

ABSTRACT

The package ccref provides a command `\ccref` parallel to cleveref's `\cref` for handling definite articles properly (especially for the article contractions in some European languages).

/ 1 /

The motivation

- 1 By default, with cleveref's `\cref` to reference theorem-like environments, the names do not
2 contain definite articles. While this might be acceptable for English, it is certainly not good
3 enough for languages such as French, Italian, Portuguese, Spanish, etc. – in these cases there
4 shall be grammatical errors and would give you a strong feeling that it is machine-generated.
5 However, even if we manually add the definite articles to the names, there would still be
6 other problems. As an example, if we define the French names to be:

```
\crefname{theorem}{le théorème}{les théorèmes}  
\crefname{proposition}{la proposition}{les propositions}
```

- 7 then when one writes (which means “*We can deduce this from ...*”)

```
On peut le déduire de \cref{thm1,thm2,prop3}.
```

- 8 the result would be:

On peut le déduire **de les** théorèmes 1 et 2 et **la** proposition 3.

- 9 which is wrong, as the correct result should be:

On peut le déduire **des** théorèmes 1 et 2 et **de la** proposition 3.

- 10 `\cref` cannot handle such cases automatically — and that is when `\ccref` comes into play.

/ 2 /

The usage

2.1 | How to load it?

- 11 Simply load the package with:

```
\usepackage{ccref}
```

TIP

- Since `ccref` uses `cleveref` internally, it should usually be placed at the last of your preamble, and notably, after `varioref` and `hyperref`.
- To handle article contractions correctly, `\ccref` shall detect the current language, thus you need to use packages such as `babel` or `polyglossia` to set your languages, and use commands like `\selectlanguage` to select them appropriately.

2.2 | How to use it ?

Then you can use the command `\ccref` as follows:

- `\ccref[\langle prep \rangle]{\langle labels \rangle}`
 - This will pass the preposition `\langle prep \rangle` to the definite articles that follows. Its behavior depends on the current language (for example, in Spanish, `\langle prep \rangle` is passed only to the first definite article, while in French it is passed to everyone).
- `\ccref-[\langle prep \rangle]{\langle labels \rangle}` and `\ccref+[\langle prep \rangle]{\langle labels \rangle}`
 - In case the automatic version does not meet your needs, here are two manual ones. The `-` version passes the preposition `\langle prep \rangle` only to the first definite article, while the `+` version passes `\langle prep \rangle` to every definite article.

TIP

There is also a starred version `\ccref*` for generating the same referencing text but without creating hyperlinks.

However, before using it, you should first define the `\crefname` carefully. The definite article in `\crefname` needs to be marked manually using `\ccmarkart`, for example:

```
\crefname{theorem}{\ccmarkart{1e} théorème}{\ccmarkart{les} théorèmes}
```

/ 3 /

Example

Let us come back to the example at the beginning, now you can do this:

```
\crefname{theorem}{\ccmarkart{1e} théorème}
{\ccmarkart{les} théorèmes}
\crefname{proposition}{\ccmarkart{1a} proposition}
{\ccmarkart{les} propositions}
```

And the sentence shall be written as:

On peut le déduire `\ccref[de]{thm1,thm2,prop3}`.

which would result in (provided that you have done `\selectlanguage{french}`):

On peut le déduire **des** théorèmes 1 et 2 et **de la** proposition 3.

Voilà !

Known issues

- 1 • `ccref` currently only works for French, Italian, Portuguese (both European and Brazilian)
2 and Spanish, certainly more would be added to this list.
- 3 • The current mechanism does not work for German. However, the author has planed to
4 adopt a more refined approach in later versions in order to support the various situations
5 in German.
- 6 • In case that the initial letter of $\langle prep \rangle$ is capitalized, `ccref` cannot yet handle the case changes
7 automatically. However, this should be a rare occurrence.
- 8 • The names of theorem-like environments are not provided — for the moment you need
9 to define them all by yourself. However, users are encouraged to use the [ProjLib](#) toolkit,
10 which already handles everything for you.