

# DummyFTPServer

Generováno programem Doxygen 1.6.3

Sun Jun 13 17:12:43 2010

## Obsah

<b>1</b>	<b>Projekt FTP server</b>	<b>1</b>
1.1	DummyFTPServer	1
1.1.1	Zadání	2
1.1.2	Autorizace	2
1.1.3	Přenos souboru	2
1.1.4	Nastavení	2
1.1.5	Screenshot	2
<b>2</b>	<b>Rejstřík tříd</b>	<b>2</b>
2.1	Hierarchie tříd	2
<b>3</b>	<b>Rejstřík tříd</b>	<b>3</b>
3.1	Seznam tříd	3
<b>4</b>	<b>Dokumentace tříd</b>	<b>4</b>
4.1	Dokumentace třídy DummyFTPServer::AccountManager::User	4
4.1.1	Detailní popis	4
4.1.2	Dokumentace konstruktoru a destruktoru	4
4.1.3	Dokumentace k metodám	5
4.2	Dokumentace třídy DummyFTPServer::AccountManager::UserList	7
4.2.1	Detailní popis	8
4.2.2	Dokumentace k metodám	8
4.3	Dokumentace třídy DummyFTPServer::Configuration	12
4.3.1	Detailní popis	12
4.3.2	Dokumentace konstruktoru a destruktoru	13
4.3.3	Dokumentace k metodám	13
4.4	Dokumentace třídy DummyFTPServer::ConfSection	15
4.4.1	Detailní popis	15
4.4.2	Dokumentace konstruktoru a destruktoru	15
4.4.3	Dokumentace k metodám	16
4.5	Dokumentace třídy DummyFTPServer::Core	17
4.5.1	Detailní popis	19
4.5.2	Dokumentace k metodám	19
4.6	Dokumentace třídy DummyFTPServer::ErrorException	25
4.6.1	Detailní popis	25
4.6.2	Dokumentace konstruktoru a destruktoru	25

4.6.3	Dokumentace k metodám	25
4.7	Dokumentace třídy DummyFTPServer::FTPCommand	26
4.7.1	Detailní popis	28
4.7.2	Dokumentace konstruktoru a destruktoru	29
4.8	Dokumentace třídy DummyFTPServer::FTPCommandLIST	29
4.8.1	Detailní popis	32
4.8.2	Dokumentace konstruktoru a destruktoru	32
4.9	Dokumentace třídy DummyFTPServer::FTPCommandRETR	33
4.9.1	Detailní popis	35
4.9.2	Dokumentace konstruktoru a destruktoru	35
4.10	Dokumentace třídy DummyFTPServer::FTPCommandSTOR	36
4.10.1	Detailní popis	38
4.10.2	Dokumentace konstruktoru a destruktoru	38
4.11	Dokumentace třídy DummyFTPServer::FTPServer	38
4.11.1	Detailní popis	40
4.11.2	Dokumentace konstruktoru a destruktoru	40
4.11.3	Dokumentace k metodám	40
4.12	Dokumentace třídy DummyFTPServer::ServerThread	41
4.12.1	Detailní popis	44
4.12.2	Dokumentace k metodám	44
4.13	Dokumentace třídy DummyFTPServer::WorkPath	45
4.13.1	Detailní popis	46
4.13.2	Dokumentace konstruktoru a destruktoru	46
4.13.3	Dokumentace k metodám	47

# 1 Projekt FTP server

## Autor

Marek Hakala <marek@hakala.cz>

## Verze

1.0.0-unstable

## Datum

2010

## 1.1 DummyFTPServer

Toto je semestrální práce do předmětu BI-PA2 na FIT ČVUT.

### 1.1.1 Zadání

Program bude sloužit k přenosu souboru prostřednictvím protokolu FTP (File Transfer Protocol). Určitě stojí za to zmínit, že program bude používat síťovou architekturu klient-server. Aplikace po startu provede načtení konfiguračního souboru. Na základě konfigurace začne program poslouchat na daném TCP portu a soketu. Jakmile se připojí k serveru klient bude vytvořeno speciální vlákno v rámci, kterého s ním bude komunikováno. Prvním krokem bude autorizační mechanismus, který provede ověření uživatelského jména a hesla oproti speciálnímu souboru. Po řádné autorizaci bude uživateli zpřístupněn obsah domovského adresáře a bude mít možnost obousměrného přenosu souborů.

Aplikace nebude obsahovat kompletní implementaci FTP protokolu podle normy RFC959 ale pouze nezbytnou část pro úplně základní operace. A to s časových důvodů.

Program bude pracovat v textovém módu. Informace o běhu ftp serveru budou vypisovány na standardním výstupu. Program bude možné vypnout stisknutím klávesy „CTRL + C“.

### 1.1.2 Autorizace

Vzdálený klient bude vyzván k zaslání uživatelského jména a hesla. Údaje budou porovnány s údaji ve speciálním souboru, sloužícímu k autorizaci. Jeden řádek odpovídá jednomu záznamu.

### 1.1.3 Přenos souboru

Uživatel si vyžádá soubor, který mu bude binárně zaslán prostřednictvím tcp socketu. V opačném směru bude klient vyzván k zaslání obsahu souboru prostřednictvím tcp socketu. Obsah bude převzat a uložen do nově vzniklého souboru na straně serveru.

### 1.1.4 Nastavení

Primárně bude využit konfigurační soubor. Ale některé věci bude možné změnit pomocí vstupních argumentů. V konfiguračním souboru uživatel nalezne proměnné port, anonymous a anonymousPath pomocí, kterých budete moct změnit nastavení.

### 1.1.5 Screenshot

#### 1.1.5.1 Technologie

## 2 Rejstřík tříd

### 2.1 Hierarchie tříd

Zde naleznete seznam, vyjadřující vztah dědičnosti tříd. Je seřazen přibližně (ale ne úplně) podle abecedy:

<b>DummyFTPServer::AccountManager::User</b>	<b>4</b>
<b>DummyFTPServer::AccountManager::UserList</b>	<b>7</b>
<b>DummyFTPServer::Configuration</b>	<b>12</b>
<b>DummyFTPServer::ConfSection</b>	<b>15</b>

<b>DummyFTPServer::Core</b>	<b>17</b>
<b>DummyFTPServer::ErrorException</b>	<b>25</b>
<b>DummyFTPServer::FTPCommand</b>	<b>26</b>
<b>DummyFTPServer::FTPCommandLIST</b>	<b>29</b>
<b>DummyFTPServer::FTPCommandRETR</b>	<b>33</b>
<b>DummyFTPServer::FTPCommandSTOR</b>	<b>36</b>
<b>DummyFTPServer::FTPServer</b>	<b>38</b>
<b>DummyFTPServer::ServerThread</b>	<b>41</b>
<b>DummyFTPServer::WorkPath</b>	<b>45</b>

### 3 Rejstřík tříd

#### 3.1 Seznam tříd

Následující seznam obsahuje především identifikace tříd, ale nacházejí se zde i další netriviální prvky, jako jsou struktury (struct), unie (union) a rozhraní (interface). V seznamu jsou uvedeny jejich stručné popisy:

<b>DummyFTPServer::AccountManager::User</b> (Třída reprezentující uživatele )	<b>4</b>
<b>DummyFTPServer::AccountManager::UserList</b> (Třída sloužící jako databáze uživatel v paměti )	<b>7</b>
<b>DummyFTPServer::Configuration</b> (Třída sloužící jako databáze nastavení v paměti )	<b>12</b>
<b>DummyFTPServer::ConfSection</b> (Objekt reprezentující jednu sekci databáze nastavení )	<b>15</b>
<b>DummyFTPServer::Core</b> (Tato třída je jádrem celé aplikace. Zajišťuje spolupráci mezi nastavením, uživatelskou databází a serverovou částí )	<b>17</b>
<b>DummyFTPServer::ErrorException</b> (Třída slouží k vyhazování chybových hlášení )	<b>25</b>
<b>DummyFTPServer::FTPCommand</b> (Zajišťuje obsluhu FTP příkazu (Abstraktní třída) )	<b>26</b>
<b>DummyFTPServer::FTPCommandLIST</b> (Zajišťuje obsluhu FTP příkazu LIST )	<b>29</b>
<b>DummyFTPServer::FTPCommandRETR</b> (Zajišťuje obsluhu FTP příkazu RETR )	<b>33</b>
<b>DummyFTPServer::FTPCommandSTOR</b> (Zajišťuje obsluhu FTP příkazu STOR )	<b>36</b>
<b>DummyFTPServer::FTPServer</b> (Třída reprezentuje FTP server )	<b>38</b>
<b>DummyFTPServer::ServerThread</b> (Objekt reprezentující serverové vlákno )	<b>41</b>
<b>DummyFTPServer::WorkPath</b> (Třída provádí veškerou práci, ze soubory a adresáři od načítání obsahu po mazání atd.. )	<b>45</b>

## 4 Dokumentace tříd

### 4.1 Dokumentace třídy DummyFTPServer::AccountManager::User

Třída reprezentující uživatele.

```
#include <user.h>
```

#### Veřejné metody

- `User` (std::string username, std::string password, PasswordType format, std::string homeDirectory)  
*Vytváří objekty reprezentující uživatele.*
- std::string `getUsername` () const  
*Metoda vrací uživatelské jméno.*
- std::string `getHomeDirectory` () const  
*Metoda vrací cestu k domovskému adresáři.*
- bool `checkPassword` (std::string password, PasswordType format) const  
*Metoda provádí porovnání hesel.*
- bool `isValidPath` () const  
*Třídní metoda provádějící kontrolu existence cesty.*
- bool `operator==` (const `User` &u) const  
*Přetížený operátor provnání. Slouží k porovnání hesel dvou uživatelských objektů.*
- bool `operator!=` (const `User` &u) const  
*Přetížený operátor porovnání. Slouží k porovnání hesel dvou uživatelských objektů.*

#### Statické veřejné metody

- static bool `isValidPath` (const QString &path)  
*Netřídní metoda provádějící kontrolu existence cesty.*

#### 4.1.1 Detailní popis

Třída reprezentující uživatele.

Definice je uvedena na řádce 43 v souboru `user.h`.

#### 4.1.2 Dokumentace konstruktoru a destruktoru

##### 4.1.2.1 DummyFTPServer::AccountManager::User::User (std::string username, std::string password, PasswordType format, std::string homeDirectory)

Vytváří objekty reprezentující uživatele.

### Parametry

*username* Uživatelské jméno  
*password* Heslo  
*format* Formát hesla  
*homeDirectory* Domovský adresář

Definice je uvedena na řádce 29 v souboru [user.cpp](#).

### 4.1.3 Dokumentace k metodám

#### 4.1.3.1 bool DummyFTPServer::AccountManager::User::checkPassword (std::string *password*, PasswordType *format*) const

Metoda provádí porovnání hesel.

### Parametry

*password* Vstupní heslo.  
*format* Formát vstupního hesla.

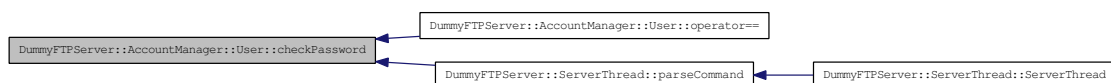
### Návratová hodnota

Vrací pravdu při shodě hesel, v opačném případě nepravdu.

Definice je uvedena na řádce 47 v souboru [user.cpp](#).

Používá se v [operator==\( \)](#) a [DummyFTPServer::ServerThread::parseCommand\(\)](#).

Tuto funkci volají...



#### 4.1.3.2 bool DummyFTPServer::AccountManager::User::isValidPath (const QString & *path*) [static]

Netřídní metoda provádějící kontrolu existence cesty.

### Parametry

*path* Nastaví cestu.

Definice je uvedena na řádce 82 v souboru [user.cpp](#).

#### 4.1.3.3 bool DummyFTPServer::AccountManager::User::isValidPath () const

Třídni metoda provádějící kontrolu existence cesty.

##### Návratová hodnota

Vrací pravdu při úspěšné validaci cesty, v opačném případě nepravdu.

Definice je uvedena na řádce 77 v souboru [user.cpp](#).

Odkazuje se na [getHomeDirectory\(\)](#).

Používá se v [DummyFTPServer::AccountManager::UserList::parseUser\(\)](#).

Tato funkce volá...



Tuto funkci volají...



#### 4.1.3.4 bool DummyFTPServer::AccountManager::User::operator!= (const User & u) const

Přetížený operátor porovnání. Slouží k porovnání hesel dvou uživatelských objektů.

##### Parametry

*u* Vstupní uživatelský objekt.

##### Návratová hodnota

Vrací nepravdu při shodě hesel, v opačném případě pravdu.

Definice je uvedena na řádce 94 v souboru [user.cpp](#).

#### 4.1.3.5 bool DummyFTPServer::AccountManager::User::operator== (const User & u) const

Přetížený operátor provnání. Slouží k porovnání hesel dvou uživatelských objektů.

##### Parametry

*u* Vstupní uživatelský objekt.

##### Návratová hodnota

Vrací pravdu při shodě hesel, v opačném případě nepravdu.



Definice je uvedena na řádce 88 v souboru `user.cpp`.

Odkazuje se na `checkPassword()`.

Tato funkce volá...



Dokumentace pro tuto třídu byla generována z následujících souborů:

- `/home/marek/Projekty/DummyFTPServer/source/user.h`
- `/home/marek/Projekty/DummyFTPServer/source/user.cpp`

## 4.2 Dokumentace třídy DummyFTPServer::AccountManager::UserList

Třída sloužící jako databáze uživatel v paměti.

```
#include <userlist.h>
```

### Veřejné metody

- `UserList` (`std::string userListFile`)  
*Vytváří seznam uživatel.*
- `User * getByUsername` (`std::string username`)  
*Slouží k získání ukazatele na uživatele.*
- `bool add` (`User *user`)  
*Slouží k přidávání uživatel do seznamu.*
- `bool remove` (`User *user`)  
*Slouží k odebrání uživatele ze seznamu.*
- `bool loadUserList` ()  
*Slouží k načtení seznamu uživatel, ze souboru.*
- `bool parseUser` (`std::string input`)  
*Slouží k načtení jednoho uživatele.*
- `bool parsePassword` (`std::string input`, `std::string *password`, `PasswordType *type`)  
*Slouží k načtení seznamu uživatel, ze souboru.*
- `bool contains` (`std::string username`)  
*Provádí kontrolu unikátnosti uživatelského jména.*
- `int count` () `const`  
*Slouží k zjištění počtu uživatel.*
- `bool isEmpty` () `const`  
*Kontrola prázdnosti seznamu.*

- bool [clean\(\)](#)  
*Smaže celý seznam uživatel.*

#### 4.2.1 Detailní popis

Třída sloužící jako databáze uživatel v paměti.

Definice je uvedena na řádku [44](#) v souboru [userlist.h](#).

#### 4.2.2 Dokumentace k metodám

##### 4.2.2.1 bool DummyFTPServer::AccountManager::UserList::add (User \* *user*)

Slouží k přidávání uživatel do seznamu.

#### Parametry

*user* Vstupní ukazatel na objekt uživatele.

#### Návratová hodnota

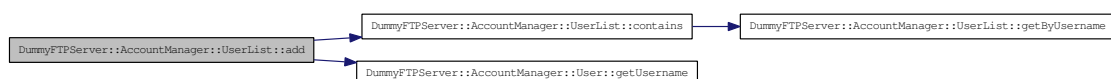
Vrací pravdu při úspěšném přidání, v opačném případě nepravdu.

Definice je uvedena na řádku [50](#) v souboru [userlist.cpp](#).

Odkazuje se na [contains\(\)](#) a [DummyFTPServer::AccountManager::User::getUsername\(\)](#).

Používá se v [parseUser\(\)](#).

Tato funkce volá...



Tuto funkci volají...



##### 4.2.2.2 bool DummyFTPServer::AccountManager::UserList::clean ()

Smaže celý seznam uživatel.

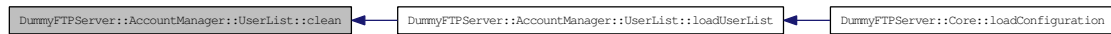
#### Návratová hodnota

Vrací pravdu při úspěšném vyprázdnění, v opačném případě nepravdu.

Definice je uvedena na řádku 181 v souboru `userlist.cpp`.

Používá se v `loadUserList()`.

Tuto funkci volají...



#### 4.2.2.3 int DummyFTPServer::AccountManager::UserList::count () const

Slouží k zjištění počtu uživatel.

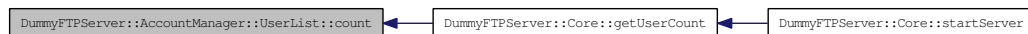
##### Návratová hodnota

Vrací celkový počet uživatel v seznamu.

Definice je uvedena na řádku 173 v souboru `userlist.cpp`.

Používá se v `DummyFTPServer::Core::getUserCount()`.

Tuto funkci volají...



#### 4.2.2.4 User \* DummyFTPServer::AccountManager::UserList::getByUsername (std::string username)

Slouží k získání ukazatele na uživatele.

##### Parametry

**username** Nastaví jméno hledaného uživatele.

##### Návratová hodnota

Vrací ukazatel na uživatele. Pokud uživatel neexistuje bude vrácen NULL pointer.

Definice je uvedena na řádku 40 v souboru `userlist.cpp`.

Používá se v `contains()` a `DummyFTPServer::Core::getUser()`.

Tuto funkci volají...





**Návratová hodnota**

Vrací pravdu při úspěšném načtení, v opačném případě nepravdu.

Definice je uvedena na řádce 138 v souboru `userlist.cpp`.

Používá se v `parseUser()`.

Tuto funkci volají...

**4.2.2.8 bool DummyFTPServer::AccountManager::UserList::parseUser (std::string *input*)**

Slouží k načtení jednoho uživatele.

**Parametry**

*input* Vstupní textový řetězec obsahující jeden uživatelský záznam.

**Návratová hodnota**

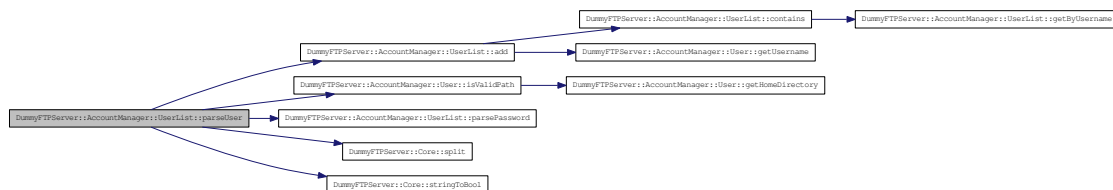
Vrací pravdu při úspěšném načtení, v opačném případě nepravdu.

Definice je uvedena na řádce 94 v souboru `userlist.cpp`.

Odkazuje se na `add()`, `DummyFTPServer::AccountManager::User::isValidPath()`, `parsePassword()`, `DummyFTPServer::Core::split()` a `DummyFTPServer::Core::stringToBool()`.

Používá se v `loadUserList()`.

Tato funkce volá...



Tuto funkci volají...

**4.2.2.9 bool DummyFTPServer::AccountManager::UserList::remove (User \* *user*)**

Slouží k odebrání uživatele ze seznamu.

### Parametry

**user** Vstupní ukazatel na objekt uživatele.

### Návratová hodnota

Vrací pravdu při úspěšném odebrání, v opačném případě nepravdu.

Definice je uvedena na řádce 58 v souboru `userlist.cpp`.

Odkazuje se na `DummyFTPServer::AccountManager::User::getUsername()`.

Tato funkce volá...



Dokumentace pro tuto třídu byla generována z následujících souborů:

- `/home/marek/Projekty/DummyFTPServer/source/userlist.h`
- `/home/marek/Projekty/DummyFTPServer/source/userlist.cpp`

## 4.3 Dokumentace třídy DummyFTPServer::Configuration

Třída sloužící jako databáze nastavení v paměti.

```
#include <configuration.h>
```

### Veřejné metody

- **Configuration** (std::string configFile=DEFAULT\_CONF\_PATH)  
*Uživatelský konstruktor, který provede otevření konfiguračního souboru pro čtení.*
- bool **appendToSection** (std::string key, std::string value, std::string sectionName="global")  
*Metoda sloužící k přidávání hodnot do sekcí. Pokud není vybrána žádná sekce je automaticky zvolena globální sekce "global".*
- bool **clean** ()  
*Provede vyprázdnění databáze.*
- bool **loadConfiguration** ()  
*Provede načtení konfiguračního souboru.*
- **ConfSection** \* **getSection** (std::string name)  
*Slouží k získání ukazatele na sekci.*

### 4.3.1 Detailní popis

Třída sloužící jako databáze nastavení v paměti.

Definice je uvedena na řádce 43 v souboru `configuration.h`.

#### 4.3.2 Dokumentace konstruktoru a destruktoru

##### 4.3.2.1 DummyFTPServer::Configuration::Configuration (std::string configFile = DEFAULT\_CONF\_PATH)

Uživatelský konstruktory, který provede otevření konfiguračního souboru pro čtení.

#### Parametry

*configFile* Nastaví cestu konfiguračního souboru.

Definice je uvedena na řádce 30 v souboru [configuration.cpp](#).

#### 4.3.3 Dokumentace k metodám

##### 4.3.3.1 bool DummyFTPServer::Configuration::appendToSection (std::string key, std::string value, std::string sectionName = "global")

Metoda sloužící k přidávání hodnot do sekcí. Pokud není vybrána žádná sekce je automaticky zvolena globální sekce "global".

#### Parametry

*key* Nastaví klíč pomocí, kterého bude hodnota přístupná.

*value* Nastaví hodnotu.

*sectionName* Vybere příslušnou sekci.

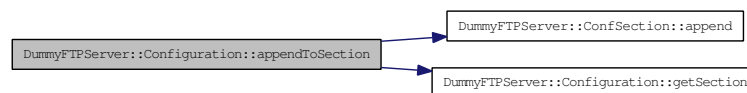
#### Návratová hodnota

Vrací pravdu při úspěšném vložení, v opačném případě nepravdu.

Definice je uvedena na řádce 39 v souboru [configuration.cpp](#).

Odkazuje se na [DummyFTPServer::ConfSection::append\(\)](#) a [getSection\(\)](#).

Tato funkce volá...



##### 4.3.3.2 bool DummyFTPServer::Configuration::clean ()

Provede vyprázdnění databáze.

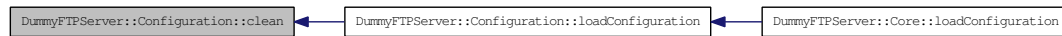
#### Návratová hodnota

Vrací pravdu při úspěšném vyprázdnění, v opačném případě nepravdu.

Definice je uvedena na řádku 141 v souboru `configuration.cpp`.

Používá se v `loadConfiguration()`.

Tuto funkci volají...



#### 4.3.3.3 ConfSection \* DummyFTPServer::Configuration::getSection (std::string name)

Slouží k získání ukazatele na sekci.

##### Parametry

**name** Nastaví jméno hledané sekce.

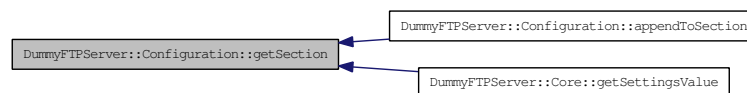
##### Návratová hodnota

Vrací ukazatel na sekci. Pokud sekce neexistuje bude vrácen NULL pointer.

Definice je uvedena na řádku 82 v souboru `configuration.cpp`.

Používá se v `appendToSection()` a `DummyFTPServer::Core::getSettingsValue()`.

Tuto funkci volají...



#### 4.3.3.4 bool DummyFTPServer::Configuration::loadConfiguration ()

Provede načtení konfiguračního souboru.

##### Návratová hodnota

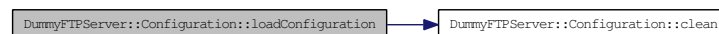
Vrací pravdu při úspěšném načtení, v opačném případě nepravdu.

Definice je uvedena na řádku 54 v souboru `configuration.cpp`.

Odkazuje se na `clean()`.

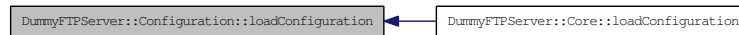
Používá se v `DummyFTPServer::Core::loadConfiguration()`.

Tato funkce volá...





Tuto funkci volají...



Dokumentace pro tuto třídu byla generována z následujících souborů:

- /home/marek/Projekty/DummyFTPServer/source/configuration.h
- /home/marek/Projekty/DummyFTPServer/source/configuration.cpp

## 4.4 Dokumentace třídy DummyFTPServer::ConfSection

Objekt reprezentující jednu sekci databáze nastavení.

```
#include <confsection.h>
```

### Veřejné metody

- [ConfSection](#) (std::string name)  
*Vytváří objekty, které reprezentují sekce.*
- std::string [getName](#) () const  
*Slouží k získání jména sekce.*
- int [getCount](#) () const  
*Slouží k získání počtu položek.*
- std::string [getValue](#) (std::string key)  
*Slouží k získání hodnoty.*
- bool [append](#) (std::string key, std::string value)  
*Slouží k přidání položky.*
- bool [remove](#) (std::string key)  
*Slouží k smazání položky podle klíče.*

### 4.4.1 Detailní popis

Objekt reprezentující jednu sekci databáze nastavení.

Definice je uvedena na řádku 36 v souboru [confsection.h](#).

### 4.4.2 Dokumentace konstruktoru a destruktoru

#### 4.4.2.1 DummyFTPServer::ConfSection::ConfSection (std::string name)

Vytváří objekty, které reprezentují sekce.

### Parametry

**name** Nastavuje název sekce.

Definice je uvedena na řádku 28 v souboru [confsection.cpp](#).

### 4.4.3 Dokumentace k metodám

#### 4.4.3.1 bool DummyFTPServer::ConfSection::append (std::string key, std::string value)

Slouží k přidání položky.

### Parametry

**key** Nastaví klíč.

**value** Nastaví hodnotu.

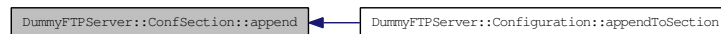
### Návratová hodnota

Vrací pravdu při úspěšném přidání, v opačném případě nepravdu.

Definice je uvedena na řádku 51 v souboru [confsection.cpp](#).

Používá se v [DummyFTPServer::Configuration::appendToSection\(\)](#).

Tuto funkci volají...



#### 4.4.3.2 int DummyFTPServer::ConfSection::getCount () const

Slouží k získání počtu položek.

### Návratová hodnota

Vrací počet položek v sekci.

Definice je uvedena na řádku 36 v souboru [confsection.cpp](#).

#### 4.4.3.3 std::string DummyFTPServer::ConfSection::getName () const

Slouží k získání jména sekce.

### Návratová hodnota

Vrací název sekce.

Definice je uvedena na řádku 32 v souboru [confsection.cpp](#).

#### 4.4.3.4 std::string DummyFTPServer::ConfSection::getValue (std::string *key*)

Slouží k získání hodnoty.

##### Návratová hodnota

Vrací hodnotu podle hledaného klíče.

Definice je uvedena na řádku 40 v souboru [confsection.cpp](#).

Používá se v [DummyFTPServer::Core::getSettingsValue\(\)](#).

Tuto funkci volají...



#### 4.4.3.5 bool DummyFTPServer::ConfSection::remove (std::string *key*)

Slouží k smazání položky podle klíče.

##### Parametry

*key* Nastaví klíč.

##### Návratová hodnota

Vrací pravdu při úspěšném smazání, v opačném případě nepravdu.

Definice je uvedena na řádku 57 v souboru [confsection.cpp](#).

Dokumentace pro tuto třídu byla generována z následujících souborů:

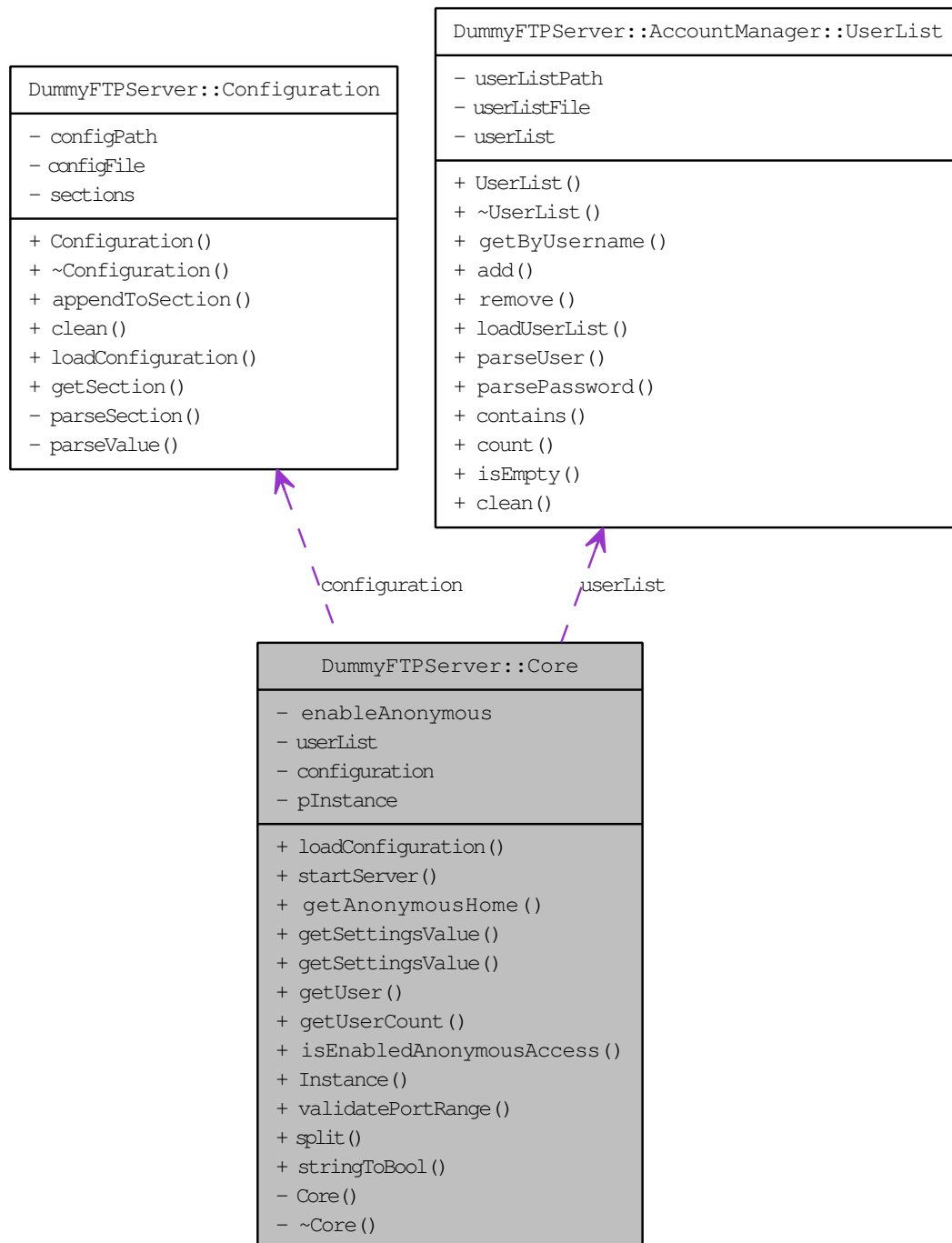
- /home/marek/Projekty/DummyFTPServer/source/confsection.h
- /home/marek/Projekty/DummyFTPServer/source/confsection.cpp

## 4.5 Dokumentace třídy DummyFTPServer::Core

Tato třída je jádrem celé aplikace. Zajišťuje spolupráci mezi nastavením, uživatelskou databází a serverovou částí.

```
#include <core.h>
```

Diagram tříd pro DummyFTPServer::Core:



### Veřejné metody

- bool **loadConfiguration** (std::string configurationPath, std::string userListPath)

*Metoda sloužící k načítání konfigurace a uživatelské databáze.*

- bool `startServer ()`  
*Metoda sloužící k zapnutí serverové části do listen režimu.*
- QString `getAnonymousHome ()`  
*Metoda slouží k získání domovského adresáře uživatele anonymous.*
- bool `getSettingsValue (std::string sectionAndKey, std::string *value)`  
*Metoda vrací hodnotu z nastavení podle zadaného klíče.*
- bool `getSettingsValue (std::string section, std::string key, std::string *value)`  
*Metoda vrací hodnotu z nastavení podle zadaného klíče.*
- User \* `getUser (std::string username)`  
*Metoda vrací ukazatel na uživatele podle zadaného jména.*
- int `getUserCount () const`  
*Metoda pro získání počtu uživatelů v databázi.*
- bool `isEnabledAnonymousAccess ()`  
*Metoda sloužící k získání příznaku, který indikuje povolení anonymního přístupu.*

### Statické veřejné metody

- static Core \* `Instance ()`  
*Metoda pro vytvoření/získání instance třídy Core. Dle návrhového vzoru Singleton.*
- static bool `validatePortRange (unsigned int port)`  
*Metoda provádějící validaci TCP portu.*
- static bool `split (std::stringstream &ss, std::string *output, char delimiter=DEFAULT_DELIMITER)`  
*Netřídní metoda sloužící k rozdělení textového řetězce podle zvoleného oddělovače.*
- static bool `stringToBool (std::string str)`  
*Netřídní metoda sloužící k převodu textového řetězce na boolovskou hodnotu podle zvoleného oddělovače.*

#### 4.5.1 Detailní popis

Tato třída je jádrem celé aplikace. Zajišťuje spolupráci mezi nastavením, uživatelskou databází a serverovou částí.

Definice je uvedena na řádce 48 v souboru `core.h`.

#### 4.5.2 Dokumentace k metodám

##### 4.5.2.1 QString DummyFTPServer::Core::getAnonymousHome ()

Metoda slouží k získání domovského adresáře uživatele anonymous.

#### Návratová hodnota

Vrací cestu k domovskému adresáři uživatele anonymous.

Definice je uvedena na řádce 108 v souboru `core.cpp`.

Odkazuje se na `getSettingsValue()`.

Tato funkce volá...



#### 4.5.2.2 bool DummyFTPServer::Core::getSettingsValue (std::string section, std::string key, std::string \* value)

Metoda vrací hodnotu z nastavení podle zadaného klíče.

#### Parametry

**section** Nastaví sekci.

**key** Nastaví klíč.

**value** Vstupní ukazatel pomocí, kterého nám bude navracena získaná hodnota.

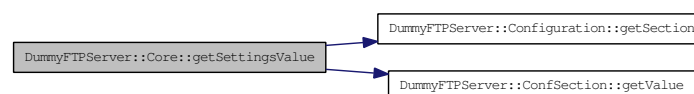
#### Návratová hodnota

Vrací pravdu při nalezení hodnoty, v opačném případě nepravdu.

Definice je uvedena na řádce 91 v souboru `core.cpp`.

Odkazuje se na `DummyFTPServer::Configuration::getSection()` a `DummyFTPServer::ConfSection::getValue()`.

Tato funkce volá...



#### 4.5.2.3 bool DummyFTPServer::Core::getSettingsValue (std::string sectionAndKey, std::string \* value)

Metoda vrací hodnotu z nastavení podle zadaného klíče.

#### Parametry

**sectionAndKey** Nastaví cestu k požadované hodnotě.

*value* Vstupní ukazatel pomocí, kterého nám bude navržena získaná hodnota.

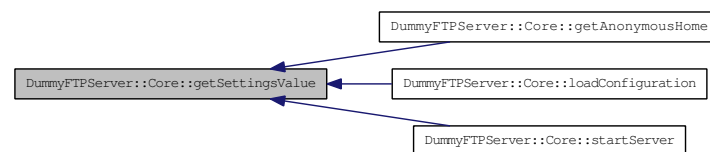
#### Návratová hodnota

Vrací pravdu při nalezení hodnoty, v opačném případě nepravdu.

Definice je uvedena na řádce 74 v souboru `core.cpp`.

Používá se v `getAnonymousHome()`, `loadConfiguration()` a `startServer()`.

Tuto funkci volají...



#### 4.5.2.4 User \* DummyFTPServer::Core::getUser (std::string username)

Metoda vrací ukazatel na uživatele podle zadaného jména.

#### Parametry

*username* Jméno hledaného uživatele.

#### Návratová hodnota

Vrací ukazatel na objekt uživatele, v případě neexistujícího uživatele vrátí null pointer.

Definice je uvedena na řádce 116 v souboru `core.cpp`.

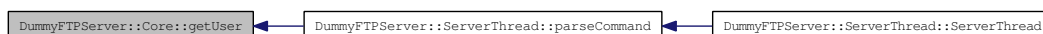
Odkazuje se na `DummyFTPServer::AccountManager::UserList::getByUsername()`.

Používá se v `DummyFTPServer::ServerThread::parseCommand()`.

Tato funkce volá...



Tuto funkci volají...



#### 4.5.2.5 int DummyFTPServer::Core::getUserCount () const

Metoda pro získání počtu uživatelů v databázi.

**Návratová hodnota**

Vrací počet uživatelů, v uživatelské databázi.

Definice je uvedena na řádce 120 v souboru `core.cpp`.

Odkazuje se na `DummyFTPServer::AccountManager::UserList::count()`.

Používá se v `startServer()`.

Tato funkce volá...



Tuto funkci volají...

**4.5.2.6 Core \* DummyFTPServer::Core::Instance () [inline, static]**

Metoda pro vytvoření/získání instance třídy `Core`. Dle návrhového vzoru Singleton.

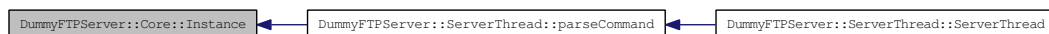
**Návratová hodnota**

Vrací instanci třídy `Core`.

Definice je uvedena na řádce 113 v souboru `core.h`.

Používá se v `DummyFTPServer::ServerThread::parseCommand()`.

Tuto funkci volají...

**4.5.2.7 bool DummyFTPServer::Core::loadConfiguration (std::string configurationPath, std::string userListPath)**

Metoda sloužící k načítání konfigurace a uživatelské databáze.

**Parametry**

*configurationPath* Nastaví cestu ke konfiguračnímu souboru.

*userListPath* Nastaví cestu.

**Návratová hodnota**

Vrací pravdu při úspěšném načtení, v opačném případě nepravdu.



Definice je uvedena na řádku 34 v souboru `core.cpp`.

Odkazuje se na `getSettingsValue()`, `DummyFTPServer::Configuration::loadConfiguration()`, `DummyFTPServer::AccountManager::UserList::loadUserList()` a `stringToBool()`.

Tato funkce volá...



#### 4.5.2.8 bool DummyFTPServer::Core::split(std::stringstream & ss, std::string \* output, char delimiter = DEFAULT\_DELIMITER) [static]

Netřídní metoda sloužící k rozdělení textového řetězce podle zvoleného oddělovače.

##### Parametry

*ss* Vstupní textový proud.

*output* Výstupní textový řetězec.

*delimiter* Nastaví oddělovač.

##### Návratová hodnota

Vrací pravdu při úspěšném rozdělení, v opačném případě nepravdu.

Definice je uvedena na řádku 132 v souboru `core.cpp`.

Používá se v `DummyFTPServer::AccountManager::UserList::parseUser()`.

Tuto funkci volají...



#### 4.5.2.9 bool DummyFTPServer::Core::startServer ()

Metoda sloužící k zapnutí serverové části do listen režimu.

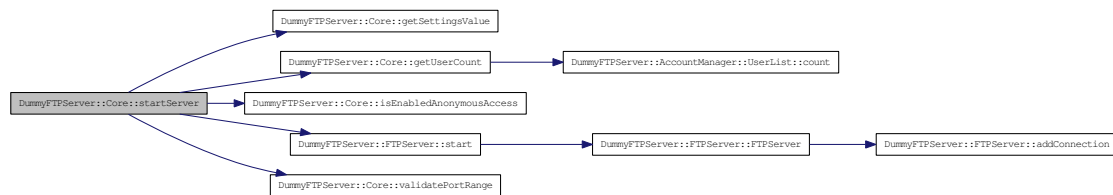
##### Návratová hodnota

Vrací pravdu při úspěšném startu v opačném případě nepravdu.

Definice je uvedena na řádku 58 v souboru `core.cpp`.

Odkazuje se na `getSettingsValue()`, `getUserCount()`, `isEnabledAnonymousAccess()`, `DummyFTPServer::FTPServer::start()` a `validatePortRange()`.

Tato funkce volá...



#### 4.5.2.10 DummyFTPServer::Core::stringToBool (std::string *str*) [static]

Netřídní metoda sloužící k převodu textového řetězce na boolovskou hodnotu podle zvoleného oddělovače.

##### Parametry

***str*** Vstupní textový řetězec.

##### Návratová hodnota

Vrací pravdu při úspěšném převodu textu na boolovskou hodnotu, v opačném případě nepravdu.

Definice je uvedena na řádku 143 v souboru `core.cpp`.

Používá se v `loadConfiguration()` a `DummyFTPServer::AccountManager::UserList::parseUser()`.

Tuto funkci volají...



#### 4.5.2.11 bool DummyFTPServer::Core::validatePortRange (unsigned int *port*) [static]

Metoda provádějící validaci TCP portu.

##### Parametry

***port*** Nastaví číslo kontrolovaného portu.

##### Návratová hodnota

Vrací pravdu při správně nastaveném portu v opačném případě nepravdu.

Definice je uvedena na řádku 124 v souboru `core.cpp`.

Používá se v `startServer()`.

Tuto funkci volají...



Dokumentace pro tuto třídu byla generována z následujících souborů:

- /home/marek/Projekty/DummyFTPServer/source/core.h
- /home/marek/Projekty/DummyFTPServer/source/core.cpp

## 4.6 Dokumentace třídy DummyFTPServer::ErrorException

Třída slouží k vyhazování chybových hlášení.

```
#include <errorexception.h>
```

### Veřejné metody

- **ErrorException** (const std::string &errorMessage, bool critical=false) throw ()  
*Vytváří chybovou zprávu.*
- virtual const char \* **what** () const throw ()  
*Přetížená metoda **what()**, která vrací chybové hlášení.*
- virtual bool **isCritical** () const  
*Příznak kritické chyby.*

### 4.6.1 Detailní popis

Třída slouží k vyhazování chybových hlášení.

Definice je uvedena na řádku 35 v souboru [errorexception.h](#).

### 4.6.2 Dokumentace konstruktoru a destruktoru

#### 4.6.2.1 DummyFTPServer::ErrorException::ErrorException (const std::string & *errorMessage*, bool *critical* = false) throw () [inline]

Vytváří chybovou zprávu.

#### Parametry

**errorMessage** Chybová zpráva.

**critical** Příznak kritické chyby.

Definice je uvedena na řádku 43 v souboru [errorexception.h](#).

### 4.6.3 Dokumentace k metodám

#### 4.6.3.1 bool DummyFTPServer::ErrorException::isCritical () const [inline, virtual]

Příznak kritické chyby.

**Návratová hodnota**

Vrací příznak kritické chyby.

Definice je uvedena na řádku 66 v souboru [errorexception.h](#).

**4.6.3.2 const char \* DummyFTPServer::ErrorException::what () const throw () [inline, virtual]**

Přetížená metoda [what\(\)](#), která vrací chybové hlášení.

**Návratová hodnota**

Vrací pole znaků, které obsahuje chybové hlášení.

Definice je uvedena na řádku 58 v souboru [errorexception.h](#).

Dokumentace pro tuto třídu byla generována z následujícího souboru:

- /home/marek/Projekty/DummyFTPServer/source/errorexception.h

**4.7 Dokumentace třídy DummyFTPServer::FTPCommand**

Zajišťuje obsluhu FTP příkazu (Abstraktní třída).

```
#include <ftpcommand.h>
```

Diagram dědičnosti pro třídu DummyFTPServer::FTPCommand

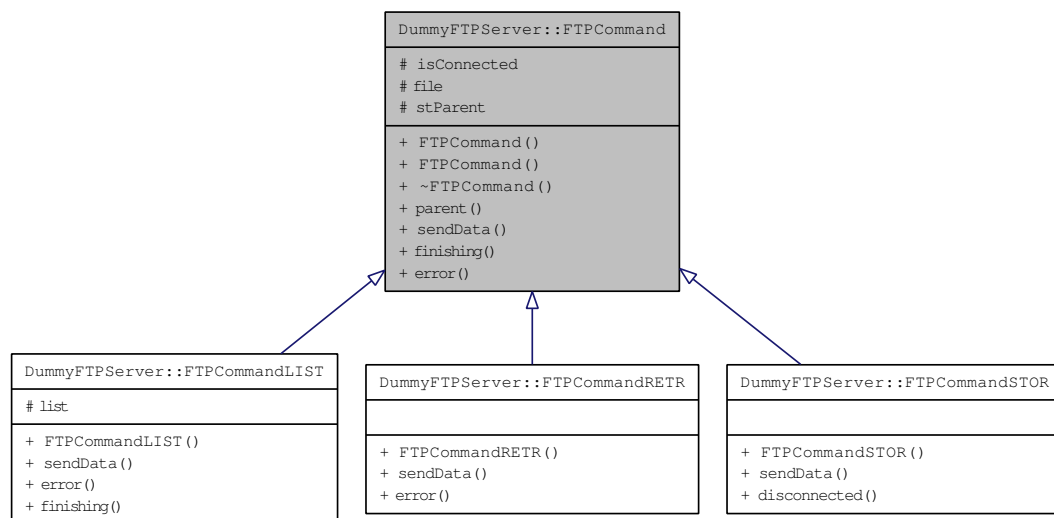
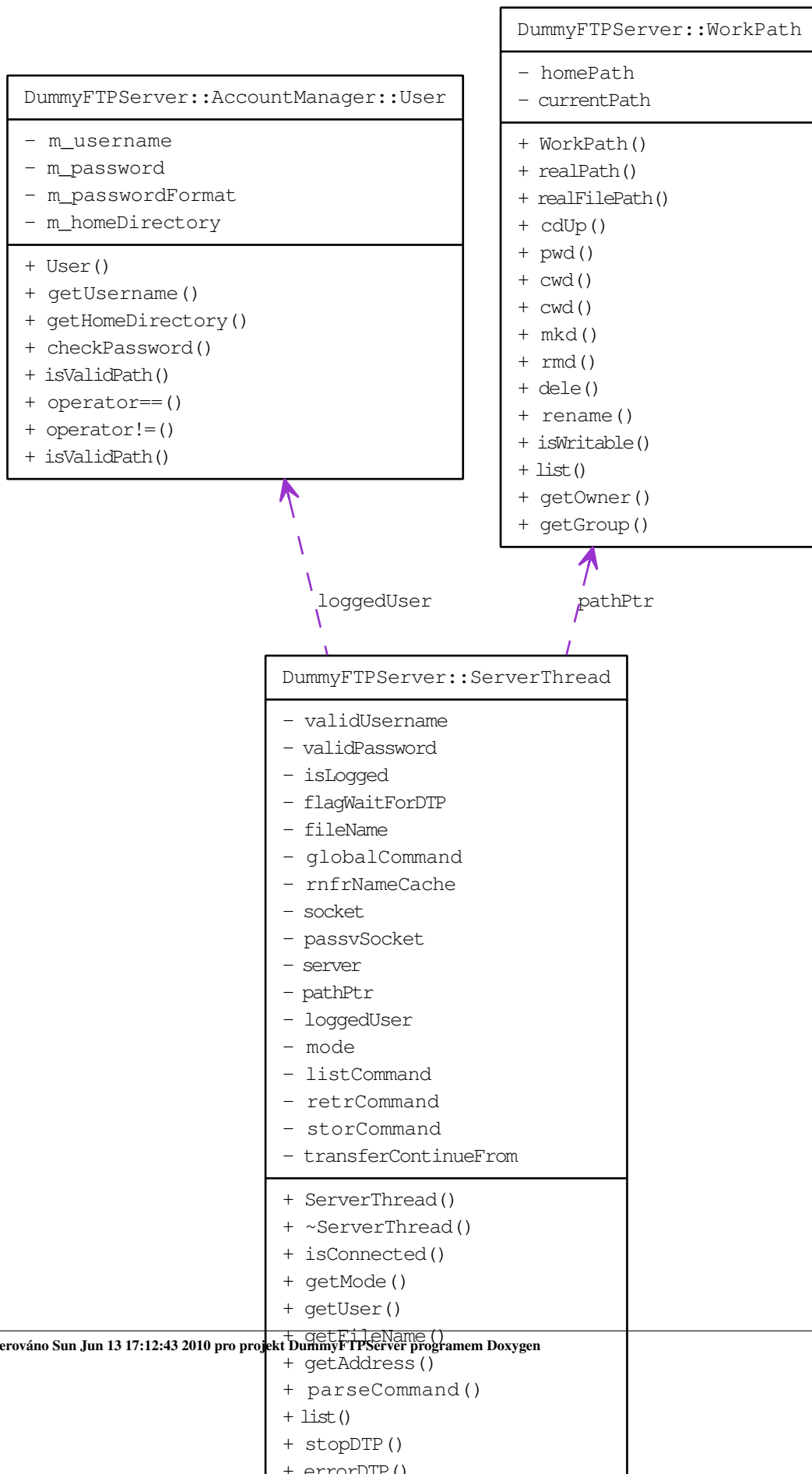


Diagram tříd pro DummyFTPServer::FTPCommand:



### Veřejné sloty

- virtual void `sendData` ()=0  
*Slot, který při zavolání naplní socket daty.*
- virtual void `finishing` ()  
*Dokončuje přenos a uzavírá spojení.*
- virtual void `error` (bool stop=true)  
*Slot, který obsluhuje chybové stavy.*

### Signály

- void `finished` ()  
*Signalizuje dokončení.*

### Veřejné metody

- `FTPCommand` (`ServerThread` \*parent=0)  
*Vytváří objekt pro zpracování FTP příkazu.*
- `FTPCommand` (const `QString` &fileName, `ServerThread` \*parent=0)  
*Vytváří objekt pro zpracování FTP příkazu.*
- virtual `ServerThread` \* `parent` () const  
*Vrací ukazatel na serverové vlákno neboli na otcovský objekt.*

### Chráněné atributy

- bool `isConnected`  
*Příznak aktivního spojení.*
- `QFile` `file`  
*Pracovní soubor.*
- `ServerThread` \* `stParent`  
*Ukazatel na předka.*

#### 4.7.1 Detailní popis

Zajišťuje obsluhu FTP příkazu (Abstraktní třída).

Definice je uvedena na řádce 41 v souboru `ftpcommand.h`.

#### 4.7.2 Dokumentace konstruktoru a destruktoru

##### 4.7.2.1 DummyFTPServer::FTPCommand::FTPCommand (ServerThread \* *parent* = 0)

Vytváří objekt pro zpracování FTP příkazu.

##### Parametry

*parent* Ukazatel na předka.

Definice je uvedena na řádce 30 v souboru [ftpcommand.cpp](#).

Odkazuje se na [isConnected](#) a [stParent](#).

##### 4.7.2.2 DummyFTPServer::FTPCommand::FTPCommand (const QString & *fileName*, ServerThread \* *parent* = 0)

Vytváří objekt pro zpracování FTP příkazu.

##### Parametry

*fileName* Název pracovního souboru.

*parent* Ukazatel na předka.

Definice je uvedena na řádce 35 v souboru [ftpcommand.cpp](#).

Odkazuje se na [isConnected](#) a [stParent](#).

Dokumentace pro tuto třídu byla generována z následujících souborů:

- /home/marek/Projekty/DummyFTPServer/source/ftpcommand.h
- /home/marek/Projekty/DummyFTPServer/source/ftpcommand.cpp
- /home/marek/Projekty/DummyFTPServer/source/moc\_ftpcommand.cpp

## 4.8 Dokumentace třídy DummyFTPServer::FTPCommandLIST

Zajišťuje obsluhu FTP příkazu LIST.

```
#include <ftpcommandlist.h>
```

Diagram dědičnosti pro třídu DummyFTPServer::FTPCommandLIST

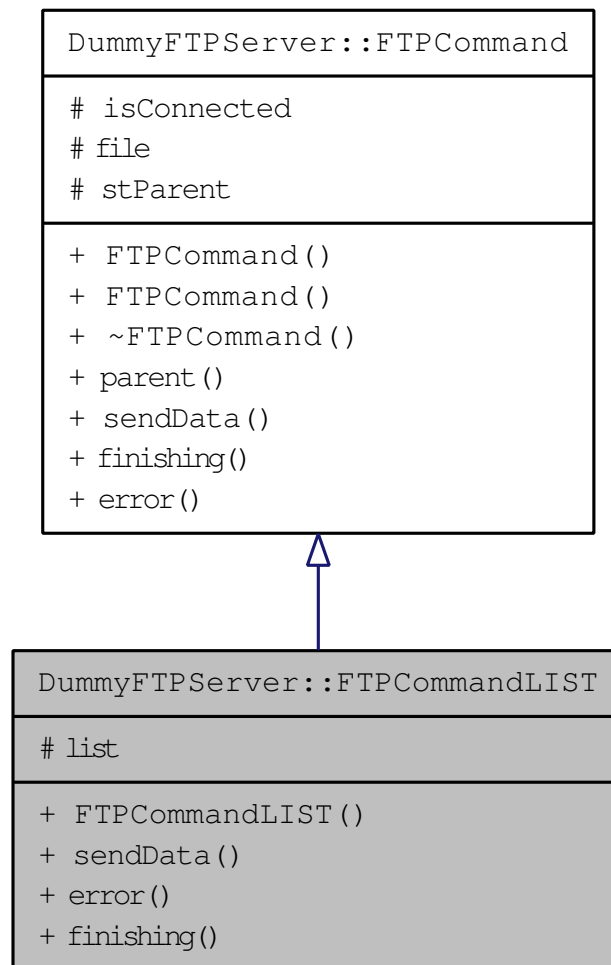
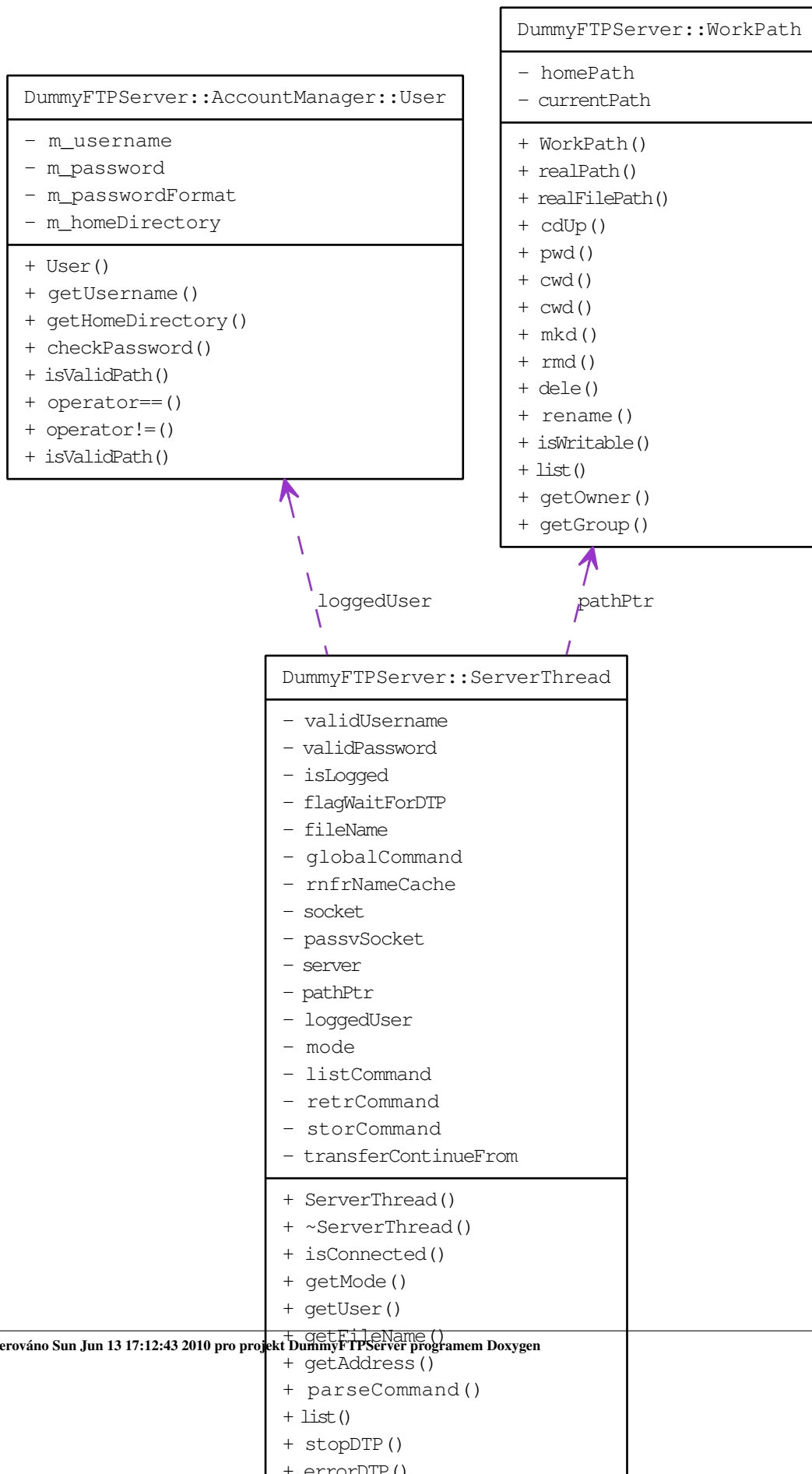




Diagram tříd pro DummyFTPServer::FTPCommandLIST:



### Veřejné sloty

- void [sendData](#) ()  
*Slot, který při zavolání naplní socket daty.*
- void [error](#) (bool stop=true)  
*Slot, který obsluhuje chybové stavy.*
- void [finishing](#) ()  
*Dokončuje přenos a uzavírá spojení.*

### Veřejné metody

- [FTPCommandLIST](#) ([ServerThread](#) \*parent)  
*Vytváří objekt pro zpracování FTP příkaz LIST.*

### Chráněné atributy

- [QList< QByteArray >](#) [list](#)  
*Seznam položek v aktuálním adresáři.*

#### 4.8.1 Detailní popis

Zajišťuje obsluhu FTP příkazu LIST.

Definice je uvedena na řádce 35 v souboru [ftpcommandlist.h](#).

#### 4.8.2 Dokumentace konstruktoru a destruktoru

##### 4.8.2.1 DummyFTPServer::FTPCommandLIST::FTPCommandLIST (ServerThread \* parent)

Vytváří objekt pro zpracování FTP příkaz LIST.

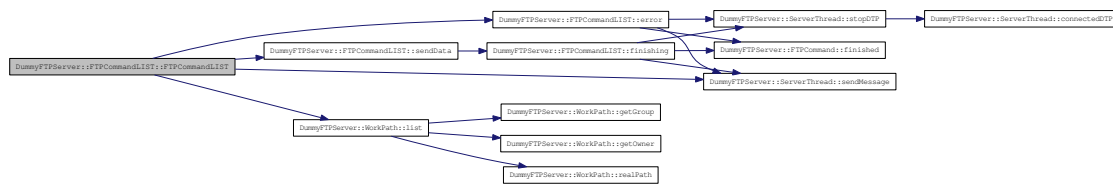
### Parametry

*parent* Ukazatel na předka.

Definice je uvedena na řádce 27 v souboru [ftpcommandlist.cpp](#).

Odkazuje se na [error\(\)](#), [DummyFTPServer::FTPCommand::isConnected](#), [list](#), [DummyFTPServer::WorkPath::list\(\)](#), [sendData\(\)](#) a [DummyFTPServer::ServerThread::sendMessage\(\)](#).

Tato funkce volá...



Dokumentace pro tuto třídu byla generována z následujících souborů:

- /home/marek/Projekty/DummyFTPServer/source/ftpcommandlist.h
- /home/marek/Projekty/DummyFTPServer/source/ftpcommandlist.cpp

## 4.9 Dokumentace třídy DummyFTPServer::FTPCommandRETR

Zajišťuje obsluhu FTP příkazu RETR.

```
#include <ftpcommandretr.h>
```

Diagram dědičnosti pro třídu DummyFTPServer::FTPCommandRETR

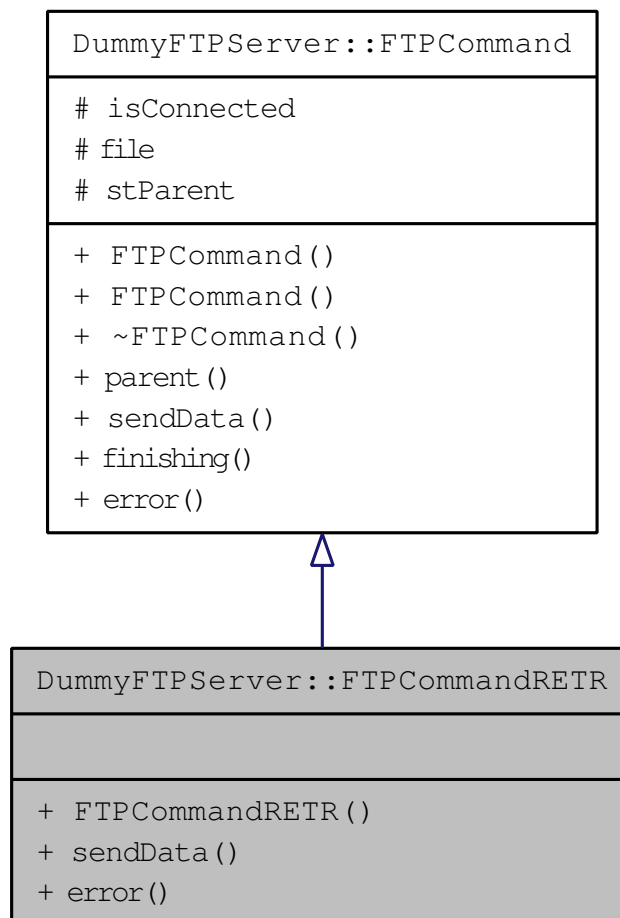
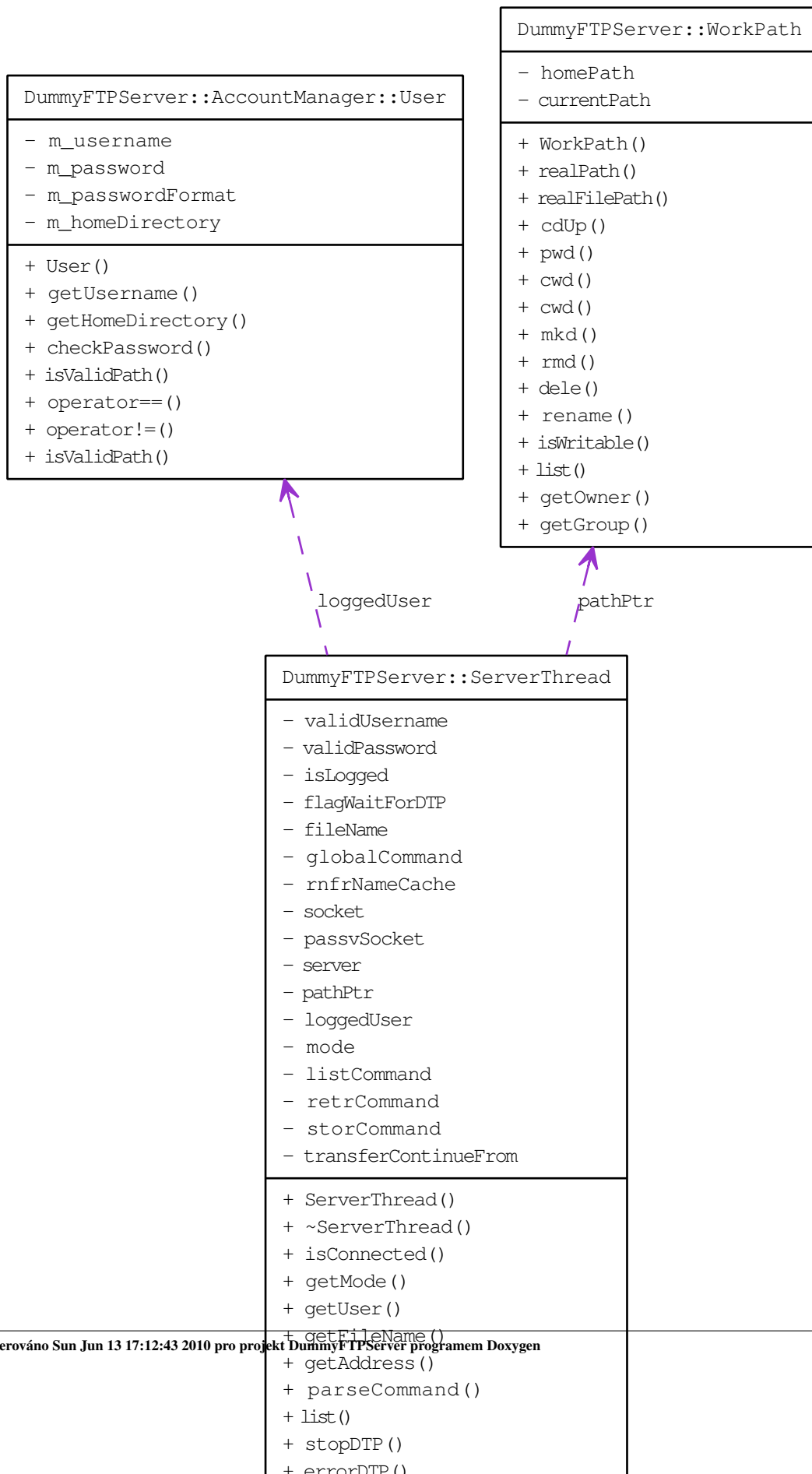


Diagram tříd pro DummyFTPServer::FTPCommandRETR:



### Veřejné sloty

- void [sendData](#) ()  
*Slot, který při zavolání naplní socket daty.*
- void [error](#) (bool stop=true)  
*Slot, který obsluhuje chybové stavy.*

### Veřejné metody

- [FTPCommandRETR](#) (const QString &fileName, [ServerThread](#) \*parent)  
*Vytváří objekt pro zpracování FTP příkaz RETR.*

#### 4.9.1 Detailní popis

Zajišťuje obsluhu FTP příkazu RETR.

Definice je uvedena na řádce 35 v souboru [ftpcommandretr.h](#).

#### 4.9.2 Dokumentace konstruktoru a destruktoru

##### 4.9.2.1 DummyFTPServer::FTPCommandRETR::FTPCommandRETR (const QString &fileName, ServerThread \*parent)

Vytváří objekt pro zpracování FTP příkaz RETR.

### Parametry

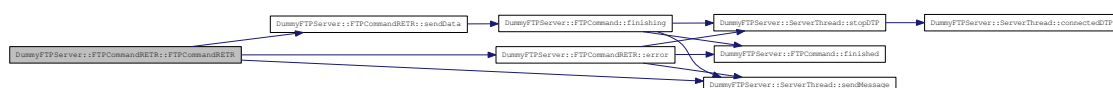
*fileName* Nastaví cestu k souboru.

*parent* Ukazatel na předka.

Definice je uvedena na řádce 28 v souboru [ftpcommandretr.cpp](#).

Odkazuje se na [error\(\)](#), [DummyFTPServer::FTPCommand::file](#), [DummyFTPServer::FTPCommand::isConnected](#), [sendData\(\)](#), [DummyFTPServer::ServerThread::sendMessage\(\)](#) a [DummyFTPServer::FTPCommand::stParent](#).

Tato funkce volá...



Dokumentace pro tuto třídu byla generována z následujících souborů:

- /home/marek/Projekty/DummyFTPServer/source/ftpcommandretr.h
- /home/marek/Projekty/DummyFTPServer/source/ftpcommandretr.cpp

## 4.10 Dokumentace třídy DummyFTPServer::FTPCommandSTOR

Zajišťuje obsluhu FTP příkazu STOR.

```
#include <ftpcommandstor.h>
```

Diagram dědičnosti pro třídu DummyFTPServer::FTPCommandSTOR

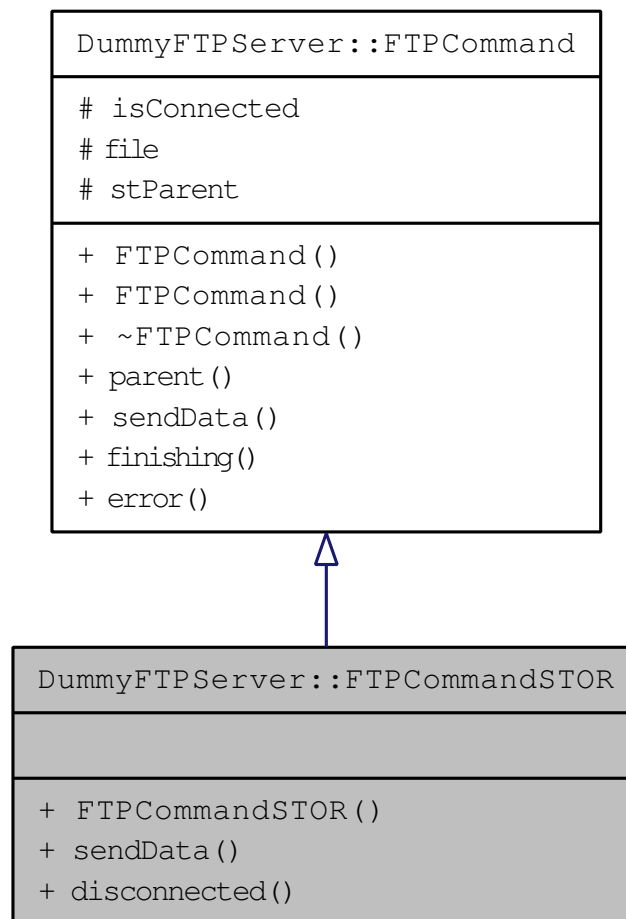
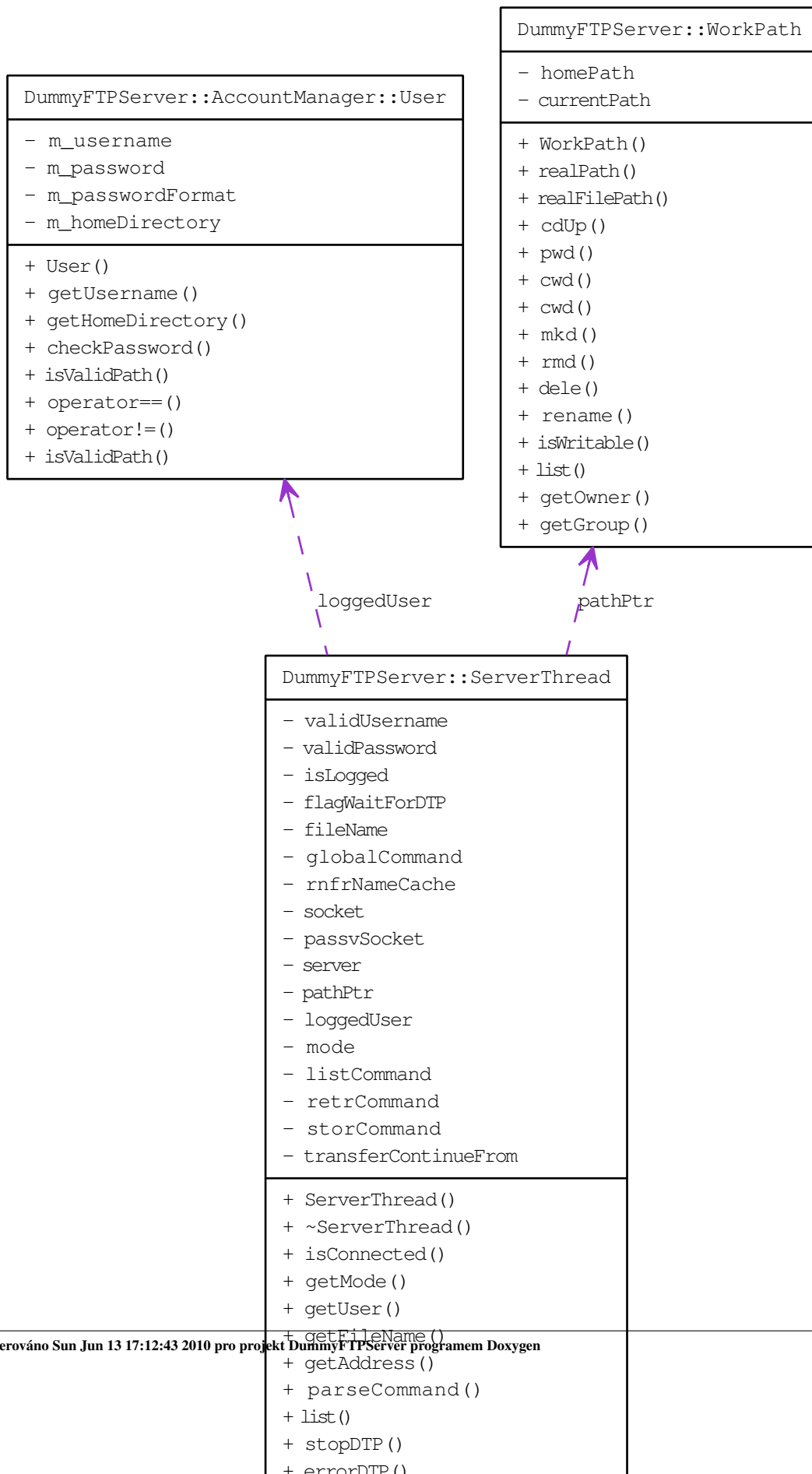


Diagram tříd pro DummyFTPServer::FTPCommandSTOR:



### Veřejné sloty

- void `sendData ()`  
*Slouží k zápisu dat, ze socketu do souboru.*
- void `disconnected ()`  
*Slouží k řádnému ukončení spojení.*

### Veřejné metody

- `FTPCommandSTOR` (const QString &fileName, bool append, `ServerThread` \*parent=0)  
*Vytváří objekt pro zpracování FTP příkaz STOR.*

#### 4.10.1 Detailní popis

Zajišťuje obsluhu FTP příkazu STOR.

Definice je uvedena na řádce 35 v souboru `ftpcommandstor.h`.

#### 4.10.2 Dokumentace konstruktoru a destruktoru

##### 4.10.2.1 DummyFTPServer::FTPCommandSTOR::FTPCommandSTOR (const QString &fileName, bool append, ServerThread \* parent = 0)

Vytváří objekt pro zpracování FTP příkaz STOR.

### Parametry

*fileName* Nastaví cestu k souboru.

*append* Append příznak.

*parent* Ukazatel na předka.

Definice je uvedena na řádce 27 v souboru `ftpcommandstor.cpp`.

Dokumentace pro tuto třídu byla generována z následujících souborů:

- `/home/marek/Projekty/DummyFTPServer/source/ftpcommandstor.h`
- `/home/marek/Projekty/DummyFTPServer/source/ftpcommandstor.cpp`

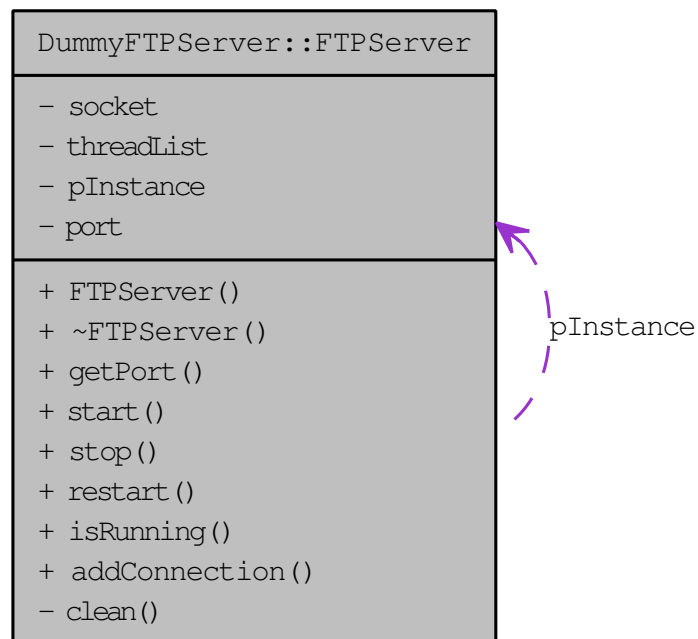
## 4.11 Dokumentace třídy DummyFTPServer::FTPServer

Třída reprezentuje FTP server.

```
#include <ftpserver.h>
```



Diagram tříd pro DummyFTPServer::FTPServer:



### Veřejné sloty

- void [addConnection](#) ()  
*Tento slot obsluhuje nová spojení.*

### Signály

- void [closing](#) ()  
*Signalizuje ukončení naslouchání.*

### Veřejné metody

- [FTPServer](#) (unsigned int tcpPort, QObject \*parent=0)  
*Vytváří objekt, který reprezentuje FTP server.*

### Statické veřejné metody

- static unsigned int [getPort](#) ()  
*Slouží k získání nastaveného portu.*
- static void [start](#) (unsigned int tcpPort)  
*Zapne server.*

- static void `stop()`  
*Zastaví server.*
- static void `restart()`  
*Provede restart serveru.*
- static bool `isRunning()`  
*Zjistí stav serveru.*

#### 4.11.1 Detailní popis

Třída reprezentuje FTP server.

Definice je uvedena na řádku 38 v souboru `ftpserver.h`.

#### 4.11.2 Dokumentace konstruktoru a destruktoru

##### 4.11.2.1 DummyFTPServer::FTPServer::FTPServer (unsigned int *tcpPort*, QObject \* *parent* = 0)

Vytváří objekt, který reprezentuje FTP server.

##### Parametry

*tcpPort* Nastaví port na, kterém bude ftp server poslouchat.

*parent* Ukazatel na předka.

Definice je uvedena na řádku 54 v souboru `ftpserver.cpp`.

Odkazuje se na `addConnection()`.

Používá se v `start()`.

Tato funkce volá...



Tuto funkci volají...



#### 4.11.3 Dokumentace k metodám

##### 4.11.3.1 void DummyFTPServer::FTPServer::start (unsigned int *tcpPort*) [static]

Zapne server.

### Parametry

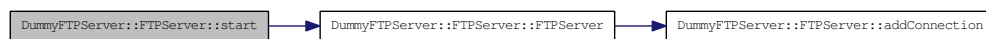
**tcpPort** Nastaví port na, kterém bude ftp server poslouchat.

Definice je uvedena na řádce 31 v souboru `ftpserver.cpp`.

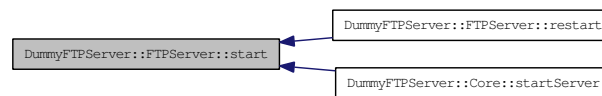
Odkazuje se na `FTPServer()`.

Používá se v `restart()` a `DummyFTPServer::Core::startServer()`.

Tato funkce volá...



Tuto funkci volají...



Dokumentace pro tuto třídu byla generována z následujících souborů:

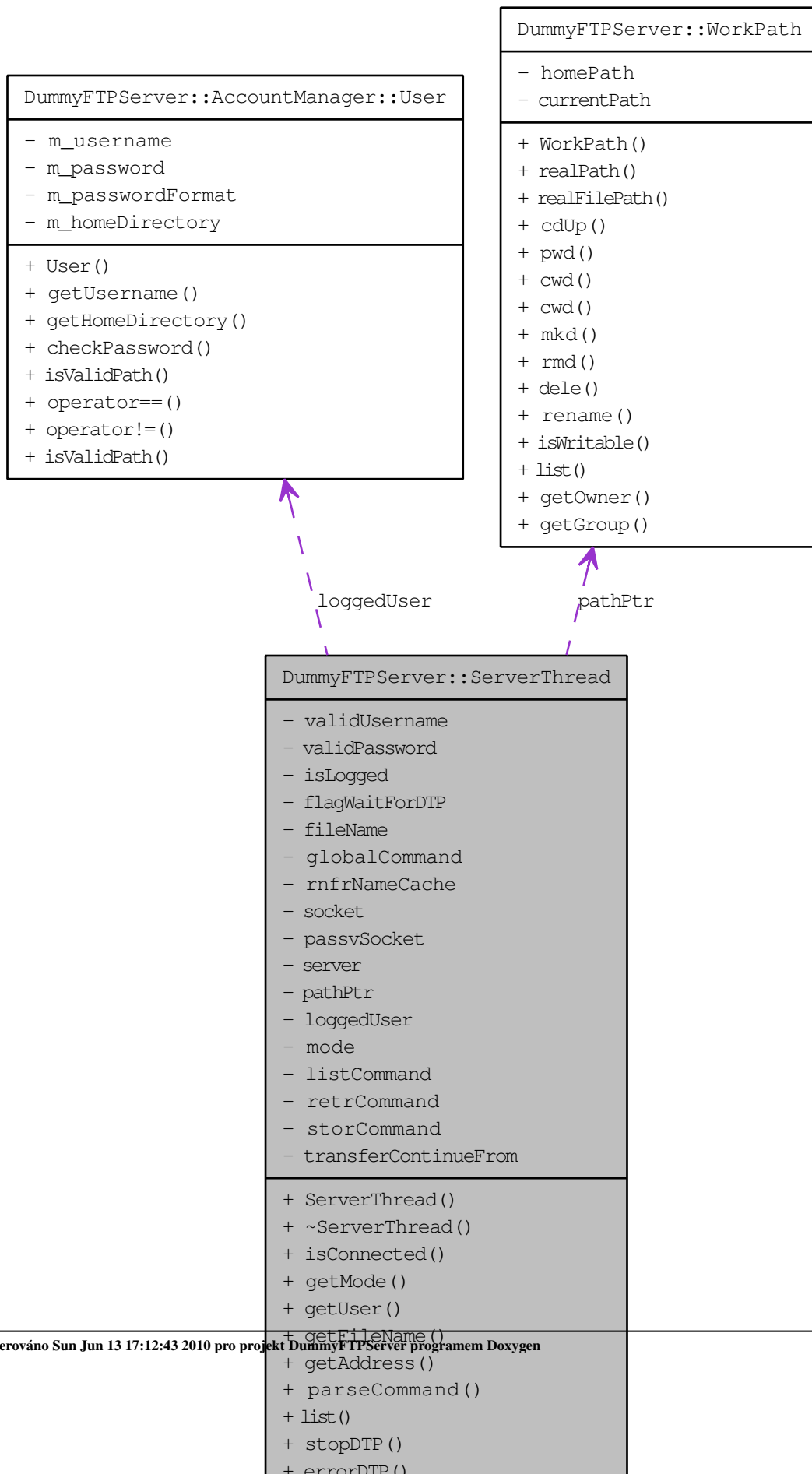
- `/home/marek/Projekty/DummyFTPServer/source/ftpserver.h`
- `/home/marek/Projekty/DummyFTPServer/source/ftpserver.cpp`
- `/home/marek/Projekty/DummyFTPServer/source/moc_ftpserver.cpp`

## 4.12 Dokumentace třídy DummyFTPServer::ServerThread

Objekt reprezentující serverové vlákno.

```
#include <serverthread.h>
```

Diagram tříd pro DummyFTPServer::ServerThread:



### Veřejné typy

- enum `MODE_DTP` { `NONE`, `STOR`, `RETR`, `APPE` }

*Výpočetní typ, který definuje podporované typy přenosů. (Data Transfer Process).*

### Veřejné sloty

- void `parseCommand` ()

*Metoda, která zpracovává příchozí požadavky.*

- void `list` ()

*Metoda, která zajišťuje zpracování seznamu souboru a zaslání do socketu.*

- void `stopDTP` ()

*Zastaví DTP.*

- void `errorDTP` ()

*Obsluhuje chyby okolo DTP.*

- void `portConnection` ()

*Obsluhuje spojení.*

- void `pasvConnection` ()

*Obsluhuje pasivní spojení.*

- void `welcomeMessage` ()

*Zasílá do socketu úplné první zprávu a to uvítací.*

### Signály

- void `connectedDTP` ()

*Signalizuje spojení s DTP. (Data Transfer Process).*

### Veřejné metody

- `ServerThread` (QTcpSocket \*\_socket, QObject \*parent=0)

*Vytváří serverové vlákno.*

- bool `isConnected` ()

*Vrací pravdu pokud spojení stále trvá, v opačném případě nepravdu.*

- `MODE_DTP` `getMode` ()

*Vrací aktuální typ přenosového režimu.*

- `User *` `getUser` ()

*Vrací ukazatel na uživatele, který je v daném vláknu obsluhován.*

- std::string [getFileName](#) ()  
*Vrací název pracovního souboru.*
- std::string [getAddress](#) ()  
*Vrací adresu vzdáleného klienta.*

### Chráněné metody

- bool [sendMessage](#) (const QString &commandCode, const QString &message)  
*Metoda, která posílá FTP zprávy do TCP socketu.*
- void [openPassive](#) ()  
*Metoda, která zapíná pasivní komunikační režim.*

### Friends

- class **FTPCommand**
- class **FTPCommandLIST**
- class **FTPCommandRETR**
- class **FTPCommandSTOR**

#### 4.12.1 Detailní popis

Objekt reprezentující serverové vlákno.

Definice je uvedena na řádce 52 v souboru [serverthread.h](#).

#### 4.12.2 Dokumentace k metodám

##### 4.12.2.1 bool DummyFTPServer::ServerThread::sendMessage (const QString & *commandCode*, const QString & *message*) [protected]

Metoda, která posílá FTP zprávy do TCP socketu.

### Parametry

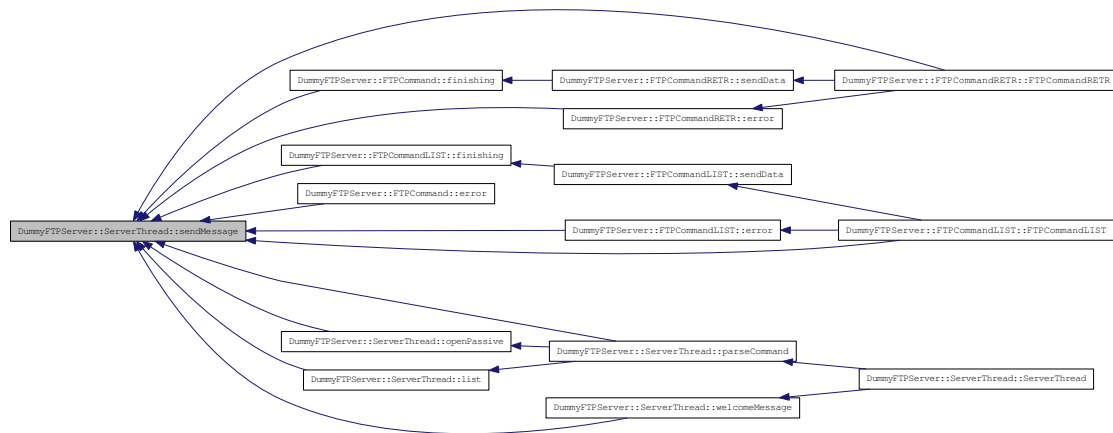
*commandCode* Číslo ftp příkazu.

*message* Textová zpráva, která bude poslána klientovy.

Definice je uvedena na řádce 384 v souboru [serverthread.cpp](#).

Používá se v [DummyFTPServer::FTPCommandRETR::error\(\)](#), [DummyFTPServer::FTPCommandLIST::error\(\)](#), [DummyFTPServer::FTPCommand::error\(\)](#), [DummyFTPServer::FTPCommandLIST::finishing\(\)](#), [DummyFTPServer::FTPCommand::finishing\(\)](#), [DummyFTPServer::FTPCommandLIST::FTPCommandLIST\(\)](#), [DummyFTPServer::FTPCommandRETR::FTPCommandRETR\(\)](#), [list\(\)](#), [openPassive\(\)](#), [parseCommand\(\)](#) a [welcomeMessage\(\)](#).

Tuto funkci volají...



Dokumentace pro tuto třídu byla generována z následujících souborů:

- /home/marek/Projekty/DummyFTPServer/source/serverthread.h
- /home/marek/Projekty/DummyFTPServer/source/moc\_serverthread.cpp
- /home/marek/Projekty/DummyFTPServer/source/serverthread.cpp

## 4.13 Dokumentace třídy DummyFTPServer::WorkPath

Třída provádí veškerou práci, ze soubory a adresáři od načítání obsahu po mazání atd...

```
#include <workpath.h>
```

### Veřejné metody

- **WorkPath** (const QString &path)  
*Vytváří objekt, který provádí manipulaci, ze soubory a adresáři.*
- QString **realPath** ()  
*Vrací reálnou cestu k aktuálnímu adresáři.*
- QString **realFilePath** (QString filename)  
*Provádí překlad ftp adresy na reálnou adresu.*
- bool **cdUp** ()  
*Nastaví aktuální adresář na nadřazený adresář.*
- QString **pwd** ()  
*Vrací cestu k aktuálnímu adresáři.*
- bool **cwd** (const QString &path)  
*Mění aktuální adresář.*
- bool **cwd** ()

*Mění aktuální adresář na domovský adresář.*

- bool [mkd](#) (const QString &name)  
*Vytváří nový adresář.*
- bool [rmd](#) (const QString &name)  
*Smaže adresář.*
- bool [delete](#) (const QString &name)  
*Smaže soubor.*
- bool [rename](#) (const QString &fromFileName, const QString &toFileName)  
*Přejmenovává soubory.*
- bool [isWritable](#) (const QString &filename)  
*Kontrola zapisovatelnosti.*
- void [list](#) (QList< QByteArray > &sList)  
*Seznam souborů.*

#### Statické veřejné metody

- static QString [getOwner](#) (const QFileInfo &info)  
*Slouží k získání vlastníka.*
- static QString [getGroup](#) (const QFileInfo &info)  
*Slouží k získání skupiny.*

#### 4.13.1 Detailní popis

Třída provádí veškerou práci, ze soubory a adresáři od načítání obsahu po mazání atd...

Definice je uvedena na řádce 36 v souboru [workpath.h](#).

#### 4.13.2 Dokumentace konstruktoru a destruktoru

##### 4.13.2.1 DummyFTPServer::WorkPath::WorkPath (const QString & path)

Vytváří objekt, který provádí manipulaci, ze soubory a adresáři.

#### Parametry

*path* Nastaví pracovní cestu.

Definice je uvedena na řádce 52 v souboru [workpath.cpp](#).



### 4.13.3 Dokumentace k metodám

#### 4.13.3.1 bool DummyFTPServer::WorkPath::cwd (const QString & path)

Mění aktuální adresář.

##### Parametry

*path* Nastaví cestu.

Definice je uvedena na řádce 81 v souboru [workpath.cpp](#).

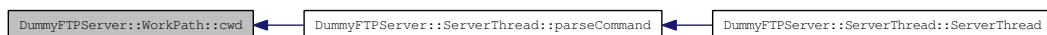
Odkazuje se na [realPath\(\)](#).

Používá se v [DummyFTPServer::ServerThread::parseCommand\(\)](#).

Tato funkce volá...



Tuto funkci volají...



#### 4.13.3.2 bool DummyFTPServer::WorkPath::delete (const QString & name)

Smaže soubor.

##### Parametry

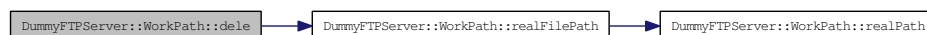
*name* Název souboru.

Definice je uvedena na řádce 142 v souboru [workpath.cpp](#).

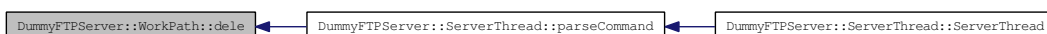
Odkazuje se na [realFilePath\(\)](#).

Používá se v [DummyFTPServer::ServerThread::parseCommand\(\)](#).

Tato funkce volá...



Tuto funkci volají...



#### 4.13.3.3 QString DummyFTPServer::WorkPath::getGroup (const QFileInfo & *info*) [static]

Slouží k získání skupiny.

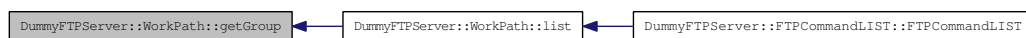
##### Parametry

*info* Ukazatel na soubor.

Definice je uvedena na řádce 43 v souboru [workpath.cpp](#).

Používá se v [list\(\)](#).

Tuto funkci volají...



#### 4.13.3.4 QString DummyFTPServer::WorkPath::getOwner (const QFileInfo & *info*) [static]

Slouží k získání vlastníka.

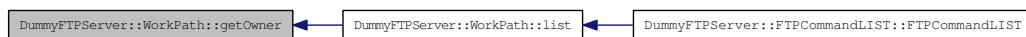
##### Parametry

*info* Ukazatel na soubor.

Definice je uvedena na řádce 32 v souboru [workpath.cpp](#).

Používá se v [list\(\)](#).

Tuto funkci volají...



#### 4.13.3.5 bool DummyFTPServer::WorkPath::isWritable (const QString & *filename*)

Kontrola zapisovatelnosti.

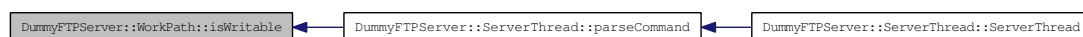
##### Parametry

*filename* Název souboru.

Definice je uvedena na řádce 178 v souboru [workpath.cpp](#).

Používá se v [DummyFTPServer::ServerThread::parseCommand\(\)](#).

Tuto funkci volají...



#### 4.13.3.6 void DummyFTPServer::WorkPath::list (QList< QByteArray > & sList)

Seznam souborů.

##### Parametry

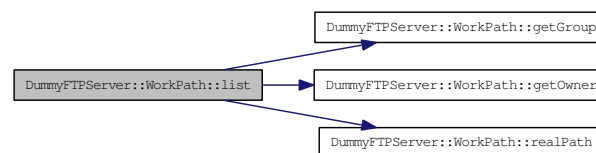
*sList* Vstupní seznam souborů.

Definice je uvedena na řádce 184 v souboru [workpath.cpp](#).

Odkazuje se na [getGroup\(\)](#), [getOwner\(\)](#) a [realPath\(\)](#).

Používá se v [DummyFTPServer::FTPCommandLIST::FTPCommandLIST\(\)](#).

Tato funkce volá...



Tuto funkci volají...



#### 4.13.3.7 bool DummyFTPServer::WorkPath::mkd (const QString & name)

Vytváří nový adresář.

##### Parametry

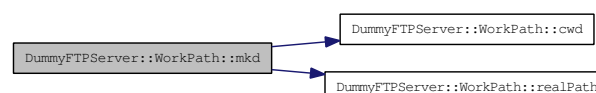
*name* Nastaví jméno nového adresáře.

Definice je uvedena na řádce 111 v souboru [workpath.cpp](#).

Odkazuje se na [cwd\(\)](#) a [realPath\(\)](#).

Používá se v [DummyFTPServer::ServerThread::parseCommand\(\)](#).

Tato funkce volá...



Tuto funkci volají...



#### 4.13.3.8 QString DummyFTPServer::WorkPath::realFilePath (QString *filename*)

Provádí překlad ftp adresy na reálnou adresu.

##### Parametry

*filename* Název souboru.

Definice je uvedena na řádku 63 v souboru [workpath.cpp](#).

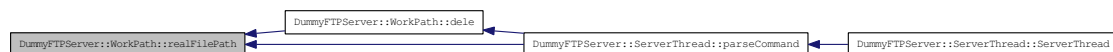
Odkazuje se na [realPath\(\)](#).

Používá se v [delete\(\)](#) a [DummyFTPServer::ServerThread::parseCommand\(\)](#).

Tato funkce volá...



Tuto funkci volají...



#### 4.13.3.9 bool DummyFTPServer::WorkPath::rename (const QString & *fromFileName*, const QString & *toFileName*)

Přejmenovává soubory.

##### Parametry

*fromFileName* Starý název souboru.

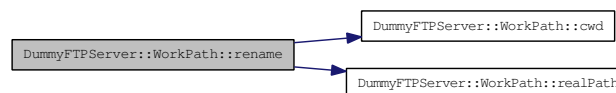
*toFileName* Nový název souboru.

Definice je uvedena na řádku 162 v souboru [workpath.cpp](#).

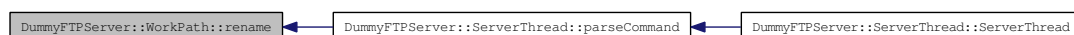
Odkazuje se na [cwd\(\)](#) a [realPath\(\)](#).

Používá se v [DummyFTPServer::ServerThread::parseCommand\(\)](#).

Tato funkce volá...



Tuto funkci volají...



#### 4.13.3.10 bool DummyFTPServer::WorkPath::rmd (const QString & name)

Smaže adresář.

##### Parametry

**name** Nastaví jméno adresáře.

Definice je uvedena na řádku 124 v souboru [workpath.cpp](#).

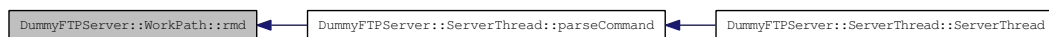
Odkazuje se na [realPath\(\)](#).

Používá se v [DummyFTPServer::ServerThread::parseCommand\(\)](#).

Tato funkce volá...



Tuto funkci volají...



Dokumentace pro tuto třídu byla generována z následujících souborů:

- /home/marek/Projekty/DummyFTPServer/source/workpath.h
- /home/marek/Projekty/DummyFTPServer/source/workpath.cpp

## Rejstřík

- add
  - DummyFTPServer::AccountManager::UserList, 7
- append
  - DummyFTPServer::ConfSection, 15
- appendToSection
  - DummyFTPServer::Configuration, 12
- checkPassword
  - DummyFTPServer::AccountManager::User, 4
- clean
  - DummyFTPServer::AccountManager::UserList, 8
  - DummyFTPServer::Configuration, 13
- Configuration
  - DummyFTPServer::Configuration, 12
- ConfSection
  - DummyFTPServer::ConfSection, 15
- count
  - DummyFTPServer::AccountManager::UserList, 8
- cwd
  - DummyFTPServer::WorkPath, 47
- dele
  - DummyFTPServer::WorkPath, 47
- DummyFTPServer::AccountManager::User, 3
  - checkPassword, 4
  - isValidPath, 5
  - operator==, 6
  - User, 4
- DummyFTPServer::AccountManager::UserList, 6
  - add, 7
  - clean, 8
  - count, 8
  - getByUsername, 8
  - isEmpty, 9
  - loadUserList, 9
  - parsePassword, 10
  - parseUser, 10
  - remove, 11
- DummyFTPServer::Configuration, 11
  - appendToSection, 12
  - clean, 13
  - Configuration, 12
  - getSection, 13
  - loadConfiguration, 14
- DummyFTPServer::ConfSection, 14
  - append, 15
  - ConfSection, 15
  - getCount, 16
  - getName, 16
  - getValue, 16
  - remove, 16
- DummyFTPServer::Core, 17
  - getAnonymousHome, 19
  - getSettingsValue, 20
  - getUser, 21
  - getUserCount, 21
  - Instance, 22
  - loadConfiguration, 22
  - split, 23
  - startServer, 23
  - stringToBool, 24
  - validatePortRange, 24
- DummyFTPServer::ErrorException, 25
  - ErrorException, 25
  - isCritical, 25
  - what, 26
- DummyFTPServer::FTPCommand, 26
  - FTPCommand, 29
- DummyFTPServer::FTPCommandLIST, 29
  - FTPCommandLIST, 32
- DummyFTPServer::FTPCommandRETR, 33
  - FTPCommandRETR, 35
- DummyFTPServer::FTPCommandSTOR, 36
  - FTPCommandSTOR, 38
- DummyFTPServer::FTPServer, 38
  - FTPServer, 40
  - start, 40
- DummyFTPServer::ServerThread, 41
  - sendMessage, 44
- DummyFTPServer::WorkPath, 45
  - cwd, 47
  - dele, 47
  - getGroup, 47
  - getOwner, 48
  - isWritable, 48
  - list, 48
  - mkd, 49
  - realFilePath, 50
  - rename, 50
  - rmd, 50
  - WorkPath, 46
- ErrorException
  - DummyFTPServer::ErrorException, 25
- FTPCommand
  - DummyFTPServer::FTPCommand, 29
- FTPCommandLIST
  - DummyFTPServer::FTPCommandLIST, 32

- FTPCommandRETR
  - DummyFTPServer::FTPCommandRETR, [35](#)
- FTPCommandSTOR
  - DummyFTPServer::FTPCommandSTOR, [38](#)
- FTPServer
  - DummyFTPServer::FTPServer, [40](#)
- getAnonymousHome
  - DummyFTPServer::Core, [19](#)
- getByUsername
  - DummyFTPServer::AccountManager::UserList, [8](#)
- getCount
  - DummyFTPServer::ConfSection, [16](#)
- getGroup
  - DummyFTPServer::WorkPath, [47](#)
- getName
  - DummyFTPServer::ConfSection, [16](#)
- getOwner
  - DummyFTPServer::WorkPath, [48](#)
- getSection
  - DummyFTPServer::Configuration, [13](#)
- getSettingsValue
  - DummyFTPServer::Core, [20](#)
- getUser
  - DummyFTPServer::Core, [21](#)
- getUserCount
  - DummyFTPServer::Core, [21](#)
- getValue
  - DummyFTPServer::ConfSection, [16](#)
- Instance
  - DummyFTPServer::Core, [22](#)
- isCritical
  - DummyFTPServer::ErrorException, [25](#)
- isEmpty
  - DummyFTPServer::AccountManager::UserList, [9](#)
- isValidPath
  - DummyFTPServer::AccountManager::User, [5](#)
- isWritable
  - DummyFTPServer::WorkPath, [48](#)
- list
  - DummyFTPServer::WorkPath, [48](#)
- loadConfiguration
  - DummyFTPServer::Configuration, [14](#)
  - DummyFTPServer::Core, [22](#)
- loadUserList
  - DummyFTPServer::AccountManager::UserList, [9](#)
- mkd
  - DummyFTPServer::WorkPath, [49](#)
- operator==
  - DummyFTPServer::AccountManager::User, [6](#)
- parsePassword
  - DummyFTPServer::AccountManager::UserList, [10](#)
- parseUser
  - DummyFTPServer::AccountManager::UserList, [10](#)
- realFilePath
  - DummyFTPServer::WorkPath, [50](#)
- remove
  - DummyFTPServer::AccountManager::UserList, [11](#)
  - DummyFTPServer::ConfSection, [16](#)
- rename
  - DummyFTPServer::WorkPath, [50](#)
- rmd
  - DummyFTPServer::WorkPath, [50](#)
- sendMessage
  - DummyFTPServer::ServerThread, [44](#)
- split
  - DummyFTPServer::Core, [23](#)
- start
  - DummyFTPServer::FTPServer, [40](#)
- startServer
  - DummyFTPServer::Core, [23](#)
- stringToBool
  - DummyFTPServer::Core, [24](#)
- User
  - DummyFTPServer::AccountManager::User, [4](#)
- validatePortRange
  - DummyFTPServer::Core, [24](#)
- what
  - DummyFTPServer::ErrorException, [26](#)
- WorkPath
  - DummyFTPServer::WorkPath, [46](#)