

## NAME

MKDoc::XML::Tagger – Adds XML markup to XML / XHTML content.

## SYNOPSIS

```
use MKDoc::XML::Tagger;
print MKDoc::XML::Tagger->process_data (
    "<p>Hello, World!</p>",
    { _expr => 'World', _tag => 'strong', class => 'superFort' }
);
```

Should print:

```
<p>Hello, <strong class="superFort">World</strong>!</p>
```

## SUMMARY

MKDoc::XML::Tagger is a class which lets you specify a set of tag and attributes associated with expressions which you want to mark up. This module will then stuff any XML you send out with the extra expressions.

For example, let's say that you have a document which has the term 'Microsoft Windows' several times in it. You could wish to surround any instance of the term with a <trademark> tag. MKDoc::XML::Tagger lets you do exactly that.

In MKDoc, this is used so that editors can enter hyperlinks separately from the content. It allows them to enter content without having to worry about the annoying <a href="..."> syntax. It also has the added benefit from preventing bad information architecture such as the 'click here' syndrome.

We also have plans to use it for automatically linking glossary words, abbreviation tags, etc.

MKDoc::XML::Tagger is also probably a very good tool if you are building some kind of Wiki system in which you want expressions to be automatically hyperlinked.

## DISCLAIMER

**This module does low level XML manipulation. It will somehow parse even broken XML and try to do something with it. Do not use it unless you know what you're doing.**

## API

The API is very simple.

```
my $result = MKDoc::XML::Tagger->process_data ($xml, @expressions);
```

Tags \$xml with the @expressions list.

Each element of @expressions is a hash reference looking like this:

```
{
    _expr      => 'Some Expression',
    _tag       => 'foo',
    attribute1 => 'bar'
    attribute2 => 'baz'
}
```

Which will try to turn anything which looks like:

```
Some Expression
sOmE ExPrEssIoN
(etccetera)
```

Into:

```
<foo attr1="bar" attr2="baz">Some Expression</foo>
<foo attr1="bar" attr2="baz">sOmE ExPrEssIoN</foo>
<foo attr1="bar" attr2="baz">(etccetera)</foo>
```

You can have multiple expressions, in which case longest expressions are processed first.

```
my $result = MKDoc::XML::Tagger->process_file ('some/file.xml', @expressions);
```

Same as *process\_data()*, except it takes its data from 'some/file.xml'.

## NOTES

MKDoc::XML::Tagger does not really parse the XML file you're giving to it nor does it care if the XML is well-formed or not. It uses MKDoc::XML::Tokenizer to turn the XML / XHTML file into a series of



MKDoc::XML::Token objects and strictly operates on a list of tokens.

For this same reason MKDoc::XML::Tagger does not support namespaces.

**AUTHOR**

Copyright 2003 – MKDoc Holdings Ltd.

Author: Jean-Michel Hiver

This module is free software and is distributed under the same license as Perl itself. Use it at your own risk.

**SEE ALSO**

MKDoc::XML::Tokenizer MKDoc::XML::Token

