

---

Subject: GetCsvLine

Posted by [Upp\\_User](#) on Tue, 12 Jun 2018 11:20:29 GMT

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

---

Hi,

Can someone give a small example of implementing GetCsvLine , i have to phrase a csv file. I believe GetCsvLine is the suggested option and i am not sure how to implement it.

Appreciate your timely help.

The csv file can be

Number\_of\_Lines,3

Line1,L1data1,L1data2,L1data3

Line2,L2data1,L2data2,L2data3

Line3,L3data1,L3data2,L3data3

Thanks

Upp\_User.

---

Subject: Re: GetCsvLine

Posted by [Oblivion](#) on Tue, 12 Jun 2018 12:21:26 GMT

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

---

Hello Upp User,

```
#include <Core/Core.h>
```

```
using namespace Upp;
```

```
CONSOLE_APP_MAIN
```

```
{  
    const String csvstr = "data1,data2,data3;data4,data5,data6;data7,data8,data9";
```

```
    stringstream ss(csvstr); // Any Stream derivative will also do (e.g. FileIn)
```

```
    auto v = GetCsvLine(ss, ';', CHARSET_DEFAULT);
```

```
    for(auto e : v)
```

```
        Cout() << e << '\n';
```

```
}
```

Best regards,

Oblivion

---

---

Subject: Re: GetCsvLine

Posted by [Upp\\_User](#) on Thu, 14 Jun 2018 08:41:39 GMT

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

---

Thanks a lot Oblivion, I am yet to try, i will keep posted on the outcome.

Best Regards

Upp User.

---

---

Subject: Re: GetCsvLine

Posted by [bozero](#) on Tue, 18 Dec 2018 11:20:04 GMT

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

---

How to parse empty string or data between ',' and '\n'.

Example:

```
stringstream ss;  
vector<string> vs;
```

```
string csvstr = "data1,data2,data3\n" data4,data5,data6\n" data7,data8,data9";  
DUMP(csvstr);
```

```
ss.open(csvstr);  
while(!ss.is_eof()){  
    vs = GetCsvLine0(ss, ',', CHARSET_DEFAULT);  
    DUMP(vs);  
    DUMP(vs.get_count());  
};
```

```
csvstr.replace("data3", "");  
csvstr.replace("data6", "\\");  
csvstr.replace("data9", "\\,"); // insert a extra comma
```

```
DUMP(csvstr);
```

```
ss.open(csvstr);  
while(!ss.is_eof()){  
    vs = GetCsvLine0(ss, ',', CHARSET_DEFAULT);  
    DUMP(vs);  
    DUMP(vs.get_count());  
};
```

Outcome:

```
csvstr = data1,data2,data3
```

```
data4,data5,data6  
data7,data8,data9
```

```
vs = [data1, data2, data3]  
vs.GetCount() = 3
```

```
vs = [data4, data5, data6]  
vs.GetCount() = 3
```

```
vs = [data7, data8, data9]  
vs.GetCount() = 3
```

```
csvstr = data1,data2,  
data4,data5,""  
data7,data8,"",
```

```
vs = [data1, data2]  
vs.GetCount() = 2
```

```
vs = [data4, data5]  
vs.GetCount() = 2
```

```
vs = [data7, data8,]  
vs.GetCount() = 3
```

---

Subject: Re: GetCsvLine  
Posted by [Oblivion](#) on Tue, 18 Dec 2018 12:00:29 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hello bozero,

There are many ways to achive this, but given your example, the below code would hopefully give you the basic idea:

```
#include <Core/Core.h>  
  
using namespace Upp;  
  
CONSOLE_APP_MAIN  
{  
    const String csvstr = "data1,data2,data3\ndata4,data5,data6\ndata7,data8,data9";  
  
    auto lines = Split(csvstr, '\n');  
  
    for(int i = 0; i < lines.GetCount(); i++) {
```

```

String line = lines[i];
Cout() << "Line: " << i + 1 << ": \n";
stringstream ss(line);
auto v = GetCsvLine(ss, ',', CHARSET_DEFAULT);
for(auto e : v)
    Cout() << e << '\n';
}
}

```

Another simple way:

```

#include <Core/Core.h>

using namespace Upp;

CONSOLE_APP_MAIN
{
    const String csvstr = "data1,data2,data3\ndata4,data5,data6\ndata7,data8,data9";

    stringstream ss(csvstr);

    int ln = 0;
    while(!ss.IsEof()) {
        auto s = ss.GetLine();
        stringstream line(s);
        auto v = GetCsvLine(line, ',', CHARSET_DEFAULT);
        Cout() << "Line: " << ++ln << "\n";
        for(auto e : v)
            Cout() << e << '\n';
    }
}

```

These output:

```

Line: 1:
data1
data2
data3
Line: 2:
data4
data5
data6
Line: 3:
data7

```

data8  
data9

Best regards,  
Oblivion

---

---

Subject: Re: GetCsvLine  
Posted by [bozero](#) on Wed, 19 Dec 2018 08:31:27 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Thank you for your reply and nice example, Oblivion. But my focus on a string with zero length ("") before new line, GetCsvLine() seems just ignore it. Even though a pair double double quotation (""") is put as space holder between (,) and (\n). Shall I need to put extra (,) to end of the each line of CSV before call GetCsvLine ?

```
void getcsvline_vs_split()
{
    StdLogSetup(LOG_COUT|LOG_FILE);
    const String csvstr = "data1,data2,\"data3\"\\ndata4,data5,\"\"\\ndata7,data8,\"\" \";
    DUMP(csvstr);
```

```
    Cout()<<"*** Get data via GetCSVLine ***\\n";
    stringstream ss(csvstr);
    int ln = 0;
    int total=0;
    while(!ss.IsEof()) {
        auto s = ss.GetLine();
        stringstream line(s);
        auto v = GetCsvLine(line, ',', CHARSET_DEFAULT);
        Cout() << "Line: " << ++ln << "\\n";
        for(auto e : v)
            Cout() << "Item " << ++total << " : " << e << '\\n';
    }
    Cout()<< "Total line: " << ln << "\\nTotal item: " << total << "\\n";
```

```
    Cout()<<"*** Get data via Split ***\\n";
    ln = 0;
    total=0;
    ss.Open(csvstr);
    while(!ss.IsEof()) {
        auto s=Split(ss.GetLine(),',');
        Cout() << "Line: " << ++ln << "\\n";
        for(auto e : s)
            Cout() << "Item " << ++total << " : " << e << '\\n';
    }
    Cout()<< "Total line: " << ln << "\\nTotal item: " << total << "\\n";
```

```
}
```

output:

```
csvstr = data1,data2,"data3" // string with ""  
data4,data5,""           // empty string  
data7,data8,"",         // extra comma at the end of line
```

```
*** Get data via GetCSVLine ***
```

```
Line: 1
```

```
Item 1 : data1
```

```
Item 2 : data2
```

```
Item 3 : data3 // string with "" is stripped
```

```
Line: 2
```

```
Item 4 : data4
```

```
Item 5 : data5 // expected empty string item "" is miss
```

```
Line: 3
```

```
Item 6 : data7
```

```
Item 7 : data8
```

```
Item 8 : // extra comma append for empty string item ""
```

```
Total line: 3
```

```
Total item: 8
```

```
*** Get data via Split *** // just compare with split
```

```
Line: 1
```

```
Item 1 : data1
```

```
Item 2 : data2
```

```
Item 3 : "data3" // string with ""
```

```
Line: 2
```

```
Item 4 : data4
```

```
Item 5 : data5
```

```
Item 6 : "" // string with ""
```

```
Line: 3
```

```
Item 7 : data7
```

```
Item 8 : data8
```

```
Item 9 : "" // string with ""
```

```
Total line: 3
```

```
Total item: 9
```