

```
1  /*
2  *
3  * Copyright (c) 2003
4  * John Maddock
5  *
6  * Use, modification and distribution are subject to the
7  * Boost Software License, Version 1.0. (See accompanying file
8  * LICENSE_1_0.txt or copy at http://www.boost.org/LICENSE_1_0.txt)
9  *
10 /*
11
12 /*
13 *   LOCATION:    see http://www.boost.org for most recent version.
14 *   FILE        regex_token_iterator_example_1.cpp
15 *   VERSION      see <boost/version.hpp>
16 *   DESCRIPTION: regex_token_iterator example: split a string into tokens
17 */
18
19
20 #include <boost/regex.hpp>
21
22 #include <iostream>
23 using namespace std;
24
25
26 #if defined(BOOST_MSVC) || (defined(__BORLANDC__) && (__BORLANDC__ == 0x550))
27 //
28 // problem with std::getline under MSVC6sp3
29 istream& getline(istream& is, std::string& s)
30 {
31     s.erase();
32     char c = static_cast<char>(is.get());
33     while(c != '\n')
34     {
35         s.append(1, c);
36         c = static_cast<char>(is.get());
37     }
38     return is;
39 }
40 #endif
41
42
43 // int main(int argc, const char*[])
44 // updated by JMH on December 5, 2012 at 12:21 PM to allow command line input
45 // note that if the command line input contains white space, it must be enclosed
46 // in double quotes
47 int main(int argc, char** argv)
48 {
49     string s;
50     do{
51         if ( argc == 1 )
52         {
53             cout << "Enter text to split (or \"quit\" to exit): ";
54             getline(cin, s);
55             if(s == "quit") break;
56         }
57         else {
58             // s = "This is a string of tokens";
59             s = argv[1] ;
60             // cout << argc << " " << s << endl ;
61         }
62
63         boost::regex re("\\s+");
64
65         // the -1 parameter below causes the token iterator to consider as tokens
66         // those parts of the input that do *not* match the regex
67         // such an iterator, if dereferenced, returns a match_results corresponding to
```

```
68     // the sequence of characters between the last match and the end of sequence
69     // see http://boost-sandbox.sourceforge.net/libs/xpressive/doc/html/boost\_xpressive/user\_s\_guide/string\_sp1
70     // and http://en.cppreference.com/w/cpp/regex/regex\_token\_iterator
71     boost::sregex_token_iterator i(s.begin(), s.end(), re, -1);
72
73     // the default-constructed sregex_token_iterator is the end-of-sequence iterator
74     // see http://en.cppreference.com/w/cpp/regex/regex\_token\_iterator
75     boost::sregex_token_iterator j;
76
77     unsigned count = 0;
78     while(i != j)
79     // while(i != s.end()) yields compiler error: no match for operator!=
80     {
81         cout << *i++ << endl;
82         count++;
83     }
84     cout << "There were " << count << " tokens found." << endl;
85
86 }while(argc == 1);
87 return 0;
88 }
```