

由 T_EX/L_AT_EX 製作中文 PDF 檔

李果正 Edward G.J. Lee

2002 年 4 月 27 日

目 錄

1	前言	2
2	工具準備	2
2.1	CJK 套件	2
2.2	pdfT _E X 系統	3
2.3	hyperref 套件	3
3	字型安裝	4
3.1	有關字型的一般注意事項	4
3.2	本文所使用的字型	5
4	製作流程	5
4.1	dvips/ps2pdf	5
4.2	dvipdfm	6
4.3	pdfT _E X/pdfL _A T _E X	6
4.4	hyperref 的設定	6
4.4.1	hyperref 的設定檔	6
4.4.2	hyperref 的設定項目	7
5	動手實作	7
5.1	文稿內的架構	8
5.2	關於交互參照(cross-reference)	9
5.3	網址超連結 (hyperlink)	9
5.4	所見即所得?	10
6	結語	11
7	授權聲明	11

1 前言

$\text{T}_{\text{E}}\text{X}/\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 的指令繁複，許多朋友都視為畏途，如果不是常在使用，那真的是會很累人，但又愛其排版品質的精細優良，而且您大概沒想到過，我還拿他排版音樂五線譜、棋譜、化學結構式……，實在是令人又愛又恨。但是有一個比較節省時間的法子，就是有空時就把一些常用的結構整理出來，或寫一篇新的格式文章時，保留檔案的結構，就好像是 `templet` 一樣，要用到時再拿來填入文章即可，這不失是一種方便的方法。

底下就是要介紹如何使用 **CJK**¹ 套件來製作中文的 PDF 檔。所使用的工具都是一般的 $\text{T}_{\text{E}}\text{X}/\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 系統裡頭就有的，而不是要去買一套所費不貲的商業軟體。當然，為達到細部品質的要求，有些套件可能需要更新一下，只要照著本文的指引去更新，應該是不會太複雜，現在不都流行 DIY 嗎？

由於基本上由 $\text{T}_{\text{E}}\text{X}/\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 系統製作 PDF 文件，主要是由 `ps` 檔來轉的，除非是使用 `pdfTEX` 系統，因此以下所說的都是要使用 `Type1` 字型或 `TTF` 字型，最好不要使用 `pk` 點陣字型，否則，製作出來的 PDF 檔，品質會很差。

2 工具準備

2.1 CJK 套件

如果您的 CJK 是 4.2.0 版的，那最好更新一下，可以到以下網址下載：

<ftp://ftp.ffii.org/pub/cjk/devel/cjk-current.tar.gz>

由於這是發展中的版本，可能一切都得手動。可參考所附說明把可執行檔編譯出來，取代系統原有的就可以了。重要的是 `$TEXMF/tex/CJK`² 目錄下的檔案要更換成新的，就是原始碼的 `/texinput` 目錄下的所有檔案。

更新 CJK 套件的目的是可以配合 `hyperref` 套件產生中文的 `bookmarks`，這在以前是只有商業軟體才能做到。而且，在 Unix-like 系統中尚無製作、閱覽中文 `bookmarks` 的商業軟體存在。

¹這是由 **Werner Lemberg** 所開發的一個 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 套件，專門用來排版中日韓文。

²這是指 Unix-like 系統，一般是在 `/usr/share/texmf/tex/CJK` 目錄下。在 Windows 系統下，那可能會在 `texmf/tex/latex/cjk` 目錄下。

2.2 pdfTeX 系統

這不僅僅是巨集套件，而是一個 $\text{T}_{\text{E}}\text{X}/\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 的代替品，可以將 `*.tex` 檔，直接編譯成 PDF 檔。這不更新也是可以，只是就無法直接取用 TTF 字型檔了。

在原有系統中也是可以只更新 pdfTeX，可以到以下網站下載各種平臺的可執行檔：

<http://www.tug.org/texlive/tlprod/Master/bin/>

別忘了一些必要的檔案，例如 `pdftex.def`, `*.pool` 等，在覆蓋原來的檔案後，執行 `texconfig init`。可參考第 3 節的網頁說明。以下為必要檔案列表：

表 1: pdfTeX 必要之檔案列表

檔案	所在位置
<code>pdftex</code>	<code>/usr/bin*</code>
<code>pdfetex</code>	<code>/usr/bin</code>
<code>ttf2afm</code>	<code>/usr/bin</code>
<code>pdftosrc</code>	<code>/usr/bin</code>
<code>pdftex.pool</code>	<code>/usr/share/texmf/web2c</code>
<code>pdfetex.pool</code>	<code>/usr/share/texmf/web2c</code>
<code>supp-mis.tex</code>	<code>/usr/share/texmf/tex/context/base</code>
<code>supp-pdf.tex</code>	<code>/usr/share/texmf/tex/context/base</code>
<code>pdftex.def</code>	<code>/usr/share/texmf/tex/latex/graphics</code>

* 不同的 distribution 或 OS 會有不同的位置，例如 Slackware 是集中在 `$TEXMF/bin` 目錄下。Windows 系統的 MiKTeX 則會在 `texmf\miktex\bin` 目錄下。

2.3 hyperref 套件

這個巨集套件主要是把 PDF 的一些功能嵌入 `*.tex` 檔內。不更新也沒關係，但就無法做出 PDF 檔中的中文 bookmarks 了。當然，中文的 bookmarks 還得配合新版的 CJK 套件才行。可在以下網址下載：

<ftp://ftp.tug.org/pub/tex/hyperref/>

版本一定要高於 2002/04/12 v6.721 否則會無法處理中文 bookmarks。這個套件的原始作者是 Sebastian Rahtz，目前是由 Heiko Oberdiek 所維護。實際例子請參考第 5 節，相關設定可參考第 4.4 小節。

3 字型安裝

3.1 有關字型的一般注意事項

這個部份請參考前曾發表的文章“使用 pdf_TE_X/pdf_LA_TE_X 讓 pdf 檔內嵌中文 TTF/TTC”：

<http://www.study-area.org/tips/latex/pdftex.html>

這裡要補充的是，目前大概還有不少朋友還留有以前的 Big-5 碼的 TTF，這可以借由 [oto](#)³ 這支程式來加入 Unicdoe name/cmap table，這樣就可以當做是 Unicode 字型來使用。不過，其中的 map 表，並不合我們使用的 Big-5 碼中有關日文的部份，我做了個 patch，只要 patch 裡頭的 BIG5.TXT 檔就可以了，不必重編譯程式，可在 [study-area](#) 網站下載：

<http://www.study-area.org/tips/latex/oto-BIG5.diff.gz>

patch 的方法如下：

```
cd /usr/local/share/oto/  
gzip -dc oto-BIG5.diff.gz | patch -s -p0
```

為了要使用 Type1 字型，\$TEXMF/dvips/config.ps 最好是修改一下，加入以下各種設定 (這裡假設您的中文 Type1 或 TTF 字型都已安裝、設定妥當)：

```
% 以下設定會預設使用 BlueSky 的 Type1 字型，等同於 dvips 的 -Ppdf 參數。  
p +bsr.map % CM/AMS fonts  
p +bsr-interpolated.map % additional sizes  
p +hoekwater.map % additional fonts from Taco Hoekwater  
h tex.pro  
h alt-rule.pro  
% 以下這個 o 要把前面的 % 拿掉，否則預設執行 dvips 會導向印表機。  
o
```

如果不加這些設定，則要執行以下的指令代替：

```
dvips -Ppdf -G0 -o your.ps your.dvi
```

千萬不要省略這個 -G0 參數，不然就得在 \$TEXMF/tex/dvips/config.pdf 中把 G 這個設定給 % 起來。這是一個古老的 acroread 的 bug 所產生的相容性問題，目前的 acroread 應該不會有這種情形，加進去反而會有其他副作用。

³這支程式是由 [Yao Zhang](#) 所開發，不使用其他函式庫，是一個獨立有用的小程式，以往的 Big-5 碼 TTF 可借由這個程式加入 Unicode table 後借屍還魂給 X 或 ttf2pt1 來使用

3.2 本文所使用的字型

本文 PDF 格式所使用的字型都是自由軟體或 Public Domain，非常感謝各提供的公司及個人，由於他們的熱心，使得這篇文件裡裡外外稱得上是自由文件。茲列表於表 2：

表 2: 本文 PDF 格式使用的字型

字型檔名稱	使用處	字型名稱	提供者	版權
bsmi00lp.ttf	內文	文鼎細上海宋	文鼎公司	GPL-like
wcl-02.ttf	內文標題	王漢宗特明體	王漢宗博士	GPL
wcl-07.ttf	Title	王漢宗粗鋼體	王漢宗博士	GPL
cm*.pfb	內文英文字	CM PS fonts	Blue Sky Research	Public Domain

使用的轉換程式是 `oto` 及 `ttf2pt1`。如果是使用 `pdfTeX` 的話，可直接抓取 TTF 而不必轉換成 Type1。

4 製作流程

以下所討論的方法應該是最常見的，當然 `TeX/LaTeX` 的世界無奇不有 (個人覺得，有點像 `perl` 的世界，一大堆的現成模組可以運用)。但以處理中文而言，除非是自行寫程式，否則這應該算是最方便的現成方法了。以下的舉例，都是以 `chpdf.tex` 為例，也是本文寫作的文稿檔案名稱。

4.1 dvips/ps2pdf

這可說是最正統，最古老的方法了。最近由於 `Ghostscript` 的大幅改進，已經使這個製作方法愈來愈完美。`Ghostscript` 要在 6.51 以上的版本才會有較佳的效果。

這個方法其實就是一般 `TeX/LaTeX` 製作的正常過程，由 `TeX/LaTeX` 指令，得到 `*.dvi` 檔，再由 `dvips` 製作成 `*.ps` 檔，然後再由 `ps2pdf` 將 `*.ps` 檔轉成 `*.pdf` 檔。當然，處理中文而言，我們使用的是 CJK 套件，在 Big-5 碼的話，指令是 `bg5latex`。實際指令如下：

```
bg5latex chpdf.tex
dvips -t a4 -Ppdf -G0 chpdf
ps2pdf13 chpdf.ps
```

其中 `ps2pdf13` 是指使用 PDF-1.25 的格式，建議使用這個格式，比上不足，比下有餘。據個人的經驗，使用這個方法製作來的 PDF 檔，會大一些，但品質會比較好，而且，在使

用 `gzip` 壓縮後會比由 `dvipdfm` 或 `pdfTeX` 做出來的還要小，很奇怪吧！這也是本文的 PDF 格式的製作方法。不過，無論如何，本文中文字型是內嵌的，這樣可攜性會比較好，但檔案就會比較大了。

4.2 dvipdfm

這個過程比第 4.1 小節的方法少了一個 `*.ps` 的步驟。`dvipdfm` 會將 `*.dvi` 檔直接生成 `*.pdf` 檔。

`dvipdfm` 標榜的是忠於原 `TeX`，但這樣一來，可加入的 PDF 特性的功能就少了許多，可以依靠 `hyperref` 套件來做擴充的功能。

實際執行指令如下：

```
bg5latex chpdf.tex
dvipdfm chpdf
```

4.3 pdfTeX/pdfLTeX

`hyperref` 套件雖然也可以加入不少的 PDF 特性，但 `pdfTeX/pdfLTeX` 可說是重寫的軟體，可以當做 `TeX/LaTeX` 來使用，並且加入相當多的 PDF 特異功能，如果目標是製作 PDF 檔的話，`pdfTeX/pdfLTeX` 無疑是首選的工具。

這個部份可以先參考第 3 節所指網頁的說明，先能動再來談應用。版本至少要 0.14h 以上的版本才能直接抓取 TTF。

當然 `pdfTeX/pdfLTeX` 的詳細功能，本篇文章並沒有做詳細說明，僅對有用到的功能略加解說。讀者可以下載使用手冊來參考。

4.4 hyperref 的設定

以目前的情形而言，`hyperref` 套件的使用範圍可能會比 `pdfTeX/pdfLTeX` 還要廣。他不僅可以配合原來的 `TeX/LaTeX` 系統，也可以和 `pdfTeX/pdfLTeX` 合用，因此對這個套件有必要多瞭解一下。

4.4.1 hyperref 的設定檔

原則上設定檔是 `hyperref.cfg`，使用 `hyperref` 套件時，一有用到其中的 macro 就是會找這個檔，其中找尋的順序是：

1. 文稿中是否有設定，這是最優先使用的，會蓋過其他設定。
2. 目前的工作目錄是否有 `hyperref.cfg`。
3. 系統的預設值，如 Unix-like 的 `$TEXMF/tex/latex/hyperref/hyperref.cfg`。Windows MiKTeX 系統中的 `$TEXMF\tex\latex\00miktex\hyperref.cfg`。

4.4.2 hyperref 的設定項目

這裡以我自己的 `hyperref.cfg` 來說明：

```
1 \hypersetup{backref,  
2             dvips,  
3             bookmarks=true,  
4             pdfauthor=Edward G.J. Lee,  
5             colorlinks=true,  
6             breaklinks=true,  
7             hyperfigures=true,  
8             pdfstartview=FitH,  
9             linkcolor=blue}
```

基本上，原來文稿使用 `\label/\ref` 及目錄、索引的部份就會自動可以交互參照。2 是指定要用其他如 `dvipdfm`，`pdfTeX` 時更改這裡的值，或設在文稿內。

3 是指要不要 bookmarks，4 是作者名字，在使用 [Xpdf](#)⁴ 所附的 `pdfinfo` 指令執行後就會顯示。或者使用 `acroread` 在功能表上叫出 Document info 時也會顯示，在 PDF 正文內則不會顯示出來。

5 及 9 是設定超連結的地方要不要顯示顏色，及顯示什麼顏色。

8 是指使用 `acroread` 閱覽這個 PDF 檔時的顯示模式，例如，是 Fit Width 還是 Fit in Window。這在 `Xpdf` 及 `gs/gv` 則沒有這個功能來判斷這個設定。這裡設的是 Fit Width，不過，這個是傳參數給 PDF View 這個 key，據個人的測試，似乎控制得不是很好。

最後，請大家饒了我吧！不要把例子裡頭的阿拉伯數字也寫進您的設定檔裡去了。

5 動手實作

這個部份可說是本文的精華了，因為要加入 PDF 特有的功能，光靠傳統的方法及工具是不夠的，例如一些連結及 bookmarks 的功能，除非很瞭解 PDF 的一些 instructions，自

⁴Xpdf 是由 [Derek B. Noonburg](#) 所開發的 PDF 閱覽器，雖不若 `acroread` 功能那麼完備，但在自由軟體來說，是除了 `Ghostscript` 外的唯一“專業”PDF 閱覽器。

行由 $\text{T}_{\text{E}}\text{X}$ 的指令 `\special` 來帶入 `*.tex` 文稿中，但這樣一來可就累了，本就夠複雜的 $\text{T}_{\text{E}}\text{X}/\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 語法，再加上 PDF 的指令，那一篇文章還沒完成，大概就打退堂鼓了。先別緊張，本文雖沒有很深入的說明，但本文這個活實例，就是您的 `templet`，直接拿來套用就對了！

本文的實際文稿及完成後的 PDF 檔可在 [study-area](http://www.study-area.org/tips/latex/chpdf.tex) 網站下載：

<http://www.study-area.org/tips/latex/chpdf.tex>

<http://www.study-area.org/tips/latex/chpdf.pdf>

這樣就可以一邊看文章一邊看實際的排版結果。其中 `chpdf.tex` 由於加入了許多注解，因此顯得比較亂，而且個人寫 `*.tex` 文稿的習慣都是不內縮的，這點請大家忍耐一下。本文沒有詳細說明的部份，應該在 `chpdf.tex` 的注解裡頭都會稍加說明。

當然，您必須具有最基本的 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 排版的經驗，例子中已經盡量採用較簡單的方式，但有一小部份，為達到所需要的效果，還是不得不要用到特殊的套件。

5.1 文稿內的架構

```
1 \documentclass[12pt,a4paper]{article}
2 \usepackage{CJK}
3 \usepackage[dvips,CJKbookmarks]{hyperref}
4 \begin{document}
5 \begin{CJK*}{Bg5}{aming}
6 \CJKtilde

.....
這裡寫文章

7 \end{CJK*}
8 \end{document}
```

重點在 2 及 3 的部份。這裡如果要改用 $\text{pdf}_{\text{T}}\text{E}_{\text{X}}$ ，那就得把 `dvips` 改成 `pdftex`，由於我的 `hyperref.cfg` 已經設定好 `dvips`，因此這裡的 `dvips` 也是可以省略。而且注意這個 `CJKbookmarks`，這就是讓 PDF 能有中文 bookmarks 的設定，要不然因為標題的部份經過編碼，最後如果沒有還原的話就變成一堆亂碼了。這裡要特別感謝 [Werner Lemberg](#) 及 [Heiko Oberdiek](#) 的幫忙，解決了中文 bookmarks 的問題，是由 Werner 提出 patch，然後 Heiko 做了一些修改後釋出。當然，這在目前只能在 Windows 系統下的 `acroread` 才會看到中文 bookmarks，拿到 Unix-like 系統的 `acroread` 閱覽則仍是一堆亂碼，這是 `acroread` 的問題，而不是 PDF 檔的問題。

其他關於目錄、交互參照、腳注、參考文獻、索引……等等，只要使用了 `hyperref` 套件，便會自動處理，不必操心。但本文並沒有談到參考文獻、索引的部份，這是更進階的課題了，留待有機會再來探討。

5.2 關於交互參照(cross-reference)

基本上 \LaTeX 本身就有交互參照的指令可用，分別是 `\label` 及 `\ref`，`\pageref`，當然這只能顯示，沒辦法由滑鼠點了以後跳來跳去。也就是說只要使用 `hyperref` 套件，縱使沒有加入新的 tags，也是會有基本的文內連結功能。當然 `hyperref` 套件另有擴充這方面的功能。請參考表 3：

表 3: 交互參照

套件	指令	功能說明
\LaTeX	<code>\label{name}</code>	以 <i>name</i> 來定位
	<code>\ref{name}</code>	顯示 <i>name</i> 之所在章節、圖表
	<code>\pageref{name}</code>	顯示 <i>name</i> 之所在頁碼
hyperref	<code>\hypertarget{name}{text}</code>	以 <i>name</i> 來定位
	<code>\hyperlink{name}{text}</code>	按 <code>hyperlink</code> 定位處，會跳至 <code>hypertarget</code> 所在之處

5.3 網址超連結(hyperlink)

這個在原本 \LaTeX 當然是沒有的， $\text{\TeX}/\text{\LaTeX}$ 是屬於平面排版系統，網路上的應用是在 PDF 格式廣為被使用後才開始有相關套件出現，這恐怕也是 \TeX 系統能長久維持一定比例使用者的原因，彈性、功能都相當大，品質也有保證，較不容易被時間所淘汰。

底下仍以圖表的方式來說明比較清楚，請參考表 4：

表 4: 網址超連結的方法

套件	定址	說明
pdf \TeX		為了相容性，請盡量直接使用 <code>hyperref</code> 的方式
hyperref	<code>\href{url}{text}</code>	連結至 <i>url</i> 所在處
	<code>\hyperref{url}{category}{name}{text}</code>	連結至 <i>url</i> # <i>category</i> 所在處
$\text{\LaTeX}2\text{HTML}$	<code>\htmladdnormallink{text}{url}</code>	連結至 <i>url</i> 所在處

5.4 所見即所得？

現在很流行“所見即所得”，這裡不想引發任何爭議或論戰，只是目前個人使用的環境是 GNU/Linux 系統，我就說說我是怎麼編譯，怎麼預視的。

首先我開了兩個終端機 (rxvt)，我的視窗管理程式是 IceWM，我已經設定好由按鍵就可以在兩個虛擬視窗間方便移動。使用的編輯器是 Vim。在工作目錄已寫好了一個 sh script 稱為 `compile` 其內容如下：

```
#!/bin/sh
if [ $# -ne 1 ]
then
    echo "Usage: 'basename $0' your.tex"
    exit 1
fi
FNAME=$1
rmttextmp
bg5latex $FNAME && \
bg5latex $FNAME && \
dvips -t a4 -Ppdf -GO ${FNAME%.tex} && \
ps2pdf ${FNAME%.tex}.ps && \
xpdf ${FNAME%.tex}.pdf
```

其中 `rmttextmp` 是另一個 sh script，內容如下：

```
#!/bin/sh
rm -f *.aux *.toc *.log *.dvi *.cjk *.out \
    *.brf *.idx *.ind *.lof *.lot
```

主要是去除一些暫時產生的檔案，可置於路徑可達之處，否則要指明路徑。這裡 `bg5latex` 要執行二次，是因為交互參照的部份要第二次編譯時才會完成。

左邊由 vim 編輯了一段，想預視的話，馬上移到右邊執行：

```
./compile chpdf.tex
```

就可以了，文稿不大，而且又使用 Type1 字型的話，速度是相當快的，而 vim 也可以由顏色來幫忙控制語法是否正確，當然前提是文稿的整個大結構要先建立好，否則編譯容易發生錯誤而停下來。而大文稿本就應該切割成幾個部份，由 make，makefile 來控制。當然要做到完全由 vim 來控制也是可以，這樣會不會更方便就看個人所持的觀點和習慣了。

在 Emacs 編輯器 (系統?) 也有現成的 `preview-latex` 可以使用。

不曉得這樣子算不算是另類 WYSIWYG？在我的經驗裡，這樣的生產量是不會輸給 MS Word，尤其是文稿份量相當大，例如幾百頁的內容，至於細部微調控制那就更不用說了。

甚至國人開發的 **cwTeX** 及 **ChiTeX** 目前都有相配合的 GUI 工具，相當方便，只可惜偏重在 Windows 系統上，在 Unix-like 系統上還是得雙手萬能一下。這裡要提醒大家一下，TeX 系統畢竟是幕後排版系統。而且，排版是一門審美藝術，大小細節要能掌握住，才能達到質的要求，我們要不是 WTGIWYG(What They Give Is What You Get)。

6 結語

好了，文章告一段落。我得事先聲明，我不算是什麼 TeX/LaTeX 的專家，只是常在使用，會有一些使用上的心得，文章裡頭所說的不敢保證百分之百正確，但至少我是親自實作過，如果有不正確的地方，希望大家提出指正。跟據我的說法去實作，如果有任何“意外”，那我不負責的。您不會認為看了這十頁的內容就變成 TeX/LaTeX 的專家了吧！千萬不能有這種想法，學海無涯，一山還有一山高……。

另外，有許多朋友來信問到 TeX/LaTeX 相關文件的問題，其實一般的 TeX 系統裡頭就會附上相當多的文件，只是有些可能沒有編譯出來，是以 *.tex 或 *.dvi 的形式存在，只要動手把他們編譯出來就成了。另外，一些 *.sty 檔裡頭都會有一些說明，這可是第一手資料喔！當然，這些都是英文，但只要找到相關的指令，照著指令的格式動手實驗一下就成了。

至於中文書籍，新版本的確實不多，但吳聰敏教授的《cwTeX 排版系統》一書應該相當具參考價值，這是個人手邊重要的工具書之一，雖然說的是 cwTeX 但其實都差不多，LaTeX 的語法是固定的，這本書網路上也可以下載。陳宏毅教授的 ChiTeX 系統也是有一些文件可以參考，也可以由網路上下載。CJK 本身的話，所附的文件也有中文的，就是 READMEb5.tex，如果沒有編譯出來的話，自行改一下裡頭使用的中文字型，手動編譯出來就是一篇 CJK 中文使用手冊了。

7 授權聲明

Copyright © 2002 李果正 **Edward G.J. Lee**

本文件除 sh script 外，為自由文件(**FDL**)，可自由複製、修改、散佈，但請保留版權聲明。sh script 的部份為 Public Domain。文件內所提及的商標皆屬其合法註冊公司所有。