

---

Subject: Re: FTP class and reference example for U++  
Posted by [Klugier](#) on Sat, 15 Apr 2017 17:23:40 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hello,

It is good to hear you introduced my proposals :)

---

I see one minor things to improvement - you could change the parameter name from whenwait to whenWait it will be more readable.

```
DirEntry&    User(const String& u)                { vm(USER) = u; return *this; }
```

Should it be SetUser(const String& u)? Moreover the u parameter is misleading for the API consumer it should be explicitly called user.

---

You should also consider moving your enums to c++ style:

```
enum Style    { UNIX, DOS };
```

```
enum class Type { UNIX, DOS }; // You could consider chaining the name to Type
```

```
// The user can refere:
```

```
DirEntry::Type::UNIX
```

instead of:

```
DirEntry::UNIX
```

---

Do we rely need this friendship declaration here?

```
ValueMap      vm;  
Bits          owner, group, other;  
friend bool    ParseFtpDirEntry(const String& in, Vector<Ftp::DirEntry>& out);
```

Why not just create

```
const ValueMap& GetVm(); // Read only access - i believe ParseFtpDirEntry doesn't set anything?
```

IMO the variable name "vm" is also misleading. I would call it dirInfo or something more verbose. The same improvment can be used in Result class - just call it resultInfo.

---

(Cosmetic - tiny) The class beginning has got problem with readability. It could be:

```
class Ftp : private NoCopy {  
public: // Just remove extra empty line.  
    class Request;  
    class Result; // Do we need forward declarations in class? I believe when we remove friendship  
we will not ;)  
  
    class DirEntry final : Moveable<Ftp::DirEntry> {
```

Sincerely,  
Klugier

---