The xr-hyper package*

David Carlisle

2026-10-09

This package implements a system for eXternal References.

It is an extension of the xr package. It was developed to support the extended label syntax of the hyperref package and to enable active links to the external documents.

In the LaTeX release 2023-06-01 the label syntax of hyperref and the LaTeX kernel have been synchronized and there is no longer a need for two packages. xr-hyper already works with all documents – it is not required to load hyperref – and its code will move in the next LaTeX release into the xr package. Then xr-hyper can be deprecated.

1 Usage

`\externaldocument[`⟨*prefix*⟩`][nocite]{`⟨*document*⟩`}[`⟨*url*⟩`]`

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` (or `\nameref` if the package nameref has been loaded 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 ⟨*prefix*⟩. 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 the prefix is expanded and so should not contain commands that are not safe in this context.

As first suggested in Enrico Gregorio's `xcite` package, the current version also allows `\cite` to reference `\bibitem` in the external document. For compatibility with `xcite`, `\externalcitedocument` is made available as an alias for `\externaldocument`

Many packages have variant citation commands (natbib, biblatex,....) and the external document may or may not have used hyperref. Because of these differences the citation linking may not always work, it can be disabled by specifying `[nocite]` after the ⟨*prefix*⟩:

```
\externaldocument[][nocite]{aaa}
```

*This file has version number v7.01s, last revised 2026-10-09.

The 'document' referred to by the main argument ⟨*document*⟩ is the file `document.aux` which must be somewhere on TeX's input path. Some packages (eg hyperref) really need to know the location of the final document rather than the aux file. By default this is assumed to be `document.pdf`. A package may redefine the command `\XR@ext` to change this default extension. However sometimes the final document may be in a position unrelated to the aux file, or the browser may not be able to find files at an arbitrary point in TeX's input path, so the final optional argument ⟨*url*⟩ allows a full URL to the final document to be specified.

```
\externaldocument{aaa}[http://here.xxx.edu/this/path/to/aaa.pdf]
```

The package stores the url of the external document in the label data. If can e.g. be retrieved with the refcount package

```
\usepackage{refcount,xr-hyper}
\externaldocument{aaa}
...
\getrefbykeydefault{intro}{url}{??} %prints aaa.pdf or ??
```

xr-hyper supports also the properties introduced in LaTeX 2023-11-01. Here the url of the external document is stored in the `xr-url` property.

```
\usepackage{xr-hyper}
\externaldocument{aaa} %aaa contains \RecordProperties{intro}{page}
...
\RefProperty{intro}{page}   %gives page number
\RefProperty{intro}{xr-url} %gives aaa.pdf
```

2 The macros

1 ⟨∗package⟩

Check for the optional argument.

```
2 \def\externaldocument{\@testopt\XR@cite{}}
3 \let\externalcitedocument\externaldocument
4 \def\XR@cite[#1]{\@testopt{\XR@[#1]}{}}
5 \def\XR@[#1][#2]#3{\@testopt{\XR@@{#1}{#2}{#3}}{#3.\XR@ext}}
```

2.1 helper definitions

To test the second optional argument

```
6 \def\XR@@nocite{nocite}
```

Needed in the processing

```
7 \long\def\@gobblefour  #1#2#3#4{}
8 \long\def\@firstoffour #1#2#3#4{#1}
9 \long\def\@secondoffour#1#2#3#4{#2}
10 \long\def\@thirdoffour #1#2#3#4{#3}
11 \long\def\@fourthoffour #1#2#3#4{#4}
```

The url is added as fifth argument. The command used here is `\XR@addURL`. The command is more complicated as needed as it tries to handle also older documents with `\newlabel`'s with two arguments.

```
12 \def\XR@addURL#1{\XR@@dURL#1{}{}{}{}\\}
```

```
\def\XR@@dURL#1#2#3#4#5\\{%
     \unexpanded{{#1}{#2}{#3}{#4}}{\XR@URL}%
  }%
```

2.2 Variables

Default file extension:

```
\providecommand\XR@ext{pdf}
```

2.3 Processing

Save the optional prefix. Start processing the first `aux` file. Version beta2 also added another improvement unrelated to the hyperref support. Olivier Michel pointed out that if the aux file was not on texinputs you could not always go `\externaldocument`/some/path/to/file specifically that worked if file.aux was a 'simple' document with one aux file, but if `\include` had been used, the 'sub' aux files would not be found by xr in the remote directory. This version calls `\filename@parse` to get the directory name of the remote directory, which is then explicitly prepended to the names of any included aux files.

```
\def\XR@@#1#2#3[#4]{{%
  \makeatletter
  \def\XR@prefix{#1}%
   \def\XR@nocite{#2}%
   \ifx\XR@nocite\XR@@nocite
     \let\XR@bibcite\vadjust
   \else
     \let\XR@bibcite\bibcite
   \fi
  \def\XR@URL{#4}%
  \set@curr@file{#3}%
  \filename@parse\@curr@file
  \XR@next\@curr@file.aux\relax\\}}
```

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`, define `\r@`⟨*label*⟩, ensure that it has five label data argument and add the url as the last one. If it is `\@input`, add the filename to the list of files to process. If it is `\bibcite`, call a `\bibcite`. If it is `\new@label@record` add the url and then call it. Otherwise ignore. Go around the loop if not at end of file. Finally process the next file in the list. Make sure the arguments are handled outside the `\ifx` test,

```
\long\def\XR@test#1#2#3#4\XR@{%
  \let\XR@tempa\@gobblefour
  \ifx#1\newlabel
    \let\XR@tempa\@firstoffour
  \else\ifx#1\XR@bibcite
    \let\XR@tempa\@secondoffour
  \else\ifx#1\@input
     \let\XR@tempa\@thirdoffour
  \else\ifx#1\new@label@record
     \let\XR@tempa\@fourthoffour
  \fi\fi\fi\fi
   \XR@tempa
    {%
     \expandafter\protected@xdef\csname r@\XR@prefix#2\endcsname{\XR@addURL{#3}}%
    }%
    {\expandafter\bibcite\expandafter{\XR@prefix#2}{#3}}%
    {\edef\XR@list{\XR@list\filename@area#2\relax}}%
    {%
      \edef\next{\noexpand\new@label@record{\XR@prefix#2}{\unexpanded{#3}{xr-url}{\XR@URL}}}%
      \next
    }
  \ifeof\@inputcheck\expandafter\XR@aux
  \else\expandafter\XR@read\fi}
```

⟨/package⟩