

Il formato PDF archiviabile con gli aggiornamenti di T_EX Live 2016

Claudio Beccari

Sommario

La distribuzione T_EX Live 2016 ha portato importanti novità per quel che riguarda la produzione di documenti PDF archiviabili a lungo termine. La novità maggiore è che ora i pacchetti disponibili permettono di usare anche X_YL^AT_EX e LuaL^AT_EX oltre che PDFL^AT_EX. Con X_YL^AT_EX bisogna fare un poco di attenzione ma le difficoltà che questo programma mostra nella produzione di documenti archiviabili si possono superare.

L'archiviabilità a lungo termine è importante in ogni settore professionale e disciplinare; le discipline umanistiche, specialmente quelle legate alle lingue, hanno avuto finora dei grossi problemi con gli alfabeti diversi da quello latino; ora sembrano superati specialmente se ci si affida a LuaL^AT_EX che permette di gestire agilmente tutti i tipi di font OpenType.

Abstract

The T_EX Live 2016 distribution contains important advances for what concerns the production of long term archivable PDF documents. The most important new functionality consists in the fact that now it is possible to use X_YL^AT_EX and LuaL^AT_EX, not only PDFL^AT_EX. With X_YL^AT_EX it is necessary to pay some special care, but these program features may be overcome.

Long term archivability is very important in any professional or research area and in any discipline; up to now humanities, especially those dealing with ancient languages that are written with alphabets different from Latin, had some real problems that now appear to be overcome if LuaL^AT_EX is used, thanks to the fact that it manages any kind of OpenType fonts.

1 Introduzione

La produzione di documenti PDF archiviabili a tempo indefinito interessa ogni attività professionale e accademica in ogni possibile settore disciplinare. Si pensi soltanto alla grande quantità di documenti legali che si producono durante i vari processi; ai progetti edili con particolare attenzione a quelli che hanno una rilevanza giuridica; alle certificazioni di ogni genere; a testi oggetto di archiviazione per l'uso libero da parte di ricercatori di ogni disciplina; all'archiviazione elettronica delle tesi di laurea e di dottorato. Si potrebbe continuare

a lungo l'elenco, ma è evidente che non esiste un solo settore in cui l'archiviazione a lungo termine non sia una necessità assoluta.

Fino a poco tempo fa il sistema T_EX disponeva di poche risorse; esisteva il pacchetto `pdfx` del 2007, ma per farlo funzionare era necessario metterci le mani dentro, come ho mostrato in un precedente articolo (BECCARI, 2009). Pur tuttavia qualcosa si riusciva a fare solo con la compilazione mediante il programma di tipocomposizione `pdflatex`; purtroppo il risultato conforme alla norma PDF/A-1b non era sempre positivo.

Esisteva anche la via attraverso il programma `ghostscript` che provvedeva alla trasformazione di un file PDF “normale” in uno conforme alle norme ISO 19005:2005-1 ma a spese di molti compromessi e con probabilità di riuscita incerte.

SCARSO (2010) ha poi presentato su `ArsTEXnica` i suoi risultati ottenuti con il programma di tipocomposizione `ConTEXt`; riusciva a produrre i file PDF nella versione *tagged PDF*, che sono la base per le norme più restrittive PDF/A-1a; ma anche con `ConTEXt` il risultato era incerto. Forse oggi `ConTEXt` offre maggiori possibilità, ma il suo uso non è così vasto come quello dei programmi basati sul mark up L^AT_EX.

Oggi il pacchetto `pdfx` rinnovato ed esteso permette di comporre documenti PDF/X conformi alle norme per la stampa in quadricromia sottrattiva secondo una moltitudine di successive norme che continuano ad aggiornarsi anno dopo anno. Similmente quel pacchetto permette di comporre documenti conformi agli standard PDF/A di diversi livelli secondo i successivi aggiornamenti della norma.

La documentazione del nuovo pacchetto `pdfx` (RADHAKRISHNAN *et al.*, 2016) descrive tutti gli standard che grazie a esso sarebbe possibile creare (al verificarsi di altre condizioni nel seguito) e mette in evidenza anche quegli standard che il pacchetto potrebbe aiutare a comporre quando sarà possibile usarlo con i programmi di tipocomposizione, basati sul mark up L^AT_EX, capaci di produrre file di tipo *tagged PDF*.

Il pacchetto `ghostscript` riesce ancora a fare il lavoro che faceva prima ma la documentazione della versione 9.16 dovrebbe venire aggiornata, perché le istruzioni sono lacunose.

Nel frattempo anche `pdflatex` con il pacchetto `pdfpages` potrebbe fare un lavoro equivalente o

migliore di quello che si può fare con **ghostscript**; l'unico vantaggio di quest'ultimo è che, volendo, riuscirebbe a trasformare direttamente un file in formato PS in un file conforme alla norma PDF/A-1b.

Inoltre con la distribuzione 2016 di TEX Live, è stata distribuita anche la versione 0.95 del programma **luatex** che introduce notevoli modifiche rispetto alla versione distribuita con TEX Live 2015; alcune di queste sono funzionali all'uso di LuaLATEX per comporre documenti conformi allo standard PDF/A.

In questo articolo cercherò di esporre succintamente come si può usare il pacchetto **pdfx** con i programmi di tipocomposizione basati sul mark up LATEX; aggiungerò qualche nota per descrivere il procedimento che si può seguire con il programma **pdflatex** insieme al pacchetto **pdfpages** e con il programma **ghostscript** per creare documenti PDF/A compatibili; questi procedimenti possono essere utili non solo per documenti interi, ma anche per singole parti da includere nei documenti completi, per esempio disegni, tabelle, e simili, da riportare come citazioni da altri documenti.

2 Caratteristiche dei documenti PDF/A

Perché un documento PDF sia conforme alla norma ISO 19005:2005 e alle sue successive modifiche, esso deve essere composto in modo che soddisfi a certi requisiti, diversi per le varianti 'a' o 'b' di ciascuna norma. La variante 'a' richiede che il file sia *tagged PDF*, mentre la variante 'b' non ha questa esigenza. Siccome a mia conoscenza, tranne ConTEXt che però non si basa sul mark up LATEX, nessuno dei programmi di tipocomposizione del sistema TEX è in grado (per ora¹) di produrre file di tipo *tagged PDF*, la specifica 'a' è fuori discussione. Mi limiterò dunque a descrivere solo la produzione di documenti conformi alla norma PDF/A-1b; questi devono avere le seguenti caratteristiche.

1. Il file PDF deve essere compilato con il linguaggio PDF di livello 1.4. Secondo standard approvati dopo il 2005, ora si potrebbero usare anche versioni più avanzate.

1. In realtà da tempo si trova nel sito indicato nella documentazione di **pdfx** un insieme di file da compilare sulla propria macchina al fine di produrre l'eseguibile di **pdftex** versione 1.50, capace di inserire i metadati richiesti; pur essendo disponibile da alcuni anni, questa versione non è mai stata resa pubblica nel senso di essere pubblicamente distribuita con TEX Live che, invece, ancora oggi contiene la versione 1.40. Quella versione sperimentale probabilmente aveva ancora dei piccoli bachi che non le consentivano di essere distribuita se non come versione beta assolutamente sperimentale e da usarsi a proprio rischio e pericolo. Io l'usai tempo fa, senza incontrare problemi particolari, ma non l'ho più né scaricata, né compilata, né integrata con il formato del linguaggio LATEX e non mi sento di raccomandarla proprio perché l'autore Hàn Thê Thành non l'ha resa pubblica lui stesso.

2. Il file deve contenere un insieme di metadati, alcuni obbligatori, altri facoltativi; il pacchetto **pdfx** provvede ai metadati obbligatori e ad alcuni facoltativi.
3. I metadati devono essere contenuti nel file PDF in chiaro, non in forma compressa né criptata.
4. Il file deve contenere una specificazione del profilo di colore per le immagini che contiene; il pacchetto **pdfx** provvede un profilo di colore RGB abbastanza generico; se però l'utente desidera usare un profilo di colore particolare lo può fare usando i comandi appositi messi a disposizione dal pacchetto **pdfx**. I profili di colore RGB vanno bene sia per le immagini colorate sia per quelle in tonalità di grigio.
5. Il file PDF deve essere autosufficiente per quel che riguarda i font, perciò deve contenere almeno tutti i glifi dei vari font usati nel documento.
6. Sono assolutamente vietati i font bitmapped (o *raster*, o a *matrici di punti*); nessun glifo deve avere larghezza nulla.
7. Non sono ammessi overlay di glifi; in particolare non sono ammessi gli accenti e gli altri segni diacritici ottenuti per sovrapposizione, come quando si usa la codifica dei font OT1.
8. Esistono delle limitazioni per quel che riguarda i collegamenti ipertestuali, se non quelli verso punti interni allo stesso documento.

Esistono altre limitazioni, descritte in documenti avanzati relativi alla norma ISO 19005, ma i punti importanti sono quelli elencati sopra.

3 Il pacchetto pdfx

Ciò premesso, quello che fa il pacchetto **pdfx** consiste proprio nel garantire che le prescrizioni siano soddisfatte, ma non può fare miracoli. Per esempio non dipende da **pdfx** se l'utente inserisce un'immagine con profilo di colore diverso da RGB, magari col profilo CMYK; non può correggere i metadati errati, anche se è in grado (specialmente con **pdflatex**, che può usare solo font codificati a 8 bit, in particolare font Type 1, con codifiche molto diverse le une dalle altre) di trasformare i glifi di entrata in modo da renderli compatibili con lo standard². Non ci sono problemi a questo proposito se si usa LuaLATEX, ma... Problemi simili si possono verificare con l'importazione di file esterni, ma si tratta sempre di sviste dell'utente.

Per i metadati facoltativi il pacchetto **pdfx** offre un certo numero di parole chiave documentate per

2. Molti anni fa ho creato i font greci a 8 bit con una codifica che di fatto è diventata quella (semi)ufficiale nel sistema TEX, cioè la codifica LGR; con mia grande sorpresa questi miei font greci non hanno mai dato nessun problema con le norme per l'archiviabilità; invece ho incontrato problemi seri con i font Computer Modern (CM) matematici usati da **pdflatex**.

esteso in RADHAKRISHNAN *et al.* (2016). Ma la documentazione stessa richiede di inserire questi metadati in un file con il nome uguale a quello del file principale del documento ma con l'estensione `.xmpdata`. Il pacchetto consiglia però di usare sempre la codifica d'entrata `utf8` sia per i metadati sia per il documento e di creare il file dei metadati mediante l'uso dell'ambiente `filecontents*` in questo modo:

```
% !TEX encoding = UTF-8 Unicode
\begin{filecontents*}{\jobname.xmpdata}
\Author{Mario Rossi}
\Title{My archivabile document}
\Publisher{University of Einstenia}
\Keywords{PDF/A\sep
pdfx.sty package\sep
long term archiving\sep
LuaLaTeX}
\end{filecontents*}
\documentclass[...]{...}
\usepackage[a-1b]{pdfx}
...
\begin{document}
...
\end{document}
```

In questo modo si è sicuri che il documento e i metadati sono composti con la stessa codifica `utf8`; questo metodo va bene per tutti e tre i principali programmi di tipocomposizione. Solo per i file da compilare con `pdflatex` è necessario specificare nel preambolo l'uso del pacchetto `inputenc` con l'opzione `utf8`, mentre con gli altri due programmi non è necessario specificare nulla del genere, ma è necessario usare solo o prevalentemente font OpenType, ricordando che i font a 8-bit, se mai dovessero assolutamente venire usati, devono essere richiamati nei preamboli in modo particolare, rispettando certe priorità che sono descritte nella documentazione del pacchetto `fontenc`.

Nei limiti del possibile raccomanderei di usare la lingua inglese per i metadati per due motivi: (a) l'inglese di solito non usa parole con diacritici, e (b) se questi metadati devono essere usati per ricerche biblioteconomiche in giro per il mondo, è meglio che siano scritti nella lingua franca, l'inglese, piuttosto che in qualsiasi lingua nazionale.

4 Font problematici

Se si evitano i font Computer Modern sia per il testo sia per la matematica si è già a buon punto; si noti che tutti i font matematici diversi dai CM che si possono usare tramite appositi pacchetti, soffrono degli stessi inconvenienti dei CM, perché i loro file metrici sono ricavati dai file metrici di quelli. Per evitare di usare i file matematici difettosi si deve evitare di usare `pdflatex`; perciò si crei il documento con `lualatex`³: è un po' più lento di

3. Per i motivi esposti più avanti si eviti di usare `xelatex`.

`pdflatex`, ma è decisamente consigliabile per la creazione di file PDF archivabili. I font OpenType matematici non sono ancora molto numerosi, ma sono sufficienti per ogni evenienza, anche se l'occhio (molto) esercitato riesce a individuare alcune piccolissime anomalie di disegno di questi font rispetto ai font testuali con cui si dovrebbero accoppiare. Qui è troppo lungo scendere nei dettagli; posso solo dire in base alla mia esperienza che io non ho mai incontrato problemi seri con i font matematici OpenType.

Se bisogna importare brani di testo, formule, tabelle o simili, per citazioni da altri testi, è possibile che anche se si traggono da file PDF, quegli elementi, una volta estratti, non siano compatibili con i requisiti dei file PDF/A. In questi casi se si tratta di testo non dovrebbe essere un problema ricomporlo; se si tratta di tabelle la ricomposizione è più laboriosa ma è fattibile; se si tratta di formule che contengono simboli strani la cosa diventa difficile se non impossibile. In questi casi conviene trasformare il codice estratto in una figura; sarebbe preferibile il formato PNG, ma siccome questo potrebbe contenere delle trasparenze, vietate nel file PDF/A compatibili, è necessario ricorrere al formato JPG con un'adeguata densità di pixel.

Talvolta si riesce a mantenere il formato vettoriale ricorrendo al programma `inkscape` che è capace di trasformare i glifi presenti nel testo nei loro disegni vettoriali; lo fa molto bene e così facendo non si perde in qualità, ma l'operazione non produce sempre un file PDF compatibile con la norma PDF/A per cui non serve a niente incorporarlo nel proprio documento se è irrimediabilmente incompatibile.

5 Figure problematiche

Gli stessi discorsi valgono per le figure problematiche quando sono in formato vettoriale e se ne vuole mantenere la vettorialità. Oppure per le figure in formato PNG che contengano trasparenze. Per questi ultimi tipi di figure si perdono le trasparenze ma non si perde quasi nient'altro se le si trasforma in formato JPG. Le figure potrebbero avere un profilo di colore diverso da profilo RGB: per controllarlo ed eventualmente modificarlo si può usare il programma freeware e open source GIMP in grado di eseguire queste verifiche e le eventuali correzioni.

6 Usare pdflatex

Per produrre un file PDF/A compatibile basta usare un preambolo e un file `.xmpdata` come mostrato nel listato della pagina 84; naturalmente perché il file sia veramente compatibile con la norma bisogna che siano rispettate tutte le cautele indicate nei paragrafi precedenti.

7 Usare lualatex

Con `lualatex` le cose sono sotto certi aspetti più semplici, nel senso che la codifica con cui il file sorgente viene registrato sul disco *deve* essere `utf8`, non si può scegliere un'altra codifica. Un discorso simile vale per i font, che *devono* essere di tipo OpenType, anche se con particolari precauzioni si potrebbero usare anche font Type 1; lo si sconsiglia vivamente perché i font OpenType coprono la quasi totalità delle necessità. Per imitare i font di default usati con `pdflatex`, i font OpenType CMU (con grazie, senza grazie, a spaziatura fissa, latini, greci, cirillici, eccetera) sono praticamente completi. Possono non essere sufficienti per l'arabo, l'ebraico, il cinese, il giapponese, il coreano, ma con ogni sistema operativo sono installati anche moltissimi font OpenType, e serve solo la pazienza di trovare quelli giusti per comporre in una lingua esotica; alla peggio si possono scaricare dalla rete altri font OpenType gratuiti o commerciali, così che si possano soddisfare tutte le esigenze. I font CMU non sono ancora dotati della collezione corrispondente di font matematici, ma i font OpenType matematici XITS Math⁴ vi si accoppiano abbastanza bene. Questo testo è composto con i font CMU testuali e i font XITS matematici: si confronti la formula seguente per constatare le minime differenze di stili fra i font matematici e quelli testuali.

$$x_{1,2} = \begin{cases} \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} & \text{se } b^2 - 4ac > 0 \\ \frac{-b}{2a} & \text{se } b^2 - 4ac = 0 \\ \frac{-b \pm i\sqrt{4ac - b^2}}{2a} & \text{se } b^2 - 4ac < 0 \end{cases}$$

8 Usare xelatex

Con `xelatex` le cose sono un po' più complesse, perché questo programma di suo non produce in uscita un file PDF, ma questo viene ottenuto dalla trasformazione di un file in formato DVI esteso mediante il programma `xdvipdfmx`. Questo è un programma completamente distinto da `xelatex` e non sa a che cosa il file DVI esteso è destinato a essere usato. Siccome il file PDF/A compatibile deve avere i metadati in formato non compresso, bisogna che `xelatex` passi il suo file di uscita a `xdvipdfmx` in modo che nulla sia compresso.

Per fare questo bisogna usare un comando dalla finestra comandi che esegua questo compito senza comprimere il file finale; in questo modo nemmeno i metadati vengono compressi. Il comando è il seguente (da scrivere in una sola riga):

```
xelatex -shell-escape
```

4. I font OpenType Latin Modern Math si accoppierebbero anche meglio, ma secondo la mia esperienza i font XITS, sebbene siano un pochino più scuri, si accoppiano bene e sono più completi.

```
-output-driver="xdvipdfmx -z 0"
<filename>.tex
```

Esso è configurato per usare la modalità `shell escape` che consente al compilatore di eseguire comandi che si potrebbero dare in una finestra comandi; come tale, quell'intera riga può venire digitata in una finestra comandi e il sistema operativo provvede a eseguire il comando principale `xelatex`; al momento opportuno questo si interrompe e ripassa il controllo al sistema operativo per trasferire il suo file di uscita all'`output-driver` a cui fornisce anche l'opzione `-z 0`; questa opzione è quella che ordina a `xdvipdfmx` di *non comprimere* il file PDF che esso produce. Ovviamente si potrebbe configurare il proprio *shell editor* perché disponga di un bottone o di una voce di menù per eseguire questa operazione, senza doversela ricordare a memoria o senza dover andare a consultare nessuna documentazione per ricavare la sintassi corretta.

Ciò produce due effetti: (a) i metadati entrano nel file di uscita non compressi, ed è quello che si vuole ottenere, e (b) l'intero file non è compresso, quindi può avere dimensioni spropositate, anche 10 volte maggiori di quelle che lo stesso PDF avrebbe se fosse compresso. Questo inconveniente può essere "curato" come si vedrà fra poco.

Tuttavia a questo punto ci si potrebbe domandare: perché usare `xelatex` con le complicazioni descritte sopra, se lo stesso file sorgente potrebbe essere compilato altrettanto bene, se non meglio, con `lualatex`? Una risposta che riesco a trovare, ma probabilmente non è l'unica, è che non si ha abbastanza dimestichezza con `lualatex` e quindi non ci si azzarda a usare un programma che non si conosce bene. Un'altra risposta potrebbe essere che `lualatex` ha un modo suo particolare per gestire la divisione in sillabe per ogni lingua presente nel documento da comporre, e che in certi rarissimi casi questa modalità non è soddisfacente. Questa seconda spiegazione può interessare gli umanisti che si occupano di linguistica; per esempio se si occupano di latino o di greco, queste due lingue hanno ciascuna tre modi diversi di dividere in sillabe; attualmente io riesco a gestirmi il latino e il greco con le tre diverse impostazioni per ciascuna lingua ricorrendo a file di descrizione di queste lingue che mi sono costruito per mio uso e consumo, ma il gruppo di lavoro `tex-hyphen` sembra che non sia d'accordo ad adottare la mia soluzione (che, lo ammetto, è piuttosto artigianale, ma funziona); può darsi che in futuro anche i file ufficiali di queste lingue saranno in grado di gestire le loro tre varianti di sillabazione.

9 Produrre un file PDF/A con pdflatex e pdfpages

Se si dispone di un file PDF comunque creato e lo si vuole rendere conforme alla norma, si può

usare anche un'altra tecnica. Devo dire subito che io non la userei per generare un intero documento, ma solo per rendere PDF/A compatibili semplici file da includere nel documento vero e proprio. Infatti per questa via si perdono i collegamenti ipertestuali interni e il documento finale non sarebbe “navigabile” se lo si legge con un dispositivo elettronico; però, lo stesso documento stampato andrebbe bene, perché a stampa i collegamenti ipertestuali attivi non servono.

Il semplice metodo consiste nel creare un piccolo file da elaborare con `pdflatex`; sia per esempio `mypdffile.pdf` il file da convertire nel formato PDF/A conforme alla norma ISO, diciamo `my-PDFA-file.pdf`. Si crea dunque il file di trasformazione in questo modo;

```
% !TEX encoding = UTF-8 Unicode
% !TEX TS-program = pdfLaTeX
%
%% File my-PDFA-file.tex
%
% esempio di metadati
\begin{filecontents*}{\jobname.xmpdata}
\Author{Mario Rossi}
\Title{My PDF/A compliant document}
\Publisher{University of Einstenia}
\Keywords{PDF/A\sep
pdfx.sty package\sep
long term archiving\sep
LuaLaTeX}
\end{filecontents*}
%
\documentclass[a4paper]{book}
\usepackage[a-1b]{pdfx}
\usepackage{pdfpages}
\nofiles
\begin{document}\pagestyle{empty}
\includepdf[pages=-]{mypdffile.pdf}
\end{document}
```

Lo si compila regolarmente con `pdflatex` come ci ricorda il secondo commento all'inizio del file. Ovviamente, come ci ricordano la prima e la quarta riga di commento, questo piccolo file va registrato con la codifica `utf8` su disco con il nome `my-PDFA-file.tex`. Se il documento originale è stato composto con un formato di carta diversa da quella indicata con l'opzione `a4paper` si cambi l'opzione corrispondentemente, ma non è necessario cambiare altro, nel senso che i comandi che seguono importano nel documento che si sta componendo una alla volta le pagine del documento originale senza cambiare le loro caratteristiche grafiche né per quel che riguarda l'impaginazione né per quel che riguarda i font.

Se il documento originale non conteneva oggetti incompatibili con lo standard PDF/A, il file di uscita risulta conforme allo standard, essendo stato arricchito di tutti i metadati necessari e di alcuni metadati facoltativi.

10 Usare ghostscript

Anche il programma `ghostscript` consentirebbe di eseguire una procedura simile a quella indicata nel paragrafo precedente; il lettore interessato può leggere la documentazione fra quella di `ghostscript`; ne ho scritto anche in altri articoli e in altri documenti in rete, ma io stesso considero questa via come un residuo dei miei primi tentativi di creare file PDF/A compatibili; era il 2008 quando le funzionalità presenti oggi nella distribuzione del sistema TEX non erano disponibili.

Il risultato finale ha gli stessi inconvenienti che si incontrano con il procedimento descritto nel paragrafo precedente; ritengo però che la gestione dei metadati e del profilo di colore sia decisamente più semplice se si ricorre alle funzionalità del pacchetto `pdfx` che, lo ricordo, si può usare solo con i programmi di tipocomposizione del sistema TEX. L'unico vantaggio che `ghostscript` potrebbe avere, sarebbe quello di poter trasformare direttamente un file in formato PS in un altro file in formato PDF/A, ma si tratta di un vantaggio molto marginale, perché la trasformazione di un file PS in un file PDF è già eseguibile molto facilmente con il programma `pstopdf` facente parte della dotazione di programmi dei sistemi operativi UNIX e Linux, ma installabile anche per i sistemi operativi Windows. Ovviamente anche `ghostscript` può trasformare un file PS in uno corrispondente in formato PDF attraverso il suo driver `pdfwrite`.

11 Il file PDF ottenuto soddisfa i requisiti PDF/A?

Ora la domanda ovvia che ci si deve porre è se il PDF ottenuto soddisfa i requisiti della norma ISO sull'archiviabilità.

Il problema è abbastanza serio: da un lato non c'è nessuna garanzia che qualche inesattezza si sia intrufolata in quanto ho descritto, dall'altro c'è la necessità di essere sicuri di aver ottenuto il risultato voluto e di poter corredare il file con un “certificato di conformità”.

Il primo aspetto è quello più delicato. Non è infrequente che si immettano nel documento finale file esterni dei quali si è sicuri che non contengano incompatibilità con lo standard PDF/A, quando invece ne contengono qualcuna.

Tuttavia il metodo per accertare questa compatibilità ed eventualmente produrre il certificato di conformità si avvale dello stesso programma: Adobe Acrobat Pro XI e versioni successive.

Esistono anche dei siti in rete che svolgono questo servizio gratuitamente; è possibile eseguire l'upload del proprio file PDF per ricevere a stretto giro di posta elettronica un messaggio che contiene la lista degli errori o il certificato di conformità. Ammetto di averne provati alcuni, ma non sono mai riuscito a ricevere verifiche positive, anche con

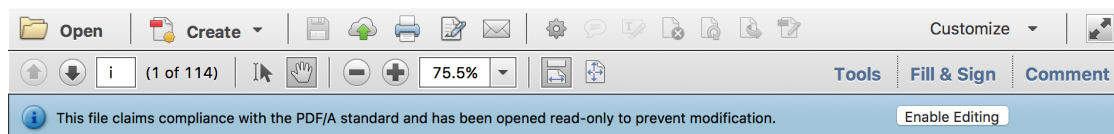


FIGURA 1: Avviso di Acrobat Pro sulla possibilità che il file appena aperto possa essere conforme con la norma PDF/A.

documenti che Adobe Acrobat Pro XI certificava come privi di problemi e quindi conformi con lo standard PDF/A.

Io credo che i tecnici dell'Adobe, che fanno parte della commissione che si occupa della norma ISO 19005, siano capaci anche di realizzare i programmi di verifica delle norme che loro stessi hanno contribuito a definire. Quindi sono propenso a fidarmi di più dell'esito della verifica effettuata con il modulo **Preflight** di Acrobat Pro XI, che non della verifica gratuita eseguita in rete.

Nello stesso tempo mi spiace che non esista un programma gratuito per verificare la conformità con la norma, come invece esistono più programmi gratuiti o commerciali che producono o dicono di produrre file compatibili. Oltre ai programmi di tipocomposizione del sistema TEX con l'aiuto del pacchetto pdfx, anche i word processor Word, di Microsoft Office, e Writer, di LibreOffice e di OpenOffice, sono in grado di esportare i loro file nel formato PDF e offrono anche la possibilità di specificare se si desidera la conformità non solo con la norma PDF/A-1b ma anche PDF/A-1a.

Infatti, tutte le volte che ho provato a esportare file in formato PDF da quei word processor, in entrambi i formati A-1b e A-1a, la funzione *Preflight* non mi ha mai risparmiato elenchi più o meno lunghi di manchevolezze. Con questo non voglio dire che quei word processor siano particolarmente inefficienti; semplicemente sembrano esserlo con i documenti non completamente testuali che io sono solito produrre.

Non sono in grado di dire niente pro o contro il word processor **Pages** del Mac, sebbene io usi un Mac, perché non l'ho mai usato per produrre file PDF che potessero ambire al rango di file archiviabili (talvolta vi scrivo lettere burocratiche o commerciali).

Prima di descrivere come procedere con Adobe Acrobat Pro XI, devo aggiungere che si tratta di un programma commerciale, che costa ben altro che qualche decina di euro. Tuttavia l'Adobe, come molte altre aziende dispone di un programma *Education* per vendere a prezzi assolutamente abbordabili i suoi prodotti ai docenti, al personale e agli studenti delle università e delle scuole secondarie superiori. Dipende dall'uso che se ne fa ma, visto che lo trovo molto utile, ritengo che quel programma sia consigliabile. Giudizio personale, naturalmente, ma nel parlare del programma *Education* non credo di aver violato alcuna norma etica.

Parlando di Adobe Acrobat posso aggiungere che consente di eseguire varie forme di editing sui file PDF; in particolare torna molto utile la sua funzione per comprimere file PDF non compressi o compressi poco; se questi file non compressi contengono i metadati per l'archiviabilità, metadati che devono rimanere in chiaro, non compressi, Acrobat Pro riesce a comprimere le parti lecite e a ridurre l'ingombro in memoria, salvando il risultato in un nuovo file; in sostanza con questo programma si supera il grosso inconveniente che si incontra quando si vuole ottenere un file PDF/A compatibile usando un file non compresso quale quello che si ottiene con **xelatex** come descritto nel paragrafo 8⁵.

12 Il modulo Preflight

Se si apre con Adobe Acrobat (o anche con Adobe Reader) un file PDF che potrebbe essere conforme col formato archiviabile, appare una riga celeste nella parte alta della schermata, figura 1, dove compare un testo che spiega che, siccome il file pretende di essere conforme allo standard PDF/A, quel file è stato aperto in modalità protetta per evitarne qualsiasi modifica che potrebbe togliergli la qualifica di compatibilità con la norma.

Non ci si lasci ingannare: quella scritta dice espressamente che è il file che pretende di essere conforme alla norma; potrebbe esserlo, ma più frequentemente non lo è. Se ne può controllare la conformità solo usando la modalità **Preflight**. Questa si attiva dal menù "Edit" e selezionando la voce "Preflight" nel menù a tendina; vedi la figura 2. Più semplicemente basta premere simultaneamente i tasti **[shift]**, **[command]** e **[X]**. Ciò fatto appare la finestra riportata nella figura 3.

Se non fosse già selezionata, si seleziona in questo menù la riga che sceglie lo standard per il quale deve essere eseguito il controllo; come appare nella figura, si è selezionata la riga che corrisponde allo standard PDF/A-1b; ora si clicca il pulsante **Analyze**. Il programma si mette a lavorare e termina con il suo responso.

Se il file non è conforme, nella finestra di **Preflight** appare una *dialog box* come quella mostrata nella figura 4; vi appaiono tutti gli errori che **preflight** ha trovato; cliccando sui triangolini

5. Merita ricordare che il modo normale di produrre file PDF compressi mediante **xelatex** funziona benissimo; è solo l'introduzione dei metadati come descritto nel paragrafo 8 che impedisce la compressione.

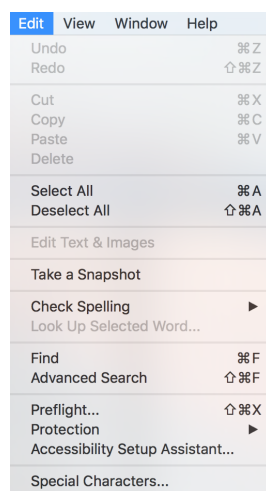


FIGURA 2: Menù a discesa della voce “Edit” dove si può selezionare il modulo **Preflight**.

grigi con la punta a destra si aprono e si leggono meglio le informazioni relative a ciascun errore; ammetto che l’interpretazione non è semplice, ma con un po’ di pratica si riesce a fare un po’ d’ordine e a capire la ragione dell’errore.

Potrebbe venire in mente di tornare alla sezione di **Preflight** mostrata nella figura 3 e di cliccare sul bottone **Analyze and Fix**; con questa azione **Preflight** rianalizza il file e, se ci riesce, corregge i (piccoli) errori alla sua portata. Talvolta questa azione ha successo, ma più spesso di quanto non si desideri essa fallisce; magari riesce a correggere qualche errore, ma non riesce a correggere gli errori importanti. La soluzione in questi casi rimane soltanto quella manuale; cercare di capire in che cosa consista ciascun errore e provvedere a eliminarlo.

Devo dire che l’esperienza che ho accumulato mi permette di affermare che la maggior parte di questi errori riguardano i font non incorporati e/o i profili di colore delle figure incluse nel testo. Se si sono seguiti i consigli indicati nei paragrafi precedenti, questi errori non dovrebbero manifestarsi, tuttavia qualche controllo sfugge all’utente, ma non sfugge a **Preflight**; l’indicazione dell’errore può essere espressa in modo criptico, ma se una parte da una di queste due ipotesi, l’informazione di **Preflight** è sufficiente a localizzare l’errore per analizzarlo e correggerlo.

Se invece **Preflight** non trova errori, esso mette un’informazione del tipo di quella indicata nella figura 5; se si vuole, si può cliccare sul pulsante **Create Report**, e il programma provvede a comporre un documento PDF che certifica tutte le caratteristiche molto dettagliate del file appena verificato, compresa, evidentemente, la conformità e lo standard al quale il file è conforme; in sostanza compila il certificato di conformità.

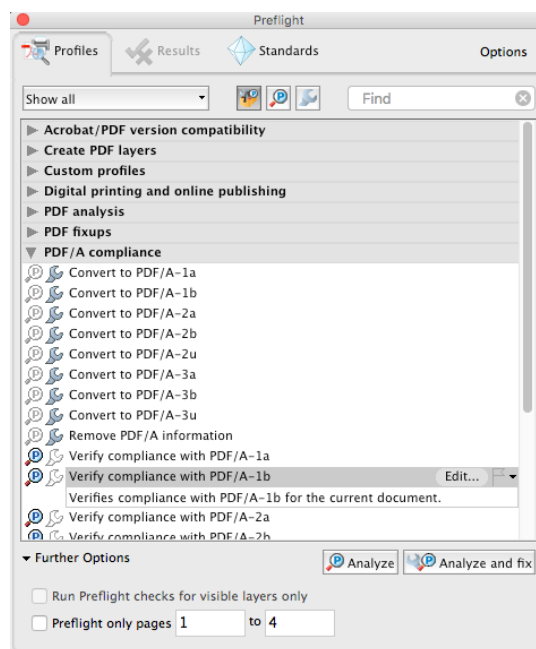


FIGURA 3: Menù delle analisi che si possono eseguire con **Preflight**.

13 Conclusione

I procedimenti indicati succintamente in questo articolo mostrano come si possano ottenere file PDF/A compatibili con i programmi di tipocomposizione del sistema $\text{T}_{\text{E}}\text{X}$. Ammetto che l’esposizione è troppo succinta per quel che riguarda la correzione degli inconvenienti che non portano a una conversione corretta, ma ci vorrebbe un intero libro per trattare questo genere di problemi. L’esperienza conta molto e ognuno deve farsela autonomamente, perché non si può trasmettere questo tipo di esperienza attraverso lo scritto.

Devo dire, però, che con l’uso di **lualatex** mi sono sempre trovato avvantaggiato perché si eliminano alla fonte quasi tutti gli errori che dipendono dai font, dai difetti dimensionali dei glifi, e dalle codifiche inconsistenti. Il pacchetto **pdfx** fornisce correttamente i profili di colore. I pochi errori che possono rimanere sono quelli veramente subdoli, ma sono rari. Come ho detto, non userei per quanto possibile il programma **xelatex** per produrre documenti conformi all’archiviabilità, perché si può quasi sempre ottenere lo stesso risultato usando **lualatex**. Il programma **pdflatex** nonostante sia il primo con il quale il gruppo di lavoro di **pdfx** si è cimentato, dà meno sicurezze di **lualatex**.

Credo che questi progressi del sistema $\text{T}_{\text{E}}\text{X}$ sulla strada dell’archiviabilità vengano incontro alle necessità di ogni tipo di archiviazione, compresa quella dei documenti di tipo umanistico a cui è stata mostrata un’attenzione particolare.

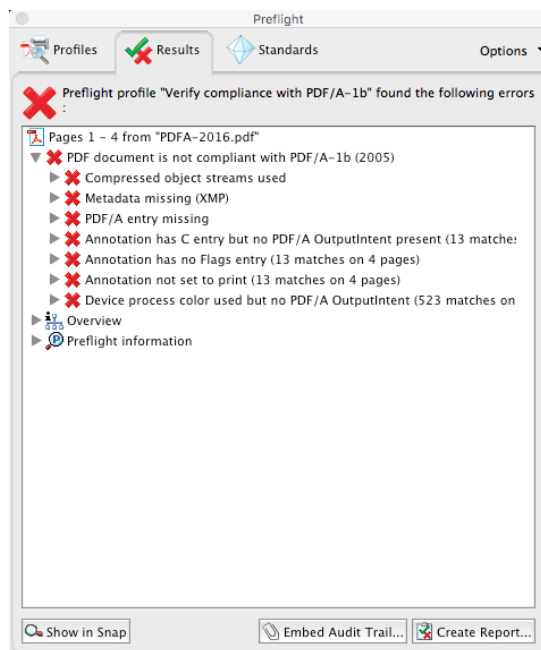


FIGURA 4: Risultato dell'analisi di Preflight su un file non conforme alla norma PDF/A.

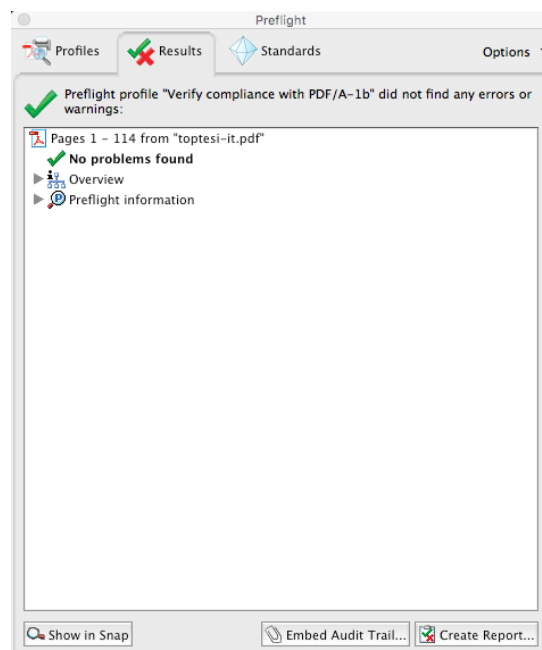


FIGURA 5: Risultato dell'analisi di Preflight su un file conforme alla norma PDF/A

Riferimenti bibliografici

- ADOBE (2006). «Adobe's free xmp toolkit for reading/writing xmp». <http://www.adobe.com/devnet/xmp/sdk/eula.html>. In basso a sinistra nella schermata puntata dall'URL c'è l'ancora con il link per il download.
- BECCARI, C. (2009). «Il formato archiviabile dei file PDF». *ArsTEXnica*, (7).
- DRÜMMER, O., VON SEGGERN, D. e OETTLER, A. (2007). *PDF/A in a Nutshell: Long term archiving*. Callas Software GmbH. ISBN: 978-398116481-7.
- HAGEN, H. (2016). «LuaTEX 0.90 backend changes for PDF and more». *TUGboat*, **37** (1), pp. 53–57.
- HÀN THẾ THÀNH (2008a). «Generating PDF/A compliant PDFs from pdftex». http://support.river-valley.com/wiki/index.php?title=Generating_PDF/A_compliant_PDFs_from_pdftex.
- (2008b). «Web message». http://sarovar.org/tracker/index.php?func=detail&aid=945&group_id=106&atid=495.
- HÀN THẾ THÀNH, RAHTZ, S., HAGEN, H., HENKEL, H., JACKOWSKI, P. e SCHRÖDER, M. (2007). «The pdfTEX user manual». [\\$TEXMFDIST/doc/latex/pdftex/manual/pdftex-a.pdf](http://$TEXMFDIST/doc/latex/pdftex/manual/pdftex-a.pdf).
- MARTINI, E. (2007). «Lo standard PDF/A: Sperimentazione di software per la verifica di conformità allo standard ISO 19005:2005». <http://>

[//www.cnipa.gov.it/site/_files/ref02_RS_3.1_martini.pdf](http://www.cnipa.gov.it/site/_files/ref02_RS_3.1_martini.pdf).

- PDF/A (2005–2008). «Technical notes». <http://www.pdfa.org/>. Il sito contiene i testi delle annotazioni tecniche rappresentati dai file `tn0001_pdfa-1_and_namespaces_2008-03-19.pdf`, `tn0003_metadata_in_pdfa-1_2008-03-18.pdf`, `tn0006_digital_signatures_in_pdfa-1_2008-03-14.pdf`, `tn0008_predefined_xmp_properties_in_pdfa-1_2008-03-20.pdf`, `tn0009_xmp_extension_schemas_in_pdfa-1_2008-03-20.pdf`.
- RADHAKRISHNAN, C. e HÀN THẾ THÀNH (2008). «Generation of PDF/X-1a and PDF/A-1b compliant PDF's with PDFTEX — pdfx.sty». <http://tug.ctan.org/tex-archive/macros/latex/contrib/pdfx/>.
- RADHAKRISHNAN, C., HÀN THẾ THÀNH, MOORE, R. e SELINGER, P. (2016). «Generation of PDF/X- and PDF/A-compliant PDFs with pdfTEX — pdfx.sty». [\\$TEXMFDIST/doc/latex/pdfx/pdfx.pdf](http://$TEXMFDIST/doc/latex/pdfx/pdfx.pdf). Version 1.5.8.
- SCARSO, L. (2010). «PDF/A-1a in ConTEXt-mk4». *ArsTEXnica*, (10).

▷ Claudio Beccari
claudio dot beccari at gmail
dot com