

XML, The Perl Way

Welcome to the XML::Twig Page.



XML::Twig is a Perl module used to process efficiently XML documents

Twig offers a tree-oriented interface to a document while still allowing the processing of documents of any size. I think the current buzzword for it would be "push-pull" processing ;-))When I was younger I wanted to grow up and write a tool that would allow people to process text the way they wanted, offering tons of feature, various ways to achieve the same result, not forcing them into any processing model but allowing them to use the one they felt the most comfortable with. Eventually I grew up and I realized a guy named Larry Wall had already written a language named Perl... Darn! So as I was quite involved in dealing with SGML, then XML, I decided to settle for the next best thing: writing a module that would allow people to process XML the way they wanted, offering them tons of feature, various ways... you get the point. So I wrote XML::Twig. XML::Twig gives you a tree interface to XML documents... if you want. It also lets you dump parts of the tree, set callbacks during processing, both on tags and on subtrees, process only part of the tree, write powerful filters... you name it. The only thing XML::Twig does not do is follow standards (except XML of course): no DOM, no XPath (well, not quite, it implements a subset of XPath), no SAX (it is planned though, as of 3.05 you can output SAX using the toSAX1 and toSAX2 methods). Consider yourself warned!

Sources

Latest development versions (Thu Dec 24 18:48:22 2015)

- [XML-Twig-3.50.tar.gz](#)

Git repo: [mirod's xmlltwig master](#), on [github](#)

Latest stable version

- [XML-Twig-3.49.README](#)
- [XML-Twig-3.49.tar.gz](#)

[XML::Twig page on seach.cpan.org](#)

Mirrors on [CPAN](#)

Documentation

References

Development version: [html](#) or [text](#), [Changes](#)

Stable version: [html](#) or [text](#), [Changes](#)

The [XML::Twig Quick Reference Page](#)

You can also have a look at the [FAQ](#) (updated Sat Jun 16 06:32:56 2012)

The [development version documentation](#) is probably the easiest to use: it is better cross-referenced, methods are sorted alphabetically and you can expand/collapse method lists.

Tutorial

A complete [tutorial](#), covering XML::Twig basics, integrating XML documents with data bases, HTML conversion and more...

Note that I wrote it back in the days of version 1.10 or so so it does not include some of the latest additions to the module. It should get you started though!

Articles

The [Ways to Rome series of articles](#) has examples of how to use XML::Twig (along with **many** other modules).

Kip Hampton also wrote an very good article: [Using XML::Twig for xml.com](#).

Tom Anderson has an interesting article: [Performance Comparison Between SAX XML::Filter::Dispatcher and XML::Twig](#). I have improved slightly the XML::Twig version of his test, see [my comments](#).

Finally, I cannot really comment on [XML-Twig](#), by Stas Orlenko, an article in Russian ;--(

Talks

A 2 hour introduction to XML::Twig: [Growing Twigs](#), gives a generic discussion on XML processing, how to install XML::Twig, code examples and a quick history of the module.

Podcast

Finally, you can hear me talk about my favourite sujet in an interview with Josh McAdams on [perlcast: Michel Rodriguez on XML::Twig](#).

Request tracking system:

You can file bug reports and view outstanding bugs throught [the XML::Twig bug list](#), using [rt.cpan.org](#)

Tools built with XML::Twig

- [xml_pp](#) is an XML pretty printer (or beautifier): [download](#) (needs XML::Twig 3.05 or above)
- [xml_spellcheck](#) runs a spell checker (aspell by default) on the text of XML files: [download](#)
- [beta]: [xmlgrep](#) greps files looking for a condition on the XML element or attributes (needs XML::Twig 3.10 or above): [download](#)
- [ooo2txt](#) reads a Star/Open Office file (only Writer is supported at the moment, .sxd/.stw files) and displays the text: [download](#). It is a rewrite using XML::Twig of the original [ooo2txt](#) by Frederic Labbe.

Test Coverage

The latest [coverage](#).

BTW, a good way to help the development of XML::Twig is to write more tests to increase this coverage. Look at [this source](#) and work on getting rid of uncovered branckes (the ones with red numbers). Just make sure the tests are portable accross various OSs and versions of Perl. Coverage is already pretty good but it is always better to increase it.

More (simple) Examples

All the examples: [twig_examples.tar.gz](#)

- [A simple example using a handler](#)
- [The example data](#)

A series of examples on a single xml file

The data is a list of perl modules in [ex_fm.xml](#). To run the examples: `ex_fm<n> ex_fm.xml`.

- [ex_fm1](#): creating a twig then extracting information from it.
- [ex_fm2](#): extracting information from the twig as the document is parsed for smaller memory usage.
- [ex_fm3](#): changing the document as it is being parsed.
- [ex_fm4](#): using `twig_roots` for maximum speed and minimum memory usage.
- [ex_fm5](#): creating elements and attributes.
- [ex_fm6](#): same as `ex_fm3` but uses `twig_roots` and `twig_print_outside_roots` for maximum efficiency.

An example using 2 documents and `$twig->finish_print` to speed up processing. To run it: `ex_insert insert_new<n>.xml insert_main.xml`

- [ex_insert](#): the script.
- [ex_insert_main.xml](#): the main file.
- [ex_insert_main.xml](#): new item 1.
- [ex_insert_main.xml](#): new item 2.
- [ex_insert_main.xml](#): new item 3.

And finally: [Examples of XML::Twig on Perlmonks](#), a list of my posts about XML::Twig on [PerlMonks](#).

Donations

If you want to support XML::Twig's development, I always appreciate anything from XML::Twig's wishlist on Amazon. Thanks.

Archives

- [XML-Twig-3.49.tar.gz](#)
- [XML-Twig-3.48.tar.gz](#)
- [XML-Twig-3.47.tar.gz](#)
- [XML-Twig-3.46.tar.gz](#)
- [XML-Twig-3.45.tar.gz](#)
- [XML-Twig-3.45.tar.gz](#)
- [XML-Twig-3.43.tar.gz](#)
- [XML-Twig-3.42.tar.gz](#)
- [XML-Twig-3.42.tar.gz](#)
- [XML-Twig-3.40.tar.gz](#)
- [XML-Twig-3.39.tar.gz](#)
- [XML-Twig-3.38.tar.gz](#)
- [XML-Twig-3.37.tar.gz](#)
- [XML-Twig-3.36.tar.gz](#)
- [XML-Twig-3.35.tar.gz](#)
- [XML-Twig-3.34.tar.gz](#)
- [XML-Twig-3.33.tar.gz](#)
- [XML-Twig-3.32.tar.gz](#)
- [XML-Twig-3.31.tar.gz](#)
- [XML-Twig-3.30.tar.gz](#)
- [XML-Twig-3.29.tar.gz](#)
- [XML-Twig-3.28.tar.gz](#)
- [XML-Twig-3.26.tar.gz](#)
- [XML-Twig-3.25.tar.gz](#)
- [XML-Twig-3.24.tar.gz](#)
- [XML-Twig-3.23.tar.gz](#)
- [XML-Twig-3.22.tar.gz](#)
- [XML-Twig-3.21.tar.gz](#)
- [XML-Twig-3.20.tar.gz](#)
- [XML-Twig-3.19.tar.gz](#)
- [XML-Twig-3.18.tar.gz](#)
- [XML-Twig-3.17.tar.gz](#)
- [XML-Twig-3.16.tar.gz](#)
- [XML-Twig-3.15.tar.gz](#)
- [XML-Twig-3.14.tar.gz](#)
- [XML-Twig-3.13.tar.gz](#)
- [XML-Twig-3.12.tar.gz](#)
- [XML-Twig-3.11.tar.gz](#)
- [XML-Twig-3.10.tar.gz](#)
- [XML-Twig-3.09.tar.gz](#)
- [XML-Twig-3.08.tar.gz](#)
- [XML-Twig-3.07.tar.gz](#)
- [XML-Twig-3.06.tar.gz](#)
- [XML-Twig-3.05.tar.gz](#)
- [XML-Twig-3.04.tar.gz](#)
- [XML-Twig-3.03.tar.gz](#)
- [XML-Twig-3.02.tar.gz](#)
- [XML-Twig-3.02.tar.gz](#)
- [XML-Twig-3.01.tar.gz](#)
- [XML-Twig-2.02.tar.gz](#)
- [XML-Twig-2.01.tar.gz](#)