



UFTP STANDALONE CLIENT

UNICORE Team

Document Version:	1.0.0
Component Version:	0.3.0
Date:	06 11 2014

Contents

1 Prerequisites	1
2 Installation	1
3 Basic Usage	1
4 Examples	2
4.1 Listing a directory: the "ls" command	2
4.2 Copying data: the "cp" command	2
4.3 Synchronizing a file: the "sync" command	4
5 Troubleshooting	4
5.1 I get "Invalid server response 500" and "Exception. . . Authentication failure" .	4
6 Getting support	4

This is a Java-based client for UFTP. It allows to

- list remote directories
- upload/download files
- sync files

It supports username/password authentication and ssh-key authentication to a UFTP Authentication Server.

1 Prerequisites

- Java 7 (OpenJDK, Oracle, IBM)
- Access to a UFTP authentication service and the corresponding UFTPD server. To use the client, you need to know the address of the authentication service.
- If you want to use the SSH-key authentication together with the ssh-agent, you will need netcat (nc). This is a standard Unix command and should be already installed on your system. If this is not the case, consult your local administrator.

2 Installation

Unzip the archive in a location of your choice. Add the *bin* directory to your path. (Alternatively, you can copy *bin/uftp* script to a directory that is already on your path, in this case edit the script and setup the required directories.)

3 Basic Usage

In this manual, we use the following format to indicate commands that you can type on the command line:

```
$> some_command
```

and assume that the bin directory of the UFTP client is on your path.

Invoking uftp without any arguments,

```
$> uftp
```

will list the available commands.

Invoking

```
$> uftp <command> -h'
```

will show help for a particular command

For password authentication, use the "-P" option. The password can be written into the URL, for example:

```
$> uftp ls -P uftp://demo:password@localhost:9000/rest/auth/TEST:/ ←  
home/demo/
```

If not given in the URL, it will be queried interactively.

If "-P" is not given, the client will attempt to use your SSH key for authentication, this has to be configured on the authentication server accordingly.

4 Examples

In the following examples, the authentication service is located at "localhost:9000/rest/auth/TEST" and the user name is *demo*. Replace these values by the correct ones for your installation.

4.1 Listing a directory: the "ls" command

```
$> uftp ls uftp://demo@localhost:9000/rest/auth/TEST:/home/demo/
```

will list the */home/demo* directory.

4.2 Copying data: the "cp" command

The *cp* command is used to copy local data to a remote server or vice versa. Remote locations are indicated by the "uftp://" prefix, and you need your user name, and the URL of the authentication server.

It has a number of features, which will be shown in the following.

4.2.1 Basic usage

Downloading a single file:

```
$> uftp cp uftp://demo@localhost:9000/rest/auth/TEST:/home/demo/ ←  
test.data .
```

will download the */home/demo/test.data* file to the current directory

Download files using wildcards:

```
$> uftp cp uftp://demo@localhost:9000/rest/auth/TEST:/home/demo/ ↵  
data/* .
```

will download all files in the */home/demo/test* directory to the current directory

Similar commands work for upload.

4.2.2 Piping data

The "cp" command can read/write from the console streams, which is great for integrating uftp into Unix pipes. The "-" is used as a special "file name" to indicate that data should be read/written using the console.

For example to tar the contents of a directory and upload the tar file using uftp:

```
$> tar cz dir/* | uftp cp - uftp://demo@localhost:9000/rest/auth/ ↵  
TEST:/archive.tgz
```

Similarly, "-" can be used to write data to standard output. As an example, consider this:

```
$> uftp cp uftp://demo@localhost:9000/rest/auth/TEST:/archive.tgz - ↵  
| tar tz
```

Or use uftp to cat a remote file:

```
$> uftp cp uftp://demo@localhost:9000/rest/auth/TEST:/foo.txt -
```

4.2.3 Byte ranges

To copy just part of a file, a byte range can be given with the "-R" option. Counting starts at "zero". For example to download only the first 1024 bytes of file (i.e. the range 0 - 1023), you would do

```
$> uftp cp -R 0-1023 uftp://demo@localhost:9000/rest/auth/TEST:/ ↵  
home/demo/test.data .
```

As an additional feature, you can use the additional "-p" flag, which will write also only the given range. For example

```
$> uftp cp -R 1024-2047-p uftp://demo@localhost:9000/rest/auth/TEST ↵  
:/home/demo/test.data .
```

will write bytes 1024-2047 of the remote file to the local file, starting at offset 1024.

The same thing works for remote files!

This feature has a number of interesting uses, for example parallel clients reading different parts of the same remote file, multiple server/client pairs for added throughput, etc.

4.2.4 Encryption and compression

The cp command supports the "-E" and "-C" options, which enable data encryption and compression (during transfer) respectively. These work only if a single data stream is used.

Data encryption uses a symmetric algorithm, which nonetheless drastically lowers the performance.

Data compression uses the gzip algorithm.

Compression and encryption can be combined.

4.3 Synchronizing a file: the "sync" command

Currently, sync only supports single files, i.e. no directories or wildcards! The syntax is

```
$> uftp sync <master> <slave>
```

For example, to synchronize a local file with a remote "master" file:

```
$> uftp sync uftp://demo@localhost:9000/rest/auth/TEST:/master.file ↔  
local.file
```

To synchronize a remote file with a local "master" file:

```
$> uftp sync master.file uftp://demo@localhost:9000/rest/auth/TEST ↔  
:/remote.file
```

5 Troubleshooting

This section lists and explains common error messages and their resolution.

5.1 I get "Invalid server response 500" and "Exception... Authentication failure"

Probably you gave a wrong username or password. Contact your site administrator if in doubt!

If using a password, make sure you give the "-P" flag.

6 Getting support

UNICORE Website: <http://www.unicore.eu>

Support list: unicore-support@lists.sf.net

Developer's list: unicore-devel@lists.sf.net