1 Building

To build in MS Visual Studio use **imap_terminal.sln** file provided with the source. To build in Linux use CMake build system. Create a separate build directory, cd into it, configure and build **imap_terminal** like this:

```
mkdir path/to/build/dir
cd path/to/build/dir
cmake path/to/source/dir
cmake --build .
```

imap_terminal depends on libcurl to exchange data with IMAP servers. For Windows the most recent stable version of libcurl is pre-built and **libcurl.dll** is put in the same directory with **imap_terminal.exe** automatically by MS Visual Studio build system. In Linux libcurl is available for installation from repositories in all modern distributions. For example, in Ubuntu-based distros install libcurl by runnning

```
sudo apt-get install libcurl4-openssl-dev
```

This must be done before trying to build **imap_terminal**.

2 Using

imap_terminal has the following command line syntax:

```
imap_terminal -h|-host=<...> [-p|-port=<...>] -u|-user=<...>
        [-P|-pass=<...>] [-s|-ssl]
```

Parameters **host, port, user, pass** are self-explanatory. Parameter **-ssl** indicates that SSL should be used to connect. Parameters **host** and **user** are required, everything else is optional. If **port** parameter is omitted then 993 is used as a default. If **-ssl** switch is not given on the command line, imap_terminal will try to connect without using SSL. For example:

```
imap_terminal -h=imap.gmail.com -p=993 -u=me@gmail.com
        -P=mypass -ssl
```

will connect to the Gmail service. When connection is established, an interactive shell-like interface is presented. This interface can be used to manipulate messages and directories in the user's mailbox. The following commands are implemented.

2.1 Working with directory tree

Use **ls, cd, pwd, mkdir, rmdir** commands to work with directories.

1. **cd** command changes the current directory. Accepts both relative and absolute paths. Understands "."and ".."special dirs. Example:

```
cd INBOX/subdir1
cd "../../[Gmail]/Sent Mail"
```

2. **pwd** command prints the current directory

3. **mkdir, rmdir** commands create and delete subdirectories inside the current directory. Example:

```
$ mkdir subdir1
$ cd subdir1
$ pwd
/subdir1
$
```

4. Use **ls** command without arguments to list the contents of the current directory. It will start by printing subdirectories in the current directory

2.2 Working with messages

Use **ls, rm, mv, head** commands to work with messages. These commands share the common syntax:

```
ls|head|rm|mv [-s|-subject=<...>] [-t|-to=<...>]
              [-f|-from=<...>] [message_uid] [destination]
```

-subject, -to, -from options or their short synonims **-s, -t, -f** can be used to filter messages in the current directory to operate on. Only messages matching the given criteria will be operated on. To do somaething to the individual message supply message UID. [destination] parameter is required for mv command. Examples:

```
ls 3
ls -from=me@gmail.com
ls -s=<<a message>> -from=me@gmail.com
ls
```

First command lists message with UID=3. Second command lists all messages from me@gmail.com in the current directory. Third command lists all messages in the current directory from me@gmail.com **AND** having subject «a message». Fourth command lists everything in the current directory.

ls, rm, mv, head commands always operate on a subset of messages in the current directory. By default the scope of these commands is limited to 20 most recent messages in the current directory. To change this value use **limit** command (described below).

1. **ls** command can be used to list messages in the current directory.

2. **mv** command can be used to move messages between subdirectories. Usage:

```
mv <what> <where>
```

<what> can be a message UID or a set of selection options like -s etc. as described above. <where> is an absolute or relative path to the destination directory

3. **rm** command removes messages from the current directory. Be careful to issue **rm** - it will remove all messages in the current directory.

4. **head** command outputs the beginning of a message body

2.3 Other commands

1. **limit** command is used to limit the scope of message commands described in section 2.2. Usage:

```
limit [N]
```

limit without arguments prints the current value. **limit N** updates the current value to **N**.

2. **exit** or **quit** commands make imap_terminal close current session and terminate.

3. **whoami** command prints the username currently used to authenticate to IMAP server.

2.4 Example session

Command	**Output**	**Explanation**
imap_terminal -h=imap.gmail.com -p=993 -u=me@gmail.com -P=mypass -ssl	none	Run the program and connect to Gmail
$ pwd	/	show the current directory
$ ls	d INBOX d [Gmail] d test_label	list the contents of the current directory
$ cd INBOX		change the current directory to **/INBOX**
$ pwd	/INBOX	show the current directory

$ ls	1 From:... Subject:...	list the contents of the current directory. Listing shows that the message with UID=1 exists
$ mv -s=... «../test_label»		Move the message with given subject to the **../test_label** directory
$ cd ../test_label		change the current directory to **/test_label**
$ ls	d test_sublabel_1 d test_sublabel_2 3 From:.. Subject:**test1** 2 From:.. Subject:... 1 From:.. Subject:...	List the contents of the current directory. Ensure that previous **mv** command succeeded