SGML-Tools 使用說明中譯版

作者: Matt Welsh. Updated by Greg Hankins.
譯者: *C.W.Huang* *cwhuang@phys.ntu.edu.tw* v0.99.0, 29 November 1996. 翻譯日期: 4-15 August 1997

這份文件是 SGML-Tools 的使用說明，一種以 SGML 爲基礎，讓你產生不同輸出格式的工具・你可以用單一的 SGML 原始文件產生純文字檔 (ASCII 與 ISO-8859-1), DVI, PostScript, HTML, GNU info, LyX, 與 RTF 檔・這份文件說明以 SGML-Tools 0.99.0. 爲準・

目錄

1 簡介

這是一份 SGML-Tools 文件處理系統的使用說明．SGML-Tools 提供一個 SGML DTD (文件形態定義，Document Type Definition) 及一組「置換檔(replacement files)」，可以將 SGML 轉換爲 groff, LaTeX, HTML, GNU info, LyX, 與 RTF 的原始檔．

SGML-Tools 主要以 *Tom Gordon* <mailto:thomas.gordon@gmd.de> 的 QWERTZ DTD 爲基礎發展的．

1.1 關於此文件

這份文件以附在 SGML-Tools 上的 linuxdoc DTD 所撰寫的． 它幾乎包含所有使用這份 DTD 來寫 SGML 文件所需知道的事情． 請參考 `example.sgml` 這份 SGML 範例，你可以拿它當作你自己文件的範本．

1.2 爲何用 SGML?

我選擇使用 SGML 是因爲 SGML 是專門用來轉換成其它格式的． SGML 是 Standard Generalized Markup Language 的簡稱，它允許你指定文件的結構—也就是，用何種事物來標示文件． 你使用一份 DTD (文件形態定義) 來指定文件的結構．linuxdoc 是一種指定 Linux HOWTOs 與其它文件的 DTD．QWERTZ 是另一種 DTD; SGML 的標準提供書，文章，及其餘一般文件所使用的 DTD．

DTD 指定文件中「組成元素」的名稱．組成元素就是一種結構—例如章，節，段落，或是更小的像是這樣的 強調文字．然而跟 LaTeX 不同的是，這些組成元素並非 SGML 本身內含的．linuxdoc DTD 就定義了類似 LaTeX 中的組成元素—你有章，節，或是「環境」等等．然而，利用 SGML 你可以照你喜歡定義文件的任何結構．就某方面來說，SGML 就像是低階的 TeX，而 linuxdoc DTD 則像是 LaTeX．

別被這類比搞混了．SGML 並不是一種文件格式化系統．沒有所謂的「SGML 格式」． SGML 的原始檔只是用來轉換成其它格式以便處理的．還有，SGML 本身是用來指定文件的結構． 沒有 SGML 本身的文字格式化工具或是「巨集」之類的．所有的一切都定義於 DTD 中． 沒有 DTD 你不能使用 SGML，一份 DTD 定義了 SGML 該做什麼．

1.3 它如何運作

這裡解釋如何使用 SGML-Tools 處理文件．首先，你需要一份 DTD．我用的是 QWERTZ DTD， 它本來是一群人因爲需要類似 LaTeX 的 DTD 而做的．我將 QWERTZ DTD 修改成 linuxdoc DTD 以符合我們的需求．DTD 不過設定了文件的結構．它看起來像是這個樣子的:

```
<!element article - -
    (titlepag, header?,
     toc?, lof?, lot?, p*, sect*,
     (appendix, sect+)?, biblio?) +(footnote)>
```

這部份設定了「文件」的整個結構，就像是 LaTeX 的「documentstyle」· 這文件包含了一個書頁名(`titlepag`)，一個可有可無的標題(`header`)，一個可選用的內容列表(`toc`)，可選用的圖形列表(`lof`)與表格列表(`lot`)，任意數量的段落(`p`)，任意數量的章節(`sect`)，可選用的附錄(`appendix`)，一個可選用的參考書目(`biblio`)以及註解(`footnote`)·

你可看出，DTD 並未說明文件應如何被格式化，或它看起來長得怎麼樣· 它只不過定義了文件由何種部份所組成·在文件的另一個地方定義了 `titlepag`, `header`, `sect` 及其餘的組成元素·

寫文件時你並不需要知道任何關於 DTD 的文法規則·我只不過讓你看一下它長得什麼樣子還有它做了什麼· 但你必需熟悉這份 DTD 所定義的文件結構·否則，在嘗試寫文件時你可能會違反結構， 而且被錯誤訊息搞昏了·下面我們會描述 SGML 文件結構的細節·

接下來就是用這份 DTD 所定義的結構來寫文件了·再說一次，linuxdoc DTD 使得文件看起來長得像 LaTeX—所以很容易遵從·用 SGML 的術語來說，使用某一個特別的 DTD 來撰寫的單一文件叫做那份 DTD 的「案例(instance)」·

爲了將 SGML 原始檔轉換成其它格式(例如 LaTex 或 groff)以便處理，你所寫的 SGML 原始檔必須經 SGML 解析器處理過·我用的是 James Clark(*jjc@jclark.com* <mailto:jjc@jclark.com>) 所寫的 `sgmls` 解析器, 他也是 `groff` 的作者·它用起來很方便·`sgmls` 讀入你的文件，檢查它遵循 DTD 所定義的結構· 它同時將你文件中的「巨集」和元素展開，以方便下一階段的 `sgmlsasp` 來處理·

`sgmlsasp` 用來將 `sgmls` 的輸出結果轉換成另一種格式(如 LaTeX)· 它使用置換檔將原始 SGML 文件中的組成元素轉換成「目標」格式(如 LaTeX 或 groff)對應的原始檔·

例如，一部份 LaTeX 的置換檔長得像這樣：

```
<itemize>    +    "\\begin{itemize}   +
</itemize>   +    "\\end{itemize}     +
```

它說明了當你在 SGML 原始檔中使用 `itemize` 元素時，應該被置換成 LaTeX 原始檔中的

```
\begin{itemize}
```

(我說過了，DTD 中的元素像極了它們在 LaTeX 中的對應物·)

因此，要將 SGML 轉換成其它格式，你所要做的就是爲此格式寫一個新的置換檔， 在新格式中爲每一 SGML 組成元素指定適當的對應·事實上，沒這麼簡單—例如， 如果你嘗試轉換成和你的 DTD 完全不同的結構，你就有麻煩·不過， 比起要爲每一種格式寫單獨的解析器與轉換程式還是要簡單的多· SGML 提供一種一般的系統將一種原始檔轉換成許多格式·

一但 `sgmlsasp` 轉換完成，你就有與原先 SGML 原始文件相對應的 LaTeX 原始檔， 你可以像一般的 LaTeX 檔一樣來使用它·下面我會示範如何做這樣的轉換與格式化· 你只要用一行指令就夠了·

不過首先，我要說明如何安裝及設定 SGML-Tools·

2 安裝

首先從這個地方

- *ftp://sunsite.unc.edu/pub/Linux/utils/text/sgml-tools-0.99.0.tar.gz*

取得 `sgml-tools-0.99.9.tar.gz`·

你可以從 *SGML-Tools WWW Page* <http://www.xs4all.nl/~cg/sgmltools/> 取得最新的資訊．

sgml-tools-0.99.0.tar.gz 包含了所有你需要用來寫 SGML 文件並轉換爲 groff, LaTeX, HTML, GNU info, LyX, 與 RTF 的東西．除此之外，你可能還需要下列的工具—SGML 系統並不需要這些，但我建議你取得這些工具以方便你轉換文件，在你公開散佈你的文件前將先確定它們看起來沒問題．

1. groff. 你必須使用 1.08 或更新的版本．你可以從 *ftp://prep.ai.mit.edu/pub/gnu* 取得．在 *ftp://sunsite.unc.edu/pub/Linux/utils/text* 可找到 Linux 的執行檔．你必需使用 groff 讓你的 SGML 文件產生純文字檔． nroff 不行!
2. TeX 與 LaTeX. 這幾乎在每個地方都有，你應該可以毫無困難地找到並安裝 (在 *sunsite.unc.edu* 可以找到 Linux 的執行檔) 當然，如果你要用 LaTeX 來格式化你的 SGML 文件你才需要 TeX/LaTeX．
3. flex. lex 可能不行．你可以從 *ftp://prep.ai.mit.edu/pub/gnu* 取得．
4. gawk 與 GNU info tools, 用來格式化與預視 info 檔案． 在 *ftp://prep.ai.mit.edu/pub/gnu*, 或 *ftp://sunsite.unc.edu/pub/Linux/utils/text* 可以找到 gawk， 還有 *ftp://sunsite.unc.edu/pub/Linux/system/Manual-pagers* 可找到 GNU info tools. awk 不行．
5. LyX (一個半所見即所得(quasi-WYSIWYG) 的 LaTeX 界面)，可在 *ftp://ftp.via.ecp.fr* 找到．

2.1 安裝軟體

安裝與設定 SGML-Tools 所需的步驟是：

1. 首先，在某處解開 sgml-tools-0.99.0.tar.gz 的壓縮檔． 它將產生 sgml-tools-0.99.0 的目錄．在那裡解開都沒有關係，只是不要把裡面的東西搬離 sgml-tools-0.99.0 目錄就行．
2. 閱讀 INSTALL 檔，它包含了安裝過程的細節．

如果一切順利，你就可以使用它了．

譯註: 其實就是執行 install 這支 shell script，回答它的問題就行了!

2.2 編排 SGML 文件

例如說你想要編排一份 SGML 文件叫 foo.sgml，這裡有一份簡短的說明如何輸出成不同的格式．完整的選項列表請參考 man pages．

2.2.1 產生純文字輸出

如果你想產生純文字檔，使用這個指令：

```
% sgml2txt foo.sgml
```

請注意我已調整 groff 轉換爲純文字輸出．也就是說，我將頁首，頁碼，邊框等去掉． 經過一些修改你可以從 groff 的輸出產生 PostScript 與 DVI 格式，但我建議你用 LaTeX 比較好．

你可以爲 man pages 產生 groff 的原始檔，而用 groff -man 將其編排． 執行這個指令：

```
% sgml2txt -man foo.sgml
```

2.2.2 產生 LaTeX, DVI 或 PostScript 輸出

要從 SGML 原始檔產生 LaTeX 文件，就用：

```
% sgml2latex foo.sgml
```

如果你希望產生 PostScript 輸出(經由 dvips)，可用 -p 選項：

```
% sgml2latex -p foo.sgml
```

或者你可以加上 -d 選項產生 DVI 檔：

```
% sgml2latex -d foo.sgml
```

2.2.3 產生 HTML 輸出

如果你希望輸出 HTML 檔，用這指令：

```
% sgml2html -img foo.sgml
```

這將產生 `foo.html`, `foo-1.html`, `foo-2.html`, 等等 — 每一章一個檔案．`foo.html` 是首頁，可以用你的 WWW 瀏覽器觀看． 確定你所有的 HTML 檔在同一目錄下，因爲它們是用相對的 URL 彼此參考的． HTML 檔的圖示目錄指向 `$SGMLTOOLSLIB/icons` 所定義的目錄． 這些檔都應被拷貝到最後的位置上．`$SGMLTOOLSLIB/` 定義在 SGML 轉換程序的最開頭．

如果你使用 `sgml2html` 而沒有加上 -img 選項的話，產生的 HTML 文件會加上導讀用的 "Previous", "Next", 以及 "Table of Contents" 標籤． 你可以藉由在 `$SGMLTOOLSLIB/rep/html/<filename>` 建立一個檔案以蓋過這些預設， 並且以你自己的話來取代它們．該檔案的格式如下所示： he following format:

```
    PrevPage:  newvalue
NextPage:  newvalue
    TOC:      newvalue
```

範例請參閱 `deutsch`．

2.2.4 產生 GNU Info 格式的輸出

如果你想要將你的檔案編排成 GNU Info 瀏覽程式接受的格式，只要使用下列指令：

```
% sgml2info foo.sgml
```

2.2.5 產生 LyX 格式的輸出

欲產生 LyX 格式的出，使用這個指令：

```
% sgml2lyx foo.sgml
```

2.2.6 產生 RTF 格式的輸出

如果你想產生 RTF 格式的輸出，執行這個指令：

```
% sgml2rtf foo.sgml
```

這將會產生 foo.rtf, 以及 foo-1.rtf, foo-2.rtf, 等等檔案—文件的每一節自成一個檔案·

2.2.7 檢查 SGML 語法

如果你只是想從 SGML 轉換中抓出你的錯誤，可以使用 sgmlcheck 這個指令稿· 例如：

```
% sgmlcheck foo.sgml
```

2.3 ISO 8859-1 字元集

ISO 8859-1 (latin1) 字元集可以用在純文字 LaTeX, HTML, LyX, 以及 RTF 輸出的國際字元(GNU Info 未來可能支援 ISO 8859-1)· 要使用這項功能的話，可以加上 -l 選項給編排指令稿，例如：

```
% sgml2txt -l foo.sgml
```

你也可以在 SGML 原始文件中使用 ISO 8859-1 字元，它們會自動被轉換成對應文件格式的適當逸出字元(escape codes)·

3 使用 SGML-Tools 撰寫文件

使用 SGML-Tools DTD 撰寫文件的過程大部分是非常容易的，而且有點類似 LaTeX· 然而，仍然有些限制要注意·這一節我將介紹 SGML 文件的撰寫· SGML 文件範例(及寫作指引)請參閱 example.sgml 檔案，你可以拿來當作自己文件的範本· 在此我將討論 SGML 的各項功能，但是以原始文件作爲範例不是很有可讀性· 因此，印出 example.sgml 的原始碼(編排後的輸出)，如此你就有眞實的參考範例。

3.1 基本概念

仔細看範例文件的原始文件，你會注意到裡面有許多以角括號(< 及 >)標示的「標籤(tags)」· 標籤單純地指出某組成元素的起始，這裡所謂的組成元素是一節，一段，一句斜體的文句，列表中的一個項目，以及諸如此類的結構· 使用標籤就像使用 LaTeX 指令如 \item 或是 \section{...} 等一般·

舉個簡單的例子，要產生這一段 **粗體文字**，我在原始碼中鍵入

```
舉個簡單的例子，要產生這一段 <bf>粗體文字</bf>, ...
```

<bf> 是這段粗體文字的開頭，而 </bf> 是結尾· 你也可以使用這個簡略形式：

```
舉個簡單的例子，要產生這一段 <bf/粗體文字/, ...
```

就是用斜線將粗體文字包住·(當然，如果被包含的文字含有線，你就必須使用長的形式， 例如 Unix 的檔案名稱·)

關於特殊字元還有其它得要注意的問題 (這就是爲什麼如果你仔細觀察原始文件就會注意到這些看起來很怪異 "&" 表示式；我將簡短地說明這些符號)·

在某些情況下，特定的組成元素之結尾標籤是可選用的· 例如，你在某一節的開始使用 <sect> 這個標籤，然而，該節的結尾標籤 (可以出現在該節主體本身結束的地方，而不只是該節的名稱後面!) 是可選用的而當你在同一個層次開始另外一節的時候也就暗示著該節的結束· 一般來說你不需要擔心這些細節；只要跟著使用在 example.sgml 指引裡的範例就可以了。

3.2 特殊字元

很明顯的，角括號它們本身就是 SGML 原始文件裡的特殊字元·另外還有其它該注意的字元· 例如，假設你想鍵入由角括號所涵蓋的表示式，像這樣: <foo>· 爲了表示左邊的角括號，你必須使用 < 這個組成元素， 這是一個會展開成左括號字元的「巨集」·因此，我在原始文件中鍵入

```
angle brackets around it, as so: <tt>&lt;foo></tt>.
```

一般來說，以結合字元(&)作爲開始表示這是一個特殊的巨集· 例如，產生 % 符號的 % 產生 | 的 | 等等· 對於所有的「特殊字元」都有結合巨集可以表示它們·

通常，你不需要使用結合巨集表示某特殊字元，然而，在某些情況下這是必須的。最常用的是：

- Use & for the ampersand (&),
- Use < for a left bracket (<),
- Use > for a right bracket (>),
- Use </ for a left bracket with a slash (</)
- Use $ for a dollar sign ($),
- Use # for a hash (#),
- Use % for a percent (%),
- Use ˜ for a tilde (˜),
- Use ‘‘ and ’’ for quotes, or use “ for “, ” for ”.

完整的特殊字元列表可以查看其中一個替換檔· 通常 LaTeX 最會抱怨特殊字元，所以查閱 $SGMLTOOLSLIB/rep/latex/general 會是個好的開始· $SGMLTOOLSLIB 定義在 SGML 轉換程序的最前面·

3.3 Verbatim 與 Code 環境

當我們正討論特殊字元這個主題時，我也必須提到「verbatim 環境」， 用來在輸出中含入不加變動的文字(空白字元以及縮排，和諸如此類)· verb 這個組成元素就是做此用途的·它看起來就像這樣：

```
<verb>
 Some literal text to include as example output.
</verb>
```

verb 環境不允許你在其內容中使用任何東西． 特別地，在 verb 環境中你必須：

- 使用 & 代表 &
- 使用 </ 代表 </
- 不要在 verb 環境中使用 \end{verbatim}， 因爲這是 LaTeX 用來結束 verbatim 環境的． (未來應該可能完全的隱藏底層的文字編排，但目前的解析器還不支援．)

code 環境非常類似 verb 環境，除了在文字周圍加入分隔線

這是 code 環境的例子．

你應該在任何 verb 環境的周圍使用 tscreen 環境，像這樣：

```
<tscreen><verb>
這是一段例句．
</verb></tscreen>
```

tscreen 是一個環境單純地縮排並設定預設字型爲 tt． 這使得範例要論在 LaTeX 以及純文字的版本中看起來好多了． 你可以使用 tscreen 而不加 verb，不過，如果你在範例中使用任何的特殊字元那麼你就需要它們兩個． tscreen 與特殊字元無關．範例請參閱 example.sgml．

quote 環境類似 tscreen，除了不將預設字型設爲 tt．所以，你可以使用 quote 於與電腦無關的引用，如：

```
<quote>
Here is some text to be indented, as in a quote.
</quote>
```

這會產生：

> Here is some text to be indented, as in a quote.

3.4 文件內容概觀

在我們深入研究細節之前，我要開始介紹 linuxdoc DTD 定義的文件概觀．文件如何設立的範例請仔細觀看 example.sgml．

3.4.1 序文

在文件的「序文」中你設立像是標題以及文件型態(style)的東西．Linux HOWTO 文件看起來應該像：

```
<!doctype linuxdoc system>

<article>

<title>Linux Foo HOWTO
<author>阿偉, <tt/cwhuang@phys.ntu.edu.tw/
<date>v1.0, 14 August 1997
<abstract>
這份文件說明了如何使用 <tt/foo/ 這個工具.
</abstract>

<toc>
```

這些組成項目可以酌予增減但須按此順序．第一行告訴 SGML 解析程式使用 linuxdoc DTD． <article> 這個標籤規定文件使用「article」此種文件型態．(原先的 QWERTZ DTD 還定義「report」以及「book」等型態；我並沒有修改它們以在 SGML-Tools 中使用)．

title, author, 以及 date 這些標籤的意義應該是很明顯的，在 date 標籤中包含文件的版本編號以及最後修改時間．

abstract 標籤設定文件頂端在內容之前印出的文字．如果你不想含入內容列表(toc 標籤)，那麼你可能不需要 abstract．

3.4.2 章節與段落

在序文之後，你就可以準備進入文件本身．有下列的分節指令可以使用：

- sect: 最頂層的章節 (也就是 1, 2, 等等)
- sect1: 第二層的次章節 (也就是 1.1, 1.2, 等等)
- sect2: 第三層的小章節
- sect3: 第四層的小小節
- sect4: 第五層的小小小節

它們相當於 LaTeX 中對應的 section, subsection 等等．

在 sect (或 sect, sect 等等) 標籤之後跟著是該節的名稱．例如，在文件的頂端，序文之後的是此標籤：

```
<sect>簡介
```

而在此節開始的地方(段落)，有個標籤是：

```
<sect2>段落
```

在一節的標籤之後，你開始文件的主體．然而，你必須以一個 <p> 標籤開始，像這樣：

```
<sect>簡介
<p>
這是一份 SGML-Tools 文件處理的使用者指引....
```

這是告訴解析程式你已完成一節的標題並且準備好開始本文主體． 其後，新的段落是以空白行開始(就如同 TeX)．例如，

```
這是第一段的結尾．

而我們從這裡開始新的一段．
```

沒有什麼理由要在每個段落的前面使用 <p> 標籤；只要在分節指令後面的第一個段落開始處加就行．

3.4.3 文件結尾

在文件結束的地方，你必須使用這個標籤：

```
</article>
```

以告訴解析程式你已經完成 article 這個組成元素(涵蓋整份文件)．

3.5 交互參考

現在我們要進入系統的其它功能．交互參考很容易．例如，如果你想要製作某一節的交互參考，你得像這樣標示該節:

```
<sect1>簡介<label id="sec-intro">
```

接著你就可以在文件中的某個地方式用這個表示式參照該節：

```
請參考<ref id="sec-intro" name="簡介">一節的介紹．
```

這將會以 sec-intro 標記的節號替換掉這個 ref 標籤． 對於 groff 以及 HTML 轉換 name 這個參數是必須的． SGML-Tools 使用的 groff 巨集設定目前並未支援交互參考，而且它常常以名稱取代數字參照某一節．

例如，本節是3.5 (交互參考)．

在全球資訊網上使用的通用資源定位器，即 URL 也有組成元素 url． 這個元素應該用來參照其它文件，FTP 上的檔案等等．例如：

```
你可從此取得 Linux HOWTO 文件
<url url="http://sunsite.unc.edu/mdw/HOWTO/"
   name="The Linux HOWTO INDEX">．
```

url 這個參數指定這個 URL 本身．連結會自動加入到 HTML 文件中． 可選用的 name 參數指定應該加上連結的文字(HTML 轉換)或是為 URL 描述命名(LaTeX 或 groff)．如果沒有給 name 參數，會使用這個 URL 本身．

例如，你可以從這裡取得 SGML-Tools 套件

ftp://sunsite.unc.edu/pub/Linux/utils/text/sgml-tools-0.99.0.tar.gz.

一個有用的變形是 htmlurl，除了 HTML 文件在其它每種文件中這個 URL 部份將不會出現． 它用在像是個人的電子郵件位址；你可以寫

```
<htmlurl url="mailto:esr@snark.thyrsus.com"
      name="esr@snark.thyrsus.com">
```

而在純文字輸出中獲得「esr@snark.thyrsus.com」 而不是重複的「esr@snark.thyrsus.com <mailto:esr@snark.thyrsus.com>」但是在 HTML 文件中仍然保有適當的 URL 部份．

3.6 字型

基本上，SGML-Tools 支援與 LaTeX 相同的字型． 注意，無論如何，純文字的轉換(經由 `groff`)不理會有關字型的資訊． 所以，你可以盡可能的使用多種字型，這對 LaTeX 的轉換會有好處． 但是在純文字的版本中不要指望能用字型點出重點．

特別是，前面描述過的 `tt` 這個標籤可以用來獲致寬度固定的「打字機」字型， 這應該用在所有的電子郵遞位址，機器名稱，檔案名稱等等． 例如：

```
Here is some <tt>typewriter text</tt> to be included in the document.
```

相當於：

```
Here is some <tt/typewriter text/ to be included in the document.
```

記得如果被包含的文字不含有斜線你可以使用簡略形式．

其它的字型有 粗體字 用的 `bf` 以及 斜體字 用的 `em`． 另外還支援數種字型，但是我不建議你使用它們，因為它們會被轉換成像 HTML 不支援的字型． 粗體，打字機體，以及斜體應該能滿足你的需要．

3.7 列表

此系統支援數種不同的列表．它們是：

- `itemize` 像這樣的告示列表
- `enum` 數字列表
- `descrip` 描述列表

在 `itemize` 或是 `enum` 裡的項目必須以 `item` 標籤加以標示． 在 `descrip` 裡的項目必須以 `tag` 標示． 例如，

```
<itemize>
<item>這是一個項目．
<item>這是第二個項目．
</itemize>
```

看起來像是這樣：

- 這是一個項目．
- 這是第二個項目．

在 enum 中

```
<enum>
<item>這是第一個項目．
<item>這是第二個項目．
</enum>
```

你應看得出來，列表可以是巢狀的．細節請參閱範例文件．

descrip 列表稍有不同，而且有點難看，但是在某些情況下你可能會想使用它：

```
<descrip>
<tag/Gnats./ Annoying little bugs that fly into your cooling fan.
<tag/Gnus./ Annoying little bugs that run on your CPU.
</descrip>
```

最後看起來像：

Gnats.
Annoying little bugs that fly into your cooling fan.

Gnus.
Annoying little bugs that run on your CPU.

3.8 更進一步的資訊

1. QWERTZ 使用者指引可以從這裡取得 *ftp://ftp.cs.cornell.edu/pub/mdw/SGML*. QWERTZ (以及 SGML-Tools) 還支援像是數學公式，表格，圖形等許多特性． 如果你想用 SGML 寫一般的文件，我建議你用原來的 QWERTZ DTD 而不要用修改過的 linuxdoc DTD， 因爲這是我特別用來寫 Linux HOWTOs 及其它類似文件用的．
2. Tom Gordon 的 QWERTZ tools 可在此找到： *ftp://ftp.gmd.de/GMD/sgml*.
3. 更多關於 SGML 的資訊可以在下面這些網頁找到：
   - *SGML and the Web* <http://www.w3.org/hypertext/WWW/MarkUp/SGML/>
   - *SGML Web Page* <http://www.sil.org/sgml/sgml.html>
   - *Yahoo's SGML Page* <http://www.yahoo.com/Computers_and_Internet/Information_and_Documentation/Data_Formats/SGML/>
4. James Clark 的 sgmls 解析器，及後來的 nsgmls 還有其它工具可在這找到： *ftp://ftp.jclark.com* 以及 *James Clark's WWW Page* <http://www.jclark.com>.
5. emacs psgml 可在此找到： *ftp://ftp.lysator.liu.se/pub/sgml*．這個工具提供許多 SGML 的功能．
6. 你可以寄電子郵件到 *majordomo@via.ecp.fr* 以參與 SGML-Tools mailing list，內容寫 subscribe sgml-tools． Mailing list 的地址是 *sgml-tools@via.ecp.fr*．
7. 關於 LyX 的更多資訊可以在此找到： *LyX WWW Page* <http://wsiserv.informatik.uni-tuebingen.de/~ettrich/>. LyX 是一個給 LaTeX 用的高階文字處理器前端． 具半所見即所得界面，會自動產生許多 LaTeX 的樣式及格局． 可以增快 LaTeX 的學習而且使複雜的格局變簡單且直接．