



Plain Text Publishing User Guide

Early Beta
2013-06-25

Platform agnostic, bare metal XML multipublishing.

Plain Text Publishing

User Guide

David Priest

Disclaimer

Copyright © 2013, David Priest.

This documentation and any accompanying software are released “as is.” David Priest makes no warranty of any kind, expressed or implied, concerning these materials, including without limitation, any warranties of merchantability or fitness for a particular purpose. In no event will David Priest be liable for any damages, including any lost profits, lost savings, or other incidental or consequential damages arising out of the use, or inability of use, of documentation or any accompanying software, even if informed in advance of the possibility of such damages.

Table of Contents

1. Welcome to Plain Text Publishing!	1
2. Library Structure	2
2.1. LIBRARY STRUCTURE	2



Chapter 1

Welcome to Plain Text Publishing!

ptp-tools combines a plain text markup translator with a industry-standard open-source publishing toolset that generates professional-quality PDF, ePub, and XHTML final documents. Pre-publication transformation and publication layout and style are fully customizable, supporting both a corporate standard style and customizations of that style for each document.

ptp-tools comprises:

- ASCIIDoc, a plain text markup processing system.
- Docbook XML and the Stayton XSL:FO transformations.
- Apache FOP, an XSL:FO to PDF rendering engine.
- ptp-tools/ptp, a set of Bash and Windows scripts guiding the transformation pipeline.
- xsltproc, an XSLT interpreter (needed by Windows platforms).
- XMLSh, a POSIX-flavoured "Bash plus XML" scripting interpreter.

ptp-tools requires pre-installation of:

- Python.
- Java.



Chapter 2

Library Structure

Each publication is maintained in a separate directory containing the source text, source images, configuration and transformation customizations, and a publication script. The ptp-tools publishing toolset is

The publication script (`publish.sh` or `publish.bat`) attempts to find the publication tools in either a sibling directory or subdirectory of the publication directory.

The tools are expected to be in a directory named `ptp-tools`. If you cloned to a different directory, you will need to set the `PTP_TOOLS` environment variable or modify your `publish.sh` or `publish.bat` file.

2.1. LIBRARY STRUCTURE

Library content consists of prose and “chunks”. Using inclusion, prose is intermixed with topical “chunks” to ultimately create a book or wiki. Books tend to take a linear approach to a topic, adding extra information to core units of information: exposition surrounding a “chunk” identified as a List, Task, Figure, Function, etc.

Documentation source directories and filenames are structured for human use:

- `_filename`
 - root document name is prefixed with an underscore.
 - assists in command line autocompletion.
 - file list placement tends to be consistently at the top or bottom
 - content mainly comprises top-level inclusions, ie. articles and chapters.
 - transformed to the ultimate deliverable: a PDF, HTML, EPub, etc.
- `filename`
 - no prefixed underscore.
 - not publishable from commandline.
 - content mainly comprises limited prose introducing mid-level inclusions, ie. chapters and sections.
 - may need to use macros to adjust inclusion title levels
- `directoryname`
 - contains fine-grained “chunks”.
 - for inclusion in mid-level content, ie. chapters and sections.

- named to indicate content/relevance.
- `directoryname/filename`
 - topical “chunks”
 - content comprises prose and lower-level inclusions, ie. sections, lists, tables, etc.
 - content comprises lowest-level chunks, ie. List, Task, Figure, etc.
 - filenames to indicate content; avoid redundancy w/directory “chunk” category

Copyright © 2013, David Priest.

DAVID PRIEST

<ptp-tools@davidpriest.ca>