

# Writing a book using the Template Toolkit

Dave Cross

Perl Template  
Toolkit



# Writing A Book

- O'Reilly has wanted a book on the Template Toolkit for some time
- Andy Wardley wanted to do it but was very busy
- Apr 2002 - Nat asks Darren Chamberlain and me to help out
- Andy writes an outline, we discuss it a bit
- Time passes...
- Contracts signed Sep/Oct 2002
- Start writing soon afterwards

Perl Template  
Toolkit



# About Writing Books

- O'Reilly accept book submissions in POD (hurrah!)
- (Actually a slightly extended version of POD)
- Books are not written linearly
- Need to coordinate who is writing what when
- Different authors have different styles
- Contents are often rearranged
- Internal references need to be maintained
- Need to include example code

Perl Template  
Toolkit



# Solutions To Problems

- Use source code control (we used CVS)
- Each chapter in multiple files
- Need a tool that can pull all these files together into one file per chapter
- And also makes it easy to rearrange chapters
- And ensure that internal references all change appropriately
- And allows us to include sample code (with its output)
- If only we knew a decent templating system...
- Oh wait!

Perl Template  
Toolkit



# Eating Our Own Dogfood

- We developed a simple series of templates to control the book's structure
- And a Perl script to build the book from these templates
- Script was like ttpage but more specialised
- Templates gave us the ability to do a number of clever things

Perl Template  
Toolkit



# Mixing Code With Code

- Our templates obviously contained a bit of TT code
- Our text also contained sample TT code
- Don't want the two to mix
- TT already had the solution
- [% TAGS star %]
- All of our files started with this directive
- The chapter discussing the [% TAGS %] directive and various tag styles was still interesting tho'

Perl Template  
Toolkit



# Useful Blocks and Macros

```
USE chnn = format("ch%02d");
USE fign = format("%02d%02d");

TT = 'Template Toolkit';

versions = {
    stable = '2.10'
    developer = '2.10a'
};

# This namespace is for plugins and such
namespace = 'TTBook';
```

Perl Template  
Toolkit



# Defining Chapters

```
Chapters = [  
    { name = 'intro'  
      title = "Introduction to the Template Toolkit"  
    }  
    { name = 'web'  
      title = "A Simple Web Site"  
    }  
    { name = 'language'  
      title = "The Template Language"  
    }  
    { name = 'directives'  
      title = "Template Directives"  
    }  
    { name = 'filters'  
      title = "Filters"  
    }  
    { name = 'plugins'  
      title = "Plugins"  
    }  
    ... etc ...  
]
```

Perl Template  
Toolkit





# Defining Chapters (cont)

```
FOREACH c = Chapters;  
    c.number = loop.count;  
    c.id = chnn(c.number);  
    Chapter.${c.name} = c;  
END;
```

```
MACRO chref(id) BLOCK;  
    THROW chapter "invalid chapter id: $id"  
    UNLESS (c = Chapter.$id);  
    seen = global.chapter.$id;  
    global.chapter.$id = 1;  
    seen ? "Chapter $c.number"  
        : "Chapter $c.number, I<$c.title>";  
END;
```

```
MACRO Chref(id) GET chref(id) | ucfirst;
```

- This would be used like this

This is covered in more detail in  
[% chref('language') %].

Perl Template  
Toolkit



# Chapter Header Template

```
=pod
```

```
=head0 NAME
```

```
Chapter [% chapter.number %] - [% chapter.title %]
```

```
=head1 ABSTRACT
```

```
[% TRY;  
    PROCESS section/abstract;  
    CATCH file;  
        "warning: $error\n" | stderr;  
    END  
%]
```

```
=head1 CHAPTER OUTLINE
```

```
[% TRY;  
    PROCESS section/outline;  
    CATCH file;  
        "warning: $error\n" | stderr;  
        "TODO\n";  
    END  
%]
```

Perl Template  
Toolkit



# Macros for Example Code

```
MACRO example(file, title) BLOCK;  
  global.example = global.example + 1;  
  INCLUDE example  
    title = title or "F<$file>"  
    id     = "$chapter.id/example/$file"  
    file   = "example/$file"  
    n      = global.example;  
  global.exref.$file = global.example;  
END;
```

```
MACRO exref(file) BLOCK;  
  # may be a forward reference to next example  
  SET n = global.example + 1  
    UNLESS (n = global.exref.$file);  
  INCLUDE exref  
    id     = "$chapter.id/example/$file";  
END;
```

```
MACRO output(file) BLOCK;  
  n = global.example;  
  "B<Output of example $n:>\n\n";  
  INCLUDE "example/$file" FILTER indent(4);  
END;
```

Perl Template  
Toolkit



# Templates for Example Code

- example

```
[% IF publishing -%]  
=begin example [% title %]  
  
    Z<[% id %]>[% INSERT $file FILTER indent(4) +%]  
  
=end  
[% ELSE -%]  
B<Example [% n %]: [% title %]>  
  
[% INSERT $file FILTER indent(4) +%]  
  
[% END -%]
```

- exref

```
[% IF publishing -%]  
A<[% id %]>  
[%- ELSE -%]  
example [% n %]  
[%- END -%]
```

Perl Template  
Toolkit



# Using the Example Macros

```
[% TAGS star -%]  
=head2 html
```

The `C<html>` filter does very basic html encoding: it replaces the most common troublesome characters, `C<E<lt>>`, `C<E<gt>>`, `C<&>`, and `C<">` with their encoded counterparts. This is enough for many encoding jobs, and this filter is very lightweight. More complex encoding will need to use the `C<html_entity>` filter, which implements a more general purpose and extended encoding filter, but which is slower and more involved. `[* Exref('html') *]` shows this filter in action.

```
[* example('html', 'Using the C<html> filter') *]
```

```
[* output('html') *]
```

Perl Template  
Toolkit



# References To Other Books

- We maintained a collection of templates containing the titles of other books we had referenced
- So we could write code like

See [% INCLUDE refs/llama %] for more details.

- Where "refs/llama" contained

```
T<Learning Perl> by Randal Schwartz and Tom Phoenix  
[%- ", page $page" IF page -%]
```

Perl Template  
Toolkit



# Conclusions

- TT is a good tool for this kind of project
- With a bit of work we were able to build a powerful tool for building complex documents
- Many of the features of a word processor, but with plain text files
- TT is not just for web sites

Perl Template  
Toolkit



# Blatent Plug

- The book will be out in late November or early December
- Just in time for Christmas
- Makes a great gift for all the family

Perl Template  
Toolkit

