

---

Subject: Re: Little .ini reading request

Posted by [mirek](#) on Fri, 08 Jan 2010 17:13:03 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

mrjt wrote on Fri, 08 January 2010 07:49The current Ini file reading util function is very handy, but it is unable to cope with sections that some people put in .Ini files. These are usually delimited by [SECTION\_NAME], but people have nasty habit of just making it up as they go along

I have a requirement to read such a file and wondered whether the following minor change could be committed for me so that I don't have to duplicate code:

Util.cpp:

```
VectorMap<String, String> LoadIniFile(const char *filename) {  
    FileIn in(filename);  
    if(!in) return VectorMap<String, String>();  
    return LoadIniStream(in);  
}
```

```
VectorMap<String, String> LoadIniStream(Stream &in) {  
    VectorMap<String, String> key;  
    int c;  
    if((c = in.Get()) < 0) return key;  
    for(;;) {  
        String k, v;  
        for(;;) {  
            if(IsAlNum(c) || c == '_')  
                k.Cat(c);  
            else  
                break;  
            if((c = in.Get()) < 0) return key;  
        }  
        for(;;) {  
            if(c != '=' && c != ' ') break;  
            if((c = in.Get()) < 0) return key;  
        }  
        for(;;) {  
            if(c < ' ') break;  
            v.Cat(c);  
            if((c = in.Get()) < 0) break;  
        }  
        if(!k.IsEmpty())  
            key.Add(k, v);  
        if(k == "LINK") {  
            in.Close();  
            if(!in.Open(v) || (c = in.Get()) < 0) return key;  
        }  
        else  
            for(;;) {
```

```
    if(IsAlNum(c) || c == '_') break;
    if((c = in.Get()) < 0) return key;
}
}
}
```

Util.h:

```
VectorMap<String, String> LoadIniFile(const char *filename);
```

```
VectorMap<String, String> LoadIniStream(Stream &in);
```

This means that I can just read .Ini file to correct section myself and then call LoadIniStream.

Cheers!

Why not.

BTW, what about adding some "real" support. Maybe we can detect sections and create keys like "SECTION.KEY"?

Mirek

---