Passepartout 0.1 User's Guide

What is Passepartout?

Passepartout is an open source desktop publishing application for X−Windows (Unix). The goal is to provide a simple and user−friendly, yet powerful tool for creating printed material. Passepartout combines a WYSIWYG layout editor with a high−quality typesetting engine. Passepartout is released under a BSD license. The Passepartout webpage can be found at http://www.stacken.kth.se/project/pptout/.

Passepartout is not a word processor

You do not use Passepartout to write text, because it is only a *layout* editor. Basically, Passepartout is in the business of taking the different parts that make up a page, such as text, photos, graphics, and "gluing" them on piece of paper. Passepartout can import from several different bitmapped image formats as well as EPS (Encapsulated PostScript) files. You write the text in your favorite text editor (e.g. Emacs or VI) in an XML−based format. The XML file is then typeset using a typesetting engine called xml2ps.

Installation

See the INSTALL file for installation instructions.

Using Passepartout

Starting the program

You start Passepartout by typing `pptout &` in your terminal window. You are then presented with an empty document window (no document is opened).

You can get a list of options accepted by Passepartout with `pptout --help`.

Creating a new document

To create an empty document, select New ... from the File menu. In the Document properties dialog you can choose between creating a document from an existing template (any Passepartout document can be used as a template) or by specifying the document format. You can also specify the page number the first page in the document should have. These settings can be changed at a later point by selecting Properties ... from the Doc menu. Press OK and the new document will appear in the main window.

Adding pages

To add a page to the document, select New Before ... or New After ... from the Page menu. If you have specified a template, you will be asked to select a template page.

Turning pages

You can change the active page with the page selector in the lower right corner of the main window.

Importing images

You can add an image to a page by selecting Import Image... from the Object menu. Passepartout accepts EPS, PNG, RAS, BMP, XPM, PNM, TIF, JPG and GIF files.

Text frames & text streams

A text frame can be added to a page by selecting New Text Frame ... from the Object menu. A text frame shows the text defined in an XML file, but the text frame is not connected directly to the file. Each text frame is connected to a *text stream*, which in turn is connected to an XML file. The reason for this is that a text may span several frames, possibly on several pages. All frames that are meant to be part of of the same text are connected to the same text stream. The stream then splits the text into parts and puts them in the appropriate frames.

Adding text frames to a document is a two−step process:

1. Create a text stream and connect it to an existing XML file in the Streams dialog, which can be opened from the Streams item on the Doc menu. You may also specify an XSLT *stylesheet*, which is used to translate the XML file into the format that xml2ps can understand (see **Creating the XML files**).

2. Create a number of text frames and link them to the text stream in the Object Properties dialog (see **Manipulating objects**).

It is also possible to begin by creating the frames and connecting them to the stream later.

Manipulating objects

An image or text frame can be moved by clicking and dragging it. Text frames and EPS images can be resized using the mouse. The properties of an object can also be manipulated explicitly in the Object Properties dialog which is accessible through the Properties item on the Object menu. When an object is selected, its properties can be edited in the dialog. In order for changes to take effect you have to push the Apply button. Not all properties apply to all types of frames. The properties that are unique to certain kinds of frame are displayed in separate tabs.

A frame can be *locked*, preventing you from accidentally moving or resizing it with the mouse. Locked frames have a gray border.

You can also choose whether text should flow around a frame or not.

Arranging & grouping

Several frames may be combined into a *group* by selecting more than one frame (using CTRL–button1) and selecting Group from the Arrange submenu in the Object menu. Frames can also be moved up or down relative to other frames using the items in the Arrange submenu.

Opening and saving files

Opening and saving files works much like in most other applications. *Note: Passepartout does not (yet) ask before writing over existing files!* The text and images imported into the document are not saved as a part of the document.

Printing

Passepartout can print PostScript or EPS files. (EPS files may only contain one page.) The file can be piped to a printing command, such as `lpr`, or saved to the filesystem. Select Print ... from the File menu. There is also a Print Preview item that will start an external PostScript viewer (if one has been defined).

Creating the XML files

XML (http://www.w3.org/XML/) is an acronym for "Extensible Markup Language". XML is a meta–language for describing different kinds of file formats in a standardized fashion. An XML file is an ordinary text file, but the text is formatted according to strict rules. If you have written HTML before, then you are already familiar with some of the key concepts. If you don't have any previous experience with XML, I would recommend reading an online tutorial:

* http://www.w3schools.com/xml/
* http://www.spiderpro.com/bu/buxmlm001.html

xml2ps – the typesetting engine

The typesetting engine of Passepartout is called xml2ps and it is a stand–alone program (This may change in the future.) . It has its own DTD (Document Type Definition.) . XSLT stylesheets can be used to translate any XML file to the format that xml2ps will accept.

There are two basic approaches to writing text for Passepartout:

1. Write the text in XML conforming directly to the typography–level xml2ps DTD.
2. Write the text in XML conforming to a high level, logically stuctured DTD (either an existing one, such as XHTML or DocBook, or one of your own design), and use an XSLT stylesheet to transform the text into a document conforming to the xml2ps DTD.

The advantage with the former approach is that you get superior control over the result, while the latter method is more suitable when authoring longer texts. Stylesheets for XHTML and DocBook are supplied as examples. This document is in fact formatted in DocBook itself. A good exercise for the reader is to modify these examples to look different. Do you want to change the font from Times to New Century Schoolbook? Do you want an empty line between paragraphs? Do you want all text to be centered? Make it happen!

A description of the xml2ps DTD

Nodes allowed in an xml2ps file:

<block–container>
Root node and parent node of <para> or other <block–container> nodes.
Parameters:
margin–top, margin–bottom, margin–left, margin–right: Margins (lengths).
All <font> parameters are also accepted.

<bp>
Mark a point in word where hyphenation is allowed.

<font>
Change font inside a <para>.
Parameters:
underline: The only accepted value is "1".
font–family: Accepted values are PostScript font (not font family) names.
font–size: Font size (a length or a percentage).
letter–spacing: Extra character spacing (a length).

<leader>
Explicit whitespace
Parameters:
width: Width of whitespace (a length).

<para>
A "paragraph". All text must be contained within paragraphs. Paragraphs may not be nested.
Parameters:
align: Text alignment. Accepted values are "left", "right", "center" and "justify".
line–height: Distance between lines as a factor relative to the font size.
All <font> and <block–container> parameters are also accepted.

Lengths may be specified in points ("12pt", "–0.3pt") or relative to the font size ("2em", "–0.3em").

A short example

```
<?xml version="1.0"?> <block-container
font-family="Bookman-Light" font-size="12pt"> <para
font-family="AvantGarde-Demi" font-size="36.3pt"
align="center" margin-bottom="20pt"> This is a centered
headline in Avant Garde demi-bold 36.3 points </para> <para
align="justify"> This is text in Bookman Light 12 points. <font
font-size="50%">This is text in Bookman Light 6 points.</font>
Some words may be typeset in an <font
font-family="Bookman-LightItalic">italic</font> typeface. The
word "hyphen<bp/>ated" may be hyphen<bp/>ated. </para>
<para margin-left="12pt" align="left" line-height="1.5"> This
is a left-justified indented paragraph with a line height of 18
points. <font underline="1">This sentence is underlined.</font>
In this sentence, there is an unnecessary <leader width="5em"/>
space. </para> </block-container>
```

Configuration

Passepartout gets its configuration options from the file .pptout in your home directory. If the file does not already exist, Passepartout will create one with default values for all settings. If you want to, you can change these settings.

Note: The configuration file format has changed since version 0.0 and the new format is incompatible with the old one.

Settings that may be of interest:

DefaultResolution
Default resolution for images, in pixels per inch

FontPath
Paths to look for Adobe Fonts Metrics (.afm) files in.

PaperName
Default paper name. (Default is A4.)

PSInterpreter
Command to run ghostscript.

PSViewer
PostScript viewer. (Default is `gv`.)

PrintCommand
Default command used for printing. (Default is `lpr`.)

StylesheetPath
Default path to look for XSLT stylesheets in.