The `cleveref-usedon` package *

Sven Pistre
sven.pistre@gmail.com

2023/03/29

Abstract

This package adds "forward-referencing" to the `cleveref` package. Any label can be referenced with the new optional argument "`UsedOn`" passed to `\cref`. Doing so, will print an info message at the original label location (in a theorem environment, say) which reads "*Used on pages ⟨pagerange⟩.*". This functionality is complementary to `hyperref`'s `pagebackref` or `biblatex`'s `backref` option for the bibliography. It might be useful for authors of longer texts such as textbooks or theses, where a lot of supplementary results and information are given in early chapters, appendices or exercises. The message on which pages these results will be used can be a helpful information for the reader of the final text. Additionally, a bug in `cleveref v0.21.4` is patched.

1 Introduction

Imagine you are reading a long mathematical text such as a text book or a thesis. There are plenty of supplementary lemmas, propositions, theorems and/or exercises throughout the whole text. You ask yourself "Gosh, while Lemma 1.12 is certainly an interesting result *where* is this result used later on in this long text? I really would find that helpful to decide *why* I should read the proof." You can, of course, use the PDF search function of your viewer to look up the string "Lemma 1.12" but wouldn't it be more helpful if Lemma 1.12 already indicates all or at least its most useful/crucial applications via an info message?

This is what the package `cleveref-usedon` tries to address. The info message "*Used on p. 40, 43-45 and 101.*" would then be printed to the header of Lemma 1.12. For example, we have given the following theorem the label

```
\label{thm:SqrtTwoIrrational}.
```

Theorem 1.1 (Used on pages 1 and 3.)
The number $\sqrt{2}$ is irrational.

Now we can reference this theorem via

```
\cref[UsedOn]{thm:SqrtTwoIrrational}:
```

A proof of Theorem 1.1 can be traced back to Euclid.

*This document corresponds to cleveref-usedon v0.1.0, dated 2023/03/29.

Let's clear the page of this PDF, so that we can see the effects of referencing Theorem 1.1 without the optional argument `[UsedOn]`, i.e.

```
\cref{thm:SqrtTwoIrrational}.
```

Note that the current page number 2 is not included in the list of page references in the header of Theorem 1.1.

2 Usage

The `cleveref-usedon` package uses `cleveref v0.21.4` as its base. To freely cite from the `cleveref` documentation:
The `cleveref-usedon` package is loaded in the usual way, by putting the line

```
\usepackage{cleveref-usedon}
```

in your document's preamble. However, care must be taken when using `cleveref` in conjunction with other packages that modify LaTeX's referencing system (see Section 13 of `cleveref`'s documentation). Basically, `cleveref-usedon` must be loaded *last* but definitely AFTER `hyperref`.

`\cref` `\cref[`⟨*UsedOn*⟩`]{`⟨*LabelName*⟩`}`
`\Cref` The `\cref` macro can be called with option `UsedOn` or the short form `uo`. This is case-insensitive, i.e. you could also write

```
\cref[UsEdOn]{⟨LabelName⟩}
\cref[uO]{⟨LabelName⟩}
```

— but why would you?

This additional option adds the text "*(Used on page(s) ... .)*" with an additional line break right after where the label has been originally set. If `hyperref` has been loaded, there will also be hyperlinks to the corresponding pages from where the label has been referenced.

If the original label has been set in a theorem-style environment such as

```
\begin{theorem} \label{thm:SqrtTwoIrrational}
    The number $\sqrt{2}$ is irrational.
\end{theorem}
```

then the info message is printed in the header of this theorem-style environment. The same functionality can be used for `\Cref`.

The package `cleveref-usedon` is implemented using `expl3`. If you are interested, I have spent some time to document and comment on the implementation in Section 6.

3 Hints and tips

If you use the `capitalise` option for `cleveref`, you might want to revert this capitalisation for page references for more visual appeal by putting

```
\crefname{page}{page}{pages}
```

in your document's preamble, after loading `cleveref-usedon`.

It is recommended to not use the optional argument for equation-style environments such as Eq. (1) because otherwise the info message will — unhelpfully — be printed inside the equation environment, like so:

$$\int_M \mathrm{d}\omega = \int_{\partial M} \omega. \textit{(Used on page 3.)} \tag{1}$$

So, for now, one should use this functionality only for theorem-style environments such as theorems, lemmas and maybe exercises.

3.1 Editing the info message

`\UsedOnMessage` `\UsedOnMessage{`⟨*PageNumberList from cpageref*⟩`}`
The standard message which gets printed to the first line of the labelled environment is "*(Used on* ⟨*PageNumberList*⟩*).*" — followed by a line break — where ⟨*PageNumberList*⟩ is generated automatically by `cleveref` via `\cpageref`. You can change this behaviour by redefining the macro `\UsedOnMessage`, e.g. as

```
\RenewDocumentCommand{\UsedOnMessage}{m}{
    \emph{(Will be cited on #1.)} \\
}
```

4 Interaction with other packages

All interactions with other packages mentioned in Section 13 of `cleveref`'s documentation also apply to `cleveref-usedon`. In fact (if `cleveref-usedon` is loaded last), `ntheorem`'s `\thref` and `varioref`'s `\vref` also obtain the additional `UsedOn` functionality because `cleveref` redefines these macros to be aliases for `\cref`.

5 Future features

It is planned to include a package option that turns on the `UsedOn` option for *all* `\cref`'s calls. Additionally, a switch package option might be included which reverses the standard behaviour, i.e. if one does not want to use `UsedOn` functionality one needs to explicitly use `\cref[NotUsedOn]{`⟨*LabelName*⟩`}`.
Let's just reference Theorem 1.1 one last time for the fun of it, check page 1 again to see the effect to the reference list in the header of Theorem 1.1.

6 Implementation

6.1 Options and requirements

The following package options currently don't do anything.

```
\bool_new:N \g_StandardBehaviour_bool
\bool_gset_true:N \g_StandardBehaviour_bool
\DeclareOption{usedon}{
    \bool_gset_true:N \g_StandardBehaviour_bool
}
\DeclareOption{notusedon}{
```

```
    \bool_gset_false:N \g_StandardBehaviour_bool
}
```

All other package options get passed on to `cleveref`.

```
\DeclareOption*{
    \PackageInfo{cleveref-usedon}
        {Passing~to~cleveref:~Option~`\CurrentOption'}
    \PassOptionsToPackage{\CurrentOption}{cleveref}
}
\ProcessOptions*
\RequirePackage{cleveref}[2018/03/27]
\RequirePackage{xparse}
```

6.2 Patches of known bugs to `cleveref`

The following fixes the range bug for `\cpageref` in `cleveref v0.21.4`
See https://tex.stackexchange.com/a/620066/267438

```
\newcommand*{\@setcpagerefrange}[3]{%
    \@@setcpagerefrange{#1}{#2}{cref}{#3}}
\newcommand*{\@setCpagerefrange}[3]{%
    \@@setcpagerefrange{#1}{#2}{Cref}{#3}}
\newcommand*{\@setlabelcpagerefrange}[3]{%
    \@@setcpagerefrange{#1}{#2}{labelcref}{#3}}
```

6.3 Overloading of label and cref

We need a branching variant of `\str_case:nn` which expands the input string token. This will be used to match options for the `\@UsedOn@Processor`.

```
\prg_generate_conditional_variant:Nnn \str_case:nn { x } { TF }
```

`\g_UsedOn_k_seq` Let's initialise a global key sequence for those label names that have been referenced via `[UsedOn]`.

```
\seq_new:N \g_UsedOn_k_seq
```

`\g_UsedOn_kv_prop` And we'll also create a global key-value property list with label names as keys and the maximal amount of times they have been referenced via `[UsedOn]` as values (possibly known from the last pdflatex run).

```
\prop_new:N \g_UsedOn_kv_prop
```

`\UsedOnMessage` The following is the standard text that gets printed in the first line of the labelled environment which later gets referenced with `[UsedOn]`.

```
\NewDocumentCommand{\UsedOnMessage}{m}{
    \emph{(Used~on~#1.)} \\
}
```

`\@UsedOn@PrintUsedOnLabel` Given a ⟨*LabelName*⟩, the following command records all references via `[UsedOn]` of this label in a temporary comma-separated list (a `clist` in `expl3` speak). This `clist` is then passed to `cleveref`'s `cpageref` and which in turn is passed to `\UsedOnMessage` to be printed after the original label.

```
\NewDocumentCommand{\@UsedOn@PrintUsedOnLabel}{ m }{%
```

First, we will check if the reference `UsedOn@`⟨*LabelName*⟩`@1` exists. Here, the `@1` means that ⟨*LabelName*⟩ has been referenced with option `[UsedOn]` at least once. If this reference does not exist, nothing happens.

```
    \cs_if_exist:cT {r@UsedOn@#1@1}
    {
```

Next, we store all the references of the form `UsedOn@`⟨*LabelName*⟩`@`⟨*Number*⟩ in a temporary comma-separated list (`clist`). We do this by looping from 1 to the value of `LastRun@UsedOn@`⟨*LabelName*⟩ (if the latter value exists, otherwise we set it to 1). Initially, this will need two consecutive runs of pdflatex.

```
        \cs_if_free:cTF {c@LastRun@UsedOn@#1}
            { \int_set:Nn \l_tmpa_int { 1 } }
            { \int_set:Nn \l_tmpa_int { \value{LastRun@UsedOn@#1} } }
        \int_set:Nn \l_tmpb_int { 1 }
        \int_while_do:nn { \l_tmpb_int <= \l_tmpa_int }
        {
            \clist_put_right:Nx \l_tmpa_clist { UsedOn@#1@\int_use:N \l_tmpb_int }
            \int_incr:N \l_tmpb_int
        }
```

Finally, we print the message that was set in the macro `\UsedOnMessage`.

```
        \UsedOnMessage{\cpageref{\l_tmpa_clist}}
    }
}%
```

`\@UsedOn@Processor` This macro takes an optional argument (a case-insensitive version of `[UsedOn]` or the shortform `[uo]`) and a mandatory argument (a single `{`⟨*LabelName*⟩`}` or a `clist` {⟨*LabelName1*⟩,⟨*LabelName2*⟩,...}).

```
\NewDocumentCommand{\@UsedOn@Processor}{ o m }{%
    \IfValueT{#1}{
```

First, we check if the option `[UsedOn]` or `[uo]` (case-insensitive) was used.

```
        \str_case:xnTF { \str_foldcase:n { #1 } }
        {
            {usedon} {}
            {uo} {}
        }
        {
            {
```

Loop through the (potential) label list in mandatory argument of `\cref` (or `\Cref`) which gets passed as the mandatory argument of the current macro.

```
                \seq_set_from_clist:Nn \l_tmpa_seq {#2}
                \seq_map_inline:Nn \l_tmpa_seq
                {
```

If the label has *not* been referenced yet via `[UsedOn]`, create a counter for the current run `ThisRun@UsedOn@##1`. If we are not in the initial run anymore, there should be a counter `LastRun@UsedOn@##1` which contains the maximal amount this specific label has been referenced via `UsedOn`. If we are in the initial run, we need to create this counter as well. Then save the label in the global container `\g_UsedOn_k_seq`.

```
                    \seq_if_in:NxF \g_UsedOn_k_seq {UsedOn@##1}
                    {
```

```
                    \newcounter{ThisRun@UsedOn@##1}
                    \cs_if_free:cT {c@LastRun@UsedOn@##1}
                        { \newcounter{LastRun@UsedOn@##1} }
                    \seq_gput_right:Nx \g_UsedOn_k_seq {UsedOn@##1}
                }
```

Increase the counter for the current run by 1 and set the counter for last run (containing the maximal amount of `UsedOn-\cref`'s) to...the maximal amount of `UsedOn-\cref`'s.

```
                \stepcounter{ThisRun@UsedOn@##1}
                \setcounter{LastRun@UsedOn@##1}{%
                    \fp_eval:n { max(%
                        \value{ThisRun@UsedOn@##1},%
                        \value{LastRun@UsedOn@##1} ) }%
                }
```

Store the value of the max counter `LastRun@UsedOn@##1` in the global container `\g_UsedOn_kv_prop`.

```
                \prop_gput:Nxx \g_UsedOn_kv_prop
                    {UsedOn@##1} {\arabic{LastRun@UsedOn@##1}}
```

Now we create a numbered auxiliary label. This label is issued at the location where we referenced the original label via `\cref[UsedOn]`⟨*LabelName*⟩. The new auxiliary label has the prefix `UsedOn@` and the suffix `@\arabic{ThisRun@UsedOn@##1}`, e.g. `UsedOn@thm:Pythagoras@4` if it is the fourth time that we called `\cref[UsedOn]{thm:Pythagoras}`.

```
                \origlabel{UsedOn@##1@\arabic{ThisRun@UsedOn@##1}}
            }
        }
    }
    {
```

Throw an error, if an unrecognised option was used for the optional argument to this macro.

```
        \msg_new:nnn {cleveref-usedon} { OptionSpellingError }
            {    \\
                Spelling~error~\msg_line_context:
                \\
                Did~you~mean~to~pass~option\\
                'UsedOn'~to~cref~or~Cref?
            }
        \msg_fatal:nn { cleveref-usedon } { OptionSpellingError }
    }
}
}%
```

`\@UsedOn@cref` This is just a wrapper around `cleveref`'s `\cref`. Additionally the `\@UsedOn@Processor` gets called.

```
\NewDocumentCommand{\@UsedOn@cref}{ s o m }{%
    \IfBooleanTF{#1}{ \origcref*{#3} }{ \origcref{#3} }%
    \@UsedOn@Processor[#2]{#3}
}%
```

`\@UsedOn@Cref` This is just a wrapper around `cleveref`'s `\Cref`. Additionally the `\@UsedOn@Processor` gets called.

```
\NewDocumentCommand{\@UsedOn@Cref}{ s o m }{%
    \IfBooleanTF{#1}{ \origCref*{#3} }{ \origCref{#3} }%
    \@UsedOn@Processor[#2]{#3}
}%
```

`\@UsedOn@ReadFromAux` From the .aux file we will read the contents of the global container `\g_UsedOn_kv_prop`. This is a key-value property list and we create and set a for each label (key) and the maximal amount (value) it was called in the last run.

```
\NewDocumentCommand{\@UsedOn@ReadFromAux}{ }{%
    \prop_map_inline:Nn \g_UsedOn_kv_prop
    {
        \newcounter{LastRun@##1}
        \setcounter{LastRun@##1}{##2}
    }
}%
```

`\@UsedOn@WriteToAux` For each label we write a line in the .aux file of the form:
⟨*LabelName*⟩ == ⟨*Maximal references via UsedOn in last run*⟩.
This information can be constructed from the global container `\g_UsedOn_k_seq` and the counters with prefix `ThisRun@` we set earlier. We need to wrap this in the on/off switch for `expl3` functionality.

```
\NewDocumentCommand{\@UsedOn@WriteToAux}{ }{%
```

First, we clear the global key-value prop list `\g_UsedOn_kv_prop` and then we rebuild it with the information from the current run.

```
    \prop_clear:N \g_UsedOn_kv_prop
    \seq_map_inline:Nn \g_UsedOn_k_seq
        { \prop_gput:Nxx \g_UsedOn_kv_prop {##1}{\arabic{ThisRun@##1}} }
    \iow_now:cx { @auxout }
        { \token_to_str:N \ExplSyntaxOn }
```

Loop through the key-val `proplist` and write contents to .aux file.

```
    \prop_map_inline:Nn \g_UsedOn_kv_prop
    {
        \iow_now:cx { @auxout }
            { \prop_gput_from_keyval:Nn \token_to_str:N \g_UsedOn_kv_prop {##1=##2} }
    }
    \iow_now:cx { @auxout }
        { \token_to_str:N \ExplSyntaxOff }
}%
```

At the hook `\AtBeginDocument` we read from the .aux file and patch commands.

```
\AtBeginDocument{%
    \@UsedOn@ReadFromAux
```

Patch label and cref to include the new `[UsedOn]` capabilities.

```
    \NewCommandCopy{\origlabel}{\label}
    \NewCommandCopy{\origcref}{\cref}
    \NewCommandCopy{\origCref}{\Cref}
    \RenewDocumentCommand{\label}{ m }{%
        \origlabel{#1}\@UsedOn@PrintUsedOnLabel{#1}
    }%
```

```
    \RenewCommandCopy{\cref}{\@UsedOn@cref}
    \RenewCommandCopy{\Cref}{\@UsedOn@Cref}
}%
```

At the hook `\AtEndDocument` we write to the .aux file.

```
\AtEndDocument{%
    \@UsedOn@WriteToAux
}%
```