.1 ClientHandler

Class for client processing.

```
#include <mdfile.h>
```

- **ClientHandler** (ErrorHandler handler)
- int autorized (int work_sock, string file_name, string file_error)

  Constructor of the ClientHandler class.
- int math (int work_sock)

  is a method for calculating mathematical operations.

4.1.1

Class for client processing.

for client processing

4.1.2

4.1.2.1 autorized()

```
int ClientHandler::autorized (
            int work_sock,
            string file_name,
            string file_error )
```

Constructor of the ClientHandler class.

handler	Error handler.

ClientHandler	ClientHandler::ClientHandler(ErrorHandler handler) { m_errorHandler = handler; } /∗∗

Client authorization method.

work_sock	is a socket for working with the client.
file_name	is the name of the file with the user's data.
file_error	is the name of the file for recording errors.

autorized	

4.1.2.2 math()

```
int ClientHandler::math (
             int work_sock )
```

is a method for calculating mathematical operations.

work_sock	is a socket for working with the client.

math	

:

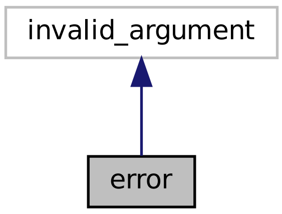
- mdfile.h
- mdfile.cpp

4.2 error

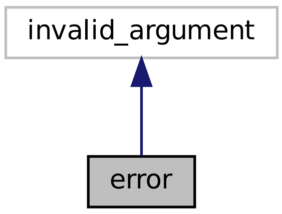
Class error.

```
#include <mdfile.h>
```

:error:



error:



- **error** (const std::string &what_arg)
- **error** (const char ∗what_arg)

4.2.1

Class error.

Output error message

:

- mdfile.h

4.3 ErrorHandler

Class for error handling.

```
#include <mdfile.h>
```

- **ErrorHandler** ()

  Class for error handling.

- static void errors (std::string error, std::string name)

  Method for writing errors to a file.
- static int er (std::string file_name, std::string file_error)

  Method for handling errors when opening a file.

4.3.1

Class for error handling.

for error handling

4.3.2

4.3.2.1 er()

```
int ErrorHandler::er (
            std::string file_name,
            std::string file_error )  [static]
```

Method for handling errors when opening a file.

file_name	File name
file_error	File name for recording errors

4.3.2.2 errors()

```
void ErrorHandler::errors (
            std::string error,
            std::string name )  [static]
```

Method for writing errors to a file.

error	Error text
name	File name for recording errors

:

- mdfile.h
- mdfile.cpp

4.4 Server

Class for the server.

```
#include <mdfile.h>
```

- Server (ErrorHandler handler)

  Class for the server.
- int self_addr (string error, string file_error, int port)

  Method for configuring the server address.
- int client_addr (int s, string error, string file_error)

  Method for setting up the client's address.

4.4.1

Class for the server.

for the server

4.4.2 ()

4.4.2.1 Server()

```
Server::Server (
            ErrorHandler handler )
```

Class for the server.

handler	handler

for	the server

-

4.4.3

4.4.3.1 client_addr()

```
int Server::client_addr (
            int s,
            string error,
            string file_error )
```

Method for setting up the client's address.

s	Server socket descriptor
error	Error text
file_error	File name for recording errors

client_addr	

4.4.3.2 self_addr()

```
int Server::self_addr (
            string error,
            string file_error,
            int port )
```

Method for configuring the server address.

error	Error text
file_error	File name for recording errors
port	Server port

self_addr	

:

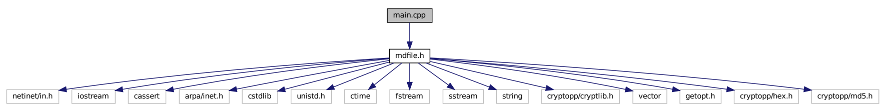
- mdfile.h
- mdfile.cpp

Chapter 5

5.1 main.cpp

Main program file.

```
#include "mdfile.h"
```
main.cpp:



- int **main** (int argc, char $*$argv[ ])

5.1.1

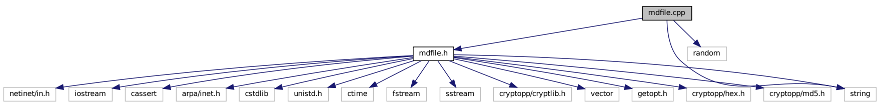
Main program file.

argc	Number of command line arguments
argv	Array of command line arguments

5.2 mdfile.cpp

```
#include "mdfile.h"
#include <random>
#include <string>
```

mdfile.cpp:

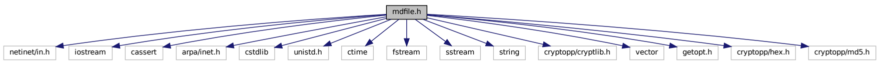


5.3 mdfile.h

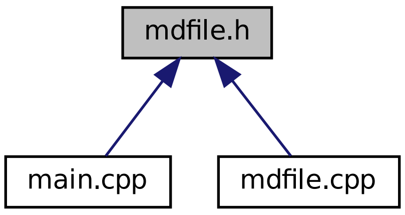
Description of the ErrorHandler, Server, ClientHandler and error_server classes.

```
#include <netinet/in.h>
#include <iostream>
#include <cassert>
#include <arpa/inet.h>
#include <cstdlib>
#include <unistd.h>
#include <ctime>
#include <fstream>
#include <sstream>
#include <string>
#include <cryptopp/cryptlib.h>
#include <vector>
#include <getopt.h>
#include <cryptopp/hex.h>
#include <cryptopp/md5.h>
```

mdfile.h:



, :



- class ErrorHandler

    Class for error handling.
- class Server

    Class for the server.
- class ClientHandler

    Class for client processing.
- class error

    Class error.

- #define **CRYPTOPP_ENABLE_NAMESPACE_WEAK** 1

5.3.1

Description of the ErrorHandler, Server, ClientHandler and error_server classes.

Arseniy Batrakov

1.0

10.12.2023

Header file for Connect

5.4 mdfile.h

```
. .

#include <netinet/in.h>
#include <iostream>
#include <cassert>
#include <arpa/inet.h>
#include <cstdlib>
#include <unistd.h>
#include <ctime>
#include <fstream>
#include <sstream>
#include <string>
#include <cryptopp/cryptlib.h>
#include <iostream>
#include <vector>
#include <getopt.h>
#include <cryptopp/hex.h> // HexEncoder
#define CRYPTOPP_ENABLE_NAMESPACE_WEAK 1
using namespace CryptoPP;
using namespace std;
#include <cryptopp/md5.h>

class ErrorHandler
```

```
{
public:
    ErrorHandler();
    static void errors(std::string error, std::string name);

    static int er(std::string file_name, std::string file_error);
};
class Server
{
public:
    Server(ErrorHandler handler);

    int self_addr(string error, string file_error, int port);

    int client_addr(int s, string error, string file_error);

private:

    ErrorHandler m_errorHandler;
};
class ClientHandler
{
public:
    ClientHandler(ErrorHandler handler);
    int autorized(int work_sock, string file_name, string file_error);
    int math(int work_sock);

private:
    std::string generate_salt(std::size_t length);

    void msgsend(int work_sock, string mess);

    std::string MD(std::string sah);

    ErrorHandler m_errorHandler;
};

class error:  public invalid_argument
{
public:
    explicit error (const std::string& what_arg):
        std::invalid_argument(what_arg) {}
    explicit error (const char* what_arg):
        std::invalid_argument(what_arg) {}
};

```