Connect.

Doxygen 1.9.4

Chapter 1

1.1

.

Chapter 2

2.1

.

Chapter 3

3.1

.

Chapter 4

4.1 Connect

Class connect server.

```
#include <Connect.h>
```

- int Connect_to_server (string str1, string str2)

  Use to server.
- void **GetLoginPassword** ()

  Recv Login and password.

- int **ip_addr**
- int **port**
- string **name_original_file**
- string **name_result_file**
- string **name_auto_file**
- string **username**
- string **pswd**

4.1.1

Class connect server.

ip_addr	addr serv
port	port serv
name_original_file	name file for read vector
name_result_file	name file for read write result
name_auto_file	name file for login and password
username	lofin client
pswd	password client

4.1.2

4.1.2.1 Connect_to_server()

```
int Connect::Connect_to_server (
              string str1,
              string str2 )
```

Use to server.

str1	addr serv
str2	port serv

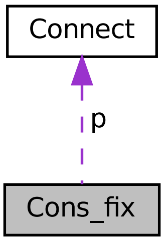
error_proj	if error

:

- Connect.h
- Connect.cpp

4.2 Cons_fix

Cons_fix:



- Connect ∗ **p**

:

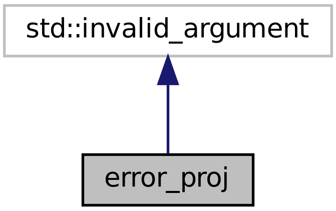
- unit_test.cpp

4.3 error_proj

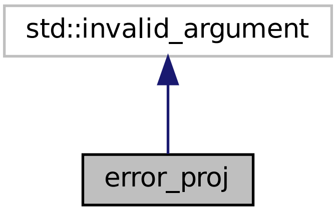
Class error_proj.

`#include <Connect.h>`

:error_proj:



error_proj:



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

4.3.1

Class error_proj.

Output error message

:

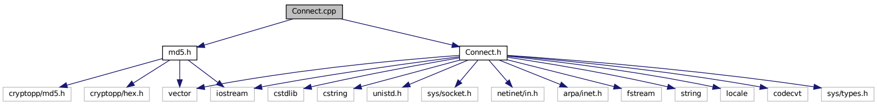
- Connect.h

Chapter 5

5.1 Connect.cpp

```
#include "Connect.h"
#include "md5.h"
```

Connect.cpp:



5.2 Connect.h

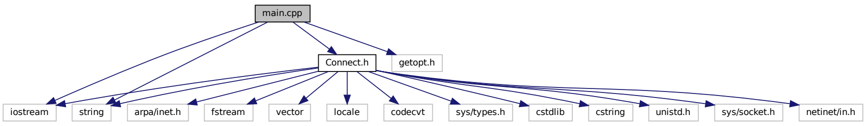
```

#pragma once
#include <sys/types.h>
#include <iostream>
#include <cstdlib>
#include <cstring>
#include <unistd.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <fstream>
#include <vector>

#include <string>
#include <locale>
#include <codecvt>

using namespace std;

class Connect{
    public:
        int ip_addr;
        int port;
        string name_original_file;
        string name_result_file;
        string name_auto_file;
        string username;
        string pswd;
        int Connect_to_server(string str1, string str2);
        void GetLoginPassword();

};


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

```

5.3 main.cpp

Main module.

```
#include <iostream>
#include <string>
#include <getopt.h>
#include "Connect.h"
```

main.cpp:



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

5.3.1

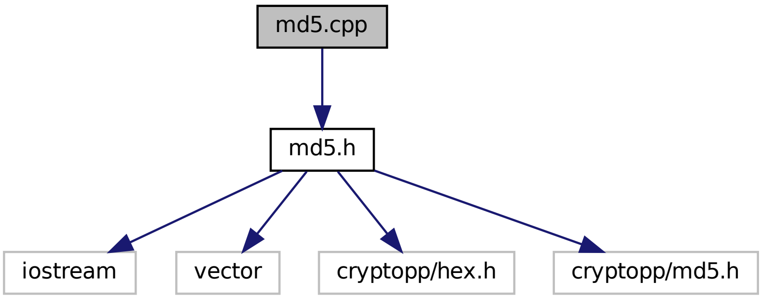
Main module.

rez	perem for work comm str
optarg	perem for work comm str

5.4 md5.cpp

```
#include "md5.h"
```

md5.cpp:



- std::string MD5_hash (std::string msg)

  Getting a hash code using the md5 algorithm.

5.4.1

5.4.1.1 MD5_hash()

```
std::string MD5_hash (
            std::string msg )
```

Getting a hash code using the md5 algorithm.

msg	messages for hashing

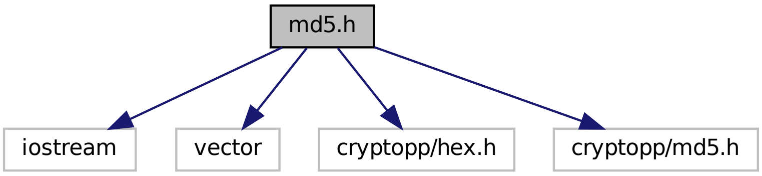
hash code

5.5 md5.h

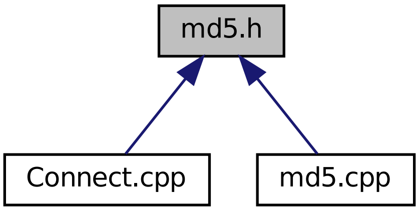
Header file for Connect.

```
#include <iostream>
#include <vector>
#include <cryptopp/hex.h>
#include <cryptopp/md5.h>
```

md5.h:



, :



- #define **CRYPTOPP_ENABLE_NAMESPACE_WEAK** 1

- std::string MD5_hash (std::string msg)

  Getting a hash code using the md5 algorithm.

5.5.1

Header file for Connect.

Header file for module md5.

Kancerov L. E.

1.0

25.11.2023

IBST PGU

5.5.2

5.5.2.1 MD5_hash()

```
std::string MD5_hash (
              std::string msg )
```

Getting a hash code using the md5 algorithm.

msg	messages for hashing

hash code

5.6 md5.h

```
..

#include <iostream>
#include <vector>

#include <cryptopp/hex.h> // HexEncoder

#define CRYPTOPP_ENABLE_NAMESPACE_WEAK 1
#include <cryptopp/md5.h> // MD%
std::string MD5_hash(std::string msg);
```