TagSoup - Just Keep On Truckin'

Index

Introduction

This is the home page of TagSoup, a SAX-compliant parser written in Java that, instead of parsing well-formed or valid XML, parses HTML as it is found in the wild: poor, nasty and brutish, though quite often far from short. TagSoup is designed for people who have to process this stuff using some semblance of a rational application design. By providing a SAX interface, it allows standard XML tools to be applied to even the worst HTML. TagSoup also includes a command-line processor that reads HTML files and can generate either clean HTML or well-formed XML that is a close approximation to XHTML.

This is also the README file packaged with TagSoup.

TagSoup is free and Open Source software. As of version 1.2, it is licensed under the Apache License, Version 2.0, which allows proprietary re-use as well as use with GPL 3.0 or GPL 2.0-or-later projects. (If anyone needs a GPL 2.0 license for a GPL 2.0-only project, feel free to ask.)

The TagSoup logo is courtesy Ian Leslie.

TagSoup 1.2.1 released

TagSoup 1.2.1 is a much-delayed bug fix release. The following bugs have hopefully been repaired:

- DOCTYPE is now recognized even in lower case.
- We make sure to buffer the reader, eliminating a long-standing bug that would crash on certain inputs, such as & followed by CR+LF.
- The HTML scanner's table is precompiled at run time for efficiency, causing a 4x speedup on large input documents.
- `]]` within a CDATA section no longer causes input to be discarded.
- Remove bogus newline after printing children of the root element.
- Allow the `noscript` element anywhere, the same as the `script` element.
- Updated to the 2011 edition of the W3C character entity list.

Download the TagSoup 1.2.1 jar file here. It's about 89K long.
Download the full TagSoup 1.2.1 source here. If you don't have zip, you can use jar to unpack it.
Download the current CHANGES file here.

Taggle, a TagSoup in C++, available now

A company called JezUK has released Taggle, which is a straight port of TagSoup 1.2 to C++. It's a part of Arabica, a C++ XML toolkit providing SAX, DOM, XPath, and partial XSLT. I have no connection with JezUK (except apparently as source of inspiration).

The author says the code is alpha-quality now, so he'd appreciate lots of testers to shake out bugs. C++ users, go to it! Having a C++ port will be a real enhancement for TagSoup.

The code is currently in public Subversion: you can fetch it with

```
svn co svn://jezuk.dnsalias.net/jezuk/arabica/branches/tagsoup-port
```

.

TagSoup 1.2 released

There are a great many changes, most of them fixes for long-standing bugs, in this release. Only the most important are listed here; for the rest, see the CHANGES file in the source distribution. Very special thanks to Jojo Dijamco, whose intensive efforts at debugging made this release a usable upgrade rather than a useless mass of undetected bugs.

- As noted above, I have changed the license to Apache 2.0.
- The default content model for bogons (unknown elements) is now ANY rather than EMPTY. **This is a breaking change**, which I have done only because there was so much demand for it. It can be undone on the command line with the `--emptybogons` switch, or programmatically with `parser.setFeature(Parser.emptyBogonsFeature, true)`.
- The processing of entity references in attribute values has finally been fixed to do what browsers do. That is, a reference is only recognized if it is properly terminated by a semicolon; otherwise it is treated as plain text. This means that URIs like `foo?cdown=32&cup=42` are no longer seen as containing an instance of the ∪ character (whose name happens to be `cup`).
- Several new switches have been added:
  - `--doctype-system` and `--doctype-public` force a `DOCTYPE` declaration to be output and allow setting the system and public identifiers.
  - `--standalone` and `--version` allow control of the XML declaration that is output. (Note that TagSoup's XML output is always version 1.0, even if you use `--version=1.1`.)
  - `--norootbogons` causes unknown elements not to be allowed as the document root element. Instead, they are made children of the default root element (the `html` element for HTML).
- The TagSoup core now supports character entities with values above U+FFFF. As a consequence, the HTML schema now supports all 2,210 standard character entities from the 2007-12-14 draft of XML Entity Definitions for Characters, except the 94 which require more than one Unicode character to represent.
- The SAX events `startPrefixMapping` and `endPrefixMapping` are now being reported for all cases of foreign elements and attributes.
- All bugs around newline processing on Windows should now be gone.

- A number of content models have been loosened to allow elements to appear in new and non-standard (but commonly found) places. In particular, tables are now allowed inside paragraphs, against the letter of the W3C specification.
- Since the `span` element is intended for fine control of appearance using CSS, it should never have been a restartable element. This very long-standing bug has now been fixed.
- The following non-standard elements are now at least partly supported: `bgsound`, `blink`, `canvas`, `comment`, `listing`, `marquee`, `nobr`, `rbc`, `rb`, `rp`, `rtc`, `rt`, `ruby`, `wbr`, `xmp`.
- In HTML output mode, boolean attributes like `checked` are now output as such, rather than in XML style as `checked="checked"`.
- Runs of < characters such as << and <<< are now handled correctly in text rather than being transformed into extremely bogus start-tags.

What TagSoup does

TagSoup is designed as a parser, not a whole application; it isn't intended to permanently clean up bad HTML, as HTML Tidy does, only to parse it on the fly. Therefore, it does not convert presentation HTML to CSS or anything similar. It does guarantee well-structured results: tags will wind up properly nested, default attributes will appear appropriately, and so on.

The semantics of TagSoup are as far as practical those of actual HTML browsers. In particular, never, never will it throw any sort of syntax error: the TagSoup motto is "Just Keep On Truckin'". But there's much, much more. For example, if the first tag is LI, it will supply the application with enclosing HTML, BODY, and UL tags. Why UL? Because that's what browsers assume in this situation. For the same reason, overlapping tags are correctly restarted whenever possible: text like:

```
This is <B>bold, <I>bold italic, </b>italic, </i>normal text
```

gets correctly rewritten as:

```
This is <b>bold, <i>bold italic, </i></b><i>italic, </i>normal text.
```

By intention, TagSoup is small and fast. It does not depend on the existence of any framework other than SAX, and should be able to work with any framework that can accept SAX parsers. In particular, XOM is known to work.

You can replace the low-level HTML scanner with one based on Sean McGrath's PYX format (very close to James Clark's ESIS format). You can also supply an AutoDetector that peeks at the incoming byte stream and guesses a character encoding for it. Otherwise, the platform default is used. If you need an autodetector of character sets, consider trying to adapt the Mozilla one; if you succeed, let me know.

The TSaxon XSLT-for-HTML processor

I am also distributing TSaxon, a repackaging of version 6.5.5 of Michael Kay's Saxon XSLT version 1.0 implementation that includes TagSoup. TSaxon is a drop-in replacement for Saxon, and can be used to process either HTML or XML documents with XSLT stylesheets.

Note: TagSoup in Java 1.1

If you go through the TagSoup source and replace all references to `HashMap` with `Hashtable` and recompile, TagSoup will work fine in Java 1.1 VMs. Thanks to Thorbjørn Vinne for this discovery.

Warning: TagSoup will not build on stock Java 5.x or 6.x!

Due to a bug in the versions of Xalan shipped with Java 5.x and 6.x, TagSoup will not build out of the box. You need to retrieve Saxon 6.5.5, which does not have the bug. Unpack the zipfile in an empty directory and copy the `saxon.jar` and `saxon-xml-apis.jar` files to `$ANT_HOME/lib`. The Ant build process for TagSoup will then notice that Saxon is available and use it instead.

In addition, if you are building on a Debian-derived distro, you will need to install not only the ant package but the ant-optional package as well.

TagSoup as a stand-alone program

It is possible to run TagSoup as a program by saying `java -jar tagsoup-1.2.1 [`*`option ...`*`] [`*`file ...`*`]`. Files mentioned on the command line will be parsed individually. If no files are specified, the standard input is read.

The following options are understood:

`--files`
: Output into individual files, with `html` extensions changed to `xhtml`. Otherwise, all output is sent to the standard output.

`--html`
: Output is in clean HTML: the XML declaration is suppressed, as are end-tags for the known empty elements.

`--omit-xml-declaration`
: The XML declaration is suppressed.

`--method=html`
: End-tags for the known empty HTML elements are suppressed.

`--doctype-system=`*`systemid`*
: Forces the output of a `DOCTYPE` declaration with the specified systemid.

`--doctype-public=`*`publicid`*
: Forces the output of a `DOCTYPE` declaration with the specified publicid.

`--version=`*`version`*
: Sets the version string in the XML declaration.

`--standalone=`[yes|no]
: Sets the standalone declaration to yes or no.

`--pyx`
: Output is in PYX format.

`--pyxin`
: Input is in PYXoid format (need not be well-formed).

`--nons`
: Namespaces are suppressed. Normally, all elements are in the XHTML 1.x namespace, and all attributes are in no namespace.

`--nobogons`
: Bogons (unknown elements) are suppressed.

`--nodefaults`
: Default attribute values are suppressed.

`--nocolons`
: Explicit colons in element and attribute names are changed to underscores.

`--norestart`
: don't restart any normally restartable elements.

`--ignorable`
: Output whitespace in elements with element-only content.

`--emptybogons`
: Bogons are given a content model of EMPTY rather than ANY.

`--any`

Bogons are given a content model of ANY rather than EMPTY (default).

`--norootbogons`
: Bogons are not allowed to be root elements; make them subordinate to the root.

`--lexical`
: Pass through HTML comments and DOCTYPE declarations. Has no effect when output is in PYX format.

`--reuse`
: Reuse a single instance of TagSoup parser throughout. Normally, a new one is instantiated for each input file.

`--nocdata`
: Change the content models of the `script` and `style` elements to treat them as ordinary #PCDATA (text-only) elements, as in XHTML, rather than with the special CDATA content model.

`--encoding=`*encoding*
: Specify the input encoding. The default is the Java platform default.

`--output-encoding=`*encoding*
: Specify the output encoding. The default is the Java platform default.

`--help`
: Print help.

`--version`
: Print the version number.

SAX features and properties

TagSoup supports the following SAX features in addition to the standard ones:

`http://www.ccil.org/~cowan/tagsoup/features/ignore-bogons`
: A value of "true" indicates that the parser will ignore unknown elements.

`http://www.ccil.org/~cowan/tagsoup/features/bogons-empty`
: A value of "true" indicates that the parser will give unknown elements a content model of EMPTY; a value of "false", a content model of ANY.

`http://www.ccil.org/~cowan/tagsoup/features/root-bogons`
: A value of "true" indicates that the parser will allow unknown elements to be the root of the output document.

`http://www.ccil.org/~cowan/tagsoup/features/default-attributes`
: A value of "true" indicates that the parser will return default attribute values for missing attributes that have default values.

`http://www.ccil.org/~cowan/tagsoup/features/translate-colons`
: A value of "true" indicates that the parser will translate colons into underscores in names.

`http://www.ccil.org/~cowan/tagsoup/features/restart-elements`
: A value of "true" indicates that the parser will attempt to restart the restartable elements.

`http://www.ccil.org/~cowan/tagsoup/features/ignorable-whitespace`
: A value of "true" indicates that the parser will transmit whitespace in element-only content via the SAX ignorableWhitespace callback. Normally this is not done, because HTML is an SGML application and SGML suppresses such whitespace.

`http://www.ccil.org/~cowan/tagsoup/features/cdata-elements`
: A value of "true" indicates that the parser will process the `script` and `style` elements (or any elements with `type='cdata'` in the TSSL schema) as SGML CDATA elements (that is, no markup is recognized except the matching end-tag).

TagSoup supports the following SAX properties in addition to the standard ones:

`http://www.ccil.org/~cowan/tagsoup/properties/scanner`
: Specifies the Scanner object this parser uses.

`http://www.ccil.org/~cowan/tagsoup/properties/schema`
: Specifies the Schema object this parser uses.

`http://www.ccil.org/~cowan/tagsoup/properties/auto-detector`
: Specifies the AutoDetector (for encoding detection) this parser uses.

Other TagSoups and related things

TagSoup is written in the world's finest imperative programming language, as opposed to my TagSoup, which is written in perhaps the world's most widely used imperative programming language. As far as I can make out, TagSoup only lexes its input, and does not attempt to balance tags in the style of my TagSoup.

BeautifulSoup is closer to my TagSoup, but is written in Python and returns a parse tree. I believe its heuristics are hard-coded for HTML. There is a port to Ruby called RubyfulSoup.

There are a variety of other HTML SAX parsers written in Java, notably NekoHTML, JTidy (a port of the C library and tool HTML Tidy), and HTML Parser. All have their good and bad points: the general view around the Web seems to be that TagSoup is the slowest, but also the most robust and reliable.

Finally, there is a full port of my TagSoup to C++, but unfortunately it is currently trapped inside IBM. The process to release it as Open Source is under way, and I hope to feature it here some time soon.

More information

I gave a presentation (a nocturne, so it's not on the schedule) at Extreme Markup Languages 2004 about TagSoup, updated from the one presented in 2002 at the New York City XML SIG and at XML 2002. This is the main high-level documentation about how TagSoup works. Formats: OpenDocument Powerpoint PDF.

I also had people add "evil" HTML to a large poster so that I could clean it up; View Source is probably more useful than ordinary browsing. The original instructions were:

SOUPE DE BALISES (BE EVIL)!
Ecritez une balise ouvrante (sans attributs)
ou fermante HTML ici, s.v.p.

There is a tagsoup-friends mailing list hosted at Google Groups. You can join via the Web, or by sending a blank email to *tagsoup-friends-subscribe@googlegroups.com*. The archives are open to all.

Online TagSoup processing for publicly accessible HTML documents is now available courtesy of Leigh Dodds.