

TFTPClient

1.0

Создано системой Doxygen 1.8.17

1 Иерархический список классов	1
1 Иерархический список классов	1
1.1 Иерархия классов	1
2 Алфавитный указатель классов	1
2.1 Классы	1
3 Список файлов	2
3.1 Файлы	2
4 Классы	2
4.1 Класс <code>client_error</code>	2
4.2 Структура <code>create_data</code>	3
4.3 Структура <code>create_req</code>	3
4.4 Структура <code>create_resp</code>	4
4.5 Класс <code>TFTPclient</code>	4
4.5.1 Конструктор(ы)	5
4.5.2 Методы	5
5 Файлы	6
5.1 Файл <code>main.cpp</code>	6
5.2 Файл <code>TFTPclient.cpp</code>	7
5.3 Файл <code>TFTPclient.h</code>	7
5.3.1 Подробное описание	8
Предметный указатель	9

1 Иерархический список классов

1.1 Иерархия классов

Иерархия классов.

<code>create_data</code>	3
<code>create_req</code>	3
<code>create_resp</code>	4
<code>invalid_argument</code>	
<code>client_error</code>	2
<code>TFTPclient</code>	4

2 Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

client_error	2
create_data	3
create_req	3
create_resp	4
TFTPclient	4

3 Список файлов

3.1 Файлы

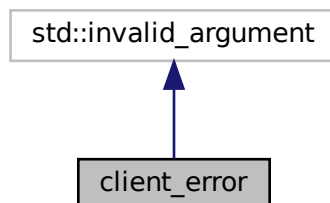
Полный список документированных файлов.

main.cpp	6
TFTPclient.cpp	7
TFTPclient.h	7
Заголовочный файл для модуля TFTPclient	

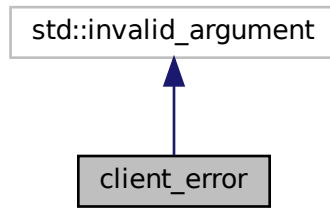
4 Классы

4.1 Класс client_error

Граф наследования:client_error:



Граф связей класса client_error:



Открытые члены

- `client_error (const std::string &what_arg)`
- `client_error (const char *what_arg)`

Объявления и описания членов класса находятся в файле:

- [TFTPclient.h](#)

4.2 Структура create_data

Открытые члены

- `bool read_block (std::ifstream &file)`
- `uint16_t get_block (void)`
- `void set_block (uint16_t block_number)`

Открытые атрибуты

- `uint16_t type = htons(TFTPclient::DATA)`
- `uint16_t block_number`
- `char data [512]`
- `size_t size`

Объявления и описания членов структуры находятся в файле:

- [TFTPclient.cpp](#)

4.3 Структура create_req

Открытые члены

- `create_req (std::string const &name, int type=TFTPclient::RRQ)`

Открытые атрибуты

- `uint16_t type = htons(TFTPclient::RRQ)`
- `char filename [512]`
- `uint16_t size = 2`

Объявления и описания членов структуры находятся в файле:

- [TFTPclient.cpp](#)

4.4 Структура `create_resp`

Открытые члены

- `void set_block (uint16_t val)`
- `uint16_t get_block (void)`

Открытые атрибуты

- `uint16_t type = htons(TFTPclient::ACK)`
- - `union {`
 - `uint16_t block`
 - `uint16_t errcode``};`
- `char err [128]`

Объявления и описания членов структуры находятся в файле:

- [TFTPclient.cpp](#)

4.5 Класс `TFTPclient`

Открытые члены

- [TFTPclient](#) (`std::string address, double rcv_timeout=2.0`)
Конструктор класса [TFTPclient](#).
- [~TFTPclient](#) ()
Деструктор класса [TFTPclient](#).
- `void Upload (std::string const &file_name, std::ifstream &file)`
Загрузка файла на сервер.
- `void Download (std::string const &name, std::ostream &file)`
Скачивание файла с сервера.

Закрытые типы

- `enum {`
 - `RRQ = 1, WRQ, DATA, ACK,`
 - `ERROR }`

Закрытые данные

- struct timeval tm
- int remoteAddress
- int remotePort = 69
- int socket

Друзья

- struct create_req
- struct create_data
- struct create_resp

4.5.1 Конструктор(ы)

4.5.1.1 TFTPclient() TFTPclient::TFTPclient (
std::string address,
double rcv_timeout = 2.0)

Конструктор класса [TFTPclient](#).

Аргументы

address	адрес TFTP сервера в формате 127.0.0.1:69.
rcv_timeout	время ожидания от сервера, по умолчанию 2.0 сек.

Создаёт сетевой сокет и привязывает его к локальному адресу.

Предупреждения

При передаче неверного адреса в верном формате подключение всё равно произойдёт.

Исключения

client_error , если	произошла ошибка.
-------------------------------------	-------------------

4.5.1.2 ~TFTPclient() TFTPclient::~~TFTPclient ()

Деструктор класса [TFTPclient](#).

Закрывает сокет, открытый в конструкторе.

4.5.2 Методы

4.5.2.1 Download() void TFTPclient::Download (
std::string const & file_name,
std::ostream & file)

Скачивание файла с сервера.

Аргументы

file_name	имя файла на сервере, который необходимо скачать.
file	файловый поток для записи принятой информации в файл.

Исключения

client_error ,если	произошла ошибка.
------------------------------------	-------------------

Предупреждения

При попытке скачать не существующий файл, TFTP сервер отправит информацию об ошибке в бинарном виде, она будет принята и записана в указанный файл.

4.5.2.2 Upload() void TFTPclient::Upload (
std::string const & file_name,
std::ifstream & file)

Загрузка файла на сервер.

Аргументы

file_name	имя файла на сервере в который будет записана загружаема информация. Создаётся автоматически.
file	файловый поток для чтения. Из него будет считана информация, загружаемая на сервер.

Исключения

client_error ,если	произошла ошибка.
------------------------------------	-------------------

Объявления и описания членов классов находятся в файлах:

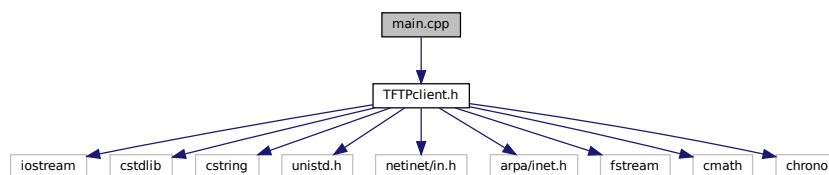
- [TFTPclient.h](#)
- [TFTPclient.cpp](#)

5 Файлы

5.1 Файл main.cpp

```
#include "TFTPclient.h"
```

Граф включаемых заголовочных файлов для main.cpp:



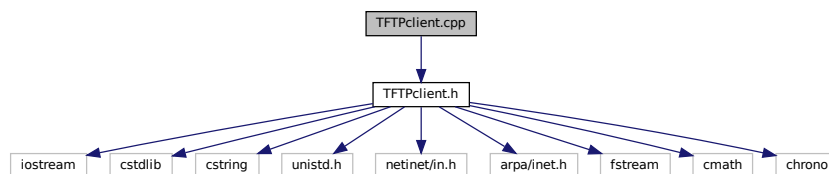
Функции

- int main ()

5.2 Файл TFTPclient.cpp

```
#include "TFTPclient.h"
```

Граф включаемых заголовочных файлов для TFTPclient.cpp:



Классы

- struct [create_req](#)
- struct [create_data](#)
- struct [create_resp](#)

5.3 Файл TFTPclient.h

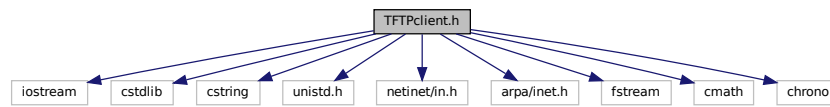
Заголовочный файл для модуля [TFTPclient](#).

```
#include <iostream>
#include <cstdlib>
#include <cstring>
#include <unistd.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <fstream>
#include <cmath>
```

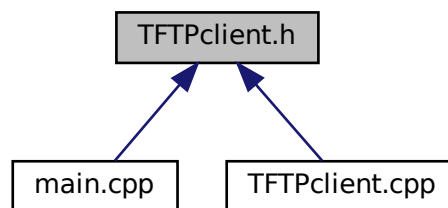


```
#include <chrono>
```

Граф включаемых заголовочных файлов для TFTPclient.h:



Граф файлов, в которые включается этот файл:



Классы

- class [TFTPclient](#)
- class [client_error](#)

Макросы

- `#define ASCII "netascii"`
- `#define INTERFACE`

5.3.1 Подробное описание

Заголовочный файл для модуля [TFTPclient](#).

Автор

Lukonin A. S.

Версия

1.0

Дата

08.06.2021

Предметный указатель

- ~TFTPclient
 - TFTPclient, [5](#)
- client__error, [2](#)
- create_data, [3](#)
- create_req, [3](#)
- create_resp, [4](#)
- Download
 - TFTPclient, [5](#)
- main.cpp, [6](#)
- TFTPclient, [4](#)
 - ~TFTPclient, [5](#)
 - Download, [5](#)
 - TFTPclient, [5](#)
 - Upload, [6](#)
- TFTPclient.cpp, [7](#)
- TFTPclient.h, [7](#)
- Upload
 - TFTPclient, [6](#)