

Boost HTML Library

Contributed by Andreas Haberstroh
 Monday, 03 September 2007
 Last Updated Wednesday, 27 February 2008

Release of the Boost HTML library, a cross-platform C++ library for reading and writing HTML documents.
 License

Boost-HTML is released under the Boost Software License. Dependencies

Boost::HTML requires Boost version 1.33 or later. Downloading

The Boost-HTML library is available here. Using Boost-HTML

The Boost-HTML library is relatively simplistic in design. All read and write operations are based on streams.

```
#include <boost/html/document.hpp>
#include <iostream>#include <fstream>
using namespace std;
int main(int argc, char* argv[])
{
    fstream ifile("index.html", ios_base::in);
    boost::html::document htmlDOM; htmlDOM.read(ifile); htmlDOM.write(cout);

    return 0;
}
}Why did I write Boost-HTML?
```

I was searching the Internet for a c++ HTML parser library that would create a tree of HTML elements and their attributes. I was looking for an c++ interface similar to JavaScript's DOM. I found plenty of XML parsers (Boost-Property Tree Library, Xerces) and a few HTML parsers that spat out information in a SAX like manner. I just simply couldn't find a C++ library that did what I wanted.

So, I did some reading of the W3C's Core DOM specification and a few other sources and decided to write my own HTML parser. The IDL interface in the Core DOM specification gave me the starting point for my c++ implementation, and that is where I started. Boost-HTML Library is simple, no frills, since it was a Labor Day weekend project. I can read in a document and write it back, almost the same as I read it. I say almost the same, because I decided to use the XHTML notation for elements that are singular in nature, for instance
.

With this library, you can also pull out a list of elements by searching for a tag name or a tag id attribute. That was really the main feature I was looking for, pull out all the <a> tags from a document.

Hopefully, you'll find this library useful and tell me about it.

Better yet, if you find a glaring bug, I hope you'll tell me about that too! What Boost Libraries are Used ?

I've used a few of my favorite Boost libraries: shared_ptr

Pretty much every element object is encapsulated as a shared_ptr. I like automagic clean up multi_index Attributes and element references are placed in multi_index_containers. Makes life easy to look up objects.hashEasier to search by a hash, and that is just what I do!

string_algo Pretty cool library giving Perl like functionality for strings. I personally use the join function a few times.