Cahiers GUTenberg

LTX2RTF : ENVOI DE DOCUMENTS LaTeX AUX USAGERS DE WORD

Daniel Taupin

Cahiers GUTenberg, nº 31 (1998), p. 28-36.

<http://cahiers.gutenberg.eu.org/fitem?id=CG_1998___31_28_0>

`ltx2rtf` : envoi de documents LaTeX aux usagers de Word

Daniel TAUPIN

laboratoire de Physique des Solides
bât. 510, centre universitaire
F-91405 Orsay Cedex

Résumé. `ltx2rtf` est un compilateur qui traduit des sources LaTeX 2ε en format RTF, format utilisé par divers éditeurs de texte, notamment Microsoft Word. Écrit initialement par Fernando DORNER et Andreas GRANZER, étudiants à Vienne (Autriche), on trouve leur version initiale sous le nom `latex2rtf` dans les serveurs CTAN.
Nous l'avons considérablement corrigé et adapté à LaTeX 2ε en 1997-98, sous le nom `ltx2rtf`. La distribution est essentiellement destinée à être utilisée dans les fenêtres MS-DOS de Win95 et Win3.11, mais le programme, écrit en C standard, peut être compilé sur n'importe quel système UNIX possédant un compilateur GCC.

Abstract.

The `ltx2rtf` compiler translates LaTeX source files into RTF, a format available in many text editors, notably Microsoft Word. Originally written by Fernando Dorner and Andreas Granzer, students in Vienna (Austria), the initial version can be found on CTAN under the name `ltx2rtf`. The distribution is mainly intended for use under MS-DOS Win95 and Win 3.11, but the program, written in standard C, can be compiled on any UNIX *system with a CGG compiler.*

1. Introduction : pourquoi un convertisseur LaTeX-RTF ?

Les utilisateurs habituels de LaTeX — dont l'auteur de ces lignes — sont toujours affrontés à de sérieux problèmes lorsque leurs articles doivent être transmis électroniquement à des non-utilisateurs de LaTeX.

1.1. Les diverses impasses de la transmission électronique de documents LaTeX

La transmission d'un document (source) LaTeX aux autres utilisateurs de LaTeX ne pose guère de problèmes, du fait que tous les formats LaTeX peuvent comprendre au moins

la codage 7-bit des lettres accentuées. Il n'en est pas de même quand le (ou les) destinataire(s) est allergique à la notation LaTeX, cas général hors du monde scientifique :

1.1.1. Envoi de texte ordinaire

Même cette solution apparemment évidente — le fichier `*.TXT` qui est la solution du pauvre — est impraticable à cause des lettres accentuées qui sont codées au moins en trois représentation différentes, le codage 850/437 pour les PC, le codage MacIntosh et le codage ISO-latin1, sans compter les codages ISO-8859* de l'Europe de l'Est. Pis que cela, même entre utilisateurs d'un même système d'ordinateur comme les PC sous Windows, certains utilitaires stockent les lettres au codage 850 (par exemple Microemacs ou MSDOS editor) mais Word et Netscape[1] les stockent au codage ISO-latin1 !

Même la solution du codage 7-bit est souvent rejetée comme « illisible » par les non-informaticiens incapables de comprendre "`r\'esum\'e`" à la place de "résumé". Et de toutes façons, les mêmes sont probablement incapables d'effectuer globalement la remplacement de la chaîne "`\'e`" par leur caractère "é" dans le document reçu.

On pourra objecter que, si on envoie le document via un système moderne de courrier électronique comme Netscape, le document arrive toujours lisible aux destinataires. Ceci est faux pour deux raisons : d'une part, même sur PC, les *attachments* doivent être codés en ISO-latin1, et non en 850 par l'émetteur car Netscape (4.05) fait la conversion 850-ISO à l'entrée sur le clavier, pas lors des *attachments*, d'autre part parce qu'à la réception — c'est-à-dire chez le destinataire peut-être incompétent — se pose le problème inverse, faute de quoi les "é" sont reçus comme des "Ú" et le reste à l'avenant.

1.1.2. Envoi d'un fichier PostScript

C'est au premier abord la « bonne » solution, utilisée par tout le monde dans le domaine scientifique ou informatique. Elle échoue pourtant avec les administrations et le grand public pour plusieurs raisons, négligées par beaucoup d'affirmations catégoriques :

1. Tous les scientifiques — au moins ceux des sciences « dures » — ont accès au moins à une imprimante PostScript, mais ce n'est pas le cas des administrations, ni *a fortiori* des personnes privées, ne serait-ce qu'à cause du prix élevé des imprimantes PostScript.

[1] Nous avons peu d'expérience d'Internet Explorer, mais quand on lui fait ouvrir un fichier texte au codage 850, il est incapable de visualiser les lettres accentuées, alors qu'il affiche très correctement l'ISO-latin1.

2. Même si les destinataires peuvent accéder à une imprimante PostScript, ceux qui reçoivent un fichier ou un courrier sous Windows ne disposent d'aucun moyen standard d'envoyer un fichier PostScript à leur imprimante : certes Windows fournit plusieurs pilotes pour les imprimantes PostScript, mais aucun d'eux ne fait la transmission sans transformation, laquelle n'est possible que par les commandes `lp` ou `lpr` sous UNIX, ou la commande `copy` de MS-DOS... qui ne marche pas si l'imprimante est connectée en réseau.
3. D'autres logiciels peuvent résoudre le problème, mais on ne peut pas demander raisonnablement à n'importe quel destinataire (à une multitude dans la cas d'une liste) d'installer GhostScript, GhostView or `prfile10`.

1.1.3. Envoi de fichiers PDF

Il est certain que la réception et l'impression de fichiers PDF ne nécessite pas d'imprimante PostScript puisque l'impression est faite par les pilotes installés sur l'ordinateur du destinataire. Mais la difficulté est la même que pour visualiser ou imprimer du PostScript : il faut que le destinataire ait installé Acrobat Reader et ce n'est pas fait dans tous les systèmes. On retombe sur l'objection précédente : si le destinataire ne l'a pas déjà installé, on ne peut pas lui demander de se procurer Acrobat Reader par `ftp` ou en utilisant son navigateur, et ensuite de l'installer, car une telle opération dépasse les capacités d'un non-informaticien.

1.1.4. Envoi de fichiers images

Plutôt que d'expédier un texte avec ses commandes de mise en page, on peut penser envoyer un fichier image du texte. C'est une solution raisonnable si le document fait une ou deux pages : on peut fabriquer un fichier GIF avec un scanner, et il existe plusieurs progiciels permettant de convertir un DVI en GIF, avec un intermédiaire BMP, PCX ou PostScript. L'avantage est que ce format est naturellement compressé, ce qui diminue les temps de transferts et l'espace de stockage. Toutefois :

1. beaucoup de destinataires ne savent pas qu'ils peuvent utiliser leur navigateur Netscape ou Microsoft Explorer pour visualiser un fichier GIF se trouvant déjà sur leur disque ;
2. même au format GIF, le bitmap d'un texte occupe bien plus de place que le texte en mode caractères : pour quelques pages, ce n'est pas un problème, mais ce n'est pas viable pour des dizaines de pages qu'il lui faudra en plus imprimer une par une ;
3. enfin, un *bitmap* possède un nombre rigide de pixels, dont l'impression peut être minuscule ou énorme selon la résolution de l'imprimante destinataire.

1.2. De la portabilité des logiciels et des documents

L'expéditeur d'un document LaTeX – tout comme l'expéditeur d'un programme C ou F77 — est donc affronté à un problème de *portabilité*.

Malheureusement, celui qui expose ces difficultés reçoit souvent des réponses du genre : « vous n'avez qu'à vous débarrasser de votre système Windows et passer à UNIX ! », « jetez donc votre vieille imprimante Epson et prenez une imprimante PostScript ! », « vous n'avez qu'à abandonner les éditeurs de texte Microsoft et utiliser LaTeX ! », « vous n'avez qu'à installer GhostScript, GhostView, `prfile10`, ou une partition Linux sur votre PC ! », etc.

Toutes ces réponses de bon sens sont justes, mais elle oublient une seule chose : le problème n'est pas ***mon*** installation personnelle quand j'envoie un document, le problème se trouve dans l'installation du destinataire, dont je ne connais souvent pas les compétences bureautiques ou informatiques, et qui est probablement incapable d'installer d'autres progiciels que ceux qu'il a trouvés sur son ordinateur lors de son acquisition ou de sa configuration par le fournisseur, ou quand il a obtenu son droit d'accès sur la station de travail multiutilisateurs.

Elle témoignent finalement d'une conception erronée de la portabilité : la portabilité ne consiste pas à adapter les systèmes aux logiciels ou aux documents, mais à adapter les logiciels et les documents aux systèmes — du moins les systèmes actuellement utilisés — tels qu'ils sont. Ceci est déjà vrai quand on parle de son propre système — car plusieurs logiciels pourraient exiger des changements incompatibles —, ça l'est encore plus quand il s'agit d'envoyer un document à un destinataire lointain, parfois inconnu.

2. L'idée sous-jacente à `ltx2rtf` : utiliser Word comme pilote

Quand on envoie un document à une multitude de destinataires, il faut se demander quel logiciel est le plus répandu parmi eux. La réponse est que, grâce à l'irrésistible publicité de Microsoft, ils possèdent tous[2] une version de Word[perfect] qui peut lire les fichiers RTF.

En fait, malgré ce qu'on peut dire de la politique de Microsoft, les spécifications de RTF sont publiées par cette société, à :

```
ftp://ftp.microsoft.com/Softlib/MSLFILES/GC0165.EXE
```

2 Peut-être l'ont-ils « piratée », mais ce n'est pas notre problème...

Il s'agit d'un fichier zippé auto-extractible qui fabrique un fichier DOC de 130 pages, imprimable par Word. En étudiant ces spécifications on obtient en principe le moyen de fabriquer du RTF à partir d'une source en LaTeX.

Nous avons bien dit « en principe » car l'expérience montre que, par exemple, Word 6.0 ne respecte pas vraiment les spécifications publiées par Microsoft, de sorte qu'il faut tester leur exactitude — sachant selon la loi de Murphy que le comportement effectif de Word supplante ce qui est écrit dans les spécifications — pour y adapter la génération du code RTF. Nous avons ainsi constaté que les *signets* n'étaient pas fiables avec Word 6.0, ni conformes à la documentation. D'autres comportements bizarres nous ont amené à penser que des bogues de principe (dans les accolades emboîtées) avaient été palliées par des bricolages adaptés aux seuls cas particuliers utilisés par Word.

La traduction de LaTeX en RTF a été commencée en 1994 par deux étudiants d'une institution qui semble être une *université technique* à Vienne (Autriche) et largement diffusée dans les CTAN sous le nom `latex2rtf`. Leur compilateur est fourni sous forme de plusieurs fichiers source C, facilement compilables grâce à un *Makefile* bien organisé.

Leur programme C est propre et bien structuré. En revanche, ces étudiants n'avaient qu'une connaissance réduite de LaTeX ; de ce fait, beaucoup de séquences ont dû être révisées par nos soins concernant la gestion des polices, les titres des niveaux de sections, les environnements `itemize`, `enumerate`, `description` et `tabular`, sans oublier les spécifications plus récentes de LaTeX 2_ε.

3. `ltx2rtf`

3.1. Logique et particularités

De la même manière que `latex2html`, `ltx2rtf` compile le code source LaTeX 2_ε et produit directement du RTF, au lieu de HTML.

Les macros intrinsèques de Word (et RTF) sont utilisées pour numéroter les sections, ce qui permet leur mise à jour quand on utilise ultérieurement Word pour insérer de nouvelles sections, c'est-à-dire des titres de niveaux divers selon la terminologie de Word. Au contraire, les environements `enumerate` produisent une numérotation figée, principalement pour contourner le fait que les macros intrinsèques de Word interdisent les sauts de paragraphe dans l'équivalent de cet environnement (Word rend chaque saut de paragraphe dans une énumération équivalent à `\item`).

Les lettres accentuées de l'Europe occidentale[3] — y compris les majuscules — sont traitées correctement, ainsi que le fameux caractère "œ" oublié dans ISO-latin1. Nous y avons rajouté les possibilités d'abréviations fournies dans le `french.sty` de Bernard GAULLE's et la commande `\scfamily` obtenue avec le `smallcap.sty` de Daniel TAUPIN, laquelle permet les petites capitales grasses et/ou penchées.

3.2. Implémentation

3.2.1. Généralités

1. Le code d'entrée peut être en 7-bit, ou ANSI (ISO-latin1) ou 850. Le codage MacIntosh est encore à l'état de projet.
2. Le code source C est compilable par n'importe quel compilateur GCC (peut de problèmes sérieux avec les compilateurs C usuels). Il a été testé avec le portage DJGPP du compilateur GCC pour DOS (natif, Windows 3.11 and Windows 95).
3. Rien d'autre n'est nécessaire, tant que l'on ne veut pas traduire des mathématiques.
4. Les parties mathématiques très simples sont traduites en utilisant les rares possibilités de RTF telles que relever ou abaisser des portions de texte et changer de police (corps et forme).

3.2.2. Les maths

Deux options permettent un traitement plus esthétique et rigoureux :

1. L'option `-m` fait appel à LaTeX pour l'*affichage des équations*, c'est-à-dire celles encadrées par des `$$` ou l'environnement `displaymath` (environnement `equation` dans le futur). Ensuite, à la manière de `latex2html` :
   - `ltx2rtf` invoque `latex` pour produire un fichier DVI pour chaque équation ;
   - `ltx2rtf` appelle une procédure externe (`DVI2PBM.BAT` sous MS-DOS) qui, à son tour,
     - soit appelle le pilote emTeX `dvidrv dvidot` pour produire des fichiers PCX, puis ensuite appelle des programmes NETPBM pour les convertir en PBM,
     - soit appelle DVIPS pour produire des fichiers PostScript file qui sont alors lus par GhostScript pour produire[4] un fichier PBM.

3 Les autres sont absentes des polices True Type utilisées par Word et RTF.

4 Grâce à l'aide d'Emmanuel BIGLER qui a proposé cette autre solution.

— Finalement, le fichier PBM file est lu par `ltx2rtf` lui-même et converti en ***wbitmap*** selon les spécifications de RTF[5];

2. L'option `-M` utilise LaTeX, non seulement pour les équations, mais aussi pour les parties mathématiques encloses entre de simples `$` ou dans l'environnement `math`.

3.2.3. Figures et musique

Le même procédé — LaTeX vers PBM en passant soit par GhostScript, soit par les pilotes emTeX — est utilisé pour traduire l'environnement `picture` et l'environnement `music` de MusiXTeX.

3.3. Les résultats

3.3.1. Les avantages

Du point de vue d'un connaisseur de LaTeX, le résultat (le texte et surtout les maths) est bien meilleur que ce qu'obtiennent les « wordistes » ordinaires, notamment en ce qui concerne les listes (cf. table 1).

Le RTF produit est donc excellent quand on veut envoyer par disquette ou courrier électronique un texte mis en page pour LaTeX à des destinataires mal connus, qui ont 95% de chances d'avoir un Word à leur disposition, alors qu'ils ont moins de 5% de chances de pouvoir imprimer un DVI ou un fichier PostScript.

3.3.2. Les inconvénients

Dès qu'il y a des maths ou des environnements nécessitant une sous-traitance partielle en LaTeX, l'installation de `ltx2rtf` présente des difficultés similaires à `latex2html` sauf que l'on n'a besoin, ni de Perl ni de GDBM/DBM, une différence qui est quand même loin d'être négligeable. Mais des inconvénients majeurs peut être relevés par les destinataire :

— Du fait de la suprématie universelle de Microsoft en matière de bureautique, tout défaut apparent dans ses produits doit être considéré comme *"not a bug, but a feature"*. Ainsi, toute mise en page différente de ce que le *wordiste* ordinaire sait faire (pensez à la difficulté d'avoir avec Word des paragraphes en retrait dans les listes, avec un retrait croissant avec l'imbrication des listes) risque d'être considéré comme un défaut, de la même manière que les majuscules accentuées qui

5 D'autres formats de figures sont décrits dans les spécifications de RTF, mais elles ne marchent pas avec Word 6.0 ; nous n'avons donc conservé que celle qui marche...

Fonction	État	Commentaires
Gestion LaTeX2ε des polices	OK	y compris `\scfamily`
Changements de corps	OK	y compris `\HUGE` et au delà
Lettres accentuées et lettres spéciales européennes	OK	... sauf oublis
`\medskip`, `\bigskip`	OK	valeurs approchées
`\parindent`, `\noindent`	OK	
`\part`, `\chapter`, `\section`, etc.	OK	numérotation par RTF
Inclusion de PCX, BMP	NON	pas d'équivalent de `\special{em:graph ...}` sans espacement. Possible à l'intérieur de `picture`.
`\label`, `\ref`, `\pageref`	NON	bogues aléatoires dans Word 6.0 qui ne respecte pas ses spécifications
Environnements		
`itemize`, `description`	OK	
`enumerate`	OK	numérotation figée par `ltx2rtf`
`tabular`	OK	largeurs arbitraires à ajuster avec Word (présente figure)
`displaymath` (ou $$)	OK	nécessite option `-m`/`-M`
`math` (ou $)	OK	nécessite option `-M`
`equation`	NON	*problème de numérotation*
`tabbing`	NON	
`picture` et `music`	OK	*sous-traités par LaTeX*
`multicols`	[oui ???]	résultats erratiques en Word 6.0

TABLE 1: Ce qui marche et ne marche pas avec `ltx2rtf` (document produit par `ltx2rtf` + Word 6.0).

sont si difficiles à réaliser (4 clics and 3 mouvements de souris pour fabriquer un æ ou un œ en français).

— Pis encore, la mise en page selon LaTeX utilise des puissantes commandes de base de RTF — elles ressemblent aux primitives de TeX et aux macros de base de plain-TeX — mais de tels résultats sont pratiquement impossibles à obtenir avec les macros toutes faites que l'on peut cliquer à la souris dans Word.

D'où l'impossibilité pour le destinataire de corriger le texte qu'on lui a envoyé, à l'exception de simples corrections textuelles (orthographe, ajout de phrases), à la rigueur introduction de sections ou sous-sections par le biais des *copier-coller*.

— Évidemment, les parties mathématiques sont figées ; tout au plus peuvent-elles être déplacées, agrandies, rétrécies, mais jamais éditées.

4. Conclusion

De même que DVIPS n'a pas pour but de permettre aux compositeurs d'abandonner LaTeX pour PostScript, `ltx2rtf` n'a pas l'intention de les inciter à passer de LaTeX à Word. Son but est seulement de leur permettre s'envoyer des textes proprement mis en page, multipliant ainsi par des dizaines le nombre des destinataires capables de les visualiser ou de les imprimer avec leurs seuls outils fournis par Microsoft.

5. Utilisation pratique

5.1. Disponibilité

`ftp://ftp.lps.u-psud.fr/pub/ltx2rtf/ltx2rtf.zip`

5.2. Installation

Le fichier `ltx2rtf.zip` contient les fichiers nécessaires ou utiles avec leurs arborescence.

Les exécutables y sont fournis pour MS-DOS (386 et plus), mais les fichiers source en C sont standard (GCC) et la génération UNIX se fait avec un simple `make`.

Les quatre fichiers `*.cfg` sont adaptables par l'utilisateur, notamment `config.cfg` qui indique à `ltx2rtf` comment il doit convertir les DVI des parties mathématiques pour produire les images PBM (commentaires explicatifs à l'intérieur).

Si l'on utilise pour les équations la conversion via GhostScript, la procédure `DVIPSGS.BAT` indique les paramètres à lui donner. Si l'on préfère utiliser les pilotes d'emTeX (plus rapides en pratique que GhostScript), on aura besoin de plusieurs programmes de la bibliothèque NETPBM, dont les exécutables pour MS-DOS sont fournis.