

Uživatelský manuál

Daniel Vršek

27. listopadu 2018

Abstrakt

Cílem tohoto dokumentu je popsat postup, který je nutné vykonat před používáním aplikace klienta protokolu LTP (dále jen klienta) a aplikace serveru protokolu LTP (dále jen serveru), a jsou zde popsána možná využití těchto aplikací v praxi s názornými ukázkami.

Obsah

I	Příprava prostředí a kompilace	3
1	Platforma	3
2	Kompilace	3
II	Server	4
3	Spuštění	4
3.1	Windows	4
3.2	Ostatní	4
4	Konfigurace	4
5	Logování	4
III	Klient	5
6	Spuštění	5
6.1	Windows	5
6.2	Ostatní	5
6.3	Argumenty	5

7	Příkazy	6
7.1	Připojení k serveru - connect	6
7.2	Autentikace - login	6
7.3	Správa účtů - user	6
7.3.1	Create	6
7.3.2	Delete	6
7.4	Změna aktuálního adresáře - cd	7
7.5	Výpis obsahu adresáře - ls	7
7.6	Vytvoření nového adresáře - mkdir	7
7.7	Stahování souboru - download	7
7.8	Nahrání souboru - upload	8
IV	Appendixy	9
8	Podporované operační systémy	9

Část I

Příprava prostředí a kompilace

1 Platforma

Aplikace pro své spuštění a běh vyžadují nainstalovanou platformu .NET Core 2.1 nebo vyšší od firmy Microsoft. Pro seznam podporovaných operačních systémů se odkažte na sekci Podporované operační systémy

2 Kompilace

Pokud máte k dispozici pouze zdrojový kód, budete potřebovat stáhnout vývojářskou verzi platformy .NET Core 2.1 nebo vyšší, která podporuje kompilaci (build) projektů. Pokud máte k dispozici verzi již obsahující binární soubory, můžete tuto sekci přeskóčit.

Vývojářskou verzi naleznete na stránkách pro stažení .NET Core pod označením SDK. Po stažení pokračujte dle návodu např. na <https://docs.microsoft.com>

Část II

Server

3 Spuštění

Ujistěte se, že máte nainstalovaný Runtime nebo SDK platformy .NET Core 2.1 nebo vyšší. Spuštění se bude lišit v závislosti na vašem operačním systému. Zadaný port nesmí být již používán a ip adresa musí být platná.

3.1 Windows

Pokud nemáte k dispozici spustitelný binární soubor (Server.exe), můžete ho získat kompilací (viz. Kompilace) nebo použít příkaz

```
dotnet Server.dll ipAddress port
```

*parametry „ipAddress“ a „port“ jsou povinné údaje

3.2 Ostatní

Pro spuštění použijte příkaz

```
dotnet Server.dll ipAddress port
```

*parametry „ipAddress“ a „port“ jsou povinné údaje

4 Konfigurace

Konfigurace je základním požadavkem pro jakýkoliv server. Konfigurace se nachází ve složce s binárním souborem pod názvem **config.cfg**. Pokud se ve složce nenachází konfigurační soubor, bude vygenerován při následujícím spuštění.

Konfigurovatelné jsou následující položky

RootFolder	Kořenový adresář pro klienty
UserSecretFilepath	Soubor s uživatelským jménem a zašifrovaným heslem klientů

5 Logování

Logování (zatím) není implementováno.

Část III

Klient

6 Spuštění

Ujistěte se, že máte nainstalovaný Runtime nebo SDK platformy .NET Core 2.1 nebo vyšší. Spuštění se bude lišit v závislosti na vašem operačním systému. Pro spuštění jsou k dispozici Argumenty, které klient přijímá.

6.1 Windows

Pokud nemáte k dispozici spustitelný binární soubor (Client.exe), můžete ho získat kompilací (viz. Kompilace) nebo použít příkaz

```
dotnet Client.dll
```

6.2 Ostatní

Pro spuštění použijte příkaz `dotnet Client.dll`

6.3 Argumenty

Argumenty klienta se rozumí jednotlivé příkazy. Každý příkaz začíná slashem (/) a parametry tohoto příkazu následují po něm až dokud nebude specifikován jiný příkaz. V případě, že nějaký příkaz vyžaduje parametry, které nebyly zadány, bude před jejich vykonáním uživatel vyzván, aby je zadal.

Příklad:

```
dotnet Client.dll /connect 192.168.0.102 1234 --no-login /  
login username
```

Klient vykoná příkazy až do doby, dokud nebude zjištěn chybějící parametr v nějakém příkazu. V tomto příkladu se jedná o heslo, které je při přihlášení vyžadováno.

7 Příkazy

7.1 Připojení k serveru - connect

Tento příkaz slouží k připojení ke vzdálenému serveru.

Syntaxe:

```
connect [-ip ipAddress] [[ipAddress] [{-p|--port} port]] [port] [--no-login]
```

Parametry:

ipAddress	Adresa (IPv4) vzdáleného serveru
port	Port vzdáleného serveru
--no-login	Pokud je zadáno, tak se nevyžaduje autentikace ihned po připojení

7.2 Autentikace - login

Tento příkaz slouží k autentizaci na vzdáleném serveru.

Syntaxe:

```
login [username] [password] [--no-force]
```

Parametry:

username	Uživatelské jméno
password	Heslo
--no-force	Pokud není zadáno, tak se bude stále dokola pokoušet získat správné přihlašovací údaje

7.3 Správa účtů - user

Tento příkaz slouží k manipulaci s uživatelskými účty na serveru.

7.3.1 Create

Vytvoří nového uživatele.

Syntaxe:

```
user create [-u|--username username] [[username] [-p|--password password] [password]]
```

Parametry:

username	Uživatelské jméno nového uživatele
password	Heslo nového uživatele

7.3.2 Delete

Odstraní uživatele.

Syntaxe:

```
user delete [-u|--username username] [[username]
```

Parametry:

`username` Uživatelské jméno mazaného uživatele

7.4 Změna aktuálního adresáře - `cd`

Tento příkaz změní **lokálně** aktuální adresář.

Syntaxe:

```
cd directory
```

Parametry:

`directory` Název adresáře, do kterého který se chcete navigovat

7.5 Výpis obsahu adresáře - `ls`

Tento příkaz vypíše obsah adresáře..

Syntaxe:

```
ls [directory]
```

Parametry:

`directory` Název adresáře, jehož obsah chcete vypsát. Pokud bude prázdné, vypíše aktuální.

7.6 Vytvoření nového adresáře - `mkdir`

Vytvoří nový adresář. Pokud cesta k tomuto adresáři neexistuje, vytvoří ji to.

Syntaxe:

```
mkdir directory
```

Parametry:

`directory` Název adresáře, který má být vytvořen

7.7 Stažení souboru - `download`

Stáhne soubor uložený na vzdáleném úložišti.

Syntaxe:

```
download [-r|--remote remoteFile] [[remoteFile] [-l|--local  
localFile] [[localFile]
```

Parametry:

`remoteFile` Název souboru na vzdáleném úložišti

`localFile` Název nového souboru na lokálním úložišti

7.8 Náhrání souboru - upload

Uloží lokální soubor na vzdálené úložiště.

Syntaxe:

```
download [-l|--local localFile] [[localFile] [-r|--remote  
remoteFile] [[remoteFile]
```

Parametry:

<code>localFile</code>	Název souboru na lokálním úložišti
<code>remoteFile</code>	Název nového souboru na vzdáleném úložišti

Část IV

Appendixy

8 Podporované operační systémy

Klient i server mají totožné požadavky na platformu, na které mohou běžet. Tyto platformy jsou omezeny podporou .NET Core na následující operační systémy.

Windows

OS	Verze	Architektura
Windows Client	7 SP1+, 8.1	x86, x64
Windows 10 Client	Version 1607+	x86, x64
Windows Server	2008 R2 SP1+	x86, x64

macOS

OS	Verze	Architektura
Mac OS X (macOS)	10.12+	x64

Linux

OS	Verze	Architektura
Red Hat Enterprise Linux	6, 7	x64
CentOS	7	x64
Oracle Linux	7	x64
Fedora	27, 28	x64
Debian	9, 8.7+	x64
Ubuntu	18.04, 16.04, 14.04	x64, ARM32*
Linux Mint	18, 17	x64, ARM32*
openSUSE	42.3+	x64
SUSE Enterprise Linux (SLES)	12 SP2+	x64
Alpine Linux	3.7+	x64

* Podpora ARM32 začíná s Debian 9 a Ubuntu 18.04. Předchozí verze těchto distribucí nejsou pro ARM32 podporovány.