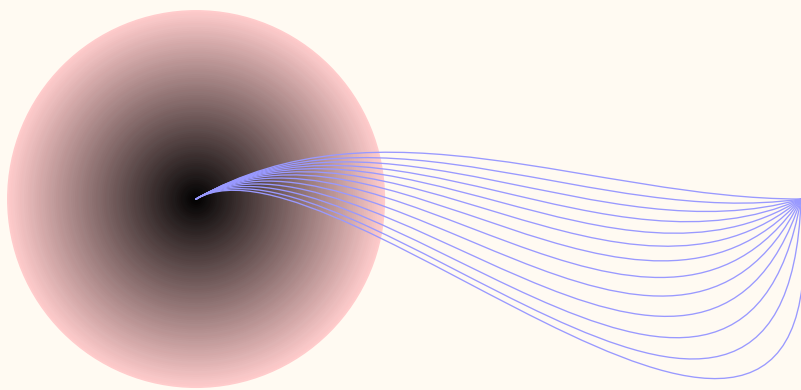


我的 CJK

Version 1.1

2004 年 5 月 18 日



By Edward G.J. Lee 李果正

Email : edt1023@info.sayya.org

本教學文件之製作，部份接受下列計劃補助：

行政院研考會委辦 「政府機關資料文件交換之電子檔案格式應用研究」

目錄

1	前言	1
2	為什麼不再使用傳統的 pk 點陣字型？	2
2.1	中文 Type1 字型	2
2.2	中文 CID-Keyed 字型	3
2.3	中文 TTF 字型	3
3	安裝說明及相關工具	4
3.1	teTeX 系統	4
3.2	其他相關套件	5
3.2.1	tlutils	5
3.2.2	ttf2pt1	5
3.2.3	ttf2pt1-chinese	5
3.2.4	FontForge	5
3.2.5	DVIPDFMx	6
3.2.6	CJK 巨集套件	7
3.2.7	hyperref	7
3.2.8	L ^A T _E X2HTML	8
3.2.9	L ^A T _E X macro/package 的安裝	8
4	製作中文 Type1 字型	10
4.1	需要哪些字型資料？	10
4.1.1	Free 的字型的取得	10
4.1.2	各種字型資訊檔的意義	11
4.1.3	尋找 L ^A T _E X 所需的資訊	12
4.2	由 mkfont.sh 來製作需要的資料	12
4.3	由 FontForge 來製作字型	13
4.4	文稿測試	15
5	CJK 語法簡介	16
5.1	L ^A T _E X 的基礎文件	16
5.1.1	一些值得參考的入門文件	16
5.2	CJK 中的特殊指令	17
5.2.1	CJK 的主要架構	18
5.2.2	變更字體	18
5.2.3	變更字距、行距	19

5.3	關於中英混合字間距及中文斷行	19
5.3.1	編輯時手動加入必要的控制	20
5.3.2	使用 CJK* 環境	20
5.3.3	由程式本身去解決	20
5.4	直排的問題	21
5.5	漢字加點、畫線	21
6	CJK 和 L ^A T _E X2HTML 的配合	23
6.1	L ^A T _E X2HTML 語法簡介	23
6.1.1	引入 L ^A T _E X2HTML	23
6.1.2	簡單的條件式指令與環境	24
6.2	L ^A T _E X2HTML 和 L ^A T _E X 配合需注意的地方	25
6.2.1	設定檔	25
6.2.2	數學模式	26
6.2.3	圖檔的處理	26
6.3	L ^A T _E X2HTML 和中文的配合	27
6.3.1	中文式網頁	27
6.3.2	把中文字轉成圖檔來顯示	27
7	CJK 和 KOMA-Script 的配合	29
7.1	KOMA-Script 和 L ^A T _E X 標準 class 的對照	29
7.2	中文章節標題的定義	29
8	CJK 中如何編輯、編譯大型文件？	31
9	如何造字？	33
9.1	為什麼需要造字？	33
9.2	選用的工具	34
9.3	定義、調整及取字	35
9.4	其他的替代方案	36
9.4.1	使用 ucs 巨集套件或 CJK 的 UTF-8 編碼	36
9.4.2	使用 Omega (Ω)	37
10	後記	38
10.1	本文的 PDF 格式檔及原始 L ^A T _E X 文稿	40
11	GNU 自由文件許可證原文	42
	使用許可證聲明	50

前言

CJK¹ 當然不是我的啦！CJK 是大家的。這裡指的是我如何使用 CJK。而這裡的 CJK 指的是由 Werner Lemberg 所發展、維護的一套 macro 系統，用於 T_EX/L^AT_EX 幕後排版系統中，排版中日韓文。

本文是個人使用 CJK 的一些心得，由於英文字型方面，最近的 T_EX 系統都已附有全套的 Type1 字型，爲了中、英文字型的配合，嘗試擺脫長久以來使用 pk 點陣字的傳統，大膽的啓用 CID TrueType Font 及中文 Type1 字型。

以下所述雖然是針對 GNU/Linux 系統，但我相信對所有 Un*x 系統及 Windows 系統，在原理上應該都是相通的。目前 Mac OS X 系統已有朋友順利的安裝成功，其他系統上有朋友願意跳出來補充一下。

請注意一下，這些心得可能會有不正確的地方，或能力不足致做法上不是很妥當，而且寫作時間斷斷續續，可能會有前後不連貫的情形，因此，僅供參考，如果你照我的方法去使用，造成任何損失，個人是不負責的。:-)

¹<http://cjk.ffii.org/>

為什麼不再使用傳統的 pk 點陣字型？

目前的 $\text{T}_{\text{E}}\text{X}$ / $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 系統，大概都轉用 Type1 向量字，好處是縮放時不會有鋸齒狀，處理上也比較方便。但在預設的設定下，還是會將 Type1 字型轉成 pk 點陣字來使用¹，原因是檔案會比較小（指低解析度的時候）。但這會發生一些缺點，那就是拿到解析度不同的印表機列印時，結果會不相同，而且不太適合在螢幕上放大閱覽。

Adobe 公司的開放規格 PDF (Portable Document File) 使用上已經非常的普遍，而製作 PDF 檔，則以 Type1 字型才能讓顯示上得到較佳的效果，如果是使用 pk 點陣字的話，製作成 PDF 檔時，會轉成 Type 3 字型，可能會看起來糊糊的，比較傷眼睛（印表機印出來則差別並不大），編譯及顯示過程也會比較慢。

另外一個很重要的課題，那就是如何使 ps/PDF 檔縮小，利於網路傳輸，又不會損失檔案的品質，最重要的，也要能像一般文字檔般的 cut&paste&search，這就是為何我會捨棄傳統 pk 點陣字的原因了。

2.1 中文 Type1 字型

由於 Type1 字型的規格，只能容納 256 個空間，也就是只能放 256 個字，這在英文語系沒有問題，但中文光 Big-5 就有一萬多字，更不用說是其他更大的編碼字集了。很不幸的， $\text{T}_{\text{E}}\text{X}$ 的字型規格也是如此，只能容納 256 個字。

因此就有許許多多的前輩們想辦法使用 subfont 的觀念，把中文字打散，分裝在不同的 Type1 字型中，由數十個中文 Type1 字型組成一個完整的中文字型，經由 NFSS(New Font Selection Scheme) 的選字機制，來組合數十個分散的 Type1 字型成為一整個的中文字集，再透過字型定義檔 *.fd 來取用字型。

¹目前新的 $\text{t}_{\text{e}}\text{T}_{\text{E}}\text{X}$ 系統已預設使用 Type1 向量字了。

2.2 中文 CID-Keyed 字型

由於 Type1 字型實在是不太適合用於中文，於是 Adobe 公司，另外研發了 CID-Keyed font 字型規格，把字型的實體資料集中在一個檔案內，再另外使用索引、對應的方式來取用所需要的字型，這樣就不會受限於 256 個字的限制了。他仍然是和 Type1/TTF 一樣是向量字，因此用來製作 PDF 檔，在顯示上不會如 pk 點陣字型那麼醜。

除了字數不受限的好處外，CID-Keyed font 的精妙處，就在於「索引、對應」，由於用索引的方式取字，因此，我們一般文字處理上要用到的 cut&paste&search 就可以很容易的達成。這在使用 subfont 方式的中文 Type1 及 pk 字型就沒有這個好處。而且，由於可以索引、對應，也代表著，實體字型資料不必一定要放在檔案中，可以放在使用者的系統中就行了，就像現在的 HTML 格式一樣，可以不必帶著字型跑。這對 ps/PDF 格式尤其重要，因為這些格式雖然顯示上很精緻，但如果嵌入字型的話檔案大了些，在網路傳輸上不很方便。

對 T_EX/L^AT_EX 系統而言，這個好處並不會帶來嚴重的缺失，因為視使用場合，也是可以把部份，甚至全部的字型給嵌入檔案中的，不像 HTML 格式，目前要內嵌字型的話，標準還不統一，IE 可以做到，但其他 web browser 就無法閱覽了。像這篇文章的 PDF 格式中，標題的顏楷體就是嵌入王漢宗博士捐贈的 TTF (CID TruType)，仍然可以 cut&paste&search。

如果各位對這些字型檔規格有興趣，可以到 [Adobe](#) 公司去下載 spec 規格檔來研究，這些規格都是公開的格式。

2.3 中文 TTF 字型

這是目前使用上相當普及的字型規格，也因此，許多 hacker 們的腦筋都會動到 TTF 上，例如 ttf2pt1 可以將 TTF 轉成 Type1 字型來用，而 DVIPDFM² 及較新版的 GhostScript 也可利用 TTF 當成是 CID-Keyed font 來使用。甚至以往的 pk 點陣字，也是由 ttf2pk 來由 TTF 轉出來的。

底下所要介紹的，就是預設使用系統上就有的中文 TTF 來當成是 CID-Keyed font 做為 CJK 預設要使用的字型，先不論高解析度時的品質問題，就我們一般的使用，這個由 TTF 來轉成其他格式的字型，作業系統及 L^AT_EX 系統共用，可真的是「一兼二顧」了！

²<http://project.ktug.or.kr/dvipdfmx/>

安裝說明及相關工具

這大概是多數人比較感頭疼的部份（包括我個人在內：），在能力所及，我盡量朝原理方面來述說，因為作業系統的煩多及各種發行套件的差異，目前實在還沒有把握做到全自動的偵測、安裝，大概只能達到半自動的目標，希望有興趣的朋友，共同來想辦法達到全自動的境界。

3.1 teTeX 系統

這應該是所有 GNU/Linux distribution 都會有，要注意的是，TeX 系統的設定相當複雜，有時一個錯誤訊息，對於剛接觸的朋友而言，會是有字天書，不知從何查起。因此，這裡建議把 distribution 裡頭，凡是和 tetex 有關的 packages 都安裝起來，一般就是開頭是 `tetex-*` 的套件。

想自行編譯的朋友，可在以下 ftp 站取得原始碼：

<ftp://tug.ctan.org/tex-archive/systems/unix/teTeX/2.0/distrib/>

原始碼會分成三個部份：

```
tetex-src-2.x.x.tar.gz
tetex-texmf-2.x.x.tar.gz
tetex-texmfsrc-2.x.x.tar.gz
```

其中 `x` 是真正的版本編號，第三個非必要，但建議抓下來參考用。

當然，前面已交待過，本文主要是說明以前的舊版本，在使用 teTeX 2.x 的版本，如果有碰到什麼問題，再提出來大家討論看看。

在其他 OS 的話，例如 MS Windows 系統，則是使用 MikTeX 或是 fpTeX 所發行包裝的發行本，都可以很方便的達到自動安裝，但是 CJK package 則不一定會安裝，有時可能得自

行選擇以才能完整安裝。在其他 Un*x 系統及 Mac OS X 的話，則可以和 GNU/Linux 一樣的安裝 $\text{teT}_{\text{E}}\text{X}$ 系統。

3.2 其他相關套件

3.2.1 t1utils

這是處理 Type1 字型的工具集，其中的 `t1asm` 會在由 TTF 製作 Type1 字型時要用到。`t1utils` 的原始網站在：

<http://www.lcdf.org/~eddieltwo/type/#t1utils>

如果，您系統上就已經有 `t1asm`，那這個 package 可以不必安裝。Mandrake 系統的話，他有收錄這個工具，因此把安裝光碟 mount 上來，安裝上去就可以了。

我們需要用到的是 `t1utils` 裡頭的 `t1asm` 這支程式，這支程式在 `ttf2pfb` 及 `ttf2pt1` 中都會附上，但在 `t1utils` 的版本比較新，建議使用這個版本，由於 `dvips` 的 partial font download 的功能很敏感，請使用最新版本的 `t1asm` 來製作中文 Type1 字型。

3.2.2 ttf2pt1

這是轉 TTF 為 Type1 的工具，原始網站在：

<http://ttf2pt1.sourceforge.net>

3.2.3 ttf2pt1-chinese

這是 `ttf2pt1` 在轉中文 TTF 時所要的 Unicode 對應表，原始網站同 `ttf2pt1`。

3.2.4 FontForge

這是由 [George Williams](#) 所發展的一個相當棒的字型編輯軟體，可以編輯大多數的向量字型及一般常用的點陣字型，在 FS/OSS 社群而言，這是一個不可多得的字型工具，集各種字

型處理的工具於一身。以前稱為 PfaEdit 但由於也可以用於其他種類的向量字型，因此改名稱為 FontForge。

這個工具不僅有 GUI 的圖型界面可以使用，而且還可以當 script interpreter 來使用，也就是說，可以寫 FontForge 的 script 檔，在命令列就可以執行，不必叫出 GUI 出來，請參考第 4 章，第 4.3 節，頁 13 的說明。

可在 FontForge 的網站下載原始碼，或打包好的 package。

<http://pfaedit.sourceforge.net/index.html>

也有其他平台的版本。但由於 FontForge 尚未對記憶體運用做最佳化，如果要處理中文字型的話，你的系統最少需要有 256MB 以上 RAM。如果是大字集的 Unicode 字型，可能需要更多的 RAM 才行。

3.2.5 DVIPDFM_x

這就是使用 TTF/CID-Keyed 字型，內嵌或不內嵌字型於 PDF 的工具，原始站在：

<http://project.ktug.or.kr/dvipdfmx/>

如果是製作 PDF 檔的話，其實，整個重頭戲就是在這個 DVIPDFM_x，這要感謝韓國及日本朋友的幫忙。

這裡要非常注意的是 te_TE_X 系統，在舊版本並沒有包括 dvipdfmx 所需要的 map 檔，所以要自行想辦法，可由 Adobe 公司下載，或者由底下這個網址取得：

<http://info.sayya.org/edt1023/tex/slk91/source/>

把他置於 \$TEXMF/dvipdfm/config 目錄下即可，如果你的 te_TE_X 是 2.x 以前的舊版本，那要在 dvipdfmx.cfg 裡頭加入：

```
f cmr.map
f psbase14.map
f lw35urw.map
```

如果是 te_TE_X 2.x 的版本，則無需以上舊版本的 map 檔，只在 dvipdfmx.cfg 加入：

```
f dvipdfm.ndl14.map
```

即可。其中那個 `ndl` 就是代表 `no download` 的意思，就是字型不會內嵌（當然，要配合一般 `Type1` 字型的套件，例如 `pslatex/times` 等），如果無論如何一定要內嵌，那要換成 `dvipdfm_dl14.map`。

中文字型的部份則加在 `cid-x.map` 檔裡頭，如果是使用 `mkfont.sh` 來製作中文 `Type 1` 字型的話，那會自動產生給 `dvipdfmx` 使用的設定，只要把這些設定加進去就可以了，請參考第 4 章，第 4.2 節的說明。如果 `cid-x.map` 沒有定義所需要的 TTF 字型，則會取自 `dvips` 所能取到的字，可能是 `pk` 或 `Type1` 字型，有 `Type1` 的話，會優先使用。

由於各種 `distribution` 置放 TTF 字型的位置不一樣，所以，這裡請手動連結一下：

```
cd /usr/share/texmf/fonts/truetype （如果沒有這個目錄，請自行建立）
ln -s yourpath/bsmi00lp.ttf bsmi00lp.ttf （請找出實際字型放在哪裡）
ln -s yourpath/bkai00mp.ttf bkai00mp.ttf
ln -s yourpath/gbsn00lp.ttf gbsn00lp.ttf
ln -s yourpath/gkai00mp.ttf gkai00mp.ttf
```

然後執行一次 `texhash`（或 `mktexlsr`）。

3.2.6 CJK 巨集套件

這就是 CJK 套件了，由於他預設並沒有附字型，字型是要由使用者想辦法，這也是本文的目的之一，原始網站在：

<ftp://ftp.ffii.org/pub/cjk/>

請參考所附文件來編譯、安裝。

3.2.7 hyperref

這是新版的 `hyperref` package，原始檔案在：

<http://www.tug.org/applications/hyperref/>

這有寫好的 `Makefile`，應該不難編譯安裝。這個 package 主要的功用是可以配合 CJK 來產生中文的目錄的超連結交互參照及 PDF 檔的 `bookmarks`。

3.2.8 L^AT_EX2HTML

這個在 Mandrake Linux 已經有附上，可以自安裝光碟安裝即可。Red Hat Linux 的話，他所附的是日本方面修改過的版本，可能會無法用於中文，其他沒有附上的 distro，請由 source code 編譯安裝，原始網站在：

<http://www.latex2html.org/>

這是 Perl 語言寫的，所以系統上要安裝有 Perl 才行。

3.2.9 L^AT_EX macro/package 的安裝

由於目前 t_EX 已全面成為自由軟體（Free Software），因此有些套件並沒有附上，有需要用到的話，要自行從 CTAN（Comprehensive T_EX Archive Network）下載安裝：

<ftp://ftp.tex.ac.uk/tex-archive/>（英）
<ftp://ftp.dante.de/tex-archive/>（德）
<ftp://ftp.tug2.cs.umb.edu/tex-archive/>（美）
<http://tug.ctan.org/CTANfind.html>（可搜尋的網頁）

絕大部份的 L^AT_EX macro 都採用 *literate programming* 的風格在書寫，也就是說文件、macro 程式碼是層次分明的混在一起的，經處理後才會分開來。簡單的 macro 通常就只有兩個檔 *.ins (installation script) 及 *.dtx (documented L^AT_EX source)，甚至只有一個 *.sty 檔。如果是大型的 macro/package¹，則可能有寫好的 Makefile，只要使用編輯器打開這個 Makefile，修改成適合自己的系統，然後：

```
make;make install
```

就可以了（舊套件要先移除）。

至於 *.ins/*.dtx 可以使用 latex/makeindex 來編譯，以下是一般的執行方法：

```
latex newpkg.ins
latex newpkg.dtx
latex newpkg.dtx（執行第二次，編譯 toc，如果有的話）
makeindex -s gind.ist -o newpkg.ind newpkg.idx
makeindex -s gglo.ist -o newpkg.gls newpkg.glo
latex newpkg.dtx
```

¹一般的說法，數個 macro 組成一個功能更完整的 package，其實裡子都是 T_EX L^AT_EX macro，因此本文並不嚴格區分。

然後手動 copy *.sty 檔至適當目錄即可（可仿原來存在的 package 來置放）。

```
dvips -Ppdf -o newpkg.ps newpkg (產生 newpkg.ps 文件)  
或  
dvi2pdf newpkg (產生 newpkg.pdf 文件)
```

將文件置放於 \$TEXMF/doc 目錄下適當位置即可。

這是在沒有 README/INSTALL 之類的文件之下的方法，如果有這些文件則以文件內的說明為準。最後，別忘了執行 texhash（或 mktexlsr），在 Windows 系統的 MiKTeX² 則執行 initexmf -u。然後，測試一下是否可以正常使用。

以上只是簡單套件的安裝，複雜的可能會需要產生字型或 *.fmt 檔，這時有時要等一下，可能會執行一段時間。沒有必要，當然是不建議手動安裝，請教一下有經驗的人會更保險。

²MiKTeX 可由選單(MikTeX Options) 安裝 MiKTeX 有收錄的 packages。

製作中文 Type1 字型

如果想使用 dvips 來產生 ps 檔，那這裡還要製作中文 Type1 字型。當然，一般使用的話，應該是可以不必製作中文 Type1 了，直接由 dvipdfmx 抓取系統上的 TTF 就可以了，但由於有時要和一些處理 eps 的 packages 配合，例如 PSTrick，所以這裡一併把 Type1 字型製作出來。

4.1 需要哪些字型資料？

給 T_EX/L^AT_EX 處理的話，只需要 *.t_fm 檔就可以運作，生成 *.dvi 檔。但要由 dvips, dvipdfmx, pdf_lat_ex 處理的話，要有實體的字型才行，因為他們的輸出不是 ps 檔，就是 PDF 檔。

4.1.1 Free 的字型的取得

目前中、日、韓都有 free 的字型可以使用，字型品質也還算不錯，可以至以下這些網站下載：

<ftp://cle.linux.org.tw/pub/fonts/>

當然，目前 GNU/Linux/*BSD 的發行版本都會附上一些字型，所以請先檢查一下你的安裝光碟或官方網站是否已有整理打包好的字型套件，這樣安裝起來會比較方便。再把字型連結到 T_EX 系統置放字型的目錄下就可以了，這樣就可以和系統共用字型。

4.1.2 各種字型資訊檔的意義

- *.tfm (T_EX Font Metric, 字型描述檔)

為二位元檔，T_EX 有這個字型描述檔就可以產生 *.dvi 檔，但其中是沒有真正的字型實體資料在裡頭，得由其他的輔助工具來完成。任何型式的字型，要給 T_EX 使用前，都一定要先產生這個 tfm 檔，而且 T_EX 只要有這個 tfm 檔就可以運作。

- *.afm (Ascii Font Metric)

這也是字型描述檔，但是以純文字的格式存在，要給 L^AT_EX 使用的話，可以由 afm2tfm 來產生 *.tfm 檔。一般的 Type 1 字型會只附 afm 檔，而沒有 tfm 檔。TTF 的話，可以由 ttf2tfm 來產生。

- *.mf

這是 Knuth 教授設計的 METAFONT source，他本身是一種 outline font，但由於顯示/成形 (render/raster) 較複雜，目前不適合像 Type 1/TTF 一樣 on-the-fly 使用，是另產生 bitmap 形式 *.gf (generic font) 再由 gftopk 產生 pk 字型來使用的。METAFONT 的可執行檔就是 mf，mf 在 font-making mode 時可以產生 tfm 檔。METAPOST¹ 也是可以用來製作字型，他的延伸檔名通常是 *.mp，可執行檔則是 mp 或 mpost。

- *.pfa (POSTSCRIPT Font Ascii)

這是 POSTSCRIPT Type 1 字型檔，但是以純文字的格式存在，可由 t1asm 來轉成 binary 格式 (即 *.pfb)。

- *.pfb (POSTSCRIPT Font Binary)

和 *.pfa 同，但是以 binary 格式存在。兩種格式目前都可以直接使用。

- *.pk (packed font)

這是給 dvi viewer/printer/dvips 使用的點陣字，以便真正產生字型嵌入 ps 檔中或供 printer 使用。當然，這個 pk 字，可能是由 METAFONT、Type 1 或 TTF 轉成的。

請注意這裡的表示方法，pk 字型是由其解析度來成為檔名的一部份的，例如 600 dpi 的 pk 字型，實際上的檔名是 *.600pk (gf 字型的表示法亦同)。

¹這是由 John D. Hobby 仿 METAFONT 開發的，經 Knuth 教授的許可借用了許多 METAFONT 的程式碼，改進了 METAFONT 的繪圖功能，並把輸出的格式改成了 POSTSCRIPT 的格式，連 Knuth 教授後來也很愛用 METAPOST。:-)

- *.fd (font description)

這是給 L^AT_EX 找字型用的字型定義檔。

4.1.3 尋找 L^AT_EX 所需的資訊

這些字型資訊的置放處都會有一定規則，這可以參考一下 TDS(T_EX Directory Structure) 文件：

```
http://tug.org/tex/tds
ftp://ftp.dante.de/tex-archive/tds/tds.pdf
```

也可以由 L^AT_EX 系統的一個函式庫 kpathsea 來找，當然，你的工具本身要有利用到這個函式庫才行，或者要直接使用 kpathsea 所附的工具，例如：

```
kpsewhich -format="dvips config" config.ps
kpsewhich econometrika.bst
kpsewhich bsmi00lp.ttf
kpsewhich -expand-var=$TEXMF （找安裝所在）
kpsewhich -show-path=texpool （找 memory dump 放在哪裡）
```

對 kpsewhich 而言，他的參數 -format 和 --format 是一樣的。

4.2 由 mkfont.sh 來製作需要的資料

這是製作中文 Type1 的一個 sh script，主要是利用 ttf2pt1。請先下載：

```
http://edt1023.sayya.org/tex/mycjk/mkfont.sh
```

並將這個 mkfont.sh 和字型置於同一個獨立的目錄中，製作方法如下：

```
./mkfont.sh your.ttf name
```

其中，*your.ttf* 就是 TTF 字型的實際名稱，後面的 *name* 指的是，在 CJK 文稿中所要使用的字型名稱。需要注意的是，這個 mkfont.sh 是使用 ttf2pt1 為工具，所以，目前只能用於 Unicode 字型，而且只能用於 Big-5 及 gb2312² 兩種編碼。這裡要多謝許多朋友的幫忙，可以到 telnet://bbs.sayya.org 的 ttfm 版查看以往的討論情形。

²這個部份 Goldencat 幫了不少忙，謝謝他。

以下是一個製作的範例：

```
mkdir aming akai
cd aming
ln -s yourpath/bsmi00lp.ttf bsmi00lp.ttf
./mkfont.sh bsmi00lp.ttf aming
cd ../akai
ln -s yourpath/bkai00mp.ttf bkai00mp.ttf
./mkfont.sh bkai00mp.ttf akai
```

其中 aming/akai 就是我們在文件中使用的字型名稱。經過以上的製作（可能會需要一點時間），會產生各種子目錄，並會產生一些定義檔，其中的作用，這裡微說明一下：

目錄或檔案	作用	置放位置
\$FNAME-afm/	ascii font metric	\$TEXMF/fonts/afm/CJK
\$FNAME-tfm/	T _E X font metric	\$TEXMF/fonts/tfm/CJK
\$FNAME-pfb/	Type1 字型檔	\$TEXMF/fonts/type1/CJK
\$FNAME-enc/	給 pdfT _E X 抓取 TTF 用的編碼對應	\$TEXMF/dvips/CJK
\$FNAME.fd	給 CJK 使用的字型定義	\$TEXMF/tex/CJK/Bg5
cid-x.map	給 dvipdfmx 使用的 map 檔	\$TEXMF/dvipdfm/config
\$FNAME.map	給 dvips 使用的 map 檔	\$TEXMF/dvips/config

其中 \$FNAME 指的是字型名稱，就是執行 mkfont.sh 的第二個參數 *name*，也是我們在文稿上使用的字型名稱。cid-x.map 的內容是附加至原來的 cid-x.map 檔中，而不是取代原來的。\$TEXMF 就是 T_EX 系統安裝的地方，包括他會去搜尋到的其他目錄。

接下來，在 \$TEXMF/dvips/config/config.ps 檔中加入：

```
p +akai.map
p +aming.map
```

最後，執行一下 texhash 即可。

4.3 由 FontForge 來製作字型

在第 3 章，第 3.2.4 小節曾提到的字型工具，他也是可以拿來製作中文 Type1 字型給 CJK/L^AT_EX 來使用。除了可以叫出 GUI 來製作外，也可以由 script 命令稿來製作。底下就是一個 FontForge script 的例子：

```
#!/usr/bin/env fontforge
```



```

# Generate CJK subfonts from master font.
# The fonts are created in the current directory.
# genfont.pe
# /usr/share/texmf/ttf2pk/UBig5.sfd
# /usr/share/texmf/ttf2pk/UGB.sfd
# $1: The master font (e.g. 'bsmi00lp.ttf')
# $2: The name stem for the subfonts (e.g. 'bsmilp')
# $3: The subfont definition file (e.g. 'UBig5.sfd')

if ($argc != 4)
    Print("usage: [fontforge -script] ", $0, " master-font name-stem SFD-file");
    Quit(1);
endif

Print("Loading ", $1, "...");
Open($1);
SetFontOrder(3)
SelectAll();
Print("Scaling to PostScript units...");
ScaleToEm(800, 200);
Print("Simplifying outlines...");
Simplify(0,3)
Print("Rounding coordinates to integers...");
RoundToInt();
ClearHints();
Print("Generating subfonts...");
Generate($2 + ".mult", "", -1, -1, $3);
Close();
Quit(0);

```

這是由 [Werner Lemberg](#) 所舉的一個簡單例子，經 FontForge 的作者 [George Williams](#) 建議改一下 `Simplify()` 函數的參數，以便符合中日韓字型。存檔成 `genfont.pe` 並 `chmod +x genfont.pe` 即可。

他會同時產生 `*.afm` 檔，這樣就可以利用 `afm2tfm` 來生成 `*.tfm` 檔了。底下是執行的例子：

```
./genfont.pe bsmi00lp.ttf aming /usr/share/texmf/ttf2pk/UBig5.sfd
```

當然，他並沒有產生相對的一些 `map`、`*.fd...` 等等額外的檔案，這些要自行參考 `mkfont.sh` 來製作，或自行寫個 `sh script` 來製作也是可以。

4.4 文稿測試

如果一切都弄妥當了，現在可以做個總測試了，先編一個測試文稿：

```
\documentclass[12pt,a4paper]{article}
\usepackage{CJK,pslatex}
\begin{document}
\begin{CJK}{Bg5}{aming}
這是 {\sf CJK} 的測試，終於可以方便製作不內嵌中文的 \LaTeX/PDF 文件了。
\end{CJK}
\end{document}
```

編好後存檔成 `test.tex`，然後：

```
bg5latex test.tex
dvipdfmx test          => 由 test.dvi 產生 test.pdf
dvips -o test.ps test  => 由 test.dvi 產生 test.ps
```

這樣 `test.pdf` 是不內嵌中文，連英文也不內嵌（引入 `pslatex`），而 `test.ps` 則是內嵌了中文 Type1 字型的。其中 `dvips` 預設會導向印表機，所以要有個 `-o` 參數，如果要改成輸出至檔案，可以修改

```
$TEXMF/dvips/config/config.ps
```

將 `o1lpr` 改成 `o`（英文字母小寫 `o`）或編輯一個 `~/.dvipsrc` 檔，檔案內容就是一個 `o` 就可以了，或乾脆就整個註解起來就可以了。

CJK 語法簡介

CJK 是一組在 \LaTeX 下使用的 macro/environment，因此最先要具備的，當然就是 \LaTeX 的語法了，再來才是 CJK 中的特殊語法。其實，只要 CJK 的大結構搞清楚，重要的語法還是在於原來的 $\text{\TeX}/\text{\LaTeX}$ 裡頭。

5.1 \LaTeX 的基礎文件

以下這些文件，都是給想接觸 \LaTeX 的朋友而寫的，相當具參考價值，請多多利用。如果不清楚一般要用到的 \LaTeX 語法，那學 CJK 的語法是沒什麼意義的。

5.1.1 一些值得參考的入門文件

系統上的 `lshort.dvi`

<ftp://ftp.dante.de/tex-archive/info/beginlatex/beginlatex.pdf>
<ftp://ftp.dante.de/tex-archive/info/simplified-latex/simplified-intro.pdf>
<ftp://ftp.dante.de/tex-archive/info/> 目錄下有許多文章可以參考

整理相當完整的網路教材

`SITE=http://www-h.eng.cam.ac.uk/help/tpl/textprocessing`
`$SITE/latex_basic/latex_basic.html`
`$SITE/latex_advanced/latex_advanced.html`
`$SITE/latex_maths+pix/latex_maths+pix.html`
請自行將 SITE 的內容取代進去連成一個完整的網址。

吳聰敏、吳聰慧兩位老師的《 cwTeX 排版系統》修訂 2 版

<ftp://ftp1.sinica.edu.tw/pub2/tex/cwTeX/cxbook-s.pdf>
雖然說的是 cwTeX 但其實道理是相通的。

〈大家來學 L^AT_EX〉一文

<http://edt1023.sayya.org/tex/latex123/index.html>
<http://edt1023.sayya.org/tex/latex123/latex123.pdf>
<http://MathNet.math.tku.edu.tw/~edt1023/tex/latex123/index.html>
<http://MathNet.math.tku.edu.tw/~edt1023/tex/latex123/latex123.pdf>

FAQ (許多問題可以從這兒查到初步資料, 可抓 pdf 檔, 比較好查詢, 網站上也有搜尋的功能)

<http://www.tex.ac.uk/faq>
<ftp://cam.ctan.org/tex-archive/help/uk-tex-faq/newfaq.pdf>

Hypertext Help with LaTeX (簡介及指令、環境、資源速查表)

<ftp://ftp.giss.nasa.gov/pub/sgreen/latex/latex.tar.gz>

安裝 Kile 就會有個 LaTeX 指令速查表 (latexhelp.html)

<http://perso.club-internet.fr/pascal.brachet/kile/>

看一些 macro 時, 會需要 TeX 指令速查表

<http://www.tug.org/utilities/plain/cseq.html>

貪多嚼不爛, 請先選定一份文件下功夫, 純熟後其他的文件自然就會很容易吸收。

5.2 CJK 中的特殊指令

其實, 依 CJK 的作者 **Werner Lemberg** 的說法, CJK macro 著重的重點, 主要是給非使用 CJK 語文為主的人士使用的, 因此是一個 environment 的形式出現, 在有需要的部份加入這個環境就可以寫入 CJK 文字, 不被這個環境包圍的部份仍然是英文模式, 而不是打算將整個 T_EX/L^AT_EX 系統中日韓化。

這樣的好處是, 多國語文可以同時並存於同一份文件, 各用各的編碼, 缺點就是常會發現和其他 package 衝突, 除非將來 L^AT_EX team 願意納入 CJK macro 的程式碼, 否則這些 macro 間的衝突是難免的, 畢竟 T_EX 系統本就不是專為多字元語系所寫的。

以下只針對常常要用到的指令來說明, 其他的可以參考 CJK 所附的文件, 也有中文的。

5.2.1 CJK 的主要架構

以下先簡單說明 CJK 的架構，當然也是一般 L^AT_EX 文件的架構。

```
\documentclass[12pt,a4paper]{article}
\usepackage{CJK}
...
這裡是 preamble 區
...
\begin{document}
...
這裡可以寫非 CJK 的文字
...
\begin{CJK}{Bg5}{aming}
...
這裡才是本文區，可以寫中日韓各種不同編碼的語文
目前是定在 Bg5，就是我們的 Big-5 編碼
...
\end{CJK}
...
這裡可以寫非 CJK 的文字
...
\end{document}
```

所以，可以看得出來，CJK 環境包住整個文稿內文區的話，那整篇文稿都可以使用中文，也可以包住部份需要中文的地方，而且這個環境可以重複使用。一篇文稿中要使用多種不同編碼的文章也是可以，只要各 CJK 環境使用不同的編碼就可以了，但有些編碼需要前置處理，有些則不必，這裡要注意一下。

5.2.2 變更字體

字體除了在進入 CJK 環境時指定，也可以途中變更，例如：

```
\CJKfamily{akai}
```

這會改變成 akai 的字體。當然，如果不是在環境中，或由大括號括住，那要記得回復原字體，否則以下的字體都會通通變成楷體。另外，也可以另起一個新的 CJK 環境，並指定不同的字型。

5.2.3 變更字距、行距

字間距是由 `\CJKglue` 所定義，他的預設值是：

```
\newcommand{\CJKglue}{\hskip 0pt plus 0.08\baselineskip} CJK 預設值
\renewcommand{\CJKglue}{\hskip -0.3pt plus 0.08\baselineskip} 本文的定義
```

可以變更他的 `\hskip` 為負值，就會縮小字間距。行距的話，和英文環境一樣，是由 `\linespread` 來控制，只要改變他的參數，就會照原來字體大小的倍數縮放。例如：

```
\linespread{1.2}
```

則會放大成字體大小的 1.2 倍，這裡所謂的行距，指的是兩 baseline 的距離，就是這一行字的底線至下一行字的底線的距離。把這些設定在 preamble 區，則對整篇文稿都會有效果。

一般的 `verbatim` 環境，常會造成無法正常折行的問題，這可以改用 `CJKverbatim` 來代替。

如果在編譯時常會有 ‘`Overfull \hbox`’ 的錯誤訊息，可以改變 `\CJKtolerance` 的容忍度，他的預設值是：

```
\newcommand{\CJKtolerance}{400}
```

另外，`CJKindent` 是專為 CJK 語文所設的，會內縮兩個中文字，不然，一般的內縮是在一至兩個中文字之間。但這要在 CJK-4.5.1 以上的版本才支援。

5.3 關於中英混合字間距及中文斷行

這個問題困擾許多人，因為 $\text{T}_{\text{E}}\text{X}$ 不會去辨識是不是中文字，以致於我們按 Enter 鍵斷行， $\text{T}_{\text{E}}\text{X}$ 會認為要加入一個空白，因為這在英文環境的確是字（word）間空白，但中文就不是了。而且，中英文間是否要有個空白？這個空白應該要多大？恐怕也沒有確定的結論。以下是幾個解決的方式。

5.3.1 編輯時手動加入必要的控制

這完全是抓 $\text{T}_{\text{E}}\text{X}$ 的特性去適應的，可能是比較笨的方法，但如果習慣了的話，問題可能比較少。也就是說中英文間，主動在編輯時就加入一個空白。而換行的話，可以在每行最後加一個百分號 %，或以我個人的習慣，中文的話是一行到底不按 Enter 鍵，這在一般的編輯器會有「假性」折行，也就是螢幕看起來有折行，實際上是沒有。

也有人就乾脆使用英文逗點，然後在英文逗點後按 Enter 鍵換行，這樣的話，單字元的英文逗點，後面再加個單字元的空白，看起來就不難看了，當然，這在中文句點就沒有替代的方法。

5.3.2 使用 CJK* 環境

也可以使用 CJK* 環境來解決，也就是說，原來的環境宣告要變成：

```
\begin{CJK*}{Bg5}{aming}  
...  
\end{CJK*}
```

加個星號，這有兩種作用：

1. 去除中英文字間的空白。
2. 按 Enter 鍵的斷行，自動去掉換行空白。

那問題來了，中英文字間距怎麼辦？這可以使用 `\CJKtile`，然後在中英文字間加個波紋號 ~，這個空白會比一般的英文空白大一點點。這在文章不長的情況下還可以忍受，長篇文章的話可能打字起來不會很方便。

5.3.3 由程式本身去解決

像 $\text{cwT}_{\text{E}}\text{X}$ 及 $\text{ChiT}_{\text{E}}\text{X}$ 就由程式本身去解決這類問題，這種處理方式可能會比較完善。有興趣的話，也可以自行寫程式或 macro 去前置處理，不過，要小心的是在 $\text{T}_{\text{E}}\text{X}$ 的世界，「地雷」很多，別誤觸「地雷」了。:-)

5.4 直排的問題

在東方語文中，會有許多機會要使用到直排。在 CJK 中，也可以處理直排的排版，但是 CJK 的版本要新一點的才行（最好是取得 CVS 中的版本），而且標點符號要另外使用直排的標點符號，這可以從 CTAN 中的 T_EXLive 檔案中取得：

<http://tug.org/ftp/texlive/Contents/live/texmf/fonts/>

只要引入 CJKvert package 就可以了，但他不會自動引入 CJK package，所以兩個都要引入：

```
...
\usepackage{CJK}
\usepackage{CJKvert}
...
```

其他就和正常橫排一樣的編輯即可，編譯過後自然會轉成直排。這裡有個實例供參考：

<http://edt1023.sayya.org/misc/big5vert.tex>
<http://edt1023.sayya.org/misc/big5vert.pdf>

他的原理是將每個中文字旋轉後再排入，這樣可以節省另一套直排字型，使用原有的一般字型就可以了，當然，也因此在操作上容易和其他的巨集套件有衝突的機會。這個套件仍在發展中，所以會有些小地方沒有注意到，但一般使用應該是可以用了。

5.5 漢字加點、畫線

東方語文也常會有加其他點標示重點或畫線的情形，例如中文的書名號、私名號……等等。這是由 Wenchang Sun 所貢獻的 macro，你必須使用 CJK-4.5.2（不含）以上的版本才有這個功能。當然，你可以從 CJK 的 cvs server 中取出最版的 CJK。

這個 macro 的名稱是 CJKfntef，他會自動載入 CJK package，所以，只要載入這個 macro 就可以使用 CJK 環境了。

```
...
\usepackage{color}      % 需要顏色放前面
```



```
\usepackage{CJKfntef}
[\usepackage{color}]    % 不需要顏色放後面
...
```

這些加點、畫線是有顏色的，如果要沒有顏色（就是和字體一樣是黑色）的話，只要把引用這個 package 的宣告時放在 color package 前面就可以了，要有顏色就一定要放在 color package 後面，這樣才能順利引用到顏色的定義。以下舉一些實例：

\CJKunderdot{中文字加點}	中文字加點
\CJKunderline{漢字畫底線}	漢字畫底線
\CJKunderwave{書名號底線}	書名號底線
\CJKunderdblline{漢字畫雙底線}	漢字畫雙底線
\CJKsout{漢字畫刪除線}	漢字畫刪除線
\CJKxout{漢字畫斜影線}	漢字畫斜影線
\CJKunderdot{\CJKunderline{也可以複合使用}}	也可以複合使用

當然，這個 macro 才剛完成，還沒有正式 release，難免會有些和原有 macro 衝突的情況發生，例如用在表格中時就會有問題，但這些變化算是滿方便的，一些臭蟲我們可以慢慢來改善。

CJK 和 \LaTeX 2HTML 的配合

HTML 格式目前還是網頁的主流， \LaTeX 2HTML 就是為此而寫的一組 macro 及 perl script 程式碼，功能很強大，也代表他的處理程序很複雜，值得一提的是，他支援 CJK package。但由於兩種格式的差異性，在使用上不免有削足適履的感覺，一些複雜的 \LaTeX macro 還是無法處理（每個 macro/package 都需要相對應的 perl 程式碼來處理），但一般使用應該是相當好用了，有必要的話，可以去修改 CSS 檔，來變更網頁的外觀，而且可以使用條件式的方式來寫 \LaTeX 文稿，達到一稿兩用，甚至多用的目的。

\LaTeX 2HTML 轉數學式為圖檔的能力很不錯，一般情形下，數學式子的表現要在目前的網頁運作下充分表現會比較困難，縱使做到了，大部分的瀏覽器也是會無法閱覽，因此做成圖檔或許是目前比較通用又不會失真太嚴重的方法。

當然，這裡只是很簡單的介紹，實際深入的運用要詳讀 \LaTeX 2HTML 的使用手冊。

6.1 \LaTeX 2HTML 語法簡介

如果是自行編譯 \LaTeX 2HTML，一定要記得 docs 子目錄下把 manual 編譯出來參考。雖然資料並不是很新，但主要的架構說明都在裡頭。另外，他的程式碼，有時也要參考一下，才能達到微調的功能。

6.1.1 引入 \LaTeX 2HTML

使用 \LaTeX 2HTML 的第一步，當然就是在 preamble 區引入¹：

¹就是宣告使用 package，載入 package 的意思。如果不需要一些 \LaTeX 2HTML 的特殊指令、環境，不引入也是可以的，但這樣一來就只能處理簡單的文稿了。

```
...
\usepackage{html}
...
```

請盡量引在最後面，就是在其他的 package 引入之後再來引入 html。

如果，有些 package，L^AT_EX2HTML 並不支援²，那得使用條件式的引入方法，以 fancyhdr package 為例，L^AT_EX2HTML 目前並沒有支援，在 preamble 區可以這樣設：

```
...
%begin{latexonly}
\usepackage{fancyhdr}
%end{latexonly}
...
```

這樣，L^AT_EX2HTML 就會忽略被 latexonly 包起來的部份。一些無法被 L^AT_EX2HTML 接受的自行定義的部份也是可以包在這個環境內，因為 L^AT_EX2HTML 目前並不完全支援自行定義指令，有些可以，有些則不行。請注意，起頭的是百分號 %，而不是一般的倒斜線³，這在 L^AT_EX 會被認為是注解，但在 L^AT_EX2HTML 則會忽略被這個環境包起來的 L^AT_EX 指令或環境。

6.1.2 簡單的條件式指令與環境

除了上一節所說的 latexonly 還有一些常用的條件式指令與環境。

- `\latexhtml{latex 敘述}{l2h 敘述}`

如果是簡短的敘述，可以使用這種方式來表示，第一個大括號是給 L^AT_EX 作用的，L^AT_EX2HTML 會忽略他，反之第二個大括號的內容則只會出現在網頁，L^AT_EX 會忽略他。長一點的內容的話，要使用條件式環境包住比較方便。

- `\htmlrule`

這是讓 HTML 格式的網頁劃一橫線。

- `\latextohtml`

這是 L^AT_EX2HTML 的 logo 特殊表現方式。

²可參考所安裝的 L^AT_EX2HTML 目錄下的檔案的檔名就可以知道是否支援。

³倒斜線也是可以，但由於計數器（counter）的行為不一樣，這裡建議使用百分號起頭的。

- `\tableofchildlinks`

這會把一些章節標題加入網頁目錄中，在使用 `\chapter*` 及 `\section*` 指令時因不會加入目錄中，可以使用這個指令強迫加入目錄，當然，這只對 HTML 格式的網頁有效。

- `\begin{htmlonly}...\end{htmlonly}`

包在這個環境內的所有指令都只能適用於 HTML 網頁，L^AT_EX 會忽略他。另一個相反的就是在第 6.1.1 小節談到的 `latexonly` 環境。

- `\begin{rawhtml}...\end{rawhtml}`

這環境是用於寫 HTML markup tags 時使用，L^AT_EX 會忽略他，只會作用於網頁。

至於更複雜的條件式判斷，可使用 `ifthen` package，詳細請參考該 package 所附文件。

6.2 L^AT_EX2HTML 和 L^AT_EX 配合需注意的地方

6.2.1 設定檔

L^AT_EX2HTML 的一些設定，可以設在 `$HOME/.latex2html-init` 這個設定檔內。例如，我個人的設定檔：

```
$ADDRESS = "<I>Compiled by Edward G.J. Lee ($address_data[1])</I>";
$charset = "big5";
$NO_NAVIGATION = 0;
$BOTTOM_NAVIGATION = 1;
$LOCAL_ICONS = 1;
$WHITE_BACKGROUND = 1;
$TRANSPARENT_FIGURES = 1;
1;
```

更複雜的設定方式，可參考 L^AT_EX2HTML 的 `manual` 或 `latex2html --help` 或 `perldoc latex2html` 來得知。

6.2.2 數學模式

在 T_EX/L^AT_EX 的數學模式，L^AT_EX2HTML 預設就是會把其中內容轉成圖檔，這對數學式子很方便，但對一些特殊符號可能就不是自己想要的了，例如 $\<$ ， $\>$ ， \backslash ... 等等，這時可以使用 L^AT_EX 的 `\textless`，`\textgreater`，`\textbackslash` 來取代，或進入 `\verb` 環境內，這樣 L^AT_EX2HTML 就不會去轉成圖檔。

6.2.3 圖檔的處理

我們一般在 L^AT_EX 使用 `\includegraphics` 所引用的圖檔都可以順利的由 L^AT_EX2HTML 來處理，但他是經由外部程式在轉換處理的，有時候會處理失誤，這時就只好做另外的處理了。

如果我們引用的是向量圖檔，包括 METAPOST 製作出來的 mps/eps 圖檔，L^AT_EX2HTML 可以轉換成 png（預設）或 gif（要另行指定）點陣圖，但有時圖檔的彩色較複雜時，如果想轉成透明圖，偶爾可能會失敗，這時只好手動使用 ImageMagick 的 `convert` 工具來事先轉好，而 L^AT_EX 文稿中引用圖檔時不使用延伸檔名，這樣一來 L^AT_EX 會去取用 eps 向量圖檔，而 L^AT_EX2HTML 則會取用點陣圖檔（png 優先），這樣就可以各取所需了。

如果講求精確性，那麼可以使用 `\htmladdimg` 這個指令來插入預先製作好的點陣圖檔，這時可以加入 HTML 引入圖檔時的一些控制選項參數，這樣就可以很精確的控制圖檔在 HTML 時的顯示方式。我們來看一下實際的例子：

```
...
%begin{latexonly}
\includegraphics{yourimg} % 這是給 latex 使用的
%end{latexonly}
\htmladdimg[WIDTH="297" HEIGHT="297" ALIGN="LEFT" VSPACE="10" HSPACE="30"
ALT="Image yourimg"]{yourimg.png}
...
```

在使用 latex 指令編譯文稿時，latex 會忽略 `\htmladdimg` 這個指令的所有敘述，這些敘述只會在 HTML 格式出現。當然，這些可能需要手動測試調整一下才會就緒。

6.3 L^AT_EX2HTML 和中文的配合

6.3.1 中文式網頁

如果想使用中文式的網頁說明，請下載以下的 style 檔：

<http://info.sayya.org/edt1023/tex/taiwan.perl>

然後，置於 \$YOURPATH/latex2html/lib/style 目錄下即可。並在同一個目錄下製作一個 taiwan.sty 的空檔案，請注意不要把這個空檔案刪除，否則轉換格式時會出錯。由於這個 package 是專為 L^AT_EX2HTML 而設的，所以引入時要由 htmlonly 環境包住。

```
\begin{htmlonly}
\usepackage{taiwan}
\end{htmlonly}
```

這樣，網頁上所呈現的一些文字按鈕說明會是中文的，但圖形按鈕則仍然是英文。這個 taiwan.perl 只完成一半，偶爾有可能會影響圖檔的產生。

由於網頁的中文斷行會發生和 T_EX 文稿一樣的情形，所以要注意按 Enter 鍵斷行所造成的英文單字間空白。

6.3.2 把中文字轉成圖檔來顯示

要讓中文字成圖檔，例如在 \mbox{} 及 \fbox{} 的時機，由於 CJK 的編碼方式，L^AT_EX2HTML 會找不到原字型的實際對應位置，這時可以利用 T_EX 的指令去指定字型對應，讓 L^AT_EX2HTML 找得到對應，就可以經由 dvips/gs 來取字轉成圖檔了。例如，本文造字的部份，由於並不在 Big-5 及 CJK 字碼裡頭，所以必需另行定義，L^AT_EX2HTML 當然會無法顯示，只能轉成圖檔來顯示。以下是定義的方法：

```
\font\newf=newfont01 scaled 1200
\def\uchr{\newf\char}
\def\Fwu{\uchr149}
\def\Shiuan{\uchr150}
\def\Kuen{\uchr151}
```

這樣一來，`\Kuen` 就會取 `newfont01` 這個字型的第 151 位置的字，不管是 L^AT_EX 或 L^AT_EX2HTML 都可以找得到字，只不過，網頁的部份是轉成圖檔而已。請注意，這裡 `scale` 成 1200 是配合文稿使用 12pt 的字，至於造字及其相關的議題，請參考第 9 章的說明。

但目前 L^AT_EX2HTML 對於字型轉成圖檔，由於面對的字型很多，很難取得統一的判斷方式，有時候會把字的上下多留了些空白，這樣在網頁上的顯示會和同一行的文字不協調，這可能還要更深入的去 hack，取得更通用的方式才行。但如果都是英文文件的話，就較不會發生這種情況。

當然，這個方法有點笨拙，像 `cwTEX` 的字碼轉換，就直接轉成取字的定義碼，這樣可以完全不必另行定義，L^AT_EX2HTML 也是取得到字轉成圖檔了。是不是有更理想的方式？應該是有吧！大家一起來努力。:-)

CJK 和 KOMA-Script 的配合

KOMA-script 不僅可以當成一般 \LaTeX 的標準 class 來使用，而且他多出了額外的指令及環境，原始創作者是 Frank Neukam，後來 Markus Kohm 大力參與改進、維護。目前使用有什麼問題的話，可以和 [Markus Kohm](#) 聯絡。

KOMA-Script package 支援 CJK 環境，主要是提供另一種 class，他的功能很多，但這裡只提和 CJK 配合運用的簡單介紹，其他的可自行參考他的使用文件（原來是德文，有英文版的說明）。為什麼會提到 KOMA-Script 呢？最主要原因是他支援 CJK 環境，因此可以很方便使用中文的章節名稱及國數字。

7.1 KOMA-Script 和 \LaTeX 標準 class 的對照

簡單使用的話，可以不必要理會那麼多，只要將 \LaTeX 標準的 class 名稱換成 KOMA-Script 的就可以了：

\LaTeX 標準	KOMA-script
article	scrartcl
report	scrreprt
book	scrbook
letter	scrlettr

當然，他的版面安排和標準的 \LaTeX class 並不太一樣，例如預設是使用 11pt 的字，a4paper 的紙張大小，內文可以運用的範圍會比較大些。

7.2 中文章節標題的定義

比較完整的方式是使用 KOMA-Script，配合 `\CJKcaption` 來達到目的，簡單例子如下：


```
\documentclass{scrreprt}
\usepackage{CJK}
\begin{document}
\begin{CJK}{Bg5}{aming}
\CJKcaption{Bg5}

\part{第一部}

\chapter{第一章}
\label{ch1}
這裡是第一章。

\part{第二部}

\chapter{第二章}
請參考 \ref{ch1}。

\end{CJK}
\end{document}
```

但如果沒有修改的話，預設 CJK 的章不會顯示中文，而是使用阿拉伯數目字。想自行修改的話，在 CJK，中文章節標題及國數字、日期格式的形式是定義在 Bg5.cap 這個檔（以 Big-5 中文為例），把這個檔由編輯器打開，會發現只定義「部」，就是 part 的，其他「章」（chapter）、「節」（section）都註解起來了，可以把這些註解拿掉，然後：

```
bg5conv < Bg5.cap > Bg5.cpx
```

來取代原來的。這樣 \ref{your-label} 時會顯示中文格式的章節、數字。在 \today 則會顯示中文的日期格式。

我本身的話是傾向使用『第 1 章』這種形式，可參考本文的原始文稿中，titlesec/titletoc 的用法。這樣 \ref{label} 的話，只會顯示阿拉伯數目字，使用上可能會比較有彈性。使用 KOMA-script 的話，會整個顯示「第一章」這種方式，而不是只有數目字，當然，這看場合而定，有時這樣顯示反而更方便，少打了兩個中文字。:-)

CJK 中如何編輯、編譯大型文件？

使用 CJK 的朋友，在編輯大型的文件時，常常會抱怨無法分開來寫，使得維護上會變得很困難，其實是可以的，以下的例子包括和 \LaTeX 2HTML 的配合。實際的文稿內容就不寫出來了，只寫出架構。

```
cjk.tex (主檔)
cjk1.tex (第一章)
cjk2.tex (第二章)
cjk3.tex (第三章)
```

cjk.tex 內容：

```
--- begin cjk.tex ---
\documentclass[12pt,a4paper]{report}
\usepackage{CJK}
\usepackage{html}           % 和 latex2html 配合

\begin{htmlonly}           % 這個環境只會在執行 latex2html 時有作用
\usepackage{taiwan}        % 執行 latex(bg5latex) 時會忽略
\end{htmlonly}

\begin{document}
\begin{CJK}{Bg5}{aming}    % 開始進入 CJK 環境

%begin{latexonly}          % 這行的第一個 % 對 latex 而言是註解
\input{cjk1.cjk}           % 對 latex2html 而言則是指令，會忽略
\input{cjk2.cjk}           % 這一整段
\input{cjk3.cjk}
%end{latexonly}

\begin{htmlonly}           % latex2html 會直接找 cjk1.tex...
\input{cjk1}
\input{cjk2}
\input{cjk3}
```

```
\end{htmlonly}

\end{CJK}
\end{document}
--- end cjk.tex ---
```

其中 *.cjk 的部份，可以事先由 bg5conv 轉出來，或由 Makefile/script 中控制。

```
bg5conv < cjk1.tex > cjk1.cjk
bg5conv < cjk2.tex > cjk2.cjk
bg5conv < cjk3.tex > cjk3.cjk
```

這樣就可以了，然後開始編譯：

```
bg5latex cjk.tex
dvips -Pcmz -o cjk.ps cjk.tex
latex2html [options] cjk.tex
```

這樣就會產生 cjk.ps 及在 cjk 子目錄下產生 HTML 格式的檔案。這樣，各個章節就可以分開維護了，不必編輯一整個大檔案。

其中的原理就是執行 bg5latex cjk.tex 時，\input 進尚未處理的 cjk1.tex 則會來不及前置處理，最後變成是亂碼。\\input{cjk1} 不加延伸檔名，預設的延伸檔名就是 *.tex，因此要寫成 \\input{cjk1.cjk}。

至於 \\include 指令，則由於他一定會認 tex 延伸檔名，因此沒有辦法這樣處理。

當然，這裡爲了方便說明，使用數目字去代表他的順序，實際運用可以命名一個代表性檔名會比較好維護，以免有時要加減幾個章節時，所有的檔名都要改變。

在 CJK package 裡頭亦有提供 \\CJKinput 及 \\CJKinclude 可以運用。

如何造字？

先不必驚訝、歡欣，這裡不是在說明從零開始的造字，只是利用現成的字型裡頭的字來加進去而已。至於從零開始的造字，往後有機會再來探討。而且，這裡以能用為原則，不討論字型品質的問題。

由於我們通常可以不受編碼影響的就是中文 Type1 subfonts，他可以透過 \TeX 內建的取字機制來取用字型裡頭的字。因此，這裡我們優先製造 Type1 字型。當然，要直接造 TTF 也是可以，但最後還是得多經一道手續來處理成 Type1 字型。其他字型的製作，已超出本文要討論的範圍，以後有機會再專文來說明。

這裡以 CJK 的中文 Type1 字型來說明，就是在第 4 章提到的字型製作所製作出來的中文 Type1 字型。中文字會加入新字，英文也當然也就會加入新字了。

9.1 為什麼需要造字？

這可真是我們長久以來的痛啊！我們目前慣用的繁體中文是使用只有一萬多個字的 Big-5 碼及其字集，大家經常會碰到，有些字硬是輸入法拼不出來，例如，煊、塋、髯……等等，雖然我們有大字集的 Unicode/CNS11643/CCCI，但是並沒有普遍的使用習慣，相對應的工具也很缺乏，這就是我們需要造字的原因。縱使有方便的大字集環境了，總是會有些新創的字，這也是需要造字。

這些字哪裡有呢？中國海字集有收錄，或一般較為完整的 Unicode 字型亦會有收錄。這裡以一般容易取得的 Unicode 字型為例，因為，如果要補其他字的時候，這種字型會收錄比較多，而且有正確的 Unicode 對應，找字較方便。

<ftp://ftp.netscape.com/pub/communicator/extras/fonts/windows/CyberCJK.ZIP>
<ftp://ftp.netscape.com/pub/communicator/extras/fonts/windows/Cyberbit.ZIP>

請注意，這個字型不是 free 的。使用時請注意他的 license 說明。或者，如果你在 Windows 系統有安裝中國海字集的話，也可以拿來使用。

9.2 選用的工具

這裡，我們選用開放原始碼的 FontForge，請參考第 3 章，第 3.2.4 節，頁 5 的安裝說明。他也可以直接讀入 Windows 系統的 TTE 的額外的字型，例如中國海字集所造的字型。

由 PfaEdiT 讀取 TTF 字型後，要調整字型的規格以符合我們所需（TTF 和 Type1 的規格並不一樣，而且我們得配合原先由 mkfont/ttf2pt1 所製作的字型），這裡以「髯」為例來說明：

```
fontforge Cyberbit.ttf
```

[Element] => [Font Info] => [General] 修改以下的值：

```
Ascent: 800      Descent: 200
EM size: 1000
Underline Position: -72  Height: 43
```

然後找到 uni9AF4（這是髯這個字的 Unicode 編碼），在這個字上 double click，然後 export 出 eps 圖檔，會產生 uni9AF4_BitstreamCyberbit-Roman.eps，這就是我們要加進去的字。這裡我們取用 mkfont.sh 製作出來的 bsmi00lp55.pfb，並把鏹改名為 newfont01.pfb。

```
mv bsmi00lp55.pfb newfont01.pfb
fontforge newfont01.pfb
[Element] => [Font Infos...] 改名為 newfont01
```

然後在 149 的格子 double click 把 uni9AF4_BitstreamCyberbit-Roman.eps import 至 149 的位置，並命名為 uni9AF4，然後存檔。這樣以後加其他的字就不必擔心名稱重複了。

這時再來重新產生字型：

[Generate Fonts] => [PS Type1(Binary)] => [Save]

產生 pfb Type1 字型的同時會產生 afm 檔，可利用 afm2tfm 產生 tfm 檔。這些檔案 copy

至 `$TEXMF/fonts/type1/CJK` 及 `$TEXMF/fonts/tfm/CJK`，執行 `texhash`。

9.3 定義、調整及取字

由於我們使用的輸入法及作業系統上的字碼對應並沒有建立，所以還是得靠 `TEX` 來輸入這些字。

```
\font\Bsmi=newfont01 scaled 1000
\def\Fwu{\Bsmi\char149}
...
山有小口，髣\Fwu{}若有光，便捨船，從口入。
```

如果是使用 12 pt 則要 scale 成 1200（視原字型的大小而定，有些需要適度調整大小及位置。那個原先沒有的字就以以上的取字方法代替，編譯後自然會取到字。

由於各種字型在造字時的大小不一樣，因此有些字型要調整一下才能配合原來的字型大小，例如，以 `Cyberbit.ttf` 而言，他的字型比較小，依我目前使用的情形，和 `bsmi00lp.ttf` 由 `ttf2pt1` 造出來的中文 `Type1` 字型要配合的話，要加入以下的調整：¹

```
\font\NewFont=newfont01 scaled 1300
\def\Fwu{\raise.1ex\hbox{\NewFont\char149}\kern.5em}
\def\Shiuan{\raise.1ex\hbox{\NewFont\char150}\kern.5em}
\def\Kuen{\raise.1ex\hbox{\NewFont\char151}\kern.5em}
```

調整他的大小，scale 成 1300 來配合 12 pt 的 `bsmi00lp` 字，而且字間距和上下位置也要調。這也可以由造字時調整，但是爲了配合其他不是 `bsmi00lp` 的字型，這裡暫時使用這種醜一點的方法。

`dvipdfmx` 是直接抓 TTF，而且用的是 Adobe CID-Keyed font 的對應，在這裡不能適用，要改的話要把字加入 TTF 中，也要修改字的對應方式，除非是把字型嵌入，不然可攜性會出問題，因此，我們這裡另造一個新的中文 `Type1` 字型，以便和原來的 TTF 區分出來，原因是，這樣才可以使用 `dvipdfmx`，這時原有的字的部份會去抓 TTF（嵌入或不嵌入），新造的字則會去取中文 `Type1` 字型（這會嵌入），這樣就不會有可攜性的問題產生了。當然，這裡要產生一個 map 給 `dvips` 找字用，可以另編輯一個 `newfont.map`：

¹其中的 `\Shiuan` 及 `\Kuen` 兩字，請 9.2 節的方法加入。

```
newfont01 newfont01 <newfont01.pfb
```

這要加在 dvips 的 config.ps 檔中。這樣一來，王建煊先生及游錫堃院長的名字就可以印得出來，桃花源記裡頭「髣髴若有光」，也可照原文印出來了。當然，每種字型的設計風格並不一樣，還是會看出一些不同。

這只是很簡陋的使用方法，並沒有納入 L^AT_EX PSNFSS 的選字機制，也就是說，他無法依 L^AT_EX 的字體大小機制來自動變化大小，也沒有粗斜體的變化，都要自行手動去另行定義，一般 L^AT_EX 裡頭的這些指令是不會生效的，因此，如果是有大量的造字時，最好就是要寫 macro，納入選字機制，使用上才會方便。

礙於字型版權的問題，這裡無法提供做好的 macro 給各位使用，如果有人有辦法和中國海字集的所有人聯絡上，而他們也同意的話，我目前已經做好的 macro 就可以釋放給大家使用，不必另行定義，已納入 PSNFSS 的自動選字。

我們也可以使用 ttf2pt1 來從其他含有這些字 TTF 字型來製作 Type1 字型，但是他的對應表就得自行想辦法才能取得到所需要的字，這對一般使用者可能會比較困難，但好處就是效率會比較好，有興趣的朋友也可以自行嘗試看看。

9.4 其他的替代方案

使用大字集，例如 UTF-8 或 CNS11643 的編碼來寫文稿。但這在目前，顯示、編譯上可能沒有問題（CJK 目前可以支援 UCS BMP²的部份，也支援我們的 CNS11643，而在 Un*x 系統的 zh_TW.euctw locale 就是使用 CNS11643），但輸入就是一個很大的問題了，我們目前並沒有很方便使用的這種大字集輸入法，而且編輯器也得支援才行。

9.4.1 使用 ucs 巨集套件或 CJK 的 UTF-8 編碼

其實 CJK 本身就有 UTF-8 編碼的環境可以利用，但是由於我們並沒有真正 free 的 Unocide 全字型，而且其中所能處理的 repertoire 並不是全面的。而 ucs 則支援較完整的 repertoire，

²Basic Multilingual Plane，基本上 UCS (Universal Character Set) 是由 128 個三維空間組成的，每個稱為一個 group，每個 group 由 256 個 plane 組成，每個 plane 又由 256 個 row 組成，每個 row 再由 256 個 cell 組成，在 Plane00, Group00 的空間就稱為 BMP，應該可以翻譯成「基本字面」，光這個字面就可以容納六萬多個字了。

而且 `ucs` 巨集不需要其他多迴語文的支援就可以獨立運作。這個部份，詳細的說明，請參考〈Ipe 與 `ucs` 巨集套件的配合〉一文：

<http://edt1023.sayya.org/tex/ipeucs/>
<http://MathNet.math.tku.edu.tw/~edt1023/tex/ipeucs/>

9.4.2 使用 Omega (Ω)

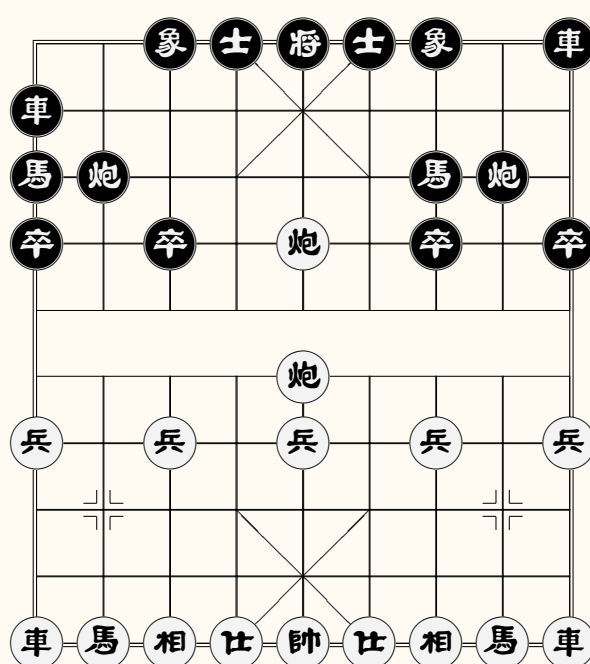
Ω 是新一代 16 位元處理的 $\text{T}_\text{E}\text{X}$ 系統，原始的 $\text{T}_\text{E}\text{X}$ 是 8 位元的方式來處理，所以，一碰到像我們中文這個多位元字元的書寫系統，就常常容易出問題。換用 16 位元的方式來處理的話，會省略許多不必要的處理過程。

但是，目前 Ω 系統對中文的處理還不是很圓滿，這裡我們可以參考李運璋博士的修正方法，詳細說明請參考〈Omega-16 位元科學排版系統之中文環境及 TrueType 字型產生〉一文：

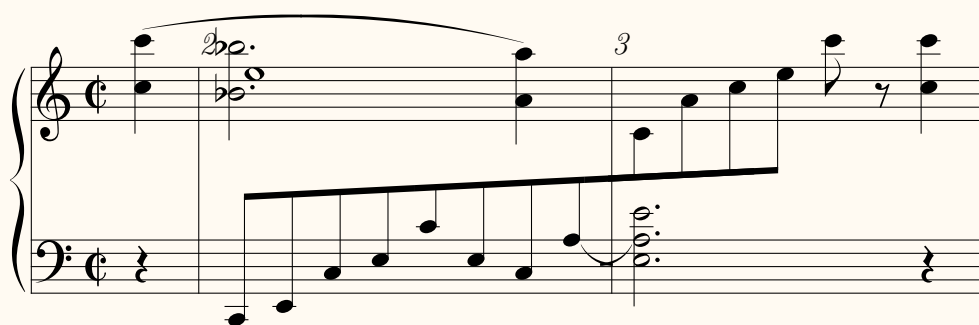
<http://edt1023.sayya.org/tex/omega/>
<http://MathNet.math.tku.edu.tw/~edt1023/tex/omega/>

後記

或許有人會認為，幹嘛不使用 Office 類的軟體就好了，不是更方便嗎？但這就是長久以來的一個誤解，把文書處理的領域和專業排版的領域混淆了，其實這是兩種不同的領域，雖然數位電子的軟體一直在進步，使得這兩個領域看起來很像，但在專業的立場而言，這仍然是不相同的領域，著重的重點並不一樣。請試試右邊的這盤棋，以一般的 Office 類軟體，要怎麼下呢？當頭炮，將軍死棋！這不是開玩笑的， $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 真的可以排版出來，其他難得一見的領域也是可以經由 $\text{T}_{\text{E}}\text{X}$ 和 METAFONT 或 METAPOST 來從無到有的製作出來，就像這些棋子、棋盤一樣。



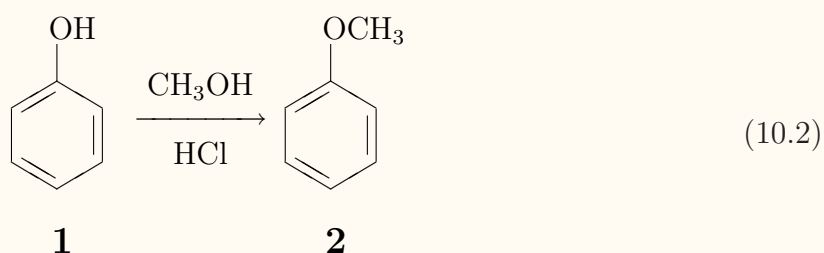
底下這個樂譜呢？



還有，數學式子呢？能排出 AMS (American Mathematical Society, 美國數學協會) 所能接受的標準嗎？這可是 $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 的拿手把戲。當初 Knuth 教授就是不滿意書商把他心愛的著作裡頭的數學式子排得很難看，他才發奮開發出 $\text{T}_{\text{E}}\text{X}$ 排版系統出來的。

$$\begin{aligned}
\mathcal{B}(t, \omega) \approx & \frac{1}{4\pi} \mathcal{D}_t^2 W_{\mathbf{S}}(t, \omega) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} t_1^2 \phi(t_1, \omega_1) dt_1 d\omega_1 \\
& + \frac{1}{4\pi} \mathcal{D}_\omega^2 W_{\mathbf{S}}(t, \omega) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \omega_1^2 \phi(t_1, \omega_1) dt_1 d\omega_1.
\end{aligned} \tag{10.1}$$

好吧，換個化學結構式好了！



曾有許多朋友問過引入程式碼的問題，以下是由 C++2_{LaTeX} 所轉成的 \TeX 文稿，`\input` 進來的。這是居士先生寫的一個測試系統地區環境支援的小工具。

```

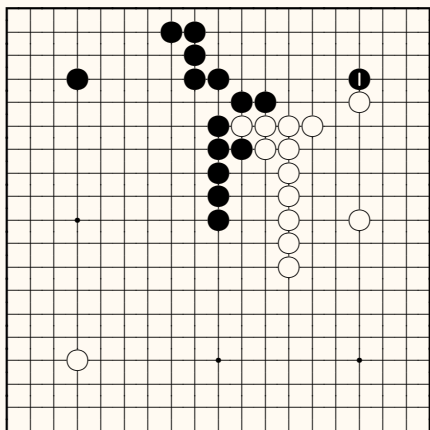
/* gcc -o localetest localetest.c -L/usr/X11R6/lib -lX11 */

#define USE_XOPEN_SOURCE

#include <stdio.h>
#include <locale.h>
#ifdef USE_XOPEN_SOURCE
# include <langinfo.h>
#endif
#include <X11/Xlib.h>
#include <X11/Xlocale.h>

main()
{
    printf("LC_CTYPE = %s\n", setlocale(LC_CTYPE, ""));
#ifdef USE_XOPEN_SOURCE
    printf("encoding = %s\n", nl_langinfo(CODESET));
#endif
    if (XSupportsLocale() == True)
        printf("X Locale = OK\n");
}

```



有人喜歡圍棋，那也成！運用之道，存乎一心。現在大概可以猜得到我選擇 T_EX/L^AT_EX 這個幕後排版系統自討苦吃的原因是什麼了吧？:-) 最重要的原因就是他很具有很低階的 T_EX 排版程式語言，他可以給我彈性、精確與品質，這才是我真正想要的東西。而且，這會產生一個「副作用」，那就是他的操作方式變動並不會很大，雖然剛開始接觸時可能會多花一些時間，但想到學到的東西能夠用一輩子，不必去擔心改版又得從頭學習，這個學習成本還算是滿值得的。

當然，T_EX/L^AT_EX 並不是沒有缺點的，例如學習曲線會比較陡，屬於幕後排版系統，記指令、查指令是免不了的，甚至特殊的需求得自行寫 macro，而這些 macro 並不會太直覺；METAFONT 及 METAPOST 的製作字型及繪圖雖功能強大，但也是需要一段時間的學習才能真正拿來用；中文直行的排版會比較麻煩；各 macro 間有時會衝突，無法同時使用，像排圍棋譜的 go macro，和 CJK 就會衝突，只能另外製作圖檔引入，排版化學式的 macro 也常會和 L^AT_EX 的數學模式衝突，要另外做調整。所以如果沒有特殊的需求，一般 GUI 的 Office 類軟體也不必去排斥，有些人製作文書，可能一輩子沒幾回，那站在學習成本的立場，你得盤算一下自己的時間成本，或者，就像我一樣，把玩電動的時間省下來改玩 T_EX。:-)

本文的 PDF 格式，所使用的字型是王漢宗博士所捐贈的 GNU GPL TTF 字型，可以使用 acroread 或 xpdf 或 gv 來閱覽。要送印表機的話，可使用 acroread/pdf2ps/pdf2ps 將他轉成 POSTSCRIPT 的格式，或直接送印表機。欲轉換成純文字檔，可使用 Xpdf 所附的 pdftotext 這個工具：

```
pdftotext mycjk.pdf
```

就可以了，會產生 mycjk.txt 檔。當然，文章的編排，由於格式差異很大，可能會有參差不齊的情形，請調整一下 pdftotext 所能使用的參數 (pdftotext -h)。

10.1 本文的 PDF 格式檔及原始 L^AT_EX 文稿

本文 PDF 格式檔及原始 L^AT_EX 文稿，可在此下載，或線上閱覽：

<http://edt1023.sayya.org/tex/mycjk/mycjk.pdf>
<http://edt1023.sayya.org/tex/mycjk/mycjk-v1.1.tar.gz>
<http://MathNet.math.tku.edu.tw/~edt1023/tex/mycjk/mycjk.pdf>
<http://MathNet.math.tku.edu.tw/~edt1023/tex/mycjk/mycjk-v1.1.tar.gz>

GNU 自由文件許可證原文

GNU Free Documentation License

Version 1.2, November 2002

Copyright © 2000,2001,2002 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, \LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats

include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or non-commercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both

covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.

- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review

or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document,

and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with. . . Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

使用許可證聲明

Copyright © 2002, 2003 李果正 Edward G.J. Lee

最後修訂日期：2004 年 5 月 18 日

本文件為自由文件（GNU FDL <http://www.gnu.org/copyleft/fdl.html>），不明示或暗示有任何的保證。本文件可自由複製、修改、散佈，但請保留使用許可證聲明，文章內所附之 GNU Free Documentation License 一章全文則不可修改其內容。程式碼的部份依其所宣告的 license，不受 GNU FDL 的規範。PDF 格式文件內所嵌入的向量字型資料，為王漢宗博士（Dr. Hann-Tzong Wang）所捐贈，字型 Copyright 屬王漢宗博士，其使用授權為 GNU GPL。因此，此部份的字型資料，亦不屬 FDL 規範範圍。文件內所提及的商標皆屬其合法註冊公司所有。