
Subject: Re: FTP class and reference example for U++
Posted by [Oblivion](#) on Wed, 14 May 2014 11:43:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Mirek,

While it is far from perfect, and I'm not sure if the below mentioned features count as advantages, aside from the basic operations, but here they are:

- 1) Ftp class is completely based on native Upp classes. It is derived from TcpSocket. Therefore supports its features (except asynchronous connections, due to FTP protocols synchronic nature). This makes bug tracking and understanding the source code easy (also ftp class source code is around 600 lines, and meant to be clean and concise). No external lib, no C code, no need for external memory management, etc..
- 2) It comes with a directory entry and file information parser (FtpDirEntry class) which makes parsing (both DOS and UNIX based directory listing) as simple as it can be (of course getting the raw Listing strings is also possible (see CLI based ftp example).
- 3) It is designed the U++ GUI in mind; For example, FtpDirEntry class perfectly fits with FileList class (see the supplied ftp browser example). and achieving a non-blocking gui is very easy.
- 4) Ftp class does work on U++ streams (any kind), not simply strings (so it can take advantage of FileIn or FileOut streams, without any other effort, which means reduced code).
- 5) Ftp class api can be extended via SendCommand method (see CLI based ftp example) by the user/developer (there are many useful FTP extension commands out there, added later).
- 6) Ftp class is not just based on RFC 959. It also takes into account some practical deviations (D. J. Bernstein's recommendations: <http://cr.yp.to/ftp.html>) from the original draft.
- 7) Proper support for aborting transfers.
- 8) Ftp class will support IPV6 (actually It already does in development branch, and I will merge it soon).
- 9) Ftp class will support proxies.
- 10) I am also planning to add SSL/TLS support.
- 11) It is documented. (API document is complete, and I am currently writing a document on its mode of operations).
- 12) It is actively developed.

I believe the code is in good shape (after all, it is in beta state), but there is always plenty of room for improvement, and bugs that I don't know of yet.

Regards.
