

Personalizzazione di LyX: Caratteristiche per l'utente avanzato

dal LyX Team¹
traduzione italiana a cura di Lino Muoio

20 gennaio 2008

¹Questa guida è aggiornata da MIKE RESSLER, mike.ressler@alum.mit.edu e mantenuta in italiano da CLAUDIO COCO, lacocio@libero.it. Potete inviare commenti o segnalare errori ad uno di loro.

Indice

1	Introduzione	7
2	File di configurazione di LyX	9
2.1	Cosa c'è in LyXDir?	9
2.1.1	Files generati automaticamente	9
2.1.2	Directories	10
2.1.3	Files che non volete modificare	11
2.1.4	Altri files che necessitano di una riga o due di spiegazioni...	11
2.2	La vostra directory di configurazione	11
2.3	Lanciare LyX con configurazioni multiple	12
3	Il popup Edit > Preferences	15
3.1	Usare il popup per la prima volta	15
3.2	Schermo & Fonts	15
3.2.1	Settaggi DPI e Zoom di Font	17
3.2.2	Comandi di definizione dei Fonts	18
3.2.3	Codifica dei Fonts	19
3.3	Bindings	20
3.4	User Interface	20
3.5	Convertitori, Formati e Visualizzatori	21
3.6	Anteprima sullo schermo	21
3.7	Opzioni ASCII-Export	22
3.8	Stampante	23
3.8.1	Changing Colors	24
3.9	TeX	24
3.10	I settaggi autorilevati	25
3.11	Il resto	25
4	Il Server LyX	27
4.1	Introduzione	27
4.2	Iniziare il Server LyX	27
4.3	Comunicazione normale	28
4.4	Notifica	28
4.5	Il Protocollo Server LyX semplice	29

5	Internazionalizzare LyX	31
5.1	Selezionare un linguaggio alternativo per l'interfaccia utente . . .	31
5.2	Usare scorciatoie per linguaggi alternativi	32
5.3	Tradurre LyX	32
5.3.1	Tradurre l'interfaccia grafica utente (messaggi di testo). . .	32
5.3.2	Tradurre la documentazione.	33
5.4	Supporto Tastiere Internazionali	34
5.4.1	Definire una propria Mappa della Tastiera: Formato File Keymap	34
5.5	Materiale sulle tastiere internazionali	36
5.5.1	Il File .kmap	36
5.5.2	Il file .cdef	37
5.5.3	Tasti morti	38
5.5.4	Salvare la vostra configurazione di Linguaggio	38
6	Installare Nuove Classi di Documento, Layouts e Templates	41
6.1	Installare un nuovo pacchetto L ^A T _E X	41
6.2	Layouts	43
6.2.1	Supportare nuove classi di documento	43
6.2.2	File sty LaTeX2e	43
6.2.3	File cls L ^A T _E X 2 _ε	44
6.2.4	File sty L ^A T _E X 2.09	44
6.3	Dichiarare una nuova textclass	45
6.3.1	Parametri generali di textclass	46
6.3.2	Sezione ClassOption	47
6.3.3	Layouts specifici di paragrafo	48
6.3.4	Descrizione dei Font	51
6.4	Creare Templates	52
7	Un Tutorial sulla Stampante	53
7.1	Introduzione	53
7.2	Il file /etc/printcap	53
7.2.1	Descrizione	53
7.2.2	Il Nome della Stampante	55
7.2.3	Il Dispositivo di Stampa	55
7.2.4	La Directory Spool e il File Log	56
7.2.5	Il Filter File	57
7.2.6	Entrate Addizionali	57
7.3	Creare un Filtro di Stampa	57
7.3.1	Stampare Files PostScript [®] Usando Ghostscript	57
7.3.2	Se Già avete una Stampante PostScript [®]	61
7.4	Il Demone Print Spool, lpd	62
7.5	Assemblando il Tutto	62

8	Il materiale esterno	65
8.1	Background	65
8.2	Come funziona?	66
8.3	La finestra di dialogo Materiale Esterno	67
8.4	Esempi	68
8.5	Il file di configurazione external template	68
8.6	Il meccanismo di sostituzione	69
8.7	Discussione sulla sicurezza	69
8.8	Il futuro dell'insieme esterno	70

Capitolo 1

Introduzione

Questo manuale illustra le caratteristiche di personalizzazione presenti in LyX. Verranno di seguito discussi temi come le scorciatoie da tastiera, le opzioni di anteprima sullo schermo, le opzioni di stampa, i comandi da inviare a LyX attraverso il LyX Server, l'internazionalizzazione, l'installazione di nuove classi di testo L^AT_EX e layouts di LyX ecc. Ovviamente non ci possiamo augurare di prendere in considerazione tutto quello che è possibile cambiare—i nostri sviluppatori aggiungono nuove funzioni più velocemente di quanto riusciamo a documentare—ma andremo a spiegare le personalizzazioni più comuni e possibilmente cercheremo di darvi qualche indicazione per quelle più complesse.

LyX 1.1.6 introduce una nuova interfaccia utente per le personalizzazioni che possono essere richiamate attraverso Edit > Preferences. Questo non rende obsoleta la seguente trattazione circa i files di configurazioni di LyX, ma si spera invece che semplifichi considerabilmente l'adattamento di LyX alle vostre necessità.

Capitolo 2

File di configurazione di LyX

Questo capitolo ha lo scopo di aiutare ad orientarsi all'interno dei file di configurazione di LyX. Prima di continuare a leggere questo capitolo, dovrete trovare dove si trova la directory delle librerie di LyX usando `Help > Version`. Questa directory è il posto in cui LyX mette i suoi files di configurazione generale, che chiameremo `LyXDir` nel prosieguo del documento.

2.1 Cosa c'è in LyXDir?

`LyXDir` e le sue sottodirectory contiene numerosi files che possono essere usati per la personalizzazione del comportamento di LyX. Nuovo in LyX 1.1.6 è la possibilità di richiamare questi files dall'interno dello stesso LyX attraverso il menu `Edit > Preferences`. La maggior parte delle personalizzazioni che si effettuano sono possibili attraverso questo menu. Tuttavia, molti altri aspetti principali di LyX possono essere personalizzati modificando gli altri file nella directory `LyXDir`. Questi sono divisi in differenti categorie, descritte nelle seguenti sottosezioni.

2.1.1 Files generati automaticamente

Questi files vengono generati quando configurate LyX. Contengono vari valori predefiniti che vengono fissati mediante una procedura di ispezione. In generale, non è una buona idea modificarli, dato che potrebbero essere sovrascritti in qualsiasi momento.

`lyxrc.defaults` contiene i valori di default per vari comandi. Se non gradite alcuni dei valori assegnati, potete reimpostarli usando la nuova voce `Edit > Preferences`.

`packages.lst` contiene la lista dei pacchetti che sono stati riconosciuti da LyX. Non è normalmente usata dal programma, ma le informa-

zioni estratte, ed anche di più sono rese disponibili attraverso **Help** ▸ **L_AT_EX Configuration**.

`textclass.lst` è la lista delle classi di testo che sono stati trovate nella vostra directory `layout/`, insieme alla classe di documento LaTeX associata e relativa descrizione.

`doc/LaTeXConfig.lyx` è generata automaticamente durante la configurazione dal file `LaTeXConfig.lyx.in`. Ogni stringa del form “@chk_foo@” verrà sostituito con “si” o “no” a seconda di dove sia stato trovato il pacchetto `foo`.

2.1.2 Directories

- `bind/` questa directory contiene i files con l'estensione `.bind` che definiscono i tasti di scelta rapida usati in L_YX (vedere la sezione ??). Se esiste una versione internazionalizzata del bindfile chiamato `$LANG_XXX.bind`, questo sarà usato per primo. Vedere il capitolo 5, e la sezione ?? per i dettagli.
- `clipart/` contiene immagini Postscript® che possono essere inclusi nei documenti.
- `doc/` contiene i files di documentazioni di L_YX (incluso quello che state ora leggendo). Il file `LaTeXConfig.lyx` è particolarmente speciale, come già detto. Se esiste una versione internazionale del documento di aiuto con `$LANG_` preposto al nome, quello sarà usato per primo. Guardare il capitolo 5 per i dettagli.
- `examples/` contiene files di esempio che spiegano come usare alcune caratteristiche. Nel file browser premere il pulsante **Example** per raggiungerlo.
- `images/` contiene files di immagine che sono usati nel menu **Document**. In aggiunta, vi sono contenute anche le icone individuali usate nella toolbar e nella schermata che viene mostrata all'avvio di L_YX.
- `kdb/` contiene i files di keymapping della tastiera. Guardare il capitolo 5.5 per i dettagli.
- `layouts/` contiene i files di classi di testo descritti nel capitolo 6
- `scripts/` contiene alcuni files che dimostrano le capacità dell'insieme **External Template**. Riferirsi a **Help** ▸ **External** ▸ **Material** per ulteriori informazioni. [*Nota dell'autore: se questo file è incorporato in un altro dei documenti standard, non sarà reperibile in questa locazione—AL*].
- `templates/` contiene i files templates standard di L_YX descritti nel Capitolo 6.4.
- `tex/` contiene alcuni files `.cls` distribuiti con L_YX.

`ui/` contiene i files con estensione `.ui` che definiscono l'interfaccia utente di `LyX`. Cioè i files definiscono quali oggetti appaiono in quali menu e quelli che appaiono nella toolbar. Riferirsi al Capitolo 3.4 per i dettagli.

2.1.3 Files che non volete modificare

Questi files sono usati internamente da `LyX` e generalmente non avete bisogno di modificarli a meno che non siate sviluppatori.

`CREDITS` questo file contiene la lista degli sviluppatori di `LyX`. Si può accedere a questo file attraverso il menu `Help` \triangleright `Credits`.

`chkconfig.ltx` questo è uno script `LATEX` usato durante il processo di configurazione. Non va lanciato direttamente.

`configure` questo è lo script usato per riconfigurare `LyX`. Crea files di configurazione nella directory in cui è lanciato.

2.1.4 Altri files che necessitano di una riga o due di spiegazioni...

`encodings ???`

`external_templates` questo file contiene i template disponibili nella nuova funzione `External Template`. Riferirsi a `Help` \triangleright `External Material` per ulteriori informazioni.

`language` questo file contiene una lista di tutti i linguaggi correntemente supportati da `LyX`.

2.2 La vostra directory di configurazione

Anche se si usa `LyX` come utente non privilegiato, è possibile voler cambiare la configurazione per un uso personale. La directory `UserDir` contiene tutti i files della vostra configurazione personale. E' questa la directory descritta come "user directory" in `Help` \triangleright `Version`. Questa directory è usata come mirror di `LyXDir`, il che significa che ogni file in `UserDir` è un sostituto del corrispondente file in `LyXDir`. Ogni file di configurazione descritto nelle sezioni precedenti può essere posizionato sia nelle grandi directory di sistema, in quel caso agirà su tutti gli utenti, sia nella vostra directory locale per un uso di carattere personale.

Per rendere chiare le cose, si forniscono alcuni esempi:

- Per personalizzare il comportamento di `LyX` prima della versione 1.1.6, l'utente doveva copiare il file `LyXDir/lyxrc.example` in `UserDir/lyxrc` e modificarlo direttamente. La nuova versione di `LyX` legge ancora questo file, se presente in `UserDir`, ma ogni cambiamento successivo fatto usando

il nuovo menu **E**dit▷**P**references sono salvate nel nuovo file `preferences`. In seguito `lyxrc` non sarà più letto e potrà essere rimosso.

- Quando si riconfigura attraverso **O**ptions▷**R**econfigure L_YX lancia configurare ed i files risultanti sono scritti nella vostra directory di configurazione locale (vedere la sezione 3.10 per avere una lista dei settaggi di questa sezione). Questo significa che ogni classe di testo addizionale che potreste aver aggiunto alla directory `UserDir/layouts` sarà aggiunta alla lista delle classi nel menu **L**ayout▷**D**ocument.
- Similmente, se avete installato alcune classi di documento L^AT_EX nella vostra home directory, che L^AT_EX può trovare nel vostro path `TEXTINPUTS`, queste verranno mostrate nella lista delle vostre classi di testo.¹
- Se acquisite della documentazione aggiornata dal sito ftp di L_YX e non potete installarla perchè non avete i permessi di amministratore sul vostro sistema, potete semplicemente copiare i files in `UserDir/doc/` e saranno raggiungibili attraverso il menu **H**elp!

2.3 Lanciare L_YX con configurazioni multiple

La libertà di configurazione della relativa directory locale può non essere sufficiente se volete avere più di una configurazione a vostra disposizione. Ad esempio, vorreste poter usare differenti scorciatoie da tastiera o settaggi della stampante in tempi diversi. Potete ottenere questo attraverso più directory che verranno richiamate al run-time.

Richiamando L_YX con lo switch da riga di comando `-userdir <some directory>` permette al programma di leggere la configurazione da quella directory e non dalla directory di default (potete scegliere la directory di default lanciando L_YX senza lo switch descritto sopra). Se questa directory non esiste, L_YX può crearla per voi proprio come fa per la directory di default la prima volta che si lancia il programma. Potete modificare le opzioni di configurazione in questa `Userdir` addizionale esattamente come fareste per la directory di default. Queste directory sono completamente indipendenti (ma continuate a leggere). Notare che settare la variabile ambientale `LYX_USERDIR_11x` allo stesso valore ha esattamente lo stesso effetto.

Avere molte configurazioni richiede anche più manutenzione: se volete aggiungere un nuovo layout a `Userdir/layouts` che volete disponibile per tutte le vostre configurazioni, dovete aggiungerlo ad ogni directory separatamente. Potete evitare questo col seguente trucco: dopo che L_YX ha creato la directory addizionale, la maggior parte delle sottodirectory (vedi sotto) sono vuote. Se volete che la nuova configurazione faccia da mirror ad una esistente, sostituite la sottodirectory vuota con un link simbolico alla relativa sottodirectory di una configurazione esistente. Fate attenzione tuttavia alla sottodirectory `doc/` dato

¹ come pure se voi o L_YX abbiate un file `.layout` per esso, naturalmente.

che contiene un file generato dallo script di configurazione (accessibile anche attraverso Options▷Reconfigure 3.10) che è specifico per ogni configurazione.

Capitolo 3

Il popup Edit ▷ Preferences

3.1 Usare il popup per la prima volta

Storicamente, è possibile personalizzare il comportamento di LyX attraverso la modifica del contenuto del file `UserDir/lyxrc`. Questo è cambiato in LyX 1.1.6 perchè ora esiste un menu Edit ▷ Preferences per semplificare il processo di semplificazione. `UserDir/lyxrc` verrà ancora letto fino a quando non si usa il nuovo menu per la prima volta. Successivamente, ogni cambiamento sarà salvato in un nuovo file, `preferences`, e `lyxrc` non sarà più letto. Potrà così essere rimosso.

Il nuovo file `UserDir/preferences` conterrà solo i cambiamenti che sono stati fatti nel comportamento di base, alcuni dei quali sono ben codificati in LyX ed altri sono contenuti nel file di sistema `LyXDir/lyxrc.defaults`. Notate che in entrambi i files le righe che iniziano con “#” sono solo commenti e non sono interpretati. Tuttavia, solo gli amministratori di sistema dovrebbero editare `LyXDir/lyxrc`. Gli utenti dovrebbero usare la voce Edit ▷ Preferences per creare e modificare il loro file personale `UserDir/preferences`.

Speriamo che il menu Edit ▷ Preferences sia ampiamente autoesplicativo. A quasi tutti i comandi sono stati associati dei commenti, così da non crearvi particolari difficoltà ad effettuare modifiche a vostro piacimento. Prima di analizzare alcuni dei comandi più importanti di seguito, vogliamo fare un avvertimento: L'applicazione di alcuni dei cambiamenti (es. fonts dello schermo) hanno effetto istantaneo. Altri (es. cambiamenti nel file associazioni) non l'avranno. Abbiate pazienza; questo è un work-in-progress. Se nulla sembra accadere come cambiamento salvate i cambiamenti e riavviate LyX.

3.2 Schermo & Fonts

Il carattere usato per visualizzare i vostri documenti sullo schermo di LyX è molto importante, dato che leggerete tutti i documenti con questo carattere. Inoltre è importante che il carattere sia leggibile ed abbia il miglior aspetto

possibile. Il LyX team ha provato a fornire il migliore carattere predefinito, ma dato che praticamente tutti i sistemi X11 sono differenti, è molto probabile che il carattere predefinito sia sub-ottimale per il vostro sistema. Fortunatamente è possibile ovviare a tale situazione, ma prima di chiarirvi le modalità secondo le quali procedere, sarebbe bene che sapeste qualcosa di più sui font in quanto la loro scelta nasce da un compromesso tra le vostre preferenze e le capacità del sistema col quale lavorate.

Notate che questa sezione tratta solo dei caratteri visibili sullo *schermo* all'interno della finestra di LyX. I caratteri che appaiono sulla *carta* in uscita dalla stampante sono indipendenti da questi e sono determinati dalla classe del documento. Consultate la *Guida dell'utente* per imparare a cambiare il carattere della versione stampata del vostro documento.

Di base, i caratteri che appaiono sullo schermo sono di due tipi differenti: caratteri scalabili e caratteri bitmap non scalabili. La distinzione può sembrare leggermente arbitraria, dato che i font non-scalabili sono generalmente scalabili nella maggior parte dei moderni renderers di font. La differenza sta nella *qualità* dello scaling, e nella *velocità* con cui essi vengono visualizzati. Da ciò ne deriva l'importanza di capire dove sia bene usare l'uno o l'altro tipo di font.

I caratteri scalabili sono costruiti a partire dai *contorni* di ciascun glifo (cioè carattere) presente nel font. Questo significa che ogni glifo è definito attraverso curve matematiche che bene si adattano allo scaling di qualsiasi dimensione. Il renderer di font ha quindi il compito di interpretare tale definizione e generare una piccola figura composta di pixels in relazione alla grandezza ed al glifo richiesto dal programmatore. Ne segue che l'aspetto dei font scalabili risulta buono per ogni dimensione, più realisticamente per quasi tutte le dimensioni. Infatti i font scalabili sono definiti in modo astratto e di conseguenza può essere difficile ottenere un buon rendering per taglie piccole, dove ogni pixel deve essere perfettamente calcolato per ottenere una buona immagine. Tale operazione è tecnicamente attuabile mediante la definizione matematica, ma per consentire un rendering ragionevolmente veloce, si preferisce raggiungere un compromesso, con il risultato è che i font scalabili sono più difficili da leggere per le piccole dimensioni.

I font bitmap invece sono definiti da grafica bitmap sin dall'inizio, così appaiono buoni per ogni dimensione richiesta. tuttavia, non scalano bene, perchè per scalare un glifo, ogni pixel è allargato in più pixel. E' quanto accade se cercate di allargare una fotografia in *xv* o in qualsiasi altro programma di manipolazione. Per ridurre questo effetto, i font bitmap sono forniti in varie dimensioni prestabilite, in genere da 8 pixel a 34 pixel o più in relazione ai possibili utilizzi. Il vantaggio derivante dall'uso di font bitmap è che per visualizzare ogni glifo non sono necessari calcoli complessi, così da rendere i font bitmap più velocemente rappresentabili sullo schermo rispetto ai font scalari. Lo svantaggio consiste nel fatto che le dimensioni non comprese in quelle predefinite, devono essere ottenute raddoppiando i pixel, e conseguentemente l'aspetto non è dei migliori.

Se ne conclude quindi che i font bitmap sono generalmente i migliori per le piccole taglie, dove disponibili, mentre i font scalabili sono da preferire per grandi dimensioni. La logica conclusione sarebbe quella di usare i font bitmap

per le piccole dimensioni, ed i font scalabili per le dimensioni superiori. Sfortunatamente questa non è una buona idea, dato che i font bitmap e i font scalabili non sono stati progettati per essere usati insieme ed il risultato derivante dal loro contemporaneo utilizzo sarebbe piuttosto sgradevole. La soluzione ottimale è dunque quella di provare entrambi gli schemi e scegliere tra i due quello che meglio si adatta alle proprie esigenze.

Di default, Lyx usa i font bitmap non-scalabili. Per i font serif, è usato *times*, per i font sans serif, è usato *helvetica*, mentre *curier* è usato come font monospazio stile macchina da scrivere.

In seguito descriveremo cosa fare se in LyX il testo non ha un bell'aspetto. Partiremo con i parametri più importanti: DPI e lo zoom dei fonts.

3.2.1 Settaggi DPI e Zoom di Font

LyX automaticamente prova a scalare i fonts in modo da farli sembrare il più possibile uguali a quelli che verranno stampati, a meno del cosiddetto fattore di zoom.

Per far funzionare su ogni sistema, questo meccanismo sfrutta i settaggi dei DPI (dots per inch) dello schermo. Il settaggio dei DPI del vostro sistema è rilevato automaticamente da LyX attraverso le informazioni che fornisce il server X. Potete controllare cosa rileva LyX circa i DPI, lanciando LyX in questo modo: `lyx -dbg 2`.

Su molti sistemi, X non è settato correttamente, quindi dovrete verificare la correttezza dei settaggi manualmente. Lanciate `xdpyinfo | more` e scrivete quali sono i DPI per la risoluzione che usate (che dovrebbe essere simile a quella che LyX rileva). E' il numero menzionato come "risoluzione". Inserite poi anche il numero di pixel che avete in larghezza (il primo numero sotto "dimensioni").

Dopodichè prendete il buon vecchio righello e misurate la larghezza visibile dell'immagine su vostro monitor. Convertite queste misure in pollici, se usate un righello in centimetri, dividendo per 2.54. Ora è possibile determinare il corretto settaggio DPI per il vostro schermo, dividendo il numero dei pixels della larghezza per la larghezza dell'immagine sullo schermo del monitor. Se questo numero è superiore a, per esempio, 5 DPI rispetto al valore autodeterminato, dovrete correggere l' X setup, o almeno dire a LyX che il valore di DPI è diverso da quello rilevato.

Se non potete correggere l'X setup (che sarebbe la soluzione migliore perchè ne beneficerebbero altri programmi oltre a LyX), è possibile comunicare a LyX il corretto settaggio DPI attraverso il menu Preferences.

Se il testo è troppo piccolo o troppo grande per il vostro gusto, dovrete cercare di rimediare attraverso lo zoom sul carattere. Questo settaggio è usato per scalare la grandezza del punto del testo. Se il vostro DPI è corretto, e lo zoom del carattere è settato a 100, questo significa che LyX proverà a mostrare il testo esattamente alla stessa grandezza della pagina stampata. Se il fattore di zoom viene settato a 200, il testo visualizzato sarà il doppio di quello che si vedrà sulla pagina stampata. Naturalmente questo accadrà solo se LyX riesce a trovare ed usare un carattere che abbia una grandezza appropriata. Ma, dato

che LyX è comunque un sistema WYSIWYM, questa limitazione, eventuale, è improbabile.

Il valore di default dello zoom è 150 dato che un monitor è generalmente più grande che un pezzo di carta, ma dovrete comunque provare a modificarlo tramite il comando `Font Zo.default` nel menu `Preferences` per trovare una grandezza che vi soddisfi. Quando avete trovato un settaggio che possa andare bene per voi (tip: usate il bottone `Apply` per lasciare la finestra aperta mentre fate esperimenti), potete poi rendere questi settaggi come default usando il comando `Save`.

Mentre è spesso possibile trovare una grandezza adatta al testo sullo schermo, questo non necessariamente vuol dire, che i fonts siano i migliori ottenibili dal proprio sistema. Per aiutarvi ad avere il massimo dal vostro sistema, potete usare i comandi di definizione dei font per raffinare l'aspetto del testo più in dettaglio.

3.2.2 Comandi di definizione dei Fonts

Come menzionato, LyX usa fonts bitmap non scalabili di default. Per i fonts serif, è usato *times*, per i fonts sans serif è usato *helvetica*, mentre *courier* è usato come font del tipo monospazio/macchina da scrivere.

Potete cambiarli attraverso il menu `Preferences`. Il numero dei fonts disponibili varia a seconda dei sistemi, ma il programma `xfontsel` dovrebbe essere reperibile ovunque. Usate questo programma per trovare i fonts adatti. Quando avete trovato un font che vi piace, provate ad inserire i primi due elementi del nome (chiamati "fndry" e "fmly" in `xfontsel`) nel campo appropriato nel menu `Preferences` e premere `Apply`. LyX poi riformatterà il vostro documento usando il nuovo font, e se il font vi piace, dovrete salvarlo con `Save`. Un punto di partenza per un nuovo font è vedere se il font scalabile "utopia" è disponibile. Tip: potete vedere se un font è scalabile o bitmap controllando i campi "resx" o "resy" in `xfontsel`. Se il valore 0 è disponibile, il font è scalabile. Se il valore 0 non è disponibile, il font è bitmap.

Prima che proviate a modificare un font bitmap perchè la larghezza più grande sembra "bloccata", dovrete provare a selezionare il bottone "Usa fonts scalabili". Questo è utile solo se usate fonts bitmap, perchè solo questi non scalano perfettamente. Se definite questo flag, LyX userà solo le grandezze fisse del carattere disponibili, e questo garantisce che tutti i fonts bitmap sembrano buoni. (Potete vedere quale grandezze individuali del carattere son o disponibili con il comando `xlsfonts`. Provate `man xlsfonts`). Tuttavia il prezzo da pagare è che la differenza tra la grandezza del font sullo schermo e la grandezza del font sulla carta sarà più grande perchè LyX dovrà accontentarsi della grandezza più vicina disponibile, e non proverà a scalare la grandezza per adeguarsi. Inoltre, c'è il rischio che alcune grandezze differenti da un punto di vista logico, come `Large` e `Larger`, saranno mappate come lo stesso font sullo schermo, rendendo difficile identificare le differenze sullo schermo. Abbiamo deciso di non usare i fonts scalabili di default a causa di questi effetti, ma dato che LyX è un sistema WYSIWYM, a molte persone piace comunque usare i flag, pur sapendo che la

grandezza del carattere sullo schermo potrebbe non essere veritiera. Ma ricordate che questo flag crea una differenza solo se usate font bitmap. I font scalabili, come ora potete capire, non verranno toccati da questo tipo di problema.

Un'ultima annotazione su questo flag: dovete sapere che non c'è nulla di sbagliato nell'usare fonts scalabili e bitmap contemporaneamente per differenti scopi. Per esempio, è abbastanza comune usare lo scalabile "Utopia" per il testo serif, assieme alla versione bitmap di "Helvetica". E potete tranquillamente selezionare "Usa fonts scalabili" senza problemi: sarà solo applicato al font Helvetica.

Qualche volta gli artefatti introdotti dall'uso del flag possono essere eliminate usando il comando `fine-detail` che definisce a quali grandezze di punto corrispondono le differenti grandezze logiche dei caratteri. Lanciate LyX con `lyx -dbg 513` per vedere come funziona concretamente e provate ad aggiustare le voci corrispondenti nel menu Preferences fino a quando i risultati vi soddisfanno. Questo è difficile da fare, perchè LyX usa i settaggi DPI e quelli di zoom del font per calcolare l'esatta grandezza del font sullo schermo da chiedere al server X. Se non riuscite ad ottenere risultati soddisfacenti con la tecnica dello "sbaglia e riprova", potete rendere il processo più trasparente se settate assieme i DPI e lo zoom a 100, anche se si sa che non è corretto. questo naturalmente renderà i font scalabili non perfetti, quindi fate attenzione.

3.2.3 Codifica dei Fonts

Di default, LyX userà dei font per scrivere un testo Western European, inclusi tutti i tipi di inglese. Viene definito attraverso il cosiddetto *font encoding*. Se volete usare Lyx per scrivere ad esempio un testo in Eastern European, Cirillico oppure ogni altro linguaggio non coperto dallo standard ISO-8859-1 di codifica dei caratteri, potete definirne uno differente con il settaggio del valore `\screen_font_encoding`. Questo richiede che siano installati speciali fonts. Potete usare `xfontsel` per queste modifiche: controllate i campi "rgstry" e "encdng" per valori ISO-8859-X diversi da ISO-8859-1, e quindi cercate quello che contiene i caratteri della lingua prescelta. Se non lo trovate è ancora possibile definirlo nel file `lyxrc`. Altrimenti bisogna cercarlo sul Web.

Quando settate LyX per l'uso di una codifica differente, dovrete considerare i cambiamenti nei fonts della finestra pop-up di LyX. Per esempio il menu Table of Contents non sarà comprensibile fino a quando non dite a LyX di usare un font differente. Di default, il menu font è settato su `*-helvetica-medium-r`, ma spesso helvetica non è disponibile nella codifica dei font di cui avete bisogno.

Inoltre potete cambiare questo font con il comando `\screen_font_emenu`.

Come potete vedere, ci sono alcune opzioni che possono essere usate per affinare l'aspetto dei vostri fonts. Questo non dovrebbe spaventarvi dal maneggiare coi settaggi, poichè dopo tutto, vi augurate di usare molto LyX in futuro. E contrariamente ai veri word processors WYSIWYG, dove siete costretti ad usare fonts che hanno un aspetto buono sia sul video che sulla carta, LyX vi dà la possibilità di usare fonts che sono disegnati per essere visti sullo schermo, se siete abbastanza fortunati da averli.

Nella versione di LyX, proveremo a supportare il font renderer T1Lib che fornisce un testo con effetto *anti-aliased*. Questo significa che ogni glifo è costituito da molte ombre colorate piuttosto che da soli due colori, creando un effetto di alta risoluzione. Questo non renderà solo il testo più leggibile sullo schermo, ma permetterà anche a LyX di avvicinare i fonts visti sullo schermo a quelli che verranno stampati, dato che sarà ad esempio possibile settare un valore di 14.4 punti al contrario di ora che è possibile solo scegliere il valore arrotondato 14.

3.3 Bindings

Le associazioni sono usate per assegnare una funzione ad un tasto. Molti file con associazioni precompilate sono disponibili: un set di associazioni CUA (simile ai tipici insiemi di assegnazioni di tastiera PC e CDE), un set Emacs di associazioni, per quelli di noi che seguono la cosiddetta “One True Way” e rifiutano di sottostare agli standards,¹ come pure associazioni speciali (broadway e hollywood) ed altri linguaggi (Francese, Tedesco, ecc..)

Se, tuttavia, volete personalizzare le associazioni secondo i vostri specifici gusti, allora copiate il file della directory `LyXDir/bind/` che meglio si adatta nella vostra `UserDir/bind/` e modificalo. Non dimenticate di caricare questo nuovo file in LyX attraverso il menu `Preferences` (per il momento dovete riavviare LyX per rendere effettivi i cambiamenti).

LyX supporta l'internazionalizzazione dell'interfaccia utente (vedere il Capitolo 5). Se il vostro *locale* viene settato con la variabile d'ambiente `$LANG`, LyX proverà ad usare i bindfile preponendo `$LANG_` al loro nome. Ad esempio, potete mettere una copia tradotta di alcuni bindfile nella vostra directory `bind/personale` e LyX li userà automaticamente. [*Nota dell'autore: non sono sicuro se è ancora vero ma forse qualcuno che utilizza questo metodo potrebbe testarlo—AL*].

La sintassi dei files `.bind` è la seguente:

```
\bind <key combination> <lyx-function>
```

Sia le combinazioni di tasti che la `lyx-function` (inclusi gli argomenti) devono essere racchiusi tra le “doppie virgolette”. Tutte le funzioni di LyX sono elencate nella *Guida di Riferimento*.

3.4 User Interface

L'aspetto del menu e della toolbar può essere cambiato attraverso il menu `Preferences`. Basta semplicemente cambiare il file `.ui` in `LyXDir/ui/`. Per il momento esiste solo un file, `default.ui`, ma sentitevi liberi di sperimentare. Copiate il file che create nella directory `UserDir/ui/` e andate! Notate che, per il momento, dovete riavviare LyX per rendere effettivi i cambiamenti.

¹Sto scherzando, ovviamente!

La sintassi dei files `.ui` è la seguente: date uno sguardo a `default.ui`. Le righe riferite alla barra dei menu, `menu` e `toolbar` devono terminare con un esplicito `end`. Possono poi contenere dei sottomenu, oggetti, separatori, icone e nel caso del menu “file”, una riga `lastfiles`. Un piccolo avvertimento. I `sottomenu` possono essere inseriti in un `menubar` o `menu`, ma sono sempre definiti come `menus` non come `submenu`.

3.5 Convertitori, Formati e Visualizzatori

LyX ha un nuovo e potente meccanismo per convertire da e verso ogni formato di file utilizzando programmi esterni. Prendiamo un paio di formati, ad es. LaTeX e PDF. Ora definiamo un convertitore da un formato all’altro. Nel nostro esempio ci sono due possibili meccanismi.

1. Una conversione diretta, da LaTeX a PDF usando `pdflatex`
2. Una via più contorta attraverso formati e convertitori intermedi: LaTeX a DVI (usando `latex`) a PostScript® (usando `dvips`) a PDF (usando `ps2pdf`)

LyX sceglierà sempre la via più veloce, così dovrete specificare due differenti nomi di Format per i files `.pdf` per poterli usare entrambi. Entrambi sono inclusi di default nel menu Preferences. Date un’occhiata e poi createne dei vostri!

Inoltre, ogni Format può essere visto con un Viewer associato ad esso. Ad esempio, potreste voler usare `ghostview` per esaminare i files PostScript®.

[Nota dell’autore: Questa sezione probabilmente ha bisogno di essere espansa, ma sostituisce completamente la prossima. In più, non sono sicuro di quanto importanti siano ancora le sezioni Stampante e Cambiare Colori. La sezione Ascii-Export è ancora importante, sebbene ha bisogno di essere usata per differenziare l’approccio `lyxc` da quello del menu Preference. — AL].

3.6 Anteprima sullo schermo

Oltre alla normale visualizzazione dei documenti attraverso LyX, è possibile avere un’anteprima del vostro documento nella forma in cui apparirà sulla carta dopo stampato. Di default LyX usa `xdvi` per l’anteprima dei files `.dvi` (l’output di LaTeX).

LyX può anche usare `dvips` per convertire i files `.dvi` in `.ps` e `ghostview` per l’anteprima. Si possono usare numerose opzioni nel file `lyxc` per personalizzare queste caratteristiche.

`\view_dvi_command` Permette di rimpiazzare `xdvi` con altri `.dvi` previewer, o per passare argomenti a `xdvi`.

`\view_ps_command` Permette di rimpiazzare `ghostview` con il vostro PostScript previewer preferito, oppure per passare argomenti a `ghostview`.

`\view_ospic_command` Definisce il programma usato per ottenere anteprime a tutto schermo delle figure PostScript®. Non è possibile includere alcuna opzione in questo tipo di settaggio.

`\ps_command` Definisce il programma per usare le figure PostScript all'interno di un documento LyX. Non è possibile includere nessuna opzione. Se viene definita una stringa vuota (""), LyX non processerà nessuna figura. Questo può essere utile se avete una macchina lenta e lavorate con molte figure.

`\default_papersize` Setta la grandezza della carta usata da LyX per fare l'anteprima dei documenti. Il valore di default è la carta USletter. Se L^AT_EX è configurato per usare una differente grandezza di default, potrete cambiare questo settaggio.

3.7 Opzioni ASCII-Export

Ci sono alcuni comandi che possono essere usati per “ripulire” i files di testo esportati in ASCII. Notate che LyX automaticamente riconosce ed usa il miglior settaggio per il vostro sistema sin dalla fase di installazione, ma è possibile modificarli se non vi piacciono.

`\plaintext_roff_command` Questo comando vi permette di settare il comando per produrre migliori tabelle ASCII con i comandi UNIX `groff/troff/nroff` (referirsi alle relative man pages per ulteriori informazioni). Non definirlo per niente dice a LyX di usare il formattatore interno (inferiore).

`\plaintext_linelen` Con questo comando potete settare la lunghezza di default della riga dell'output di un file ASCII. Settarlo a 0 significa righe infinite. Il Default è 75 caratteri.

Examples:

```
# This let's you generate ASCII tables with latin1
# characters, using the groff formatter
#
\plaintext_roff_command groff -t -ms -Tlatin1"

# This let's you generate ASCII tables using the
# nroff formatter
#\plaintext_roff_command "nroff"
# This sets the line length to 80 characters
#
\plaintext_linelen 80
```

3.8 Stampante

Ci sono una serie di comandi che vi aiutano a settare il comando della stampante da LyX. Normalmente si usano solo alcuni di questi, ma vengono riportati tutti per questione di completezza.

`\printer` Questo valore definisce il nome di default della vostra stampante.

`\print_command` Questo valore definisce quale comando LyX deve essere usato per stampare il vostro documento.

`\print_evenpage_flag` Questo valore definisce quale flag del comando di stampa comporterà la stampa delle sole pagine pari.

`\print_oddpage_flag` Questo valore definisce quale flag del comando di stampa si occuperà di stampare solo le pagine dispari.

`\print_reverse_flag` Con questo comando, potete definire quale comando di stampa si occuperà di stampare le pagine in ordine inverso.

`\print_landscape_flag` Questo comando definisce quale flag è settata quando si richiede di stampare il proprio documento in landscape. se il documento è stampato in portrait, nessuna opzione è passata.

`\print_to_printer` Se specificate un esplicito nome per la stampante, dovrete preporre questo nome con un flag dalla linea di comando della stampante. Questo comando definisce quale flag dovrebbe apparire prima del nome della stampante.

`\print_to_file` Se volete stampare su un file, dovrete utilizzare un flag prima del nome del file. Questo comando vi permette di definire come apparirà questo flag.

`\print_file_extension` Quando stampate su un file, con questo comando potete definire quale estensione potrà avere il file stampato.

`\print_extra_options` Con questo comando potete definire una serie di flags che saranno inseriti nei comandi di stampa dopo gli altri flags.

`\print_spool_cmd` Se si specifica qualcosa in questo comando, LyX stamperà in due fasi: Prima sarà eseguito il programma specificato nel comando `\print_command` con le direttive per stampare su un file. Poi, il comando specificato con `\print_spool_cmd` sarà eseguito col file di output col precedente comando sulla riga di comando, e, se specificato, la direttiva (`\print_to_printer`) per mandare l'output ad una stampante definita prima del nome del file.

Tutto sommato, speriamo che questo vi permetta di stampare i vostri documenti senza problemi.

3.8.1 Changing Colors

E' possibile cambiare i colori dello schermo usando il nuovo menu `Preferences`. Alternativamente, se vi sentite particolarmente perversi potete usare la funzione associabile `set-color` (vedere la *Guida di Riferimento*). L'input avrà il seguente formato:

```
set-color LyXName X11Color
```

Di seguito è presentata una lista (parziale) delle funzioni e dei colori di default:

Nome LyX	Scopo	Colore di default (X11)
background	text background	black
foreground	text foreground	linen
latex	\LaTeX code	DarkRed
math	Mathed formulae	DarkBlue
mathline	fraction Lines, brackets, etc.	Blue
mathbag		AntiqueWhite
mathframe		black
selection	selection background	LightBlue

La lista completa si trova correntemente solo nel file sorgente `LColor.cpp`

3.9 \TeX

Alcuni siti² hanno ancora la versione 2.09 di \LaTeX . Sebbene non possiamo immaginare perchè questo dovrebbe essere ancora il default dato che la versione 2.09, essendo questa obsoleta dal 1994, dovremo comunque continuare a riferirci con LyX a tale ambiente. Non bisogna *pensare* però ad un "fixing" di LyX per girare sotto \LaTeX 2.09. Aggiornare il vostro \LaTeX sarà più facile, ed è comunque la soluzione migliore. Molti grandi sistemi dovrebbero avere, anche se di default hanno la 2.09, una versione disponibile di \LaTeX 2, ma forse l'eseguibile potrebbe avere un altro nome. Potete cambiare il nome del comando \LaTeX appropriato, come ad esempio "lyxrc"³ con la linea:

```
\latex_command latex2e
```

nel vostro file `lyxrc`. Al momento dell'installazione, LyX proverà a rilevare automaticamente se questo settaggio è necessario, quindi normalmente, non ve ne dovrete preoccupare.

²come la mia macchina da ufficio – Dave johnson

³come la mia macchina da ufficio – Dave johnson

3.10 I settaggi autorilevati

Ci sono numerosi oggetti che sono rilevati quando lanciate `Options▷Reconfigure`. In questa sezione elencheremo quelle che si riferiscono a `lyxrc`. Quello che segue, tutte le chiamate `*_command` sono settate a “niente” se non viene stato trovato il comando opportuno. In questo caso, la corrispondente funzione non è disponibile.

`\latex_command` è settato su un processore $\text{\LaTeX} 2_{\epsilon}$ valido (programmi testati sono `latex` e `latex2e`).

`\ps_command` è settato su `gs` se avete Ghostscript installato.

`\view_ps_command` è settato su `gv` o `ghostview` se sono stati trovati nel sistema (`\view_pspic_command` è settato per usare lo stesso programma).

`\plaintext_roff_command` usa `groff` oppure `nroff+tbl`, a seconda di quello disponibile.

`\chktex_command` è settato su `chktex` con in più una serie di opzioni.

`\fax_command` è settato su `fax send` o `sendfax`, che dipende da che pacchetto `fax` è stato trovato. Le necessarie opzioni del comando vengono anche aggiunti.

`\print_spool_command` viene settato to `lp` su sistemi (chiamati System V) che hanno questo comando e `lpr` altrimenti (sistemi BSD).

`\print_spool_printerprefix` viene settato a `-d` o su `-p` a seconda della presenza di `lp` o `lpr`.

`\font_encoding` è settato su `T1` se i fonts `ec` sono rilevati e \LaTeX ha un supporto interno per questi fonts. Potete settarli manualmente se avete i cosiddetti fonts `dc`.

3.11 Il resto

Ci sono molti altri comandi `lyxrc` che possono essere usati per personalizzare il comportamento di `LyX`. Abbiamo attualmente ancora necessità di documentarli in questa sede, ma ancora, il file `lyxrc.example` è riccamente commentato, così consigliamo di dare un’occhiata per avere un’idea di quello che potete personalizzare. Potete infine tentare la fortuna nella *Guida di Riferimento*, ma non ci sono garanzie dato che è molto difficile tenerla aggiornata.

Capitolo 4

Il Server L_YX

4.1 Introduzione

Il server L_YX è un metodo implementato in L_YX che abilita altri programmi a parlare con L_YX, a chiamare comandi L_YX e ad ottenere informazioni sullo stato interno di L_YX. E' uno strumento principalmente indirizzato ad utenti esperti, che dovrebbero trovarlo utile.

4.2 Iniziare il Server L_YX

Il server L_YX lavora attraverso l'uso di un paio di canali. Queste sono di solito locate nella vostra directory home e si chiamano ".lyxpipe.in" e ".lyxpipe.out". I programmi esterni vanno a scrivere in .lyxpipe.in e legge i dati da .lyxpipe.out. L'inizio di questi canali può essere specificato nel file lyxrc aggiungendo una linea del tipo:

```
\serverpipe "/home/myhome/.lyxpipe"
```

L_YX aggiungerà '.in' and '.out' per creare i canali. La precedente riga ha inoltre l'effetto di attivare il server L_YX. se uno dei canali già esiste, L_YX assumerà che c'è un altro processo L_YX in esecuzione e non lascerà partire il server. Per avere molti processi L_YX con rispettivi servers in contemporanea, dovete cambiare il lyxrc tra gli inizi dei programmi.

Se state sviluppando un programma client, potreste trovare utile abilitare una modalità di debugging dal server L_YX. Potete farlo facendo partire L_YX con `lyx -dbg 8192`.

Attenzione: se L_YX va in crash, non può gestire la rimozione dei canali, e quindi bisognerà farlo manualmente. Se L_YX parte e i canali sono già esistenti, non farà partire alcun server.

Potete trovare degli esempi completi scritti in C nella distribuzione dei codici come `development/server_monitor.c`.

4.3 Comunicazione normale

Per assegnare una chiamata L_YX, il client scrive una riga di codice ASCII nel canale di input. Questa linea ha il seguente formato:

LYXCMD:*clientname:function:argument*

In questo caso *clientname* è un nome che il client può scegliere arbitrariamente. Il suo unico uso è che L_YX gli risponderà se questi manda una risposta - così un client può mandare risultati da differenti richiedenti.

function è la funzione che volete che L_YX esegua. E' lo stesso dei comandi che dovrete usare in minibuffer.

argument è un argomento opzionale rilevante solo per alcune funzioni (ad esempio "auto-inserimento" che inserisce l'argomento come testo nella posizione del cursore.)

La risposta da L_YX arriverà nel canale di output e sarà del tipo

INFO:*clientname:function:data*

Dove *clientname* e *function* sono richiamati dalla richiesta di comando, mentre *data* è un'informazione più o meno utile a seconda dell'esecuzione del comando. Alcuni comandi restituiscono informazioni circa lo stato interno di L_YX, come il "font-state", mentre altri restituiscono risposte "vuote". Questo significa che l'esecuzione del comando ha avuto fine.

In caso di errori, la risposta da L_YX avrà questa forma

ERROR:*clientname:function:error message*

dover l'*error message* dovrebbe contenere la spiegazione del perchè il comando è fallito.

Esempi:

```
echo "LYXCMD:test:beginning-of-buffer:" >~/lyxpipe.in
echo "LYXCMD:test:get-xy:" >~/lyxpipe.in
read a <~/lyxpipe.out
echo $a
```

4.4 Notifica

L_YX può notificare ai clients gli eventi asincroni. Correntemente lo farà soltanto se l'utente assegna una sequenza di tasti alla funzione "notifica".

Il formato della stringa che L_YX manda è:

NOTIFY:*key-sequence*

dove *key-sequence* è la rappresentazione stampata della sequenza di tasti pre-muta dall'utente.

Questo meccanismo può essere usato per estendere il set di comandi L_YX ed implementare macro: assegnare una sequenza di tasti a “notifica”, lanciare un client che ascolta sul canale d'uscita, fornisce il comando relativo alla sequenza e inizia una funzione che può usare le chiamate e le richieste di L_YX per assegnare un comando o una serie di comandi a L_YX.

4.5 Il Protocollo Server L_YX semplice

Lyx implementa un semplice protocollo che può essere usato per la gestione delle sessioni. Tutti i messaggi sono della forma:

LYXSRV:*clientname:protocol message*

dove *protocol message* può essere “hello” o “bye”. Se “hello” è ricevuto da un client, L_YX ritornerà indietro per informare il client che sta ascoltando il suo messaggio, mentre “bye” mandato da L_YX informa i clients che L_YX sta chiudendo.

Capitolo 5

Internazionalizzare LyX

LyX supporta l'uso di interfacce tradotte. L'ultima volta che abbiamo controllato, LyX forniva testi in 14 linguaggi diversi oltre l'inglese di base. Il linguaggio scelto è definito linguaggio *locale*. (per ulteriori letture sui settaggi locali, riferirsi alla documentazione del vostro sistema operativo. Per Linux, le manual page di locale(5) sono un ottimo punto di partenza).

Notate che queste traduzioni funzionano, ma contengono alcuni difetti. In particolare, tutti i menu di pop-up sono stati progettati in inglese, questo significa che alcune traduzioni potrebbero essere troppo grandi rispetto allo spazio ad esse assegnate in fase di progettazione. Questo è solo un problema di visualizzazione che non causa molti problemi. Inoltre, potrete notare che molte delle traduzioni non definiscono le scorciatoie per tutto. A volte, molto semplicemente, non ci sono abbastanza lettere disponibili. Altre volte, il traduttore non lo ha ancorafatto. Cercheremo di ovviare a questi inconvenienti nelle prossime versioni di LyX.

5.1 Selezionare un linguaggio alternativo per l'interfaccia utente

Questa possibilità è disabilitata di default, ciò significa che il testo sarà in inglese. Per abilitare un linguaggio alternativo bisogna settare la variabile ambientale appropriata. Usate `setenv LANG xx` per le interfacce di classe `csh` o `export LANG=xx` per la classe `sh`. Sostituite le `xx` con il codice di due lettere del linguaggio che volete. Ad esempio `no` è Norvegese.

Evidentemente vorrete mettere la linea appropriata in `~/.Xsession` or `~/.xinitrc` in modo che la traduzione sia attivata di default. Ricordate che se LyX è configurato e compilato con `--disable-nls`, questo non funzionerà.

5.2 Usare scorciatoie per linguaggi alternativi

Una causa principale di inconsistenza nelle versioni tradotte di L_YX è il fatto che tutte le scorciatoie da tastiera per le funzioni del menu sono basate sulla versione inglese del testo. Ad esempio, l'entrata nel menu `M-f t` aprirà il popup `nuovo-file-da-template`. Questo è semplice da cambiare, come tutte le scorciatoie delle operazioni del menu definite nel file unico `LyXDir/bind/menus.bind`. Ogni utente può avere la propria copia di questo file nella propria directory privata `~/.lyx/bind`, che avrà precedenza rispetto agli altri file di sistema. In questa copia privata ognuno può facilmente cambiare le definizioni, ad esempio per la versione tedesca avremo `M-f t` in `M-d v` (per `Date▷Vorlage`). Per alcuni linguaggi, gli esempi per adattare i files `menus.bind` si possono trovare in `LyXDir/bind/xx_menus.bind` (`xx` è il codice del linguaggio). Questi, se esistono (e se il vostro settaggio locale è `$LANG=xx`), saranno automaticamente usati da L_YX.

Dovete fare attenzione a che le nuove definizioni per le scorciatoie non coincidano con altre prima definite. Ad esempio, l'associazione `M-d v` genererà un errore per quelli che usano `(x)emacs.bind`, dove `M-d` è unito a `word-delete-forward`.

Dalla versione 1.0 di L_YX, i nomi tradotti per i menu top level (tradotti nei files "po") sono usati dal meccanismo di assegnazione, quindi il menu *tradotto* deve essere usato nel `menus.bind` tradotto.

5.3 Tradurre L_YX

5.3.1 Tradurre l'interfaccia grafica utente (messaggi di testo).

L_YX usa la libreria GNU `gettext` per gestire l'internazionalizzazione dell'interfaccia. Per far parlare a L_YX il vostro linguaggio preferito in tutti i menu e le caselle di dialogo, avete bisogno di un `po`-file per quel determinato linguaggio. Quando questo è reperibile, dovrete creare da questi un `mo`-file ed installarlo. Tutti i passaggi necessari ad eseguire questa operazione sono spiegati nella documentazione di GNU `gettext`, ma, in breve, questo è quello che dovete fare (`xx` rappresenta il codice del linguaggio):

- Copiare `LYX-SOURCE-DIR/po/lyx.pot` in `xx.po` (se `lyx.pot` non esiste, può essere rifatto con `make lyx.pot` in quella directory, oppure potete usare un `po`-file esistente per altri linguaggi come se fosse un template).
- Editare `xx.po`¹. Per alcune etichette di menu, ci sono anche scorciatoie da tastiera che dovrebbero essere tradotte. Questi tasti sono segnati dopo un `'|'`, e dovrebbero essere tradotti in relazione alle parole e alle frasi del linguaggio `xx`. C'è uno strumento chiamato `scgen.pl` scritto in Prolog in `LYX-SOURCE-DIR/development/tools/` che può essere utile per creare

¹Raccomandiamo di usare Emacs per fare questo, dato che la distribuzione `gettext` include un appropriato supporto.

determinate scorciatoie. Notate che XForms (almeno la versione 0.86) non può manipolare altro che scorciatoie di caratteri di 7 bit. Dovreste inoltre aggiungere delle informazioni all'inizio del nuovo file po come il vostro indirizzo email ecc, così gli altri sanno dove possono rintracciarvi.

- Create `xx.mo`. Questo può essere fatto con `msgfmt -o xx.mo < xx.po`
- Copiare il file mo nella vostra directory locale, precisamente nella directory dei messaggi dell'applicazione per il linguaggio `xx`, e, sotto il nome `lyx.mo` (es. `/usr/local/share/locale/xx/LC_MESSAGES/lyx.mo`)

Aggiungere un nuovo file po alla *distribuzione* di L_YX comprende il cambiamento degli script di configurazione ed oltre, ma per come lavora gettext, attualmente non avete bisogno di codice L_YX per tradurlo—se si ha `lyx.pot` (o un file po esistente), bastano gli strumenti di gettext.

Se avete scritto un file di traduzione per un linguaggio che attualmente L_YX non supporta, ritenetevi liberi di introdurlo magari mandando una patch. In questo caso vi raccomandiamo di leggere il README della directory `LYX-SOURCE-DIR/po/` per ulteriori informazioni.

5.3.2 Tradurre la documentazione.

La documentazione online (nel menu Help) può (e dovrebbe!) essere tradotta. Se ci sono versioni tradotte della documentazione disponibile², e il locale è adeguatamente settato, queste saranno usate automaticamente da L_YX. L_YX cerca di tradurre le versioni come `LyXDir/doc/xx_DocName.lyx`, dove `xx` indica il linguaggio settato dalla variabile ambientale `$LANG`. Se non c'è nulla, sarà mostrato il linguaggio di default ovvero l'inglese. Notate che le versioni tradotte devono avere gli stessi nomi dei files come gli originali (`DocName` di cui sopra). Se ve la sentite di tradurre la documentazione (un buon modo per leggersi la documentazione originale BTW!), ci sono alcune cose che dovrete fare:

- Leggere `DocStyle.lyx`, la guida per scrivere documentazione su L_YX. Fare molta attenzione alla sezione traduzioni.
- Controllare la pagina web delle traduzioni The LyX Developer's Web Site <http://www.devel.lyx.org>. In questo modo potete trovare (se ce ne sono) i documenti tradotti e quelli da tradurre. Inoltre troverete un responsabile organizzatore (se c'è) che vi aiuterà nella vostra opera. Se non c'è un organizzatore fateci sapere se siete interessati.

Una volta che iniziate a tradurre ci sono alcune cose che dovete ricordare:

²Ad Aprile 1999, quasi tutti i documenti sono stati tradotti in tedesco. Il tutorial è stato tradotto in francese, polacco, portoghese e svedese, con altre traduzioni in corso. La libreria dei documenti tradotti cresce rapidamente.

- Unirvi al team di documentazione! Ci sono informazioni su come farlo in `Intro.lyx` (Help paese. Inventare una vostra propria terminologia potrebbe solo creare confusione al lettore. *(Attenzione! La tipografia da assuefazione!)*)
- Fate una copia del documento. Questo sarà la vostra copia di lavoro. Potrete inoltre usarla come vostra copia personale tradotta mettendola nella vostra directory `~/.lyx/doc/`.
- Qualche volta il documento originale (dal team L_YX) viene aggiornato. Usate lo strumento CVSweb reperibile presso il sito dello sviluppatore per vedere i cambiamenti avvenuti³. In questo modo potrete facilmente vedere le parti del documento che vanno aggiornate.
- Se per caso trovate un errore nel documento originale, correggetelo e comunicate al documentation team i cambiamenti! (non avete dimenticato di unirvi al documentation team vero?).

5.4 Supporto Tastiere Internazionali

Nota dell'editore: la seguente sezione è a cura di Ivan Schreter. Necessita di essere corretta in relazione al nuovo Documentation Style ed al fine di utilizzare le caratteristiche della versione 1.0. Il tutto va inoltre unito con le sezioni successive.-jw

5.4.1 Definire una propria Mappa della Tastiera: Formato File Keymap

Diamo uno sguardo al file di definizione della tastiera più da vicino. E' un file di testo ASCII che definisce

- key-to-key or key-to-string translations
- dead keys
- dead keys exceptions

Per definire una traduzione tasto a tasto o tasto a stringa, usare questo comando:

```
\kmap key outstring
```

dove `key` è il tasto che deve essere tradotto e `outstring` è la stringa che deve essere inserita nel documento. Per definire i tasti morti, usare:

```
\kmod key deadkey
```

³Alternativamente, potete procurarvi una copia dell'ultima versione del documento inglese che avete tradotto.

dove `key` è il tasto della tastiera e `deadkey` è il nome del tasto morto. Sono supportati i seguenti tasti morti (le scorciatoie sono tra parentesi):

<i>Name</i>	<i>Example</i>
acute (acu)	áéíóú
grave (gra)	àèìòù
macron (mac)	ō
tilde (til)	ñÑ
underbar (underb)	ǒ
cedilla (ced)	çÇ
underdot (underd)	ḡ
circumflex (circu)	âêîôû
circle (circl)	ĂăŮů
tie (tie)	\t oo
breve (bre)	ăǒ
caron (car)	č̈, š̈
hungarian umlaut (hug)	őű
umlaut (uml)	äöü
dot (dot)	žš

Dato che in molte tastiere internazionali ci sono delle eccezioni alle funzioni di alcuni tasti morti, potete definirli usando

```
\kxmod deadkey key outstring
```

Ad esempio, sulle tastiere slovacche se immettete “caron-o”, verrà scritto “circumflex-o”, così dovrete cambiare

```
\kxmod caron o "\^o"
```

per farlo funzionare correttamente. Inoltre, dovrete definire come eccezione i tasti morti su i e j , per rimuovere il punto prima di inserire un accento. Lo cambierò appena possibile in quanto non ho ancora avuto il tempo.

Circa i caratteri, backslash non è presente, quindi per inserirlo avete bisogno del doppio backslash. Inoltre virgolette e # hanno differenti significati. # segna i commenti, le virgolette iniziano e finiscono i comandi in stile L^AT_EX. Per immettere le virgolette dovete usare \", per #, usare \#.

Se create un file di descrizione della tastiera che va bene per la vostra lingua, vi prego di mandarmelo, in modo che io possa includerlo nella prossima distribuzione delle mappe di tastiera.

Molte parole chiave saranno presenti nelle future distribuzioni come:

- `\kinclude filename` include un altro file
- `\kprog program` define un programma esterno di traduzione della tastiera

Inoltre dovrebbe apparire nel file `lyxrc` di default (ad esempio un'opzione `\kinclude` per includere la tastiera di default).

5.5 Materiale sulle tastiere internazionali

Le prossime due sezioni descrivono la sintassi dei file `.kmap` e `.cdef` nel dettaglio. Queste sezioni dovrebbero aiutarvi a disegnare la vostra mappa di caratteri se quelle che avete non vi soddisfano.

5.5.1 Il File `.kmap`

Un file `.kmap` mappa le pressioni dei tasti sia di caratteri che di stringhe. Come suggerisce il nome, esso setta la mappa della tastiera. Il file `.kmap` contiene delle parole chiavi come `kmap`, `kmod`, `ksmod`, e `kcomb` qui di seguito descritte.

`kmap` Mappa il carattere rispetto a una stringa

```
\kmap char string
```

Questo mapperà *char* in *string*. Notate che in *string*, le due virgolette (") e il backslash (\) devono essere evitate con un precedente backslash (\).

Un esempio di `kmap` che serve affinché il simbolo / venga generato dal tasto & è:

```
\kmap & /
```

`kmod` Specifica un carattere accentato

```
\kmod char accent allowed
```

Questo farà sì che il carattere *char* sia *accentato sul carattere ammesso*. Questo è un meccanismo di tasto morto⁴.

Se premete *char* e poi un altro tasto non in *allowed* (*non permesso*) avrete un *char* seguito da un altro tasto, non abilitato. Ricordate che un Backspace cancella un tasto morto, così se premete *char* *Backspace*, il cursore non andrà una posizione indietro, bensì cancellerà l'effetto che il *char* dovrebbe avere sul tasto successivo.

Il seguente esempio specifica che il carattere ' dovrà essere un accento acuto, abilitato sui caratteri a, e, i, o, u, A, E, I, O, e U:

```
\kmod ' acute aeiouAEIOU
```

`ksmod` Specifica un'eccezione al carattere accentato

```
\kxmod accent char result
```

⁴Il termine tasto morto si riferisce ad un tasto che non produce un carattere da solo, ma soltanto quando è seguito da un altro tasto. Ad esempio, si può riprodurre un carattere tedesco con un umlaut come *ä*.

Questo definisce un'eccezione per l'*accent* sul *char*. L'*accent* deve essere stato assegnato ad un tasto con una precedente dichiarazione `\kmod` e *char* non deve appartenere al set degli accenti (*accent*) abilitati (*allowed*). Quando immettete la sequenza del carattere accentato (*accent char sequence*), il risultato (*result*) viene prodotto. Se non esiste una dichiarazione nel file `.kmap` e voi immettete l'*accent char*, avrete un'*accent_key char* dove *accent_key* è il primo argomento della dichiarazione `\kmod` declaration.

La seguente procedura di comandi fa in modo che venga generato äi immettendo acute-i ('i):

```
\kxmod acute i "\'{"i}"
```

`kcomb` Combina due caratteri accentati

```
\kcomb accent1 accent2 allowed
```

Questo effetto ha qualcosa di esoterico. Infatti vi permette di combinare l'effetto di *accent1* e *accent2* (in quell'ordine!) su caratteri abilitati (*allowed*). I tasti per *accent1* e *accent2* devono essere stati settati con un comando `\kmod` in un punto precedente del file.

Consideriamo questo esempio dal file `greek.kmap`:

```
\kmod ; acute aeioyvhAEIOYVH \kmod : umlaut iyIY \kcomb acute umlaut iyIY
```

Questo vi permette di premere `;;i` ed ottenere l'effetto di `\'{"i}`. Un backspace in questo caso cancella l'ultimo tasto morto, così se premete `;; Backspace i` avrete `\'{"i}`.

5.5.2 Il file `.cdef`

Dopo la mappatura fatta con `.kmap`, il file `.cdef` mappa le stringhe che i simboli generano rispetto ai caratteri del font corrente. La corrente distribuzione di `LYX` include almeno i files `iso8859-1.cdef` e `iso8859-2.cdef`.

In generale il file `.cdef` è una sequenza di dichiarazioni di questa forma:

```
char_index_in_set string
```

Ad esempio, per mappare `\'{"e}` sul corrispondente carattere nel set `iso-8859-1` (233), si usa la seguente dichiarazione

```
233 "\'{"e}"
```

con `\` e `"` estranei alla stringa. Notate che lo stesso carattere può essere applicato a più di una stringa. Nel file `iso-8859-7.cdef` avete

```
192 "\'{"\{"i}"
```

```
192 "\'{"\\'{"i}"
```

Se `LYX` non trova una mappatura per la stringa prodotta dalla pressione del tasto o da una sequenza con tasti morti, controllerà se assomiglia ad un carattere accentato e proverà a disegnare un accento sul carattere sullo schermo.

5.5.3 Tasti morti

C'è un altro modo per aggiungere il supporto a caratteri internazionali, ed è quello che usa la tecnica dei cosiddetti tasti morti. Un tasto morto lavora in combinazione con una lettera per produrre un carattere accentato. In questa sezione spieghiamo come si crea un semplice tasto morto per far vedere come funziona.

Supponiamo di avere bisogno di un carattere circonflesso, “^”. Si potrebbe associare il tasto `^` [a.k.a. `Shift-6`] al comando L_YX `accent-circumflex` nel vostro file `lyxrc`. Ora, se digitate il tasto `^` seguito da una lettera, la lettera avrà un accento circonflesso su di essa. Per esempio, la sequenza “`^e`” produce la lettera `e` con `^`. Se provate a digitare “`^t`”, tuttavia, L_YX segnalerà con un beep che l'operazione non è possibile dato che la `t` non ha mai accento. Premere `Spazio` dopo un `deadkey` produrrà un accento semplice. Attenzione a quest'ultimo punto! Se associate un tasto ad un `deadkey`, dovrete riassociare il carattere su quel tasto ad un altro tasto ancora.

Un altro modo comune di associare tasti morti è quello di usare `Meta-`, `Ctrl-`, e `Shift-` in combinazione con un accento come “`~`” o “`,`” o “`^`”. Un'altra tecnica consiste nell'uso di `xmodmap` e `xkeycaps` per settare il tasto speciale `Mode_Switch`. Il `Mode_Switch` si comporta in alcuni casi proprio come lo `Shift` e permette di associare tasti a caratteri accentati. Potete inoltre convertire tasti semplici in tasti morti associandoli ad oggetti del tipo `usldead_cedilla` e successivamente, associando questo tasto simbolico al corrispondente comando L_YX.⁵ Potete fare pressochè tutto con il tasto `Mode_Switch`: uno dei tasti `Ctrl-`, una funzione libera, ecc. Come per i comandi L_YX che producono accenti, consultate il *Manuale di Riferimento* per l'accento acuto. Troverete lì la lista completa.

5.5.4 Salvare la vostra configurazione di Linguaggio

Potete editare il vostro file `lyxrc` in modo che il linguaggio da voi preferito sia configurato e caricato automaticamente allo startup di L_YX. Questa sezione descrive i comandi `lyxrc` da specificare:

- `default`, `primary`, and `secondary` keymaps
- `character set`

Guardate il vostro file `lyxrc` per i vari esempi su come configurarlo. In particolare, potete settare una keymap così:

```
\kmap true
\kmap_primary german
\kmap_secondary american
```

⁵Nota da JOHN WEISS: Questo è esattamente quello che faccio nei miei files `~/lyx/lyxrc` e `~/xmodmap`. Ho il mio tasto `Scroll Lock` settato come `Mode_Shift` e un sacco di questi tasti simbolici “`usldead_*`” che consentono di ottenere cose del tipo `Scroll Lock-^` e `Scroll Lock-~`. Questo è il modo in cui genero i miei caratteri accentati.

e una font-mapping (.cdef file) in questo modo:

```
\screen_font_encoding iso8859-2
```


Capitolo 6

Installare Nuove Classi di Documento, Layouts e Templates

In questo capitolo si descrivono le procedure per creare ed installare nuovi layouts di LyX e files template, come anche si procederà ad illustrare la corretta installazione di nuove classi di documento L^AT_EX. Alcune definizioni: una classe di documento è un file L^AT_EX (che di solito hanno estensione in `.cls` o `.sty`) che descrive il formato di un documento come un articolo, report, pre stampa di giornale, ecc. e tutti i comandi di cui si ha bisogno per realizzare quel formato. Un file layout è un file LyX che corrisponde ad una classe di documento L^AT_EX e che dice a LyX come “disegnare” le cose sullo schermo in modo da rendere le cose il più possibile vicine alla pagina finale stampata. Più precisamente, un file layout descrive una “textclass” che è il costrutto interno che LyX usa per fare il rendering dello schermo. “Layout” e “textclass” possono a volta essere usate come sinonimi, ma è meglio riferirsi al file come il layout, e all’oggetto residente nella memoria di LyX come textclass. Un file template è semplicemente un documento LyX che contiene un set di righe predefinite per una data classe di documento che sono generalmente richiesti per quella classe. I templates sono in particolare utili per cose come i manoscritti di giornale che devono essere processati elettronicamente.

6.1 Installare un nuovo pacchetto L^AT_EX

Alcune installazioni possono non includere un pacchetto L^AT_EX che vorreste usare in LyX. Ad esempio, vorreste avere FoilT_EX, un pacchetto comune (e molto potente) per preparare slides o grafici per capiprogetto.¹Ecco di seguito tutti i

¹Vedere il manuale *Extended LyX Features* per ulteriori informazioni su FoilT_EX

passaggi necessari per avere il pacchetto e lanciarlo se si usa una recente versione di TeX^2 o altre distribuzioni basate su `web2c`.

1. Procurarsi il pacchetto da CTAN o simili.³
2. Leggere il file `texmf.cnf` (che di solito risiede nella directory `$TEXMF/web2c`, sebbene possiate lanciare `kpsewhich texmf.cnf` per localizzarlo). Questo descrive come aggiungere una directory locale `texmf`; seguire le istruzioni. Avete bisogno di inserire il nome della vostra directory locale `texmf` in `texmf.cnf`. Sotto Linux, `/usr/local` è una posizione logica dove installare software che non è presente nelle normali distribuzioni, così dovete usare `/usr/local/texmf`. Di solito dovete modificare solo due cose:
 - (a) Settare `TEXMFLOCAL` sulla directory che avete scelto; es.
`TEXMFLOCAL = /usr/local/texmf`
 - (b) Assicuratevi che `TEXMF` includa la variabile `TEXMFLOCAL` variable; es.
`TEXMF = {$HOMETEXMF, !!$TEXMFLOCAL, !!$TEXMFMAIN}`
3. Create la vostra directory locale `texmf` (es. `/usr/local/texmf`). Dovete seguire la struttura dell'esistente directory `texmf` (ad esempio, i pacchetti `latex` dovrebbero andare sotto `/usr/local/texmf/tex/latex/`).
4. Installare il pacchetto. Ad esempio, vogliamo scompattare l'archivio `FoilTeX` e creare `/usr/local/texmf/tex/latex/foiltex`. La directory `foiltex` contiene vari files.
5. Lanciare: `texhash`. Questo dovrebbe creare `/usr/local/texmf/ls-R` tra le altre.
6. Dall'interno di `LAX`, fate: `Options` \triangleright `Reconfigure`. Riavviate `LAX`.

Ora dovrete vedere il vostro nuovo pacchetto—ad esempio `slides` (`FoilTeX`)—sotto `Layout` \triangleright `Document`, campo `Class`. Notate che ci sono modi più semplici per installare pacchetti: potete aggiungere un link alla nuova directory del pacchetto nella directory di sistema TeX (`$TEXMF/tex/latex`, non dimenticate di lanciare dopo `texconfig`), o qualche volta basta semplicemente settare la variabile ambientale `$TEXINPUTS` per includere un nuovo pacchetto. Tuttavia, la procedura formale descritta in `texmf.cnf` è garantito che funzioni, così dovrete provare a seguirla fino a che se ne ha la possibilità (che potrebbe perdersi se non si ha accesso da superuser).

²Questa procedura è definitivamente valida per la versione v1.0 e tutte le versioni beta v0.9. Commenti sul funzionamento per le versioni v.04 sono graditi, anche se è comunque preferibile aggiornarsi alla versione v1.0.x.

³Vedere il manuale *Inventary of your TeX configuration* per i dettagli su cosa è CTAN e dove possono essere trovate le classi di documento supportate.

6.2 Layouts

Questa sezione descrive come scrivere ed installare i propri files di layout di LYX (anche detti *textclass*), utilizzando la *textclass* `article` come esempio. I files `.layout` descrivono quali stili di paragrafi sono disponibili per una data classe di documenti e come LYX dovrebbe mostrarli. Proveremo qui a fornire un'accurata descrizione del processo; tuttavia, ci sono così tanti tipi di documenti supportati dalle classi $\text{L}\text{A}\text{T}\text{E}\text{X}$ da renderci impossibile fornire una descrizione esaustiva dell'argomento e delle possibilità che si possono manifestare.

Quando programmate di scrivere un nuovo layout, è estremamente utile guardare prima i layout di esempio distribuiti con LYX . Se usate una classe di documento $\text{L}\text{A}\text{T}\text{E}\text{X}$ particolarmente bella che può essere di interesse anche per altre persone e che abbia un buon corrispondente layout in LYX , siete liberi di mandarcelo in modo da poterlo inserire nella distribuzione.

Tutte le righe di codice di questo paragrafo sono case-insensitive; ciò significa che `Style`, `style` e `StYle` sono lo stesso comando. I possibili valori sono scritti tra parentesi dopo il nome della caratteristica. Il valore di default se una caratteristica non è specificata nella descrizione della *textclass* è scritto in *corsivo*. Se l'argomento è un tipo di dati come "string" o "float", il valore di default è scritto come `float=default`.

6.2.1 Supportare nuove classi di documento

Ci sono tre situazioni che potete incontrare se volete supportare una nuova classe di documento $\text{L}\text{A}\text{T}\text{E}\text{X}$. I due tipi più semplici coinvolgono classe (`.cls`) e style (`.sty`) $\text{L}\text{A}\text{T}\text{E}\text{X} 2_{\epsilon}$. Se siete sfortunati potreste incappare nella terza situazione di supportare un file di stile $\text{L}\text{A}\text{T}\text{E}\text{X} 2.09$. Il processo di creazione di un file di layout per queste tre situazioni è simile; tuttavia un passo in più è richiesto per permettere a LYX di usare $\text{L}\text{A}\text{T}\text{E}\text{X} 2.09$.

6.2.2 File sty LaTeX2e

Se la vostra nuova classe di documento è sotto forma di style file, usato in aggiunta ad uno esistente, la classe di documento supportata, inizia copiando il layout della classe esistente nella vostra directory locale. A causa della mancanza di esempi assumeremo che il file style si chiami `myclass.sty` e sarà usato con `report.cls` con una classe standard.

```
cp report.layout ~/.lyx/layouts/myclass.layout
```

Poi editiamo `myclass.layout` e cambiamo la linea:

```
\DeclareLaTeXClass{report}
```

in modo da avere

```
\DeclareLaTeXClass[report, myclass.sty]{report (myclass)}
```

Poi aggiungiamo:

```
Preamble \usepackage{myclass}EndPreamble
```

presso l'inizio del file.

Lanciamo LyX e selezioniamo **Options**▷**Reconfigure**. Riavviamo LyX e proviamo a creare un nuovo documento. Dovreste vedere “report (myclass)” come una classe di documento opzione nel menu **Layout**▷**Document**. E’ probabile che alcuni dei comandi di sezionamento ed altri differiscano dal modo di lavorare delle classi base⁴, così potrete giocare con i settaggi per le differenti sezioni se vorrete. In seguito l’argomento sarà ripreso.

6.2.3 File cls L^AT_EX 2_ε

In questo caso, dovrete “vederla da soli” con il vostro layout. Raccomandiamo fortemente di copiare un file di layout esistente che usa una classe di testo L^AT_EX simile e modificarlo per quanto possibile. oppure si può almeno usare un file esistente come punto di partenza in modo da capire quali oggetti servano ai vostri scopi. Ancora, più avanti ci saranno altri dettagli.

6.2.4 File sty L^AT_EX 2.09

Se il file `.sty` è specifico per l’ormai defunto L^AT_EX 2.09 potete stabilirlo attraverso uno di questi due modi:

1. Un file di esempio o la documentazione nel file `sty` stabilisce l’uso con `\documentstyle{myclass}` (opposto a `\documentclass`).
2. Lanciare L^AT_EX su un file di esempio produrrà un banner con l’avvertimento che si sta entrando nel “compatibility mode”.

LyX tecnicamente supporta solo documenti L^AT_EX 2_ε. Ma non mollate ancora! Dove c’è volontà c’è un modo (oppure “adattarsi è impossibile, ma l’abbiamo comunque fatto.”). Il supporto esistente per ReV_TE_X 3.1, ad esempio, comprende il codice 2.09 da usare nei documenti formattati 2e. Potete riutilizzare l’adattatore ReV_TE_X per la vostra `myclass.sty` (se risulta essere di codice 2.09).

Bene, ecco cosa dovete fare:

```
cp LyXDir/tex/revtex.cls ~/tex/myclass.cls
```

oppure dove avete i vostri files locali T_EX. Ora editate `myclass.cls` e fate un trova-e-sostituisci tra “revtex” e “myclass”. E’ probabile che alcune delle dichiarazioni `\DeclareOption` nel file non sono rilevanti per il vostro file di stile, ma non ci interessano per il momento.

Il prossimo passo è quello di ripetere il processo di creazione del file di layout `myclass.layout` di prima; solo per questa volta usate:

⁴Report nell’esempio

```
\DeclareLaTeXClass{myclass}
```

e non dovrete aver bisogno di definizioni di tipo `Preamble`. Lanciate `LyX` e riconfigurate, riavviate `LyX` e con un po di fortuna avrete un sistema che ha bisogno solo di qualche accorgimento allo schermo e null'altro. Questo implica il manipolare dei valori usati da `LyX` per lo spazio, lo stile dei fonts e le altre cose definite nei files `.layout`.

Avrete inoltre bisogno di controllare che avete tutte le variabili d'ambiente da `myclass.sty` disponibili nel file `myclass.layout`. Se mancano alcune bisognerà inserirle. Copiate un altro stile che sembra simile e modificatelo. Il modo più semplice per farlo è con:

```
Style MissingStyle
  CopyStyle SimilarStyle
  # whatever is different you now redefine
End
```

Guardare in seguito per i dettagli circa quello che fa ogni elemento nel file `.layout`.

Dovrebbe essere abbastanza per mettervi sulla giusta strada. Fate riferimento alle successive sezioni per i dettagli sui contenuti attuali dei files `layout` e ricordate che se ci sono problemi potete chiedere aiuto alla mailing list di `LyX`. Siamo sempre lieti di avere nuove classi di documento.

6.3 Dichiarare una nuova textclass

Quando è finalmente giunta l'ora di sporcarvi le mani e creare un vostro personale file di layout, la seguente sezione descrive a cosa andrete incontro. Il nostro consiglio è di procedere piano, salvare e testare spesso, ascoltare musica soffusa e bere un paio di bevande per adulti che gradite; qualcuna in più se siete particolarmente disorientati. Non è molto difficile, eccetto per la moltitudine di opzioni che possono creare problemi se cercate di molte cose contemporaneamente. Andate a prendervi un'altra bibita, solo per precauzione.

Andiamo!

Le linee in un file di layout che iniziano con `#` sono commenti. C'è una eccezione a questa regola: tutti i layouts dovrebbero iniziare con queste linee:

```
#!/ Do not delete the line below; configure depends on this
# \DeclareLaTeXClass{article}
```

La seconda linea è usata quando configurate `LyX`. Il file di layout è letto dallo script `LATEX chkconfig.ltx`, in un modo speciale e tale che `#` venga ignorato. La prima linea è solo un commento `LATEX`, e la seconda contiene la dichiarazione della classe di testo. Se queste linee appaiono in un file chiamato `article.layout`, definiranno una `textclass` dal nome `article` (il nome del file di layout) che usa la classe di documento `LATEX article.cls` (Il default è quello

di usare lo stesso nome del layout). La stringa “article” che appare sopra è usata come descrizione della textclass nel menu Layout▷Document .

Assumiamo che scrivete la vostra textclass utilizzando la classe di documento `article.cls`, dove avete cambiato le sezioni di testa. Se lo mettete in un file `myarticle.layout`, le intestazioni di questo file dovrebbero essere:

```

% Do not delete the line below; configure depends on this
# \DeclareLaTeXClass[article]{article (with my own headings)}

```

Questi dichiarano una textclass `myarticle`, associata con la classe di documento \LaTeX `article.cls` e descritta come “article (con le vostre intestazioni)”. Se la vostra textclass dipende da molti pacchetti, potete dichiararla come:

```

% Do not delete the line below; configure depends on this
# \DeclareLaTeXClass[article,foo.sty]{article (with my own headings)}

```

Queste indicano che la vostra textclass usa il pacchetto `foo.sty`. Infine è anche possibile dichiarare classi per codice DocBook. Le dichiarazioni tipiche avranno questa forma:

```

% Do not delete the line below; configure depends on this
# \DeclareDocBookClass[article]{SGML (DocBook article)}

```

Notate che a queste dichiarazioni può essere anche dato come parametro opzionale che dichiara il nome della classe di documento (ma non una lista).

Quando la textclass è stata modificata secondo i vostri gusti, tutto quello che dovete fare è copiarla in `$LyXDir/layouts/` o in `$UserDir/layouts` e lanciare Options▷Reconfigure. Chiudete LyX e rilanciatelo; a questo punto la vostra nuova textclass dovrebbe essere disponibile insieme alle altre.

6.3.1 Parametri generali di textclass

Questi sono i parametri generali che descrivono la struttura dell’intero documento:

Columns [*1*, *2*] A seconda che la classe di default abbia una o due colonne. Può essere cambiato nel menu Layout▷Document. Questo settaggio (a volte si confonde con **Sides**) è importante: se le vostre textclass hanno due colonne di default, ma dimenticate di settarlo correttamente, l’opzione \LaTeX `twocolumn` non sarà mostrata quando selezionate **Two Columns** in Layout▷Document.

Sides [*1*, *2*] A seconda che la classe di default debba stampare su uno o entrambi i lati del foglio. Può essere cambiato nel menu Layout▷Document .

PageStyle [*plain*, *empty*, *headings*] La classe di default pagestyle. Può essere cambiata nel menu the Layout▷Document .

MaxCounter [*Counter_Chapter*, *Counter_Section*, *Counter_Subsection*, *Counter_Subsubsection*, *Counter_Paragraph*, *Counter_Subparagraph*, *Counter_EnumI*, *Counter_EnumII*, *Counter_EnumIII*, *Counter_EnumIV*] Il più alto livello di enumerazione automatica. Valori utili sono *Counter_Chapter* per un libro o un report e *Counter_Section* per un articolo. Se non c'è bisogno di enumerazione, ad esempio all'interno di una classe di lettere, **MaxCounter** dovrebbe essere settato su *Counter_EnumI*.

ClassOptions...End Questa sezione descrive varie opzioni globali supportate dalla classe di documento. Guardare la sezione .

ProvidesAmsmath [*0*, *1*] Se la classe già carica il pacchetto **amsmath**. Questo è il caso delle classi di documento **amsart** e **amsbook**.

ProvidesMakeidx [*0*, *1*] Se la classe fornisce la funzione del pacchetto **makeidx**. Questo è il caso delle classi di documento **amsart** e **amsbook**.

ProvidesUrl [*0*, *1*] Se la classe fornisce la funzione del pacchetto **url**. Questo è il caso della classe di documento **AASTeX**.

DefaultFont Questo è usato per descrivere il font di default del documento. Guardare la sezione 6.3.4 per una descrizione.

Preamble...EndPreamble Un set di macro definizione che saranno emesse all'inizio dei files **L^AT_EX**. Usatelo per definizioni globali.

Input Come implica il nome, questo comando permette di inserire un altro file di definizione di **textclass** per evitare una ripetizione dei comandi. Molti files di **textclass**, per esempio, mettono **stdclass.inc**, che contiene molti dei layouts di base.

NoStyle Questo comando elimina uno stile esistente. E' particolarmente utile quando si vuole sopprimere un layout definito in un input file.

6.3.2 Sezione ClassOption

La sezione **ClassOption** può contenere le seguenti righe:

FontSize [*string="10|11|12"*] Una lista delle grandezze dei font disponibili per il font principale del documento, separato da "|".

PageStyle [*string="empty|plain|headings|fancy"*] La lista degli stili di pagina disponibili, separati da "|".

Other [*string=""*] Alcune opzioni di classi di documento, separate da una virgola, che sarà aggiunta alla parte delle opzioni del comando **\documentclass**.

6.3.3 Layouts specifici di paragrafo

La descrizione di un layout description è di questo tipo⁵:

```
Style name
...
End
```

dove sono consentiti i seguenti comandi:

CopyStyle Questo è usato per copiare tutte le caratteristiche di un layout esistente in quello corrente.

LatexType [*Paragraph*, *Command*, *Environment*, *Item_Environment*, *List_Environment*] Come dovrebbe essere tradotto il layout in L^AT_EX. *Paragraph* non significa nulla in particolare. *Command* significa `\LatexName{...}` e *Environment* significa `\begin{LatexName}...\end{LatexName}`. *Item_Environment* è lo stesso di *Environment*, eccetto che per `\item` che è generato per ogni paragrafo di questo ambiente. *List_Environment* è lo stesso di *Item_Environment*, eccetto *LabelWidthString* che è passato come argomento per l'ambiente. *LabelWidthString* può essere definito nel menu Layout▷Paragraph.

InTitle [1, 0] Se il valore è 1, segna il layout come necessitante di un `\maketitle`.

LatexName Il nome del corrispondente materiale L^AT_EX. Può essere sia di ambiente che di comando.

LatexParam Il parametro opzionale per il relativo materiale *LatexName*. Questo parametro non può essere cambiato da L^AX.

Margin [*Static*, *Manual*, *Dynamic*, *First_Dynamic*, *Right_Address_Box*] Il tipo di margine che il layout ha sul lato sinistro. *Static* significa un margine statico. *Manual* significa che il margine sinistro dipende dalla stringa immessa nel menu Layout▷Paragraph. Questo è usato per intestazione enumerate automatiche. *Dynamic* significa che il margine dipende dalla grandezza dell'etichetta. E' ovvio che l'intestazione "5.4.3.2.1 Very long headline" deve avere un margine sinistro più ampio (ampio quanto "5.4.3.2.1") di "3.2 Very long headline", anche se altri word processors non sono in grado di farlo. *First_Dynamic* è s simile, ma solo la prima riga del paragrafo è dinamica, mentre le altre sono statiche. usato ad esempio per le descrizioni. *Right_Address_Box* significa che il margine è scelto in modo tale che la riga più lunga di questo paragrafo arriva fino al margine destro. Usato per scrivere indirizzi in alto a destra della pagina.

⁵Notate che questo può definire un nuovo layout o modificarne uno esistente.

`NextNoIndent` [`1, 0`] Se si vuole consentire al paragrafo successivo di staccare dal margine la prima riga, 0 consente di farlo, 1 no.

`ParIndent` [`string=""`] Il capoverso della prima linea del paragrafo. L'argomento è passato come una stringa. Ad esempio "MM" significa che il paragrafo rientra con una larghezza di "MM" nel font normale. Potete anche avere una larghezza negativa premettendo il valore "-". Questo al fine di ottenere lo stesso effetto per ogni font. Il `Parindent` sarà fisso per un certo layout. L'eccezione è lo Standard layout, dato che il rientro del paragrafo di layout standard può essere proibito con `NextNoIndent`. Inoltre, i paragrafi con Standard layout, usano il `Parindent` dell'ambiente, non il loro nativo. ad esempio, i paragrafi Standard dentro una numerazione non hanno il rientro.

`Parskip` [`float=0`] L_YX permette di scegliere sia il "rientra" che il "salta" per redigere un documento. Quando si sceglie "rientra", questo valore è completamente ignorato. Quando si sceglie "salta" l'indentazione di un layout "Paragrafo" in stile L_AT_EX è ignorato e tutti i paragrafi sono additionally separati da questo argomento `parskip`. Lo spazio è verticale è calcolato con `value*DefaultHeight()` dove `DefaultHeight()` è l'altezza di una riga con il font normale. In questo modo l'aspetto rimane lo stesso indipendentemente dai fonts.

`TopSep` [`float=0`] Lo spazio vericale con cui il primo di una catena di paragrafi con questo layout è separato dal precedente. Se il paragrafo precedente ha un altro layout, le separazioni non sono semplicemente ma è preso il massimo.

`BottomSep` [`float=0`] La stessa cosa di `TopSep` per l'ultimo paragrafo.

`Parsep` [`float=0`] Lo spazio verticale tra due paragrafi di questo layout.

`Itemsep` [`float=0`] Questo è uno spazio extra tra i paragrafi di un ambiente di layout. Se immettete altri layouts nello stesso ambiente, ognuno sarà separato con `Parsep`. Ma la totalità degli oggetti presenti nell'ambiente sono separati additionally con `Itemsep`.

`LeftMargin` [`string=""`] Se immettete layouts nell'ambiente, i margini sinistri non sono semplicemente aggiunti, ma aggiunti con un fattore $\frac{4}{depth+4}$. Notate che questo parametro è usato anche quando il bordo è definito come `Manual` o `Dynamic`. Poi è aggiunto al bordo manuale o dinamico. Questa stringa ha lo stesso significato di `ParIndent`.

`RightMargin` [`string=""`] Simile a `LeftMargin`.

`Labeltype` [`No_Label`, `Manual`, `Static`, `Top_Environment`, `Centered_Top_Environment`, `Sensitive`, `Counter_Chapter`, `Counter_Section`, `Counter_Subsection`, `Counter_Subsubsection`, `Counter_Paragraph`, `Counter_Subparagraph`, `Counter_EnumI`,

`Counter_EnumII`, `Counter_EnumIII`, `Counter_EnumIV`]

`Manual` significa che l’etichetta è la primissima parola (lettera). `Static` viene definito nel layout (vedere `LabelString`). `Top_Environment` e `Centered_Top_Environment` sono casi speciali di `Static`. L’etichetta sarà stampata sopra il paragrafo, ma solo in cima all’ambiente o in cima ad una catena di paragrafi con questo layout. L’uso è ad esempio il layout `Abstract` o il `Bibliography`. E’ questo anche il caso di etichette `Manual` con tipo `LATEX Enviroment`, per creare layouts per teoremi messi giù correttamente. `Sensitive` è un caso speciale per le etichette di testo “Figura” e “Tabella”. `Sensitive` significa che la stringa dell’etichetta dipende dal tipo di oggetto flottante. I tipi di etichetta `Counter_xxx` sdirebbero essere abbastanza chiare.

`Labelsep` [`string=""`] Lo spazio orizzontale tra l’etichetta e il corpo del testo. Usato solo per etichette che non sono sopra il corpo del testo.

`LabelBottomsep` [`float=0`] Lo spazio verticale tra l’etichetta e il corpo del testo. Usato solo per etichette che sono sopra il corpo del testo (`Top_Environment`, `Centered_Top_Environment`).

`LabelString` [`string=""`] La stringa usata per un’etichetta del tipo `Static`. Quando il bordo è manuale questa stringa è anche usata come suggerimento per `LabelWidthString` che può essere settato nel menu `Layout > Paragraph`. Quando il tipo label è `Counter_xxx`, questa stringa viene preposta al contatore.

`Align` [`block`, `left`, `right`, `center`] Allineamento del paragrafo.

`AlignPossible` [`block`, `left`, `right`, `center`] Sono *liste*, separate da una virgola, di possibili allineamenti. Alcuni stili `LATEX` proibiscono certi allineamenti, dato che questi non hanno senso. Ad esempio una numerazione allineata a destra o centrata non è possibile.

`Fill_Top` [`0,1`] Con questo parametro il valore `Fill` della lista “Spazi verticali sopra” del menu `Layout`⁶.

`Fill_Bottom` [`0,1`] Simile a `Fill_Top`.

`NeedProtect` [`0,1`] Qualora comandi fragili in questo layout necessitassero di essere protetti `\protetti`.

`Newline` [`0, 1`] Se le nuove linee sono tradotte in nuove linee `LATEX` (`\`) o no. La traduzione può essere evitata per facilitare l’editing di `LATEX` all’interno di `LyX`.

`FreeSpacing` [`0, 1`] Di solito `LyX` non vi permette di inserire più di uno spazio tra le parole, dato che lo spazio è considerato come una separazione tra

⁶Nota di Jean-Marc: Non sono sicuro che questo settaggio sia molto usato, e sarà probabilmente rimosso nelle versioni successive.

due parole, non un carattere o un simbolo a se stante. Questa è una cosa molto simpatica ma a volte annoiante, ad esempio quando si scrive il codice di un programma o codice L^AT_EX. Per queste ragioni, `FreeSpacing` può essere abilitato. Notate che LyX creerà bianchi protetti per i bianchi addizionali se in un modo diverso dal modo-L^AT_EX.

`KeepEmpty` [*0*, *1*] Di solito LyX non vi consente di lasciare un paragrafo vuoto, dato che porterebbe ad un output L^AT_EX vuoto. Ci sono alcuni casi in cui però questo è auspicabile: in un template di lettera i campi richiesti possono essere rappresentati da spazi vuoti, così da agevolare le persone; in alcune classi speciali, un layout potrebbe essere usato come un tipo di interruzione, che non contiene testo, ecc.

`Spacing` [*single*, *onehalf*, *double*, *other value*] Questo definisce la spaziatura di default del layout. Gli argomenti *single*, *onehalf* e *double* corrispondono rispettivamente ad un valore multiplo di 1, 1.25 e 1.667. Se specificate l'argomento *other*, dovrete poi fornire un argomento numerico che sarà l'attuale moltiplicatore. Notate che, contrariamente agli altri parametri, `Spacing` implica la generazione di uno specifico codice L^AT_EX, usando il pacchetto `setspace.sty`.

`Font` Il font usato *sia* per il corpo del testo *che* per l'etichetta. Vedere la sezione .Notare che definire questo font automaticamente definisce `LabelFont` allo stesso valore.

`TextFont` Il font usato per il corpo del testo. Vedere la sezione 6.3.4.

`LabelFont` Il font usato per le etichette. Vedere la sezione 6.3.4.

`Preamble...EndPreamble` Un set di macro definizione che saranno emesse all'inizio dei files L^AT_EX quando il layout è usato. Usatelo per definire le macro di cui si hanno bisogno per questo particolare layout.

6.3.4 Descrizione dei Font

Una descrizione di un font appare così:

```
Font or LabelFont
```

```
EndFont
```

e i seguenti comandi sono disponibili:

```
Family [Roman, Sans, Typewriter]
```

```
Series [Medium, Bold]
```

```
Shape [Up, Italic, SmallCaps]
```

```
Size [tiny, small, normal, large, larger, largest, huge, giant]
```

Color [*none*, black, white, red, green, blue, cyan, magenta, yellow]

Latex [*no_latex*, latex] *no_latex* significa che LyX tradurrà tutto in L^AT_EX.

Latex significa una stampata rossa e nessuna traduzione.

6.4 Creare Templates

I templates vengono creati come normali documenti. L'unica differenza è che i documenti normali contengono tutti i possibili settaggi, inclusi schemi di font e dimensioni della carta. Di solito l'utente non vuole che il template sovrascriva il suo default, per questo motivo il progettatore di template dovrebbe rimuovere i corrispondenti comandi come `\fontscheme` or `\papersize` dal file di template di LyX. Questo si può fare con un qualsiasi editor di testo come `vi` o `xedit`.

Mettete il template editato che avete creato in `$UserDir/templates/`, copiate quello che usate dalla directory globale dei template in `$LyXDir/templates/` nello stesso posto, e ridefinite la directory dei template in `lyxrc` usando il comando `\template_path`.

Capitolo 7

Un Tutorial sulla Stampante

7.1 Introduzione

Questo capitolo descrive come sia possibile fare il setup di una stampante sotto Unix, in modo da poterla usare in LyX. Se avete un sistema Linux, e avete bisogno di settare la vostra stampante, potete anche leggere il file `Printing-HOWTO` [spesso nella directory `/usr/doc/FAQ/HOWTO/` o `/usr/doc/HOWTO/` a seconda del sistema]. Inoltre dovrete installare il pacchetto Ghostscript dal vostro CD-ROM, se non lo avete ancora o mai fatto.

Alcune distribuzioni Linux attualmente hanno i cosiddetti “filtri magici” che settano automaticamente il vostro sistema e lo rendono pronto all’uso. Potete usare questi strumenti in alternativa o in aggiunta a quello che andremo a descrivere, anche se i risultati con questi strumenti saranno molto inferiori rispetto alle rispettive potenzialità. Se la vostra distribuzione ha un tool di configurazione anche in fase di installazione, potete certamente usarlo, ma dovrete comunque leggervi il `Printing-HOWTO` quando vorrete personalizzare il setup.

Così, se vi sentite avventurosi, o non avete altra scelta, continuate a leggere!

7.2 Il file `/etc/printcap`

7.2.1 Descrizione

Il file `/etc/printcap` è quello che permette di comunicare a Linux i dati della vostra stampante. Ogni entrata definisce una stampante, le da un nome con cui gli utenti possono chiamarla, e dice a Linux come gestire il relativo output. Potete ovviamente avere più “stampanti” nel vostro sistema anche se gli utenti accedono ad una stessa stampante fisica, però con differenti setup o funzioni abi-

litate.¹ Naturalmente, avrete bisogno di un'entrata `printcap` per ogni stampante fisica che avete.

Ecco un esempio `/etc/printcap`:

```
lp|Text on an Epson:\
:lp=/dev/lp1:\
:sd=/var/spool/lpd/lp:\
:lf=/var/spool/lpd/lp/errs:\
:mx#0:\
:sh:\
:sf:
draft|Postscript on an Epson Stylus Color IIs, 180dpi:\
:lp=/dev/lp1:\
:sd=/var/spool/lpd/draft:\
:lf=/var/spool/lpd/draft/errs:\
:if=/var/spool/lpd/draft/draft.filter:\
:mx#0:\
:sh:\
:sf:
ps|Postscript on an Epson Stylus Color IIs, 360dpi:\
:lp=/dev/lp1:\
:sd=/var/spool/lpd/ps:\
:lf=/var/spool/lpd/ps/errs:\
:if=/var/spool/lpd/ps/ps.filter:\
:mx#0:\
:sh:\
:sf:
```

Prima di avventurarci nella spiegazione di questi geroglifici, abbiamo un messaggio importante:

ATTENZIONE! ATTENZIONE! ATTENZIONE!

Usate solo una linea tra le entrate!!!

Assicuratevi che la linea fra le entrate sia **COMPLETAMENTE BIANCA!!!**

Anche se lasciate uno spazio bianco tra le entrate, possono accadere cose strane, ed il sistema non interpreterà correttamente il file `/etc/printcap`.

Date uno sguardo ad una delle entrate dell'esempio precedente. Ogni parte dell'entrata è seguita da un carattere ":". Il carattere "\" alla fine di ogni linea

¹Nota di JOHN WEISS: Io ho una Epson Stylus Color IIs. Quando c'è bisogno di cambiare le testine per usare il colore sulla Stylus IIs, faccio un setup per una stampante separata, che chiamo `color`. Ho anche creato una stampante `draft` che usa una risoluzione di stampa minore e meno caratteristiche di stampa. Quindi potete intuire la funzionalità di `printcap`.

dice a Linux che l'entrata continua alla riga successiva. Non mettete nulla dopo il “\”, nemmeno uno spazio. Il “\” dovrebbe seguire il “:” dell'ultima parte dell'entrata sulla linea che avete interrotto.

Se scegliete di interrompere un'entrata di `printcap` in linee multiple, assicuratevi di mettere un “:” all'inizio del *testo* della nuova linea, dopo ogni spazio. Ad esempio, potremmo aver scritto la prima entrata dell'esempio precedente come:

```
lp|Text:lp=/dev/lp1:sd=/var/spool/lpd/lp:lf=/var/spool/lpd/lp/errs:
```

Notate che c'è un solo “:” tra ognuna delle parti dell'entrata. Se volete dividere questa entrata in due linee:

```
lp|Text:lp=/dev/lp1:sd=/var/spool/lpd/lp:\
:lf=/var/spool/lpd/lp/errs:
```

Abbiamo interrotto la linea al punto “:” aggiungendo un “\”, e iniziando la nuova linea che degli spazi e poi un “:” prima di continuare. E' uso comune quello di far rientrare la linea successiva di un pochettino in questi casi.

Andremo ora a spiegare ogni parte di un'entrata di `/etc/printcap`. Notate che ci sono altre opzioni che potete aggiungere ad un'entrata. Noi ci occuperemo soltanto delle quattro basi qui, e vi preghiamo di riferirvi alle man page su `printcap` per le informazioni addizionali.

7.2.2 Il Nome della Stampante

La prima linea di un'entrata di `printcap` è il nome della stampante. Contiene il nome della stampante, seguito da un carattere “|”, seguito da un testo che descrive la stampante, e infine, naturalmente, il carattere “:”. Ad esempio:

```
draft|OurPrinter in Draft Mode:\
```

è una stampante chiamata “`draft`” con la descrizione “Nostra stampante in modalità bozza” [Notate che l' “\” alla fine della linea, indica che continueremo l'entrata alla riga successiva.] Il nome `draft` è quello che in genere passate a molte stampanti con l'opzione `-P` dei comandi. [Il programma `lpr` è come un comando.]

Di solito si usa chiamare la vostra stampante principale PostScript® come “`ps`” e chiamare la vostra stampante di testo “`lp`”.

7.2.3 Il Dispositivo di Stampa

La prossima parte del nostro esempio di entrate `printcap` identifica il dispositivo di stampa. in tutti gli esempi citati c'è una linea:

```
:lp=/dev/lp1:\
```

[Ancora, notate che “\” alla fine della linea, così come l’inizio con “:” con relativi spazi di rientro.]

Il tag “lp=” porta il nome del dispositivo di stampa. In questo caso è /dev/lp1, ovvero la porta parallela. Ci sono stampanti che si connettono tramite porta seriale. se ne avete una, dovrete usare /dev/ttyS0 per la prima porta seriale, /dev/ttyS1 per la successiva e così via. Se volete aggiungere altri comandi relativi alla stampante nelle entrate di `printcap`, come il baud rate della stampante, dovrete riferirvi alle man pages di `printcap`, che possono aiutarvi.

Molte stampanti, tuttavia, usano la porta parallela. In aggiunta, la porta stampante sulla *maggior parte* dei computer PC-compatibili sarà /dev/lp1. Se volete sapere perchè, guardatevi questa nota.² Il vostro sistema non sarà differente, tuttavia controllate sempre a cosa corrisponde la vostra porta parallela.

7.2.4 La Directory Spool e il File Log

Oltre a dare un nome alla vostra stampante e a dire a Linux quale dispositivo è, avete ancora bisogno di una directory per il print spooling. la sigla “sd=” identifica la directory di spool per la stampante. E’ inoltre una buona idea avere un log file per gli eventuali errori. la sigla “lf=” lo identifica.

Molti sistemi mettono tutte le code di stampa nella directory /var/spool/lpd; alcuni usano solo /var/spool. In ogni caso, entrata del `printcap` tradizionalmente ha la sua propria spool directory dichiarata dopo il nome della stampante. Dovrete essere “root” e creare queste directory da soli. Dovrete inoltre cambiare la proprietà di queste directory, cosicchè il loro gruppo sia il gruppo `daemon`. Potete farlo col comando, “`chgrp daemon dirname`”, dove *dirname* è il nome della directory di spool.

Ad esempio, guardiamo l’entrata per la stampante “ps” nell’esempio. Contiene le linee:

```
:sd=/var/spool/lpd/ps:\
:lf=/var/spool/lpd/ps/errs:\
```

Questo indica che la directory di spool è /var/spool/lpd/ps e che il log file è /var/spool/lpd/ps/errs. Dovete “creare” questo file con il comando “`touch errs`” nella directory di spool.

Similmente, l’entrata per la stampante “lp” contiene:

²Perchè è /dev/lp1 e non /dev/lp0 la prima porta stampante? Va rilevato che l’indirizzo I/O di ogni porta parallela è codificato nel kernel di linux. L’indirizzo 0x3BC è /dev/lp0; l’indirizzo 0x378 è /dev/lp1; e 0x278 è /dev/lp2. Dato che molti dei computer PC-compatibili assegna la prima porta parallela all’indirizzo I/O 0x378 di default, /dev/lp1 sarà quindi la porta parallela sulla maggior parte dei sistemi.

Se avete più di una porta parallela, la situazione cambia. Inoltre se alterate l’indirizzo I/O della porta per qualsiasi motivo, poi uno degli altri dispositivi lp potrebbe puntare sulla porta parallela. Su alcuni sistemi questo può creare molti problemi. in alcuni casi potete andare in DOS e controllare quale indirizzo è stato assegnato alla vostra porta. Dopodichè potete usare questo indirizzo per stabilire se la porta parallela è /dev/lp0, /dev/lp1, o /dev/lp2.


```
:sd=/var/spool/lpd/lp:\
:lf=/var/spool/lpd/lp/errs:\
```

Ancora, la spool directory è `/var/spool/lpd/lp` e il log file è `/var/spool/lpd/lp/errs`.

7.2.5 Il Filter File

A differenza di “lp”, che passa semplicemente il testo non formattato alla stampante, le stampanti “draft” e “ps” hanno bisogno di un filtro [ecco perchè c’è una riga addizionale]. La riga `printcap` per il programma filtro è “if=” [sta per `input filter`, significa l’input dello spool].

Ci sono generalmente molte righe per i programmi filtro, che permettono di specificare un filtro differente per diversi scopi. Potete selezionare i filtri come opzioni a `lpr`, ma non ci inoltriamo per questa strada.

Diamo un’occhiata alle righe per la stampante “ps”:

```
:if=/var/spool/lpd/ps/ps.filter:\
```

Il file “`ps.filter`” è uno script di shell che risiede nella directory di spool per la stampante “ps”. Potete anche specificare l’indirizzo e il nome del file del programma; tuttavia, chiamando un programma con uno script di shell vi permette di specificare le opzioni dalla riga di comando.

Tutti i filtri, sia programmi o script di shell, dovrebbero prendere l’input da `stdin` [il flusso di input standard] e mandare il loro output a `stdout` [il flusso standard di output, the standard output stream].

7.2.6 Entrate Addizionali

Ci sono tre cose addizionali che dovete mettere nel vostro `printcap`. Potete vederle alla fine dei tre esempi, e sono:

```
:mx#0:\
:sh:\
:sf:
```

Il “`mx#0`” significa “non limitare la grandezza del file che possiamo stampare.” Il “`sf`” previene che il form-feed sia mandato dopo la fine del documento. Infine, l’ “`sh`” previene che l’intestazione sia mandata all’inizio di ogni job. Probabilmente voi volete sempre aggiungere queste entrate al vostro `printcap`.

7.3 Creare un Filtro di Stampa

7.3.1 Stampare Files PostScript® Usando Ghostscript

Il modo più semplice per far capire alla vostra stampante il PostScript® è quello di usare il programma Ghostscript come filtro di stampa. Ci sono probabilmente altri modi, ma questo è quello più comune. Se, per qualche ragione

non potete o volete usare Ghostscript, dovrete provvedere da soli [guardatevi `Printing-HOWTO` per iniziare].

La prima cosa che dovete fare è installare Ghostscript. C'è almeno una versione in ogni distribuzione Linux. se non c'è la potete ottenere. The first thing you should do is install Ghostscript. There should be a version in your Linux da uno dall'archivio FTP di Linux `sunsite.unc.edu`. Una volta installato, lanciatelo usando

```
gs -h | more
```

dalla linea di comando. Questo farà una lista dei dispositivi di stampa disponibili con relativi formati. Qui di seguito riportiamo una lista che vi aiuta a capire meglio a quale dispositivo Ghostscript corrisponde ogni stampante.

- Stampanti Apple:

appledmp Apple dot matrix printers
iwhi Imagewriter in high-res mode
iwlo Imagewriter in low-res mode
iwlq Imagewriter LQ in 320 × 216 dpi mode

- Stampanti Canon:

bj10e BubbleJet BJ-10e
bj200 BubbleJet BJ-200
bjc600 Color BubbleJet BJC-600, BJC-4000 and BJC-70.³
bjc800 Color BubbleJet BJC-800
lbp8 LBP-II8 laser printer
lips3 LIPS-III laser printer in CaPSL mode

- Stampanti DEC:

la50 LA50
la70 LA70
la70t LA70 with low-res text enhancement
la75 LA75
la75plus LA75 Plus
ln03 LN03
lj250 LJ250 color printer
declj250 Alternate LJ250 driver0

³Buona anche per alcune Apple come la StyleWriter 2x00

- Stampanti Epson:
 - ap3250** ActionPrinter AP3250 [ed ogni stampante che usa il linguaggio ESC/P2]
 - escp2** Tutte le stampanti Epson che usano i comandi ESC/P2 [questo driver non è più disponibile, usare **ap3250**.]
 - epson** All 9- and 24-pin dot matrix printers [e compatibili]
 - eps9mid** All 9-pin dot matrix printers [e compatibili] in intermediate resolution
 - eps9hi** All 9-pin dot matrix printers [e compatibili] in triple resolution
 - epsonc** LQ-2550 color printer [buono anche per Fujitsu 1200, 2400, e3400]
 - st800** Stylus 800 inkjet
 - stcolor** the Stylus Color, Stylus Color Pro, Stylus Color II and IIs color inkjet printers
- Stampanti Hewlett Packard [HP]:
 - lp2563** 2563B line printer
 - cdeskjet** DeskJet 500C with 1 bpp color
 - cdjcolor** DeskJet 500C [and 540C] with 24 bpp color and Floyd-Steinberg dithering
 - cdjmono** DeskJet 500C [and 510C, 520C, 540C], monochrome only.
 - cdj500** same as **cdjcolor**
 - cdj550** DeskJet 550C and 560C
 - deskjet** DeskJet and DeskJet Plus
 - djet500** DeskJet 500
 - djet500c** Alternate DeskJet 500C driver [does not work with the 550C or the 560C]
 - dnj650c** DesignJet 650C
 - ljet2p** LaserJet IId, LaserJet IIp, and LaserJet III* with TIFF compression
 - ljet3** LaserJet III* with Delta Row compression
 - ljet3d** LaserJet IIID with duplex capability
 - ljet4** LaserJet 4 [default res=600dpi]
 - lj4dith** LaserJet 4 with Floyd-Steinberg dithering
 - ljetplus** LaserJet Plus
 - pj** PaintJet XL
 - pjetxl** Alternate PaintJet XL driver
 - pjxl** PaintJet XL color

pjxl300 PaintJet XL300 color [and 1200C]

paintjet Alternate PaintJet color driver

- Stampanti Tektronix:

t4693d2 4693d color printer with 2 bits/RGB component [=6 bpp]

t4693d4 4693d color printer with 4 bits/RGB component [=12 bpp]

t4693d8 4693d color printer with 8 bits/RGB component [=24 bpp]

tek4696 4695 and 4696 inkjet plotters

- Stampanti varie:

cp50 Mitsubishi CP50 color printer

ibmpro IBM 9-pin Proprinter

imagen Imagen ImPress printers

jetp3852 IBM Jetprinter ink-jet color printer (Model #3852)

m8510 C.Itoh M8510 printer

necp6 NEC P6/P6+/P60 printers at 360 dpi

nwp533 Sony Microsystems NWP533 laser printer [Sony only]

oce9050 OCE 9050 printer

oki182 Okidata MicroLine 182

okiibm Okidata MicroLine IBM-compatible printers

r4081 Ricoh 4081 laser printer

sj48 StarJet 48 inkjet printer

sparc SPARCprinter

xes Xerox XES printers (2700, 3700, 4045, etc.)

Se il pacchetto Ghostscript che avete nella vostra distribuzione non include la vostra stampante potete sempre scaricarvi il codice e compilarvelo da soli. Se la vostra stampante non è *direttamente* supportata da Ghostscript, provate a scegliere una compatibile.

Notate che alcune stampanti richiedono comandi speciali. Ad esempio il driver **stcolor** richiede che il file **stcolor.ps** sia letto prima dell'input file. Vi conviene leggere la documentazione di Ghostscript per i dettagli.

Poi, create il filtro. Sugeriamo di dargli il nome "*pname.filter*", dove "*pname*" è il nome che avete dato alla stampante nel file **printcap**. Sugeriamo inoltre di mettere lo script di filtro nella directory di spool della stampante. Il filtro dovrebbe contenere quanto segue:

```
#!/bin/sh
/usr/bin/gs -q -dSAFER -dNOPAUSE -sDEVICE=devname \
[-rResNum ExtraOptions] -sOutputFile=- [precmd.ps] -
```

dove *devname* è il nome del dispositivo Ghostscript; *ResNum* è la risoluzione che volete usare; *s* the resolution you want to use; *ExtraOptions* indica ogni opzione addizionale passata a Ghostscript; e “*precmd.ps*” è il nome del file che deve essere letto prima dell’input [come il file *stcolor.ps* per il driver *stcolor*]. La “\” spezza il comando in linee multiple.

Notate che gli esempi fatti assumono che *gs* sia nella directory */usr/bin*. Il vostro sistema potrebbe avere una configurazione diversa.

Differenti drivers Ghostscript trattano l’argomento “-r” diversamente. Alcuni cercano un numero singolo, ad esempio, “-r600”. Altri vogliono risoluzioni DPI separate per le direzioni x e y, es. “-r360x360”. Il driver della stampante può avere altre opzioni, quindi è meglio consultare la documentazione di Ghostscript per maggiori informazioni.

Dopo che avete creato il printer filter, usate “*chmod a+x*” sullo script file per renderlo eseguibile. [dovete essere root per farlo].

Ecco un esempio print file Ghostscript, usato da JOHN WEISS per la sua Epson Stylus Color IIs:

```
#!/bin/sh
/usr/bin/gs -q -dSAFER -sDEVICE=stcolor \
-r360x360 -sDithering=fsmono \
-dNOPAUSE -dnoWeave \
-sOutputFile=- stcolor.ps -
```

Questo filtro genera un output monocromatico a 360dpi su una Epson Stylus Color IIs.⁴

7.3.2 Se Già avete una Stampante PostScript®...

Molte stampanti laser moderne oggi hanno il supporto per PostScript®. Su alcune stampanti laser, PostScript® è attualmente il linguaggio nativo della stampante. Se avete una stampante, non avete bisogno del filtro e potete passare un file PostScript® direttamente alla stampante. Di contro, se volete stampare un file di testo, avrete comunque bisogno di formattarlo in qualche modo in PostScript®.

Siamo a conoscenza di tre programmi che traducono il testo ASCII in PostScript®: *nenscript*, *a2ps*, e *mpage*. *mpage* possono anche stampare pagine multiple su un singolo foglio di carta. Dovete quindi creare uno script di filtro usando uno di questi programmi per la stampante *lp*. [La stampante *lp* è tradizionalmente il nome della stampante che stampa file di testo ASCII.] Leggete il *Printing-HOWTO* per maggiori informazioni.

⁴Nota da JOHN WEISS: Questa è la mia stampante ps. Io uso un filtro simile sulla stampante *draft*, che ha l’opzione *-r180x180* invece di quello che avete visto. E per la mia stampante *color*, uso un valore differente dell’opzione *-sDithering*.

7.4 Il Demone Print Spool, lpd

Il programma che generalmente fa lo spooling e la stampa dei files è `lpd`. E' anche il programma che legge il file `/etc/printcap`. Da qualche parte nei vostri scripts di bootup,⁵ dovete eseguire `lpd` per iniziare lo spooling ad ogni avvio del sistema. Potete anche eseguire `lpd` come un job di background dalla linea di comando se loggate come root.

Una volta che `lpd` è in esecuzione sul vostro sistema, potete controllare il suo comportamento e quello del print spool usando il comando `lpc`. Ancora una volta, dovete essere root per pare ciò.

7.5 Assemblando il Tutto

Concluderemo questo mini-HOWTO con i passaggi che dovete seguire per avere un setup minimale. Quando avrete finito, avrete due stampanti, una chiamata “`lp`” per stampare testoASCII, e una seconda chiamata “`ps`” per i files PostScript®. Il file filtro per la stampante “`ps`” [incluso sotto] funziona su ogni stampante compatibile Epson [ovvero pressochè la totalità delle stampanti dot-matrix attualmente in commercio], usando la risoluzione di stampa di default.

1. Entrate come “root” o usate “`su root`”.
2. Eseguite i seguenti comandi:
 - (a) `cd /var/spool`
 - (b) `mkdir lpd`
[se non esiste già]
 - (c) `chgrp lp lpd`
[potreste dover usare “`daemon`” al posto di “`lp`”]
 - (d) `cd lpd`
 - (e) `mkdir lp ps`
[fatelo solo se queste directory non esistono già]
 - (f) `touch lp/errs ps/errs`
 - (g) `chgrp daemon lp ps lp/errs ps/errs`
3. Ora editate il file `/etc/printcap`
 - (a) Se il file esiste, quindi...
 - i. ... rinominate la vecchia versione in qualcosa tipo “`/etc/printcap.orig`”
 - ii. ... oppure, commentate tutto il file mettendo “`#`” all’inizio di ogni riga.

⁵Alcune macchine tengono i loro scripts di bootup nella directory `/etc/rc.d`. Altri usano `/etc/init.d`. Dovete capire da voi che sistema è usato.

- (b) Mettete quanto segue in `/etc/printcap`:

```
lp|Text printer:\
:lp=/dev/lp1:\
:sd=/var/spool/lpd/lp:\
:lf=/var/spool/lpd/lp/errs:\
:mx#0:sh:sf:
ps|Postscript printer:\
:lp=/dev/lp1:\
:sd=/var/spool/lpd/ps:\
:lf=/var/spool/lpd/ps/errs:\
:if=/var/spool/lpd/ps/ps.filter:\
:mx#0:sh:sf:
```

- (c) ASSICURATEVI CHE:

- i. le linee bianche tra due entrate non abbiano nulla al loro interno, nemmeno spazi bianchi.
- ii. niente segue il carattere “\” alla fine di ogni linea.

4. Poi, create lo script filtro per la stampante “ps”.

- (a) Mettete quanto segue nel file `/var/spool/lpd/ps/ps.filter`:

```
#!/bin/sh
/usr/bin/gs -q -dSAFER -dNOPAUSE \
-sDEVICE=epson -sOutputFile=- -
```

[Se non avete una stampantedot-matrix compatibile Epson, rimpiazzate “epson” con l’appropriato nome del dispositivo Ghostscript. Inoltre, se il programma `gs` non è in `/usr/bin`, mettete l’indirizzo corretto.]

- (b) Eseguite questi comandi:

- i. `chmod a+x /var/spool/lpd/ps/ps.filter`
- ii. `chgrp daemon /var/spool/lpd/ps/ps.filter`

5. Infine, lanciate il demone della stampante, `lpd`:

- (a) Se non è già in uso, usate “`lpd &`” per avviarlo.
- (b) Se è già in esecuzione, eseguite “`lpc`”. Poi, da dentro `lpc`, date un “`restart all`” e un “`up all`” per attivare i nuovi spools di stampa. [Il comando “`quit`” abbandona `lpc`].
- (c) Assicuratevi di aggiungere il comando appropriato ai vostri scripts di bootup in modo che `lpd` sia eseguito ad ogni avvio del sistema.

6. Potete ora uscire da `root` e testare le vostre nuove stampanti per assicurarvi che funzionino. Date il comando “`lpr -P1p`” seguito dal nome di un piccolo file di testo ASCII per verificare che la stampante “`lp`” funzioni. Poi usate “`lpr -Pps`” seguito dal nome di un piccolo file PostScript® per testare la stampante “`ps`”.

Un avvertimento: quando testate “ps”, se qualcosa va storto, il sistema potrebbe passare dall’immondizia alla stampante. Se avviene questo, spegnete la stampante ed usate i comandi `lpq` e `lprm` per killare il print job. Dopo aver killato il print job, potete riaccendere la stampante.

Se nessuna di queste stampanti funziona, controllate di nuovo tutto, consultate le pagine `man` per `printcap` e `lpd`, e leggete il `Printing-HOWTO` per aiuto.

Se tutto funziona, congratulazioni!

Capitolo 8

Il materiale esterno

8.1 Background

Una caratteristica spesso richiesta dagli utenti LyX è quella di interfacciare LyX con Xfig, Dia, o altre applicazioni specializzate nella produzioni di un certo tipo di diagrammi, figure, schemi e simili cose che possono essere rilevanti per il vostro documento. Prima in LyX era solo possibile includere immagini fisse, statiche, noiose con l'insieme figure, ma ci sono numerose limitazioni relative a questo approccio:

- Se volete cambiare la figura, dovete richiamare manualmente un programma esterno
- LyX non notifica i cambiamenti ai riferimenti dei files, così quello che è visualizzato sullo schermo può rapidamente diventare obsoleto, e questo è aggravato da una media di aggiornamenti dello schermo
- L'insieme delle figure supporta solo materiale PostScript
- L'insieme delle figure non fornisce nessun meccanismo per copiare con differenti formati esportati come DocBook, HTML, o raw Ascii

L'insieme del materiale esterno prova a risolvere tutti questi problemi¹. Effettua ciò offrendo un metodo generale per interfacciare LyX alle applicazioni esterne. Invece di introdurre una lunga lista di insiemi ritagliati per ogni specifica applicazione, abbiamo scelto di sacrificare la visualizzazione in line del materiale incluso al fine di fornire uno strumento generale per affrontare una grande quantità di applicazioni. Il risultato è l'insieme esterno. Un insieme esterno presenta se stesso nel documento semplicemente come un bottone, ma non lasciatevi ingannare. Quando ci cliccate sopra, apparirà una finestra di

¹Anche se l'insieme delle figure non può risolvere tutti i problemi, è ancora valutabile perchè fornisce un'anteprima in linea della figura, e supporta trasformazioni geometriche avanzate con una semplice interfaccia utente.

dialogo che vi permetterà di scegliere esattamente quale materiale includere, è più avanti imparerete che in realtà è un potente meccanismo che può risolvere molti problemi ed anche di più.

8.2 Come funziona?

L'insieme esterno è basato sul concetto di *template*. Un template è una specificazione di come L_YX dovrebbe interfacciarsi con un certo tipo di materiale. In genere L_YX viene distribuito con una serie di template predefiniti per figure XFig, diagrammi Dia, varie immagini in formato raster, gnuplot, ed oltre. Potete controllare la lista attuale attraverso il comando `Insert▷Insert external material`. Inoltre, è possibile adattare i vostri personali template per supportare uno specifico tipo di materiale. Più avanti descriveremo più in dettaglio gli elementi coinvolti, e si spera che possiate creare dei template che poi ci manderete per includerli nelle successive versioni di L_YX.

Un'altra idea di base dell'insieme esterno è di distinguere tra il file originale che serve come base per il materiale finale e il file prodotto che è incluso nel vostro documento esportato o stampato. Per esempio, consideriamo il caso di una figura prodotta con XFig. L'applicazione XFig lavora su un file originale con estensione `.fig`. All'interno di XFig create e modificate la vostra figura, e quando avete finito salvate il file `fig`. Quando vorrete includere la figura nel vostro documento, basta richiamare `transfig` al fine di creare un file PostScript che può essere prontamente incluso nel vostro file L^AT_EX. In questo caso, il file `.fig` è il file originale, il file PostScript è quello prodotto.

Questa distinzione è importante al fine di effettuare l'aggiornamento del materiale mentre si è in fase di scrittura del documento. Inoltre, ci fornisce quella flessibilità che è necessaria per supportare molteplici formati di export. Ad esempio, nel caso di un file Ascii, non è esattamente una buona idea quella di includere la figura come un raw PostScript; dovrete invece preferire l'includere un riferimento alla figura oppure provare a richiamare qualche convertitore grafico da Ascii per rendere il risultato finale accettabile. L'insieme materiale esterno vi permette di farlo, perchè è parametrizzato su differenti formati di export supportati da L_YX.

Oltre a supportare la produzione di differenti prodotti in relazione al formato esportato, l'insieme esterno supporta una stretta integrazione con le applicazioni di editing e di visualizzazione. Nel caso di una figura XFig, potete richiamare `xfig` sul file originale con un singolo clic all'interno di L_YX oltre che avere un'anteprima del file PostScript prodotto con `ghostview` sempre con un semplice clic. Non c'è più bisogno di smanettare con la riga di comando o con il browser per trovare e manipolare i files originali o quelli prodotti. In questo modo potrete finalmente utilizzare a pieno molte differenti applicazioni che risultano rilevanti nella scrittura di un documento, al fine di essere maggiormente produttivi.

Quindi, in definitiva, L_YX ha informazioni circa un numero di programmi differenti da usare dietro le quinte per realizzare tutto questo marchingegno. Questa informazione, infatti, è esattamente ciò che è contenuto nei templates.

Per ogni template, c'è associata una lista di linee di comando usate per irrichiamare le applicazioni, convertire il file originale nel file prodotto ed altro ancora. Questo meccanismo permette all'utente avanzato di estendere le capacità di LyX senza smanettare col codice sorgente. Questo richiede un imponente lavoro per definire tutti i differenti comandi e flags, ma fortunatamente, il team di LyX ha già fatto questo lavoro per voi!

Ma prima che gli alberi diventino grattacieli, dobbiamo ammettere che abbiamo fatto alcuni piccole scorciatoie. Siccome il set esterno può produrre molti tipi di files prodotti per i relativi formati di export, ci si potrebbe anche aspettare che sia possibile vedere l'anteprima di ogni prodotto. Ma, ancora una volta, il team di LyX ha deciso per una linea che mantenesse l'interfaccia utente semplice. Per evitare di mettere un pulsante per ogni formato di file esportato, si è deciso di introdurre il concetto di formato primario di file ed avere solo un bottone. Quando si preme View result nella finestra dell'insieme esterno, verrà visualizzata un'immagine del file prodotto nel formato di file primario. E questo è specificato dalla vostra classe di documento. Per la maggior parte delle classi di documento il formato primario di file è L^AT_EX, ma per la classe di documento DocBook, il formato di file primario è DocBook. Così quando si visualizza il file prodotto, ricordatevi che sarà solo un'anteprima di quello che sarà il risultato principale. Se volete vedere come lavorano gli altri formati, dovete esportarli e fare l'anteprima manualmente.

8.3 La finestra di dialogo Materiale Esterno

Inserite un insieme esterno dal menu Insert. Quando fate questo, un bottone viene inserito nel vostro documento e viene mostrata la finestra di dialogo dell'insieme materiale esterno. Questa finestra vi permette di descrivere esattamente quale materiale deve essere incluso. Inoltre, fornisce un accesso alle applicazioni esterne per vedere, editare e produrre il materiale usato nel file risultante.

In cima a questa finestra, c'è una lista a cascata dove è possibile scegliere quale template l'insieme deve usare. Immediatamente sotto troviamo un'area di testo con una descrizione del template scelto che dovrebbe aiutarvi ad usarlo. Più spesso, fornisce oltre ad una descrizione del template anche alcune indicazioni su come parametrizzarne l'uso. Ancora più in basso si trova un campo di inserimento di nome di file con un bottone browse che permette di scegliere quale file deve essere incluso. Questo campo specifica il file originale. Siccome il file prodotto è generato automaticamente quando se ne ha bisogno, non c'è motivo per consentirne l'accesso nell'interfaccia utente.

In basso alla finestra di dialogo troverete una casella di immissione chiamata parameters. Questo box è generalmente usato per parametrizzare uno specifico template. L'uso specifico dovrebbe essere indicato e chiarito nella casella di definizione e aiuto associata al template, ma in genere permette tipicamente di definire le variazioni su come il file prodotto dovrebbe essere generato.

Nella parte destra della finestra ci sono tre bottoni: Edit file, View result e Update result. Questi rispettivamente permettono di editare il vostro file origi-

nale con l'applicazione di editing opportuna, vedere il file prodotto come incluso nel formato primario di documento, ed infine forzare un aggiornamento del materiale risultante nel formato primario. Normalmente il bottone **Update result** sarà disabilitato perchè molti template sono configurati per aggiornare automaticamente il file prodotto quando ce ne è bisogno. In questi casi non c'è bisogno di forzare la produzione di un nuovo file prodotto. Tuttavia alcuni templates sono configurati per non produrre automaticamente il prodotto residuale, in quanto il costo di produrre un file prodotto potrebbe essere così grande da renderne impossibile l'attuazione. Questi insiemi sono conosciuti come insiemi *manuali* esterni. In questi casi potete usare il bottone per forzare la produzione del file prodotto proprio quando ne avete bisogno, controllando così il lavoro che viene fatto. Infatti, è sotto la *vostra* responsabilità fare questo per mantenere i files prodotti costantemente aggiornati: prima di stampare, prima di esportare, prima di visualizzare, ecc. Forse, in futuro, sarà possibile che LyX vi aiuti in questo compito. Sarebbe infatti simpatico poter usare un comando **Edit>Update all external inset** per aggiornare tutti gli insiemi esterni che usano un template manuale. Ma vi anticipiamo che potrebbe richiedere molto tempo per l'implementazione.

In basso alla finestra di dialogo troverete i soliti bottoni OK e Cancel. L'unica cosa utile da ricordare circa questi è che ogni cambiamento nel template, nome del file o parametri sono attualmente applicati quando si premono i bottoni **Edit file**, **View result** e **Update result**. Questo implica che dopo aver usato uno di questi, potrete solo annullare i cambiamenti che sono stati effettuati attraverso quei bottoni premendo cancel. Fortunatamente, potete usare la funzione undo generale in LyX per ritornare allo stato precedente.

8.4 Esempi

In questa sezione, dovremmo includere alcuni esempi circa l'uso dell'insieme del materiale esterno. Questi esempi potrebbero includere:

- Immagini raster esterne
- figure esterne XFig
- Diagrammi a scacchi
- Campioni di suoni
- Uso dei makefiles
- Templates LyX esterni ricorsivi

8.5 Il file di configurazione external template

E' relativamente semplice aggiungere definizioni di template esterni a LyX. Tuttavia state attenti poichè fare queste cose in maniera approssimativa *potrebbe*

probabilmente creare un buco di sicurezza facilmente sfruttabile. Così prima di procedere, leggete il successivo capitolo sulla sicurezza.

Dopo aver detto queste cose, vi incoraggiamo a farci pervenire ogni template interessante che create.

I templates sono definiti nel file `lib/external_template`. Forse, prima o poi qualcuno documenterà i contenuti dei templates e la sintassi usata per definirne di propri.

8.6 Il meccanismo di sostituzione

Quando l'insieme del materiale esterno richiama un programma esterno, è fatto sulla base di un comando definito nel file di configurazione del template. Questi comandi possono contenere varie macro che vengono espanse prima dell'esecuzione. L'esecuzione avviene sempre nella directory del documento contenuto.

Inoltre, ogni volta che viene visualizzato un insieme esterno, il nome sarà prodotto dal meccanismo di sostituzione.

Le macro disponibili sono le seguenti:

\$\$FName Il nome del file specificato nella finestra dell'insieme esterno.

\$\$Basename Il nome del file senza estensione.

\$\$Tempname Un nome e un indirizzo completo di un file temporaneo che sarà automaticamente cancellato quando si chiude il documento contenuto, oppure quando è cancellato l'insieme esterno.

\$\$Contents("filename.ext") Questa macro espanderà il contenuto del file col nome `filename.ext`.

\$\$Sysdir Questa macro espanderà all'indirizzo assoluto della directory di sistema. E' tipicamente usato per puntare i vari script di aiuto che sono inseriti in L_X .

In aggiunta a questi gli strumenti espanderanno le variabili d'ambiente generali con una sintassi del tipo `$$PATH`.

8.7 Discussione sulla sicurezza

L'insieme del materiale esterno si interfaccia con molti programmi esterni e lo fa automaticamente, per questo vanno considerate le implicazioni di sicurezza di questo. In particolare, siccome avete l'opzione di includere dei vostri nomi dei file e/o stringhe di parametri, e questi vengono aperti in un comando, sembrerebbe che sia possibile creare un documento maligno che può eseguire comandi arbitrari quando un utente vede o stampa un documento. Questo lo vogliamo definitivamente evitare.

Tuttavia, dato che i comandi dei programmi esterni sono specificati solo nel file di configurazione del template, non ci sono problemi di sicurezza se LyX è configurato bene solo con templates sicuri. E' così perché i programmi esterni sono richiamati con la chiamata di sistema `execvp` piuttosto che con `system`, quindi non è possibile eseguire comandi arbitrari dal file o dalla sezione dei parametri attraverso la shell.

Tutto ciò implica che siete limitati nelle stringhe di comando che potete usare nei template dei materiali esterni. In particolare canali e redirection non sono subito disponibili. Ciò dovrebbe essere così se LyX dovesse rimanere sicuro. Se volete usare alcune funzioni della shell, dovrete scrivere uno script sicuro per farlo in maniera controllata, dopodiché richiamare lo script dalla linea di comando. Nella directory `lib/scripts` dell'installazione di LyX, potrete trovare uno script sicuro preconfezionato `general_command_wrapper.py` che supporta la redirection di input e output. Questo può essere un esempio di come scrivere script di template sicuri. Per un esempio più avanzato che usa `fork` e simili, date uno sguardo allo script di conversione `pic2ascii.py`.

E' possibile progettare un template che interagisce direttamente con la shell, ma dato che questo potrebbe permettere ad un utente con cattive intenzioni di eseguire comandi arbitrari scrivendo intelligenti filename e/o parametri, raccomandiamo in generale di usare solo script sicuri che lavorano solo con le chiamate di sistema `execvp` in maniera controllata. Naturalmente se si lavora in un ambiente controllato potrebbe essere allettante ritornare ad usare gli script di shell ordinari. Se lo fate state attenti poiché *andrete a creare* un buco di sicurezza facilmente sfruttabile nel vostro sistema. Naturalmente è ragionevole che questi templates non sicuri non verranno mai inclusi nella distribuzione LyX standard, sebbene incoraggiamo le persone a sottoporre nuovi templates nella tradizione open source. Ma LyX come rilasciato nei canali ufficiali di distribuzioni non avrà mai templates non sicuri.

L'insieme del materiale esterno fornisce molta potenza, perciò dovete stare attenti a non introdurre elementi di azzardo sulla sicurezza a questa potenza. Un lieve errore in una singola riga di un semplice script può creare grandi problemi di sicurezza. Così, se non capite completamente questo tema, vi consigliamo di consultare un professionista della sicurezza oppure il team di sviluppo di LyX se avete domande sulle implicazioni di sicurezza circa uno script. E, ovviamente, fate questo prima di usare lo script in un ambiente non sicuro.

8.8 Il futuro dell'insieme esterno

La corrente implementazione dell'insieme esterno è un costrutto potente e stabile che fornisce un accesso diretto alle parti più interne di LyX, ma come ogni caratteristica di LyX, dovrebbe essere sempre considerata un work-in-progress. Quando e se qualcuno ha il tempo di continuare a lavorarci su, ecco alcune direzioni generali che possono essere seguite:

- Supportare l'anteprima in linea in vari formati, piuttosto che il bottone di testo a cui si è obbligati attualmente.

- Supportare l'editing in linea attraverso l'uso di OpenParts o altri protocolli rilevanti.
- Estendere l'informazioni dinamica per avere campi di parametri opzionali per i comandi di conversione in tutti i formati di export, ed avere campi di parametri opzionali per ciò che è prodotto all'interno di tutti i differenti formati esportati. Al momento siamo costretti ad usare una sola stringa di parametri che è multiplata tra queste numerosi applicazioni. Inoltre un cambiamento come questo ci permetterebbe di liberarci delle strane restrizioni del formato principale.
- Estendere l'area di lavoro per fornire delle opzioni di personalizzazione più intelligenti in aggiunta alla piuttosto semplicistica stringa di parametri. Con un linguaggio di script adatto, sarà possibile implementare versioni user friendly di molti insiemi di personalizzazione che supportano una grande quantità di formati, pacchetti \LaTeX , editors, ecc.