

Notes from Well House Consultants

These notes are written by Well House Consultants and distributed under their Open Training Notes License. If a copy of this license is not supplied at the end of these notes, please visit

*<http://www.wellho.net/net/whcotnl.html>
for details.*

1.1 Well House Consultants

Well House Consultants provides niche training, primarily but not exclusively in Open Source programming languages. We offer public courses at our training centre and private courses at your offices. We also make some of our training notes available under our "Open Training Notes" license, such as we're doing in this document here.

1.2 Open Training Notes License

With an "Open Training Notes License", for which we make no charge, you're allowed to print, use and distribute these notes provided that you retain the complete and unaltered license agreement with them, including our copyright statement. This means that you can learn from the notes, and have others learn from them too.

You are NOT allowed to charge (directly or indirectly) for the copying or distribution of these notes, nor are you allowed to charge for presentations making any use of them.

1.3 Courses presented by the author

If you would like us to attend a course (Java, Perl, Python, PHP, Tcl/Tk, MySQL or Linux) presented by the author of these notes, please see our public course schedule at

<http://www.wellho.net/course/index.html>

If you have a group of 4 or more trainees who require the same course at the same time, it will cost you less to have us run a private course for you. Please visit our onsite training page at

<http://www.wellho.net/course/otc.html>

which will give you details and costing information

1.4 Contact Details

Well House Consultants may be found online at

<http://www.wellho.net>

graham@wellho.net

technical contact

lisa@wellho.net

administration contact

Our full postal address is

404 The Spa

Melksham

Wiltshire

UK SN12 6QL

Phone +44 (0) 1225 708225

Fax +44 (0) 1225 707126

Plain Old Documentation

A file that contains a Perl program or a Perl module should also contain comments to help the maintainer in the future. It can also contain POD directives, which provide documentation for the user, either a user manual (for a program), or a description of the API provided (for a module).

<i>Own-line directives</i>	5
<i>Embedded Directives</i>	6
<i>Including POD within programs and modules</i>	6

Hopefully, you already comment your Perl programs well so that anyone who has to modify or enhance them (or fix any little problems) is clear on what your program does, and when. Comments start from a hash character (#), and run to the end of the line.

But there's another type of documentation you may require called "user documentation". For a Perl program as a whole, that's documentation for the person who'll be running the program. For a Perl module, that's documentation for the programmer who's going to use your module.

Two questions.

- a) What format would you like the documentation to be provided in? Plain Text? HTML as a web page? Postscript form printing? Manpage format?
- b) Where do you want to keep the documentation files?

Different users are likely to want different documentation formats (and you won't want to maintain more than one). Nor will you want to keep it in a separate file; it's much better in the same file as the program if you can!

Perl provides "Plain Old Documentation" which will meet all the needs of the programmer and of the user. POD format is such that it can be mixed in with a Perl program. The program can be run with Perl ignoring the documentation, or it can be passed through various pod converters to produce documentation in whatever format.

Here's the source file of an example:

```
#!/usr/bin/perl

# This is a demonstration of a POD

=head1 A HELLO world program

All training courses include a "hello
world" program, and we would hate to
make this the exception.

Of course, you know far too much Perl
already for this to be I<just> a
program that greets you.

=cut

# This is nothing to do with POD

print "Hello World\n";

# And so on
```

The output files:

```
<!-- $Id$ -->
<HTML><HEAD>
<CENTER><TITLE>pod1</TITLE>
</HEAD>
<BODY></CENTER><p><hr>

<H1>
<A NAME="pod1_a_0">
A HELLO world program</A>
</H1>
```

```

All training courses include a "hello
world" program, and we would hate to
make this the exception.
<p>Of course, you know far too much Perl
already for this to be <EM>just</EM> a
program that greets you.
<p>
</BODY>
</HTML>

```

Figure 1 Running our "Hello World" Perl program and creating "pod.txt" using PODs

```

seal% pod1
Hello World
seal% pod2html pod1
Creating pod1.html from pod1
seal% pod2text pod1 > pod.txt
seal%

```

and the text as a result:

```

A HELLO world program
  All training courses include a "hello world" program, and we
  would hate to make this the exception.

  Of course, you know far too much Perl already for this to be
  *just* a program that greets you.

```

And the result in HTML:

2.1 Own-line directives

POD formats text in paragraphs. Paragraphs are separated by an empty line. Directives can be placed on their own line between paragraphs which means that there must be a blank line before them. All directives start with an "=" character.

- =head1** starts a level 1 headline (and you place the headline after the directive)
- =head2** starts a level 2 headline (again, place the headline after the directive)

- =over** lets you indent text 'over' a certain number of columns (specify a number – 4 is recommended as the default for some formatters)
- =item** lets you specify a "bullet" for the item
- =back** lets you back out of the previous "over"
- =cut** lets you indicate to the various programs that handle POD that this is the end of a block of POD (and perhaps the start of a block of code)

Special formatting directives

You can also use special formatting directives in pod for a particular translator.

If I wanted a diagram in HTML, or a description in plain text, I could write:

```
=for html
```

```
<BR><IMG SRC=demo.gif><BR>
```

```
=end html
```

```
=for text
```

```
This text describes the demo.gif picture
```

```
=end text
```

You can specify any interpreter you wish. Here we've used "html" and "text", but it's also possible to provide a block which is purely for comment (unless you yourself write a comment interpreter!). This is a very useful way of adding blocks of comments into Perl programs.

2.2 Embedded Directives

You want bold or italic? Easy enough to do. That would be written **B<bold>** or **I<Italic>** within the POD paragraphs. You can also specify **S<Non breaking space>** text so the POD interpreters will not add a line break between the words "Non" and "breaking", nor between the words "breaking" and "space".

With some of the translators (notably HTML), you may well want links within your documentation, and the format allows that. **L<pod1>** will link to the file *pod1.html*. **L<pod1/"World">** will link to the headline "World" within the file *pod1*.

In order to vary clickable text, you may place some text in the link such as **L<Press this|pod1>**.

2.3 Including POD within programs and modules

The simplest way to write POD documentation is after an **__END__** or **__DATA__** directive in Perl. Although if you do this, your ability to provide data within the Perl is limited.

Otherwise (at the slight risk of confusing) you can mix up your POD and your Perl program itself within the body of the source or module file. A good scheme might be to include the description of each use-callable subroutine in a module just in front of the code that defines that subroutine.

License

*These notes are distributed under the **Well House Consultants Open Training Notes License**. Basically, if you distribute it and use it for free, we'll let you have it for free. If you charge for its distribution of use, we'll charge.*

3.1 Open Training Notes License

Training notes distributed under the **Well House Consultants Open Training Notes License** (WHCOTNL) may be reproduced for any purpose PROVIDE THAT:

- This License statement is retained, unaltered (save for additions to the change log) and complete.
- No charge is made for the distribution, nor for the use or application thereof. This means that you can use them to run training sessions or as support material for those sessions, but you cannot then make a charge for those training sessions.
- Alterations to the content of the document are clearly marked as being such, and a log of amendments is added below this notice.
- These notes are provided "as is" with no warranty of fitness for purpose. Whilst every attempt has been made to ensure their accuracy, no liability can be accepted for any errors of the consequences thereof.

Copyright is retained by Well House Consultants Ltd, of 404, The Spa, Melksham, Wiltshire, UK, SN12 6QL - phone number +44 (1) 1225 708225. Email contact - Graham Ellis (graham@wellho.net).

Please send any amendments and corrections to these notes to the Copyright holder - under the spirit of the Open Distribution license, we will incorporate suitable changes into future releases for the use of the community.

If you are charged for this material, or for presentation of a course (Other than by Well House Consultants) using this material, please let us know. It is a violation of the license under which this notes are distributed for such a charge to be made, except by the Copyright Holder.

If you would like Well House Consultants to use this material to present a training course for your organisation, or if you wish to attend a public course is one is available, please contact us or see our web site - <http://www.wellho.net> - for further details.

Change log
Original Version, Well House Consultants, 2004

Updated by: _____ on _____
Updated by: _____ on _____
Updated by: _____ on _____
Updated by: _____ on _____
Updated by: _____ on _____
Updated by: _____ on _____

License Ends.