NAME

pud – Portable Unix Documentation for manual pages and FAQ documents

DESCRIPTION

Portable Unix Documentation or PUD currently provides two mini-languages for authoring in the UNIX environment. The two mini-languages are for writing UNIX manual pages and FAQ documents. Source documents in PUD languages can be compiled to either cross-linked html or to troff. The troff output can be further compiled into PostScript, PDF, and plain text.

Table of Contents

Portable Unix Documentation extends zoem

Portable Unix Documentation is built on top of **zoem** (http://micans.org/zoem), an all-purpose macro/programming language. PUD documents are processed by compiling them with the zoem processor. The documents themselves are generally well-structured, relatively free of formatting statements and compact to write. They can be easily extended since the full zoem language is available from within a portable unix document.

Portable Unix Documentation is currently shipped with zoem. To use it you need zoem and if you install zoem you automatically get PUD. The upside is that the language files locations (which must be imported in a PUD document) are compiled into zoem and zoem and PUD will work seamlessly together.

Getting started

i. Get and install **zoem** (http://micans.org/zoem). All PUD files will be installed as well. If you are reading this locally on your system, chances are zoem and PUD are installed.

ii. On this page read the pointers in section *Section 6* if you want to write a manual page. Read the pointers in section *Section 7* if you want to write an FAQ. The fastest way to get up to speed is to copy and modify a template or existing source document.

iii. While writing your document, consult the **pud-man(7)** documentation, the **pud-faq(7)** documentation, and the **pud-base(7)** documentation as necessary.

iv. Off you go. If you need macro facilities or programming facilities, zoem is there to assist you. Simple macro tasks are easy to accomplish. For more involved stuff you might want to consult the Zoem User Manual (or ZUM). ZUM should be installed locally. Alternatively view the latest ZUM **at micans** (http://micans.org/zoem/doc/zum.html) or subscribe to the mailing list (http://micans.org/zoem/index.html#list).

Unix manual pages in html, troff and PostScript

With the **pud-man(7)** package you create manual pages for output in either *troff* (groff, nroff) or html. The first can be viewed from a terminal, the second in a browser.

The fictitious *buzzz* utility is described in a PUD manual page. It is shipped with every zoem distribution and the buzzz manual page should be installed *locally* in the same location as *its source*. If the location is hard to find you can just obtain the PUD source from the zoem source distribution, or alternatively you may view the latest **buzzz source** (http://micans.org/zoem/mac/buzzz.azm) upstream at micans. Further local links are to the *PostScript version* and the *plain text format*.

For other examples consider the oldest PUD manual page ever written: the **MCL manual page** (http://micans.org/mcl/man/mcl.html), the same in **PostScript output** (http://micans.org/mcl/man/mcl.ps), and the **source for all this** (http://micans.org/mcl/man/mcl.azm). By using the venerable *col* program, the troff output can be converted to nice looking **plain text format** (http://micans.org/mcl/man/mcl.txt). Find the **troff output** (http://micans.org/mcl/man/mcl.1) disclosed as well.

There are some 20+ manual pages for **different utilities in the mcl family** (http://micans.org/mcl/man/).

FAQ documents in html, troff and PostScript

Create FAQ documents with **pud-faq(7)** for output in either *troff* (groff, nroff) or html. The former can be viewed in a terminal via the UNIX man page system, the latter can be viewed in a browser.

The *PUD FAQ mini-language* is described as a rather trivial FAQ itself. It can be viewed in *PostScript* (compiled from troff compiled from *the zoem source* and in *plain text* (again compiled from troff).

For examples behold the **browsing delight** (http://micans.org/mcl/man/mclfaq.html) that is the mcl FAQ, and the **PostScript pleasure** (http://micans.org/mcl/man/mclfaq.ps). Find the **noblest format** (http://micans.org/mcl/man/mclfaq.txt), the **impregnable troff** (http://micans.org/mcl/man/mclfaq.7), and the **source** (http://micans.org/mcl/man/mclfaq.azm) for all that jazz.

Manuals and FAQ examples elsewhere

Other people exist writing PUD. Not many yet. Joost van Baal has used the pud-faq package and the pud-man package to create documentation for **GnuPG (in Dutch)** (http://mdcc.cx/gnupg/), **caspar** (http://mdcc.cx/pub/caspar/caspar-latest/doc/), and the **strong (fire)walls of uruk** (http://mdcc.cx/pub/uruk/uruk-latest/man/).

DocBook considered harmful

People justly wonder why PUD turns away from the blazing light of goodness that is DocBook. DocBook does provide manual page elements and it does support gazillions of output devices. Nevertheless DocBook man pages are a cruelty, a curse and the bane of all things good and pure.

DocBook cannot be written, it cannot be maintained, it cannot be programmed. Yes, XML and DocBook are not *supposed* to be programmed, but where is the decree that man should toil and suffer so that his documentation would be transmogrifyable into all eternity?

DocBook provides some sort of manual page ontology, describing supposedly every element you might ever need. Inevitably you will want to do things that are not provided and then you are stuck. DocBook lists and enumerations are painful and limited. The verbosity of DocBook makes a mountain out of what should be a mole hill.

Zoem manual pages are concise and can be easily cross-referenced. The source is a pleasure to read and output from self-documenting commands can be imported. Zoem IO, macro and programming facilities make the source extendable so that new requirements can be coped with.

Wise people argue that one cannot fathom the needs of future generations and urge the good people of UNIX to use DocBook. The fool knows that this particular premise disproves the thesis and that joy begets joy. Factor the present into the authoring sustainability equation and the scales tip.

At this juncture, I am hesitantly willing to bet that the PUD languages can easily be ported to DocBook. None of the pain, all of the gain. The PUD *itemize* environment is a sticking point though. It provides, horrors, a few formatting options. Optional paragraphs skips, compact mode, right-alignment of items, automatic enumeration, and the fantabulous intermezzo feature.

Info evil

The good people of info consider manual pages obsolete. What more is there to say? It is all written **here** (http://micans.org/stijn/views/infoinferno.html).

AUTHOR

PUD was written by Stijn van Dongen.

SEE ALSO

pud-man(7)
pud-faq(7)
pud-base(7)
pud-ref(7)