



## paper $\TeX$ : Creating newspapers using $\LaTeX 2\epsilon$

Ignacio Llopis Tortosa and  
María José Castro Bleda

### Abstract

The first author has been working in an Internet newspaper office for a year. He was asked to create a special printable edition of an on-line newspaper. The idea was to get an easy-to-print newspaper containing the main daily news. The solution involved two things: a new  $\LaTeX$  class, that we called paper $\TeX$ , and a Perl-based system that gets all the information from the database and composes a new  $\LaTeX$  document.

### 1 Overall

The final system consists of a Perl script which controls the entire process, from the data collection through the document compilation using PDF $\LaTeX$ . There is also a web wizard that lets the user set up the news and the information that he or she would like to show at the newspaper. The configuration file stores the SQL queries which take the information from the database, thus it can be kept for a long time.

Together with this Perl script, there is a new  $\LaTeX 2\epsilon$  class called paper $\TeX$ ,<sup>1</sup> specially created for this purpose. This class provides many macros to create a document in a newspaper style. It has a front page and the inside part that contains all the news we would like to include. Every piece of news appears just below the one before. Headings use the entire page width and the text can be split into several columns. paper $\TeX$  also provides commands for adding outstanding titles, images, timestamps, etc.

Our last page includes an automatically generated Sudoku, a cultural agenda and a humor drawing. Finally, the system is scheduled to run several times a day creating the PDF version of the newspaper so that users can easily download it. Neither manual design help nor computer programming help are needed. The system determines where images can be placed, if there is enough place for a new piece of text and so on.

You can see a comparison between the newspaper web site and the output generated automatically by the Perl script in Figure 1. The main news items are the same as the ones in the web site and their contents appear in the inner pages of the PDF newspaper.

### 2 Why $\LaTeX$ ?

When creating newspapers it is well known you have many high specific applications which do the work. Most of these applications let you create a newspaper and publish it in several ways at the same time: print version, HTML version, etc. *Protec* has three different publishing systems called Millenium, Edicomp and Arcano.<sup>2</sup> *Unisys* has a system called Hermes.<sup>3</sup> Of course there is also Adobe InDesign and Quark Express but this system does something different than designing layouts.

Why use  $\LaTeX$ ? First of all, as  $\LaTeX$  users, we wanted to use it for handling this big project. Secondly,  $\LaTeX$  is a tag based language which lets you create documents without taking care of design—quite the opposite of Quark Express and InDesign. The idea was for the system to take care of the design, from simple text input. Thirdly,  $\LaTeX$  is a free and open source application and a huge number of packages have been written for it.

Once the system was proposed, we had to prove that it was a good option and it could be used for creating newspapers. We created several documents using the `multicol` package, and we added images and capital letters using the `lettrine` and `graphics` packages. We also designed documents using the `textpos` package, placing items in any desired spot. By this time, it was clear that  $\LaTeX$  could do the job.

### 3 paper $\TeX$ : A new class for creating newspapers

The first idea was to look for something similar. Has anyone tried this before? We posted the question in some forums and mailing lists with disappointing results. But someone told us about the `newsltr` class and the  $\TeX$  capability to handle newsletters. This class was a good starting point but we had some problems. The class was created for plain  $\TeX$  and that was a problem for including  $\LaTeX$  packages, and other matters such as embedding Spanish characters directly or correct hyphenation.

So we decided to develop a new  $\LaTeX$  class for creating newspapers. This class provides commands

<sup>1</sup> Freely available from CTAN in <http://www.ctan.org/tex-archive/macros/latex/contrib/papertex>

<sup>2</sup> <http://www.protecmedia.com/>

<sup>3</sup> <http://www.unisys.com/>



Figure 1: The newspaper web site and the self-generated PDF using paperTeX and Perl.

to create a new paperTeX journal: from the front page until the last.

To set up a new document, you have to load the class as usual and use its own commands to define the contents. paperTeX also includes many style macros which the user can customize as desired: font sizes and styles, colors, headings, etc.

### 3.1 The front page

The front page is quite individualized since it was designed using the textps package that has the capability to place things at absolute positions on the page. We did this because the front page needed to have a different style from the rest of the newspaper, being the first thing a reader sees. It includes a main image or photo, three news blocks, an index which links to the inner news, the weather forecast for a locality and some information about the editor. It has also a banner heading like every newspaper does.

Just after the `\begin{document}`, you can start filling the front page inside its own environment. The main image and each of the main news items have their own commands which get all the information: image path and caption, heading, subheading, opening paragraph, section and time stamp. The table of contents has its own environment and the only thing we have to do is to add entries using the `\indexitem` macro. This command requires two parameters: a short text and a reference to a piece of

news inside the newspaper which allows paperTeX to calculate the exact page.

The weather forecast block has three different positions to specify three different weather conditions. Each weather item has an image, maximum and minimum temperatures and a short description. Finally, the editor block includes the usual contact names, email, logo, etc.

As mentioned above, paperTeX includes a set of macros that you can redefine to change default format and layout. For example, redefining the front page logo is as simple as this:

```
\renewcommand{\logo}{
\mylogo{\includegraphics[width=\textwidth]{img}}
}
```

The `\mylogo` command removes the paperTeX default logo and changes heading elements' positions to make your new logo fit well. Other style aspects are even easier to redefine. You can find all of them in the paperTeX manual [1].

### 3.2 The news pages

Once you have introduced all the front page information, the next thing to do is to include all the news in your new paperTeX. News items are easy to include and they can have different shapes: *news*, *editorial* and *short news*. Each one of these types has its own environment definition.

To include a normal news item you use the *news* environment, which needs five parameters: number

```

\begin{news}{3}
{...NEWS HEADING...}
{...News subheading...}
{SPORTS}
{1}
\authorandplace{Name Surname}{Place}
\image{img}{Image caption}
\noindent\timestamp{08:25}
Lorem ipsum dolor sit amet, ...
... platea dictumst.
\end{news}

```

Figure 2: Example of a compilation of news items.

of columns, heading, subheading, section name and reference id. The main news text must appear inside the environment and we can also use the following commands within the text in order to add other useful information:

- `\authorandplace`: inserts the name of the editor and where the news happened. To be used at the beginning of the text.
- `\timestamp`: inserts the time and a separator just before the text. It should be used at the beginning of the text.
- `\image`: inserts an image within the text. Since the `multicol` package does not support floating elements, this macro inserts the image only if there is enough space, otherwise you can get images outside the page boundaries.
- `\columntitle`: inserts a single column title or heading, using one of five different shapes via the `fancybox`<sup>4</sup> package: `shadowbox`, `doublebox`, `ovalbox`, `Ovalbox` and `lines`.
- `\expandedtitle`: similar to `\columntitle` except the text extends across the entire page, above all the news columns.

Using all these commands a news source code and its respective result would look like the example in Figure 2.

Editorial news and short news require fewer parameters but are generally similar to the news environment. The `paperTEX` manual [1] has more information.

Once the news items are included, when compiling `paperTEX` will create the corresponding PDF

<sup>4</sup> You can find more information in the `fancybox` package manual available on CTAN.



bookmarks inside each section. In order to generate a new group of items we can use the `\newsection` command which takes the section name as a parameter. From this point, all news PDF bookmarks will be grouped under this section name. Another useful and simple command is `\newsep`, which draws a thin line between two items as a separator.

#### 4 The core system

When the `paperTEX` class was finished, we implemented a new system to carry out the entire newspaper creation process. We decided to use Perl as programming language because we are familiar with it, and because Perl has a lot of modules freely available on CPAN.<sup>5</sup>

Before coding, it was very important to select which news items have to be included each time that `paperTEX` is run. The idea was to define these items using the SQL queries which get all the information from the database; thus, `paperTEX` would not need to be configured each time. The only time when it is necessary to modify something is if the editor of the newspaper wants to change the source of a piece of news or add or remove particular content.

First of all, a configuration file was created for the Perl script to get all SQL queries, execute them, and extract the useful information from all fields. It worked well enough, but was not very user-friendly. Therefore we decided to create a web wizard which gets all the information from the user. This application also lets the users change any parameter or SQL instruction. After this, we asked the editor of the newspaper to give us the list of news he wanted

<sup>5</sup> The Comprehensive Perl Archive Network (<http://www.cpan.org/>).

to appear in the paper $\TeX$  newspaper; then we ran the web wizard and filled in all SQL queries and database information. This config file has not been changed in more than six months and is working well.

The news items are stored in HTML format, so we created a script to get all the items included in the config file, convert them to  $\LaTeX$ , create a new paper $\TeX$  document and, in the end, compile it using PDF $\LaTeX$ . We also decided to add a final page including an events list, an automatically generated Sudoku and a humour drawing. To develop this script some Perl modules were used. For example:

- `HTML::Latex` by Peter Thatcher, which converts any HTML text to  $\LaTeX$  in every way you like. This module was very important for us because all the news items were stored in HTML format just as they were published on the website.
- `Weather::Com` by Thomas Schnuecker, which retrieves the weather forecast included on the paper $\TeX$  front page.
- `Games::Sudoku::Component` by Kenichi Ishigaki, which creates and solves a new Sudoku each time paper $\TeX$  is created. This module was very useful (after few modifications) with the `sudoku`  $\LaTeX$  package by Paul Abraham.

When executing the final script, we had some problems composing the newspaper front page, because the database contained texts that could be longer than the space available. To prevent text overflow, a function to trim texts at certain position (always after a period) was implemented. Some tests in order to get the right threshold were made. Although the perfect solution would have been to have short texts, the system is working quite well as it stands. Another problem was that, at first, the editor of the newspaper office did not want to hyphenate headings in the front page. So we tried to avoid hyphenation, but we got very bad results when there were long words which ran off the page.

Finally, we installed  $\LaTeX$  and the script on the company server and use a *cron*-based application to schedule execution three times a day. We also linked the PDF output in the newspaper web site.

## 5 Conclusions

The goal of this project was to create a special printable edition of an on-line Spanish newspaper. The solution involved a new  $\LaTeX$  class, that we called paper $\TeX$ , and a Perl-based system that automatically extracts, composes and creates a final PDF

file. This system works off-line and does not need any human assistance in order to generate the publication several times a day. It is also worth emphasizing that this system is working today with the same configuration as in August 2006. The final application is running for a Spanish on-line newspaper called `Panorama-Actual.es` and it creates a publication which name is *papelDigital*. You can freely access them through <http://www.panorama-actual.es/pdigital/>.

Finally, there is another system in Spain which does something similar to the application described in this document. It appeared at the beginning of this project but, as far as we know, it does not use  $\LaTeX$  at all. The publication is called *24 Horas*<sup>6</sup> and it is used by EL PAIS, one of the most widely-circulated newspapers in Spain.

## Acknowledgments

We would like to gratefully acknowledge the support of Robert Fuster, from the Polytechnic University of Valencia (Spain). Thanks for reading the various drafts and making useful suggestions.

We would also especially like to thank the members and regular participants on the Spanish  $\TeX$  user list *es-tex* list. They have contributed with comments during the whole development process. Finally, the first author would like to thank Mar for always being *there*.

## References

- [1] Ignacio Llopis Tortosa. paper $\TeX$  class: Creating newspapers using  $\LaTeX$ . Technical Report DSIC-II/03/07, Department Sistemas Informáticos y Computación, Polytechnic University of Valencia, 2006.

- ◊ Ignacio Llopis Tortosa  
Department Sistemas Informáticos  
y Computación  
Univ. Politécnic de Valencia  
Camino de Vera s/n  
46071 Valencia, Spain  
lloptor (at) gmail dot com
- ◊ María José Castro Bleda  
Department Sistemas Informáticos  
y Computación  
Univ. Politécnic de Valencia  
Camino de Vera s/n  
46071 Valencia, Spain  
mcastro (at) dsic dot upv dot es

<sup>6</sup> *24 Horas* is free to download through <http://www.elpais.com/24horas/>