

The `xr` package*

David Carlisle[†]
carlisle@cs.man.ac.uk

1994/05/28

This package implements a system for eXternal References.

If one document needs to refer to sections of another, say `aaa.tex`, then this package may be loaded in the main file, and the command

`\externaldocument{aaa}`

given in the preamble.

Then you may use `\ref` and `\pageref` to refer to anything which has been given a `\label` in either `aaa.tex` or the main document. You may declare any number of such external documents.

If any of the external documents, or the main document, use the same `\label` then an error will occur as the label will be multiply defined. To overcome this problem `\externaldocument` has an optional argument. If you declare `\externaldocument[A-]{aaa}` Then all references from `aaa` are prefixed by `A-`. So for instance, if a section of `aaa` had `\label{intro}`, then this could be referenced with `\ref{A-intro}`. The prefix need not be `A-`, it can be any string chosen to ensure that all the labels imported from external files are unique. Note however that if your style declares certain active characters (`:` in French, `"` in German) then these characters can not usually be used in `\label`, and similarly may not be used in the optional argument to `\externaldocument`.

1 The macros

```
1 \langle*package\rangle
```

Check for the optional argument.

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

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

```
3 \def\XR@[#1]#2{\%
```

```
4 \makeatletter
```

```
5 \def\XR@prefix{#1}%
```

```
6 \XR@next#2.aux\relax\}}
```

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

```
7 \def\XR@next#1\relax#2\{\%
```

```
8 \edef\XR@list{#2}%
```

```
9 \XR@loop{#1}}
```

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

```
10 \def\XR@aUX{\%
```

```
11 \ifx\XR@list\@empty\else\expandafter\XR@explist\fi}
```

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

```
12 \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.

```
13 \def\XR@loop#1{\openin\@inputcheck#1\relax
```

*This file has version number v5.02, last revised 1994/05/28.

[†]The Author of Versions 1–4 was Jean-Pierre Drucbert

```

14 \ifeof\@inputcheck
15   \PackageWarning{xr}{^^JNo file #1^^JLABELS NOT IMPORTED.^^J}%
16   \expandafter\XR@aux
17 \else
18   \PackageInfo{xr}{IMPORTING LABELS FROM #1}%
19   \expandafter\XR@read\fi}

```

Read the next line of the aux file.

```

20 \def\XR@read{%
21   \read\@inputcheck to\XR@line

```

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

```

22   \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.

```

23 \long\def\XR@test#1#2#3#4\XR@{%
24   \ifx#1\newlabel
25     \newlabel{\XR@prefix#2}{#3}%
26   \else\ifx#1\@input
27     \edef\XR@list{\XR@list#2\relax}%
28   \fi\fi
29   \ifeof\@inputcheck\expandafter\XR@aux
30   \else\expandafter\XR@read\fi}
31 \</package>

```