

Galerie de fichiers DTX*

Will Robertson

2007-08-07 (PDF original)

Sommaire

I	Extraire le fichier ins du dtx	4
II	Extraction conditionnelle de code	5
III	Réordonnement du code	6
IV	La galerie dtx	7

1 Introduction

Cette collection de fichiers montre des choses simples qu'il est possible de faire à l'aide du format, flexible et sous-estimé, de fichier docstrip. Il est conçu pour accompagner l'excellent et influent « dtxut » de Scott PAKIN, exemple de production, de cette façon, d'extensions LaTeX.

*Traduction et adaptation françaises par le TeXnicien de surface, version 1, 2008-05-01.

2 Fichiers de la galerie

On donne ici la liste des exemples de DTX de cette galerie en ordre approximatif de pertinence et de complexité.

single-source.dtx

Voici un exemple où les fichiers `.sty` et `.ins` sont extraits tous deux du fichier `.dtx`. Cela permet de ne maintenir qu'un seul fichier source et de produire un couple DTX/INS au « comportement normal » pour le dépôt sur le CTAN.

conditional-code.dtx

Cet exemple montre comment on peut extraire le code dans plusieurs fichiers ce qui permet, p. ex., de créer facilement des versions de « débogage » de l'extension.

rearrange.dtx

Cet exemple montre comment le code source peut être écrit et documenté dans un ordre logique différent de celui dans lequel il faut qu'il soit exécuté. Cela peut être utile, par exemple, lorsque les valeurs par défaut sont fournies au moment même du chargement de l'extension mais que l'on préfère les décrire d'abord dans la partie principale (non technique) de la documentation.

dtxgallery.dtx

Ce document-ci est un exemple de fichier DTX qui est lui-même le résultat de la compilation de nombreux fichiers DTX externes. Cette possibilité est utile pour créer un seul fichier de documentation pour une grande collection de logiciels — p. ex. les sources même de LATEX.

3 Utilisation de cette galerie

Il sera plus clair de cesser ici la lecture de ce document et de jeter un œil sur les fichiers DTX individuels — et leurs versions compilées PDF — présentés ci-dessus. La suite de ce fichier contient une compilation des fichiers DTX de la galerie, ce qui fournit un exemple par lui-même. On en lira plus à ce sujet dans le fichier [IV](#), page 7.

4 Interlude : éliminer les sentinelles

Puisque je suis en position de faire quelques commentaires, je peux en profiter pour présenter quelques tuyaux que j'utilise quand j'écris un fichier DTX.

Les « sentinelles » sont ce qui délimite les sections de code dans un fichier DTX. De nombreuses extensions se passeront de toute fantaisie en ce qui les concerne et on trouvera souvent quelque chose comme :

```
% ...
% \begin{macrocode}
%<*package>
<Tout le code commenté de l'extension>
%</package>
% \end{macrocode}
% ...
```

avec d'autres sentinelles utilisées pour détacher des choses comme le code du pilote au début et peut-être d'autres petits morceaux. Dans de tels cas, on n'a pas besoin d'inclure les sentinelles dans la documentation mise en page — si on essaie de faciliter la tâche du lecteur — aussi j'écris plutôt ceci :

```
% ...
% \iffalse
%<*package>
% \fi
% \begin{macrocode}
<Tout le code commenté de l'extension>
% \end{macrocode}
% \iffalse
%</package>
% \fi
% ...
```

Cela est utile également de toute façon lorsque l'on utilise un marquage logique en parties et sections pour séparer les fichiers dans le DTX.

PREMIÈRE PARTIE — EXTRAIRE LE FICHER INS DU DTX

5 *Code source documenté*

```
1 ⟨*package⟩
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{single-source-fr}
4   [2005/07/15 v0.1 dtx: single-source-fr]
5 \def\mymacro{salut :)}
6 ⟨/package⟩
```

6 *Contenu de single-source-fr.sty*

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{single-source-fr}
  [2005/07/15 v0.1 dtx: single-source-fr]
\def\mymacro{salut :)}
```

7 *Contenu de single-source-fr.ins*

```
\input docstrip.tex
\keepsilent\askforoverwritfalse
\nopreamble\nopostamble
\generate{\file{\jobname.sty}{\from{\jobname.dtx}{package}}}
\endbatchfile
```

8 *Contenu de single-source-lisezmoi.txt*

```
-----
single-source-fr.dtx

Ceci donne un exemple où tout est extrait
d'un seul fichier. dtx.
-----
```

DEUXIÈME PARTIE — EXTRACTION CONDITIONNELLE DE CODE

9 *Inclusion conditionnelle de code*

D'après les exemples précédents, la syntaxe `%<*guard>... %</guard>` devrait être un peu plus familière. Cet exemple montre l'extraction conditionnelle de tels éléments du document source et leur placement dans différents fichiers. Cette technique rend plus facile, p. ex., la maintenance d'une version de « débogage » de l'extension sans pollution du source public par le code en phase d'essai.

```
1 <A> code dans 'A'
2 <B> code dans 'B'
3 <!A> code qui n'est pas dans 'A'
4 <!B> code qui n'est pas dans 'B'
5 <A & B> code dans 'A' et 'B'
6 <A | B> code dans 'A' ou 'B'
7 <(A | B) & !(A & B)> code dans 'A' xor 'B'
8 <*A>
9 <B> 'B' imbriqu\`e dans 'A'
10 </A>
```

10 *Fichiers produits verbatim*

10.1 *Créé à partir de « A »*

```
code dans 'A'
code qui n'est pas dans 'B'
code dans 'A' ou 'B'
code dans 'A' xor 'B'
```

10.2 *Créé à partir de « B »*

```
code dans 'B'
code qui n'est pas dans 'A'
code dans 'A' ou 'B'
code dans 'A' xor 'B'
```

10.3 *Créé à partir de « A » et « B »*

```
code dans 'A'
code dans 'B'
code dans 'A' et 'B'
code dans 'A' ou 'B'
'B' imbriqu\`e dans 'A'
```

TROISIÈME PARTIE — RÉORDONNEMENT DU CODE

11 Exemple de réordonnement du source docstrip

Ceci est un essai pour montrer comment ça marche : `\test` → “1”

12 Code documenté

Voici le code des valeurs par défaut que l'utilisateur peut avoir envie de voir :

```
1 <*defaults>
2 \mytest{1}
3 </defaults>
```

Et voici du code interne dont l'utilisateur n'a pas grand chose à faire :

```
4 <*package>
5 \ProvidesPackage{rearrange-fr}
6   [2005/07/15 v0.1 docstrip: rearranging]
7 \def\mytest#1{\def\test{‘#1’}}
8 </package>
```

13 Fichiers produits verbatim

13.1 *rearrange-fr.sty*

```
\ProvidesPackage{rearrange-fr}
  [2005/07/15 v0.1 docstrip: rearranging]
\def\mytest#1{\def\test{‘#1’}}
\mytest{1}
```

13.2 *rearrange-fr.ins*

```
\input docstrip.tex
\keepsilent\askforoverwritefalse
\nopreamble\nopostamble
\generate{\file{\jobname.sty}{
  \from{\jobname.dtx}{package}
  \from{\jobname.dtx}{defaults}
}}
\endbatchfile
```

QUATRIÈME PARTIE — LA GALLERIE DTX

Ce fichier-ci, `dtxgallery.dtx`, est un exemple de fichier DTX qui ne contient aucun code propre mais charge *d'autres* fichiers DTX.

On notera que l'on pourrait inclure du code dans ce fichier comme dans tout autre fichier DTX et qu'il serait composé avec les exemples qui suivent mais j'ai choisi d'alléger ce fichier DTX particulier aussi fortement que possible dans le seul but de montrer que c'est possible. Aussi, il ne produit aucun fichier, et ne fournit pas de code. Il ne sert qu'à documenter les autres fichiers DTX.

On notera que ce fichier met en relief les différences entre les deux endroits où l'on peut placer des « sources de documentation » dans un fichier DTX :

```
example.dtx
% \iffalse
% ...
%<*driver>
\documentclass{ltxdoc}
...
\begin{document}
  <Source de la documentation — spécifique>
  \DocInput{example.dtx}
\end{document}
%</driver>
% \fi
% <Source de la documentation — générale>
% ...
```

Le source de documentation « spécifique » est utilisé quand on ne compile que ce fichier DTX-ci alors que le source de documentation « général » apparaît *seul* quand il est incorporé par une commande `\DocInclude`. Dans le cas d'une compilation de fichier DTX — et c'est ce que montre ce document-ci —, `\DocInclude` fait référence à un hypothétique fichier `example.dtx` complètement différent ce qui fait que la partie spécifique du code est invisible.

Aussi, il vaut mieux définir le formatage et les métadonnées du document — telles que `\title` et `\maketitle` — à l'endroit « spécifique » et n'inclure que le corps de la documentation dans la partie « générale ».

Ce fichier donne un exemple de ce que le formatage de code dans la partie « spécifique », ici, change les fontes et la mise en page qui ne sont utilisées que dans la galerie DTX elle-même.

Table des matières

1	Introduction	1
2	Fichiers de la galerie	2
3	Utilisation de cette galerie	2
4	Interlude : éliminer les sentinelles	2
I	Extraire le fichier ins du dtx	4
5	Code source documenté	4
6	Contenu de single-source-fr.sty	4
7	Contenu de single-source-fr.ins	4
8	Contenu de single-source-lisezmoi. txt	4
II	Extraction conditionnelle de code	5
9	Inclusion conditionnelle de code	5
10	Fichiers produits verbatim	5
III	Réordonnement du code	6
11	Exemple de réordonnement du source docstrip	6
12	Code documenté	6
13	Fichiers produits verbatim	6
IV	La galerie dtx	7