Le package xr[*]

David Carlisle[†]
(traduction française: Jean-Pierre Drucbert
Jean-Pierre.Drucbert@onecert.fr)

1994/05/28

Ce package permet de gérer des Références eXternes

Si un document se réfère à certaines sections d'un autre, par exemple `aaa.tex`, alors charger ce package dans le fichier principal et placer la commande
`\externaldocument{aaa}`
dans le préambule.

Vous pouvez alors utiliser `\ref` et `\pageref` pour vous référer à tout ce qui a été défini par `\label` dans chaque `aaa.tex` ou dans le document principal. Il est possible de déclarer autant de documents externes que l'on veut.

Si l'un des documents externes ou le document principal utilise le même `\label`, cette étiquette (label) étant définie plusieurs fois, une erreur survient. Afin de contourner ce problème, la commande `\externaldocument` a un argument optionnel. Si vous déclarez

```
\externaldocument[A-]{aaa}
```

alors toutes les références dans le document `aaa` sont préfixées par `A-`. Par exemple, si une section de `aaa` contient `\label{intro}`, alors ce label pourra être référencé par `\ref{A-intro}`. Le préfixe peut être différent de `A-`; il peut être toute chaîne choisie pour que tous les étiquettes importées depuis des fichiers externes soient uniques. Notez cependant que si vos packages déclarent certains caractères actifs (`:` en français, `"` en allemand, alors ils ne peuvent pas en général être utilisés dans l'argument de `\label` ni dans l'argument optionnel de `\externaldocument`.

1 Les macros (section non traduite)

⟨*package⟩

Check for the optional argument.

```
\def\externaldocument{\@ifnextchar[\XR@{\XR@[]}}
```

Save the optional prefix. Start processing the first `aux` file.

```
\def\XR@[#1]#2{{%
  \makeatletter
  \def\XR@prefix{#1}%
  \XR@next#2.aux\relax\\}}
```

[*] Cette version a comme numéro v5.02, dernière revision 1994/05/28.
[†] L'auteur des versions 1 à 4 est Jean-Pierre Drucbert (Jean-Pierre.Drucbert@onecert.fr)

Process the next **aux** file in the list and remove it from the head of the list of files to process.

```
\def\XR@next#1\relax#2\\{%
  \edef\XR@list{#2}%
  \XR@loop{#1}}
```

Check whether the list of **aux** files is empty.

```
\def\XR@aux{%
  \ifx\XR@list\@empty\else\expandafter\XR@explist\fi}
```

Expand the list of aux files, and call `\XR@next` to process the first one.

```
\def\XR@explist{\expandafter\XR@next\XR@list\\}
```

If the **aux** file exists, loop through line by line, looking for `\newlabel` and `\@input`. Otherwise process the next file in the list.

```
\def\XR@loop#1{\openin\@inputcheck#1\relax
  \ifeof\@inputcheck
    \PackageWarning{xr}{^^JNo file #1^^JLABELS NOT IMPORTED.^^J}%
    \expandafter\XR@aux
  \else
    \PackageInfo{xr}{IMPORTING LABELS FROM #1}%
    \expandafter\XR@read\fi}
```

Read the next line of the aux file.

```
\def\XR@read{%
  \read\@inputcheck to\XR@line
```

The `...` make sure that `\XR@test` always has sufficient arguments.

```
  \expandafter\XR@test\XR@line...\XR@}
```

Look at the first token of the line. If it is `\newlabel`, do the `\newlabel`. If it is `\@input`, add the filename to the list of files to process. Otherwise ignore. Go around the loop if not at end of file. Finally process the next file in the list.

```
\long\def\XR@test#1#2#3#4\XR@{%
  \ifx#1\newlabel
    \newlabel{\XR@prefix#2}{#3}%
  \else\ifx#1\@input
     \edef\XR@list{\XR@list#2\relax}%
  \fi\fi
  \ifeof\@inputcheck\expandafter\XR@aux
  \else\expandafter\XR@read\fi}
```

```
⟨/package⟩
```