- Dream.In.Code> Programming Tutorials
- > C++ TutorialsPage 1 of 1

Stringstream tutorial Using a stringstream to read input from a csv file Rate Topic: ★ ★ ★ ★ ★ 2 Votes

r.stiltskin
Posted 28 March 2009 - 12:17 AM

You can think of a stringstream as a file loaded into something resembling a string, or alternatively, as a sort of string that you can write to and read from like a file. It's not exactly either of those things, but read on and it should become clearer...

A stringstream works essentially the same as an input/output file stream. You need the preprocessor directive #include <sstream>, declare a stringstream just like an fstream, for example
stringstream ss;
and, like an fstream or cout, you can write to it:
ss << myString; or
ss << myCstring; or
ss << myInt;, or float, or double, etc.

and you can read from it:
ss >> myChar; or
ss >> myCstring; or
ss >> myInt; This is also an easy way to convert strings of digits into ints, floats or doubles.

You can get the entire contents of the stringstream as a single C++ string:
string s = ss.str();

And it also inherits many other members from istream and ostream like get, getline, read, write, put, ... In this code I used the stringstream as an intermediate step between an input (text) file and an int array to help deal with the fact that the data within each line of the file was separated by commas, but the lines are separated only by newlines.

Following is a sample program using a stringstream to read numbers from a csv file named "input.txt" into a 6 row by 5 column int array and then prints the array.

It can take data from an input (text) file that looks like this:

```
1  1,3,10,3,1
2  1,10,10,3,10
3  1,,,,
4  0,0,0,1,10
5  1,1,0,2,2
6  10,1,0,1,10
```

and this is what ends up in the array:

```
1  1 3 10 3 1
2  1 10 10 3 10
3  1 0 0 0 0
4  0 0 0 1 10
5  1 1 0 2 2
6  10 1 0 1 10
```

C++ Tutorials

- **Network programming under UNIX**
- **Implementation Inheritance**
- **Change Theme in Code::Blocks**
- **A New Webcam Api Tutorial in C++ for Windows**
- **External Sorting with C**
- **C++ Windows Charting Library Part 6**
- **Bezier Curves Part 1 [Linear Algebra Series]**
- **STL Algorithms ~ Tutorial 1: Using sort()**
- **C++ Windows Charting Library Part 5**
- **C++ Windows Charting Library Part 4**
- **342 More C++**

General Discussion ▼

- ▶ Caffeine Lounge
- ▶ Corner Cubicle
- ▶ Student Campus
- ▶ Software Development
- ▶ Industry News
- ▶ Introduce Yourself
- ▶ Nightmare.In.Code

Programming Help ▼

- ▶ C and C++
- ▶ VB.NET
- ▶ Java
- ▶ C#
- ▶ ASP.NET
- ▶ .NET Framework
- ▶ VB6
- ▶ PHP
- ▶ Ruby
- ▶ Python
- ▶ ColdFusion
- ▶ Databases
- ▶ Other Languages
- ▶ Game Development
- ▶ Mobile Development
- ▶ 52 Weeks Of Code

Web Development ▼

- ▶ Web Development

It uses both forms of the istream getline member.
Both of them take characters from the stream,
copy them into a char array and terminate them
with a '\0'.
The first one reads up to n-1 characters or until it
reaches a newline or eof:
istream& getline (char* s, streamsize n);
The second one reads up to n-1 characters or until
it reaches the character specified as 'delim' or eof:
istream& getline (char* s, streamsize n, char delim);

```
01  #include <iostream>
02  #include <fstream>
03  #include <sstream>
04  using namespace std;
05
06  const int ROWS = 6;
07  const int COLS = 5;
08  const int BUFFSIZE = 80;
09
10  int main() {
11    int array[ROWS][COLS];
12    char buff[BUFFSIZE]; // a buffer to
    temporarily park the data
13    ifstream infile("input.txt");
14    stringstream ss;
15    for( int row = 0; row < ROWS; ++row )
    {
16      // read a full line of input into
    the buffer (newline is
17      //  automatically discarded)
18      infile.getline( buff,  BUFFSIZE );
19      // copy the entire line into the
    stringstream
20      ss << buff;
21      for( int col = 0; col < COLS; ++col
    ) {
22        // Read from ss back into the
    buffer.  Here, ',' is
23        //  specified as the delimiter so
    it reads only until
24        //  it reaches a comma (which is
    automatically
25        //  discarded) or reaches the
```

The above code was set up explicitly to load data
into a 6x5 array because that's what a particular
application required, but it can be easily modified to
read an entire input file of indeterminate length
using a while loop, i.e.
while(infile.getline(buff, 50)) , containing lines with
any number of comma-separated values (as long
as a big enough buffer is provided). Here's an
example of that (it only prints the first 10 columns
of the first 10 rows of the array so as not to fill
your screen with 0's:

Reference Sheets

Code Snippets

- **C Snippets**
- **C++ Snippets**
- **Java Snippets**
- **Visual Basic Snippets**
- **C# Snippets**
- **VB.NET Snippets**
- **ASP.NET Snippets**
- **PHP Snippets**
- **Python Snippets**
- **Ruby Snippets**
- **ColdFusion Snippets**
- **SQL Snippets**
- **Assembly Snippets**
- **Functional Programming Snippets**
- **Perl Snippets**
- **HTML/CSS Snippets**
- **Javascript Snippets**
- **Flash/ActionScript Snippets**
- **ASP Snippets**
- **Linux, Unix, and Bash Snippets**
- **Other Languages Snippets**
- **Regex**

Tutorials...

▶ HTML & CSS
▶ JavaScript
▶ Graphic Design
▶ Flash & ActionScript
▶ Blogging
▶ SEO & Advertising
▶ Web Servers & Hosting
▶ Site Check

```
01  #include <iostream>
02  #include <fstream>
03  #include <sstream>
04  using namespace std;
05
06  const int ROWS = 100;
07  const int COLS = 80;
08  const int BUFFSIZE = 80;
09
10  int main() {
11    // The array must be big enough to
      fit the input data.
12    int array[ROWS][COLS] = {0};
13    int row, col;
14    char buff[BUFFSIZE]; // a buffer to
      temporarily park the data
15    ifstream infile("input.txt");
16    stringstream ss;
17    // Read input into the buffer a line
      at a time, until
18    //  end of file is reached (newlines
      are automatically
19    //  discarded).  The buffer must be
      big enough to fit
20    //  an entire line from the file.
21    // Notice that while reading from the
      file we check how
22    //  many rows have been read, to
      avoid writing beyond
23    //  the end of the array.
24    row = 0;
25    while( infile.getline( buff,
      BUFFSIZE ) && row < ROWS ) {
```

This post has been edited by **r.stiltskin**: 02 April 2009 -
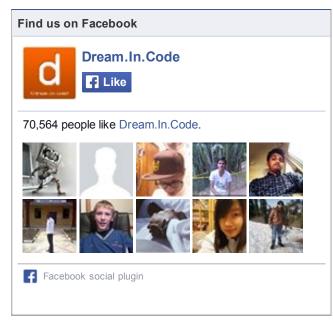10:42 PM

Replies To: Stringstream tutorial

markkenzy
Posted 15 April 2011 - 08:00 PM
  Nice tuts. Thank you, guy!
Page 1 of 1

Related C++ Topics^{beta}

🗀 **Is There A Substitute
   For Stringstream
   That Doesn't Return
   A Const**

🗀 **Print To Beginning Of
   Deque**

🗀 **Need Help In
   Displaying Words
   Non Similar Words
   In String**

- **Encrypting And Decrypting Text File.**

- **File Extraction To Variables**

- **Converting A String Of Numbers(Characters) Into Characters(ASCII)**

- **Cannot Read In Next Line From Data File**

- **C++ Basic Program Help Need - There Is This Problem Regarding Length Of String...**

- **Pig Latin Converter C++ - Problem With Converting Whole Sentences**

- **How To Read .csv Files In C++?**

FAQ | **Team Blog** | **Feedback/Support** | **Advertising** | **Terms of Use** | **Privacy Policy** | **About Us**