Trasformazione in altri formati

Spesso ci si trova di fronte alla necessità o all'utilità di trasformare un documento scritto in un certo modo, per esempio in LaTeX, in qualcosa di diverso, per esempio in HTML. In generale, queste cose andrebbero pianificate prima, per decidere lo stile del documento in base alle forme in cui questo deve poi concretizzarsi. Meglio ancora sarebbe l'utilizzo di strumenti appositi, di solito SGML/XML, pensati in anticipo per la produzione di documentazione in formati differenti.

Questo capitolo serve a raccogliere la descrizione di strumenti che possono aiutare a trasformare un documento realizzato con sistemi di composizione tradizionale, pensati principalmente per la stampa su carta, e viceversa.

Non ci si possono fare illusioni: gli strumenti di questo tipo non funzionano sempre, ma solo quando le caratteristiche del sorgente lo consentono.

DLH: trasforma LaTeX in HTML

«

DLH [1] è uno strumento relativamente semplice per la conversione di sorgenti LaTeX in HTML. La trasformazione avviene con successo solo quando si tratta di un sorgente LaTeX in cui non si usano ambienti matematici e soprattutto non si usano comandi particolarmente sofisticati (ciò inteso dal punto di vista di DLH).

DLH utilizza un insieme personalizzato di stili LaTeX, collocato normalmente nella directory '`/usr/share/dlh/inputs/dlh/`'. Si tratta dei soliti '`article.sty`', '`epsfig.sty`' e altri, ma il contenuto di questi file è ridotto rispetto a quelli equivalenti di LaTeX. Se nel sorgente LaTeX si utilizzano altri stili particolari occorrerebbe creare un file corrispondente anche in questa directory, cercando di adattarlo a DLH (cosa che potrebbe risultare difficile, dal momento che bisogna ragionare in termini di TeX limitato secondo le possibilità di DLH).

Il programma eseguibile è '**`dlh`**' che accetta l'indicazione di alcune opzioni e in particolare un elenco di file LaTeX:

```
dlh [opzioni] file_latex...
```

In corrispondenza dei file indicati come argomento vengono create altrettante directory contenenti una serie di file HTML che rappresentano il risultato della trasformazione (a partire da '`index.html`'

che normalmente è un collegamento simbolico al primo di questi file).

DLH utilizza una serie di icone per rappresentare i pulsanti per lo scorrimento del documento secondo la sua struttura. I file di queste icone si trovano normalmente nella directory '`/usr/share/dlh/icons/`' e andrebbero copiati nella directory '`../icons/`', rispetto a quella in cui si trovano i file HTML.

Tabella u96.1. Alcune opzioni.

Opzione	Descrizione
`-f` `--force`	Questa opzione serve a creare tutti i file che compongono il documento, in particolare le immagini. Ciò può creare un rallentamento nel funzionamento di DLH, ma in generale serve a garantire un risultato più sicuro.
`-i` ***uri*** `--icon-dir=`***uri***	Permette di definire esplicitamente la collocazione dei file che rappresentano le icone utilizzate da DLH per rappresentare i pulsanti per lo scorrimento del documento.

Segue la descrizione di alcuni esempi.

- `$` **`dlh prova.tex`**[*Invio*]

  Crea la directory '`./prova/`' e al suo interno inserisce una serie di file HTML che riproducono il documento '`prova.tex`'. In questo caso, i file HTML fanno uso delle icone che si trovano nella directory '`./icons/`', relativa al nodo di rete in cui si trovano.

- `$` **`dlh -f prova.tex`**[*Invio*]

Come nell'esempio precedente, ma viene forzata la creazione di tutti i file, nel caso ce ne fosse bisogno.

- `$` **`dlh -i icone prova.tex`**[*Invio*]

  Come nel primo esempio, con la differenza che i file delle icone devono trovarsi nella directory '`./prova/icone/`'.

Help2man: genera una pagina di manuale dalle informazioni fornite dal programma

«

Help2man [2] è un programma in grado di generare una pagina di manuale a partire dalle informazioni che restituisce un altro programma attraverso le opzioni '**`--help`**' e '**`--version`**'.

Help2man è predisposto principalmente per gestire convenientemente il risultato generato da un programma che segue le convenzioni GNU (ovvero della Free Software Foundation).

`help2man` [***opzioni***] ***programma_eseguibile***

Lo schema sintattico permette di vedere che si tratta dell'eseguibile '**`help2man`**', che oltre alle opzioni eventuali richiede l'indicazione di un programma da avviare con le opzioni '**`--help`**' e '**`--version`**' per ottenere le informazioni necessarie. In modo predefinito, il risultato viene emesso attraverso lo standard output.

Tabella u96.2. Alcune opzioni.

Opzione	Descrizione
`-o` ***file*** `--output=`***file***	Permette di definire il nome del file da generare, evitando così di emettere il risultato attraverso lo standard output.
`-s` ***n_sezione*** `--section=`***n_sezione***	Permette di specificare il numero della sezione della pagina di manuale.

Segue la descrizione di alcuni esempi.

- `$` **`help2man ls > ls.1`**[*Invio*]

  Genera il file '`ls.1`', contenente la pagina di manuale di '**`ls`**'.

- `$` **`help2man -o ls.1 ls`**[*Invio*]

  Esattamente come nell'esempio precedente, utilizzando esplicitamente l'opzione '**`-o`**'.

Pstotext: estrae il testo da un file PostScript o PDF

Pstotext [3] è un programma molto semplice per l'estrazione del testo contenuto all'interno di un file PostScript o PDF, per mezzo di Ghostscript.

```
pstotext [opzioni] file
```

Tutto il lavoro viene svolto dall'eseguibile '**`pstotext`**'. Il risultato dell'elaborazione viene emesso attraverso lo standard output, a meno che sia stato stabilito diversamente con le opzioni.

Tabella u96.3. Alcune opzioni.

Opzione	Descrizione
`-cork`	Specifica che il file PostScript utilizza la codifica «cork», ovvero ciò che si ottiene da Dvips quando questo converte file DVI generati da TeX con la codifica T1.
`-landscape` `-landscapeOther`	Queste due opzioni indicano che il testo è ruotato a 90 gradi in un senso, oppure nell'altro.
`-portrait`	In questo caso si intende che il testo scorre come di consueto, su un foglio orientato in modo verticale.
`-output` ***file***	Consente di indicare il file di testo da generare, senza bisogno di ridirigere lo standard output.

Mswordview

«

Mswordview [4] è un programma il cui scopo è quello di convertire file di MS-Word in HTML. La conversione non può essere perfetta, ma il progetto è condotto con impegno e i risultati che dà questo programma sono buoni.

L'eseguibile di questo programma corrisponde a '**`mswordview`**' e la sintassi per il suo utilizzo si può schematizzare secondo il modello seguente:

```
mswordview [opzioni] file_doc
```

Mswordview è in grado di convertire solo un file alla volta, pre-

cisamente quello che viene indicato alla fine degli argomenti. Se non viene richiesto qualcosa di particolare attraverso le opzioni, Mswordview tenta di creare un file con lo stesso nome di quello che viene convertito, con l'aggiunta dell'estensione '`.html`'. Inoltre, se il file contiene delle immagini incorporate, queste vengono trasferite su file esterni.

Tabella u96.4. Alcune opzioni.

Opzione	Descrizione
`-o` ***file_html*** `--outputfile` ***file_html***	Permette di indicare esplicitamente il file HTML che si vuole generare.
`-g` ***file_errori*** `--errorfile` ***file_errori***	Permette di annotare gli errori incontrati durante la conversione nel file indicato.

Catdoc

Catdoc [5] è un programma molto semplice, che si sostituisce idealmente a '**`cat`**' quando si tratta di visualizzare il contenuto di file scritti in formato MS-Word. Il suo funzionamento è intuitivo e in generale non servono opzioni: il file indicato come argomento, o fornito attraverso lo standard input, viene emesso dallo standard output dopo una conversione in formato testo. Se il file originale contiene in realtà solo testo puro, non avviene alcuna conversione.

```
catdoc [opzioni] file_doc
catdoc [opzioni] < file_doc
```

Tabella u96.5. Alcune opzioni.

Opzione	Descrizione
`-b`	Cerca di elaborare anche file MS-Word che apparentemente non lo sono, a causa di una firma iniziale errata.
`-m`***n***	Specifica il margine destro del testo ottenuto. Il margine predefinito è a colonna 72. Si osservi che l'opzione '**`-m0`**' equivale a '**`-w`**'.
`-w`	Specifica il margine destro del testo ottenuto di lunghezza indefinita, in modo da ottenere che i paragrafi occupino una riga intera.
`-v`	Genera alcune informazioni diagnostiche prima del testo trasformato.

Per quanto semplice possa essere questo programma, è prevista una configurazione, composta dal file '`/etc/catdocrc`' per il sistema e dai file '`~/.catdocrc`' per gli utenti. Senza entrare nel dettaglio delle direttive di configurazione, è il caso di descrivere quella che rappresenta l'impostazione comune:

```
charset_path=/usr/lib/catdoc
map_path=/usr/lib/catdoc
source_charset=cp1252
target_charset=UTF-8
unknown_char='?'
```

Come si può intuire, le direttive '**`charset_path`**' e '**`map_path`**' servono a indicare la collocazione di file utilizzati da Catdoc per la conversione. La direttiva '**`source_charset`**' permette di stabilire la codifica predefinita del file sorgente, quando questo non appare utilizzare la UTF-16. La direttiva '**`target_charset`**' permette di

definire la codifica da usare per il testo generato; come si vede nell'esempio viene usata la codifica UTF-8. Infine, è possibile stabilire in che modo mostrare i caratteri che non possono essere rappresentati, attraverso la direttiva '**unknown_char**', che in questo caso usa il punto interrogativo.

Segue la descrizione di alcuni esempi.

- `$` **`catdoc pippo.doc | less`** [*Invio*]

  Visualizza il contenuto del file '`pippo.doc`', con l'aiuto di '**`less`**' per scorrerlo.

- `$` **`catdoc pippo.doc > pippo.txt`** [*Invio*]

  Genera il file '`pippo.txt`' a partire da '`pippo.doc`'.

Antiword

Antiword [6] è un programma molto semplice per convertire file dal formato MS-Word in testo puro e semplice, oppure in PostScript , estrapolando anche le immagini. Il suo funzionamento è intuitivo e in generale non servono opzioni: il file indicato come argomento, viene emesso attraverso lo standard output dopo la conversione.

`antiword` [***opzioni***] ***file_doc***...

Tabella u96.7. Alcune opzioni.

Opzione	Descrizione
`-t`	Genera una conversione in formato testo puro e semplice. L'uso di questa opzione è implicito.

Opzione	Descrizione
`-w` ***n_colonne***	Permette di specificare, nell'ambito di una conversione in formato testo, l'ampiezza del testo in caratteri. Se si utilizza il valore zero, si ottiene ogni paragrafo in una sola riga.
`-m` ***file_mappa***	Consente di indicare la codifica del file di testo che si vuole ottenere.
`-p` ***dimensioni_carta***	L'utilizzo di questa opzione richiede implicitamente la conversione in formato PostScript, mentre in condizioni normali si ottiene un testo puro e semplice. L'argomento dell'opzione stabilisce la dimensione della carta e può trattarsi delle parole chiave seguenti, con il significato intuitivo che hanno: '**`10x14`**', '**`a3`**', '**`a4`**', '**`a5`**', '**`b4`**', '**`b5`**', '**`executive`**', '**`folio`**', '**`legal`**', '**`letter`**', '**`note`**', '**`note`**', '**`quarto`**', '**`statement`**', '**`tabloid`**'.
`-L`	Nell'ambito di una conversione in PostScript, indica un orientamento orizzontale del foglio.
`-i` ***livello_di_visualizzazione_immagini***	Consente di specificare cosa fare delle immagini che fossero eventualmente contenute nel file di partenza. L'argomento è un numero.

Opzione	Descrizione
`-i 0`	Genera un file compatibile con Ghostscript, ma non adatto a stampanti PostScript comuni. Tuttavia, in condizioni normali, se si arriva alla stampa, si passa generalmente per Ghostscript, per cui questo valore è quello che può essere adatto.
`-i 1`	Non estrapola le immagini.
`-i 2`	PostScript livello 2.
`-i 3`	PostScript livello 3.
`-s`	Include anche il testo nascosto, indicato come tale nel file originale.

Segue la descrizione di alcuni esempi.

- `$` **`antiword pippo.doc | less`**[*Invio*]

  Visualizza il contenuto del file '`pippo.doc`', con l'aiuto di '**`less`**' per scorrerlo.

- `$` **`antiword -p a4 pippo.doc > pippo.ps`**[*Invio*]

  Genera il file '`pippo.ps`' (PostScript, A4) a partire da '`pippo.doc`'.

xlHTML

«

xlHTML[7] è un programma per convertire file dal formato MS-Excel in HTML, come suggerisce il nome, oppure in testo puro. Se non si usano le opzioni, si ottiene un file HTML, contenente una tabella con ciò che appare nel foglio elettronico indicato nella riga di comando, emesso attraverso lo standard output:

```
xlhtml [opzioni] file_xls > file
```

Tabella u96.8. Alcune opzioni.

Opzione	Descrizione
`-fw`	In condizioni normali, le celle che contengono delle espressioni vengono valutate e ne viene mostrato solo il risultato, con una nota sulla possibilità che il valore mostrato non sia preciso. Per fare sparire questa nota si usa l'opzione '**`-fw`**'.
`-c`	Fa in modo che la tabella contenente i dati del foglio elettronico appaia centrata nel corpo della pagina HTML.
`-asc`	Genera un risultato in formato testo puro, se però si abbina anche una delle opzioni che iniziano per '**`-x`**'.
`-csv`	Genera un risultato in formato testo, dove i campi sono delimitati da apici doppi e separati da una virgola. Anche in questo caso, l'opzione funziona solo in abbinamento con una delle opzioni che iniziano per '**`-x`**'.
`-xp:`***n***	Converte solo la pagina ***n***, contando a cominciare da zero.

Opzione	Descrizione
`-xc:`***m*****-*****n***	Converte solo le colonne da ***m*** a ***n***, contando a cominciare da zero.
`-xr:`***m*****-*****n***	Converte solo le colonne da ***m*** a ***n***, contando a cominciare da zero.

Per comprendere le possibilità di xlHTML viene mostrato un solo esempio, di un foglio elettronico realizzato con Gnumeric, salvando in formato XLS. Due figure mostrano il contenuto del foglio, sia nel suo aspetto finale, sia nel contenuto effettivo delle celle.

Figura u96.9. Il foglio elettronico di esempio, nel suo aspetto finale.

Figura u96.10. Il foglio elettronico di esempio con le espressioni contenute nelle celle.

	A	B	C	D	E
1	Straordinario				
2	Data	dalle ore	alle ore	durata	
3	=date(2005;1;15)	=time(19;30;0)	=time(21;30;0)	=C3-B3	
4	=date(2005;1;16)	=time(19;30;0)	=time(21;30;0)	=C4-B4	
5	=date(2005;1;17)	=time(19;30;0)	=time(21;30;0)	=C5-B5	
6			Totale straordinario:	=sum(D3:D5)	
7					

Supponendo che il file si chiami ‘`esempio.xls`’, si può procedere con il comando seguente per generare il file ‘`esempio.html`’:

```
$ xlhtml -fw esempio.xls > esempio.html[Invio]
```

Il file che si ottiene dovrebbe avere l’aspetto seguente; si osservi che le date non sono state rappresentate in modo corretto:

Sheet1

Straordinario			
Data	dalle ore	alle ore	durata
38367 *	19:30	21:30	2:00
38368 *	19:30	21:30	2:00
38369 *	19:30	21:30	2:00
Totale straordinario:			6:00

Spreadsheet's Author: Unknown
Last Updated with Excel 97
* This cell's format is not supported.

Created with xlhtml 0.5.1

[1] **DLH** GNU GPL

[2] **Help2man** GNU GPL

[3] **Pstotext** licenza speciale

[4] **Mswordview** GNU GPL + alcuni file con licenza speciale

[5] **catdoc** GNU GPL

[6] **Antiword** GNU GPL

[7] **xlHTML** GNU GPL