

Liú Hǎiyáng (Leo Liu)
leoliu.pku@gmail.com

2014/04/06

1 Introduction

When preparing index, the macros `\see` and `\seealso` define by $\text{\LaTeX 2}_{\epsilon}$'s standard package `makeidx` are handy to use. They are used as special page format command, to present cross reference of index entry. For example, if we use

```
\index{math|see{mathematics}}
```

to reference the index entry “mathematics”, we will get

math, *see* mathematics

in the output index.

However, `\see` and `\seealso` commands in $\text{\LaTeX 2}_{\epsilon}$ cannot produce page numbers. Moreover, if we use multiple `\see` or `\seealso`'s for one entry, there will be multiple reference targets produced. Therefore we may get this unexpected output:

math, *see* mathematics, *see* mathematics, *see* mathematics

if we have used `\index{math|see{mathematics}}` at 3 different places in the document.

The **seealso** package solve the problem. If we use

```
\usepackage{seealso}
```

all cross references of index entry will produce page numbers, and the reference targets, which we call *see list*, will be merged. The previous example will produce an output like this:

math, 2, 4, 5, *see* mathematics

2 Reference

2.1 Package Loading

The basic usage of the package is to simply load the package

```
\usepackage{seealso}
```

then `\see` and `\seealso` commands will be redefined to support page number output and reference target merging.

The **seealso** package should be loaded after `makeidx`, `imakeidx`, etc. Or the redefinition of `\see` and `\seealso` may be broken.

2.2 Package Options

`override` `override` is a boolean option, with a default value `true`. When it is set the original `\see` and `\seealso` will be overridden. However, when it is set to be `false`, say

```
\usepackage[override=false]{seealso}
```

then `\see` and `\seealso` will not be overridden, and `\seepage` and `\seealsopage` should be used for page number output.

`activecr` `activecr` is a boolean option, with a default value `true`. When it is set, the see list will be output at the end of an index entry line (in a `.ind` file). If we set `activecr=false`, however, the see list will not be output automatically, one should use `\SeealsoPrintList` manually at the end of the index entry instead. Usually `\SeealsoPrintList` is added via `delim_t` style of `Makeindex`.

2.3 The `\see` and `\seealso` Commands

`\seepage` `\seepage` and `\seealsopage` provide the main functions of this package. They
`\seealsopage` provide the same faculty of cross reference, but output page numbers and prevent redundancy. For example, suppose that there are

```
\index{foo|seepage{bar}}
```

in page 1, 2, and 3, we will have a `.ind` file as below:

```
foo, \seepage{bar}{1}, \seepage{bar}{2}, \seepage{bar}{3}
```

And the output of the index will be:

```
foo, 1, 2, 3, see mathematics, bar
```

It is possible to use different references for one index entry. For example we can use

```
\index{foo|seepage{bar}}
```

and

```
\index{foo|seepage{foobaz}}
```

several times, the generated `.ind` file may be:

```
foo, \seepage{bar}{1}, \seepage{bar}{2}, \seepage{foobaz}{3}
```

And the output of the index will be:

```
foo, 1, 2, 3, see mathematics, bar, foobaz
```

`\seenopage` The macros `\seenopage` and `\seealsonopage` save the original definitions of `\see`
`\seealsonopage` and `\seealso` before loading `seealso`, which do not produce page numbers.

`\see` When option `override` is `true`, which is the default, `\see` and `\seealso` works
`\seealso` just as `\seepage` and `\seealsopage`.

`\SeealsoPrintList` `\seepage` and `\seealsopage` will collect the see lists, which will be output at the end of the line under default `activecr` option. However, when `activecr` is set to be `false`, the see lists won't be output at the end of source line where `\seepage` and `\seealsopage` are. In the latter case, we can add `\SeealsoPrintList` manually at the end of line to output the see lists. Usually we can set `delim_t` variable in the style file of `Makeindex`, for example:

```
% example.ist
delim_t "\\SeealsoPrintList"
```

Then a `\SeealsoPrintList` will be appended at the end of every non-empty index entry, and the see lists will be output properly.

`\DeclareSeealsoMacro` `\DeclareSeealsoMacro` can be used to define a new macro like `\see` and `\seealso`. The syntax is

```
\DeclareSeealsoMacro\⟨macro⟩{⟨seelist⟩}{⟨name⟩}
```

where `\⟨macro⟩` is the macro name, `⟨seelist⟩` is the new see list, and `⟨name⟩` is the name of the see list to be output (like `\seename`). Using `\DeclareSeealsoMacro`, two macros will be defined, the one is `\⟨macro⟩`, the other is `\⟨macro⟩name`. If there has no `\⟨macro⟩name` predefined, it is defined to be `⟨name⟩`, or the definition remains unchanged.

For example, `\seepage` and `\seealsopage` are defined like this:

```
\DeclareSeealsoMacro\seepage{see}{see}
\DeclareSeealsoMacro\seealsopage{also}{see also}
```

The definitions above also provide the default value of `\seename` and `\alsoname` to be `see` and `see also`, respectively.

2.4 Output Styles

`\seealsosetup` The macro `\seealsosetup` is used to configure the output style of see lists. The syntax is:

```
\seealsosetup[⟨seelists⟩]{⟨kv-options⟩}
```

where `⟨seelists⟩` is a comma separated list, each item of which is a see list to be setup. `⟨seelists⟩` can also be omitted or set to be empty, which means we are setting up the default output style of all kind of see lists. `⟨kv-options⟩` is the options in key-value syntax.

All possible output styles are summarized in table 1.

For example, if we want to set all page numbers produced by `\seealsopage` to be italic, we can use:

```
\seealsosetup[also]{pageformat=\textit{#1}}
```

`\SeealsoGobble` `\SeealsoGobble` takes a character `⟨c⟩` as argument. It checks the following character, gobble it and ignore spaces if it is `⟨c⟩`, or do nothing if it is not `⟨c⟩`. This macro is useful in `pageformat` style to prevent outputting the page numbers, like this:

```
\seealsosetup[see,also]{pageformat=\SeealsoGobble{,}, listsep=}
```

However, `\see` and `\seealso` will be encapsulated by `\hyperindexformat` if we use `hyperref`, then `\SeealsoGobble` will be invalid.

3 Known Issues

Here are some known problems and their solutions:

Key	Meaning	Argument	Default Value
<code>name</code>	The name of see list to be output, e.g. <code>\seename</code>	<code>none</code>	(invalid)
<code>listsep</code>	The separator before a see list	<code>none</code>	<code>,\space</code>
<code>itemsep</code>	The separator between see list items	<code>none</code>	<code>,\space</code>
<code>nameformat</code>	Output format of the see list name	<code>#1</code>	<code>\emph {#1}\space</code>
<code>itemformat</code>	Output format of the see list items	<code>#1</code>	<code>#1</code>
<code>pageformat</code>	Output format of the page numbers	<code>#1</code>	<code>#1</code>

Table 1: Output styles used by `\seealsosetup`

- `Makeindex` wraps the output lines with respect to `line_max` variable. The default value of `line_max` is only 72, therefore the entry line containing `\see` and `\seealso` may be wrapped. However, `activecr` is `true` by default, then package `seealso` will print the see lists at each end of lines, and in the wrapped lines the see lists are output multiple times, which is unexpected.

To solve the problem, it is better to set larger `line_max` in the `.ist` style file to prevent wrapping. Another approach is to set `activecr=false` option, and set `delim_t` to be `\SeealsoPrintList` explicitly to output the see lists.

- Since the output format of page numbers of `\seepage`, `\seealsopage` is the same as normal page numbers, it may be ambiguous if we use normal `\index` and `\index{...|seepage}` simultaneously. Usually we should avoid using different style of `\index` with or without `\seepage` and `\seealsopage`. If it is impossible, we can set special page number format to identify them, or we can use `\SeealsoGobble` to disable page numbers.
- When `hyperref` package is used, the function of avoiding page numbers via `\SeealsoGobble` is invalid. That is because that `\see` and `\seealso` will be encapsulated by `\hyperindexformat` in the `.ind` file. To solve the problem, we can set `hyperindex=false` option of `hyperref` to disable hyperlinks in the index, or some special patching of `hyperref` can be considered.

刘海洋

leoliu.pku@gmail.com

2014/04/06

4 简介

在索引生成时, $\text{\LaTeX} 2_{\epsilon}$ 的标准宏包 `makeidx` 定义了 `\see` 与 `\seealso` 两个宏, 它们通常是在 `\index` 中作为一种特殊的页码格式使用, 表示索引项的引用。例如使用

```
\index{math|see{mathematics}}
```

会使索引项 `math` 引用到 `mathematics`。生成类似

`math, see mathematics`

的索引条目。

不过, $\text{\LaTeX} 2_{\epsilon}$ 的 `\see` 与 `\seealso` 命令并不会为交叉引用的索引项输出页码。更严重的问题是, 如果对一个索引项使用了多次交叉引用, 则会多次输出相同的目标引用。因此可能出现这样的尴尬效果:

`math, see mathematics, see mathematics, see mathematics`

其中可能在三个不同的地方使用了 `\index{math|see{mathematics}}`。

`seealso` 宏包即为解决这种问题而编写。只要使用了

```
\usepackage{seealso}
```

则交叉引用的索引项会同时显示页码, 同时多个交叉引用的目标 (我们称为参见列表) 会合并显示。前面的例子就可能会输出:

`math, 2, 4, 5, see mathematics, bar, foobaz`

这样的格式。

5 参考手册

5.1 宏包载入

基本的用法就是直接使用

```
\usepackage{seealso}
```

这样 `\see` 和 `\seealso` 命令就会被重定义为支持输出页码和合并目标的格式。

注意 `seealso` 宏包的载入应该晚于 `makeidx`、`imakeidx` 等定义 `\see`、`\seealso` 的宏包, 否则其功能将会失效。

5.2 宏包选项

`override` `override` 是布尔型选项，默认为 `true`。表示覆盖已有的 `\see` 与 `\seealso` 定义。如果设置

```
\usepackage[override=false]{seealso}
```

则不对 `\see` 与 `\seealso` 覆盖，要使用 `\seepage` 和 `\seealsopage` 命令调用相应的功能。

`activecr` `activecr` 是布尔型选项，默认为 `true`。表示打开在（.ind 文件中）索引项行末自动输出所有参见列表的功能。如果设置 `activecr=false`，则必须手工（通常通过设置 `Makeindex` 的 `delim_t` 格式）在索引项末加上 `\SeealsoPrintList` 输出参见列表。

5.3 \see 与 \seealso 命令

`\seepage` `\seepage` 与 `\seealsopage` 提供了宏包的主要功能，它们与原始的 `\see`、`\seealso` 命令类似提供索引项的交叉引用功能，但同时会输出页码，并避免重复输出交叉引用。例如，在文档中第 1, 2, 3 页分别多次使用

```
\index{foo|seepage{bar}}
```

则在 .ind 文件中会生成：

```
foo, \seepage{bar}{1}, \seepage{bar}{2}, \seepage{bar}{3}
```

而输出的结果将会是：

```
foo, 1, 2, 3, see mathematics, bar, foobaz
```

同一个词可以有不同的参见引用项目，例如在不同的位置几次使用

```
\index{foo|seepage{bar}}
```

和

```
\index{foo|seepage{foobaz}}
```

则 .ind 文件中可能生成：

```
foo, \seepage{bar}{1}, \seepage{bar}{2}, \seepage{foobaz}{3}
```

于是将得到输出结果

```
foo, 1, 2, 3, see mathematics, bar, foobaz
```

`\seenopage` `\seenopage` 与 `\seealsonopage` 这两个宏则保存了在调用 `seealso` 宏包之前，`\see` 与 `\seealso` 的原始定义，它们不会输出页码。

`\see` 在默认的 `override` 选项下，`\see` 与 `\seealso` 的功能与 `\seepage`、`\seealsopage` 相同。

`\SeealsoPrintList` `\seepage` 和 `\seealsopage` 会收集参见列表，在默认的 `activecr` 选项为 `true` 时会在行末自动输出。但如果 `activecr` 选项设置为 `false`，则 `\seepage` 与 `\seealsopage` 所在行结束后不会自动输出参见列表。此时，可以手工在此行最后加上 `\SeealsoPrintList` 宏，输出参见列表。通常可以在 `Makeindex` 的格式文件中，输出格式选项 `delim_t` 里面加上 `\SeealsoPrintList` 宏，如：

```
% example.ist
delim_t "\\SeealsoPrintList"
```

这样，每个非空索引项输出的最后都会加上 `\SeealsoPrintList`，按需要输出参见列表。

`\DeclareSeealsoMacro` `\DeclareSeealsoMacro` 命令用于生成新的类似 `\see` 与 `\seealso` 的宏。其语法为：

```
\DeclareSeealsoMacro\<macro>\{<seelist>\}{<name>}
```

其中 `\<macro>` 是新的宏名称，`<seelist>` 是新的参见列表名，`<name>` 是输出参见列表时的名称。使用该命令会生成两个宏，一个是 `\<macro>`，一个是对应的名称 `\<macro>name`。`\<macro>name` 如果事先没有定义，则定义其值为 `<name>`，否则保持不变。

例如，`\seepage` 和 `\seealsopage` 就是这样定义的：

```
\DeclareSeealsoMacro\seepage{see}{see}
\DeclareSeealsoMacro\seealsopage{also}{see also}
```

上述命令同时定义了 `\seename` 的备选值为 `see`，以及 `\alsoname` 的备选值为 `see also`。

5.4 格式设置

`\seealsosetup` `\seealsosetup` 命令用于设置参见列表的输出格式。其语法为：

```
\seealsosetup[<seelists>]{<kv-options>}
```

其中，可选参数 `<seelists>` 是一个逗号分隔的列表，每一项表示要设置其格式的参见列表，如果不设置 `<seelists>` 或设为空，则设置默认的输出格式。而选项 `<kv-options>` 则是输出格式项。

可用的输出格式见表 2。

选项	意义	参数	默认值
<code>name</code>	输出参见列表的名称	无	(不可用)
<code>listsep</code>	参见列表前的分隔符	无	<code>,\space</code>
<code>itemsep</code>	参见列表项间的分隔符	无	<code>,\space</code>
<code>nameformat</code>	输出参见名“see also”的格式	<code>#1</code>	<code>\emph {#1}\space</code>
<code>itemformat</code>	参见列表项的格式	<code>#1</code>	<code>#1</code>
<code>pageformat</code>	页码格式	<code>#1</code>	<code>#1</code>

表 2: `\seealsosetup` 使用的输出格式

例如，如果要让所有 `\seealsopage` 命令生成的页码以斜体显示，就可以使用

```
\seealsosetup[also]{pageformat=\textit{#1}}
```

`\SeealsoGobble` `\SeealsoGobble` 宏接受一个字符作为参数，它检查后面是否有此字符，如果是则吞掉此字符及后面的空格，否则什么都不做。在 `pageformat` 选项中使用这个宏可以用来禁止显示参见项的页码，像下面这样：

```
\seealsosetup[see,also]{pageformat=\SeealsoGobble{,}, listsep=}
```

但注意使用 `hyperref` 时，`\see`、`\seealso` 等命令会被 `\hyperindexformat` 所包裹，使这种方法会失效。

6 已知问题

以下是一些已知的问题和解决方案：

- Makeindex 会按输出选项 `line_max` 对输出的内容进行折行，默认的 `line_max` 只有 72，所以当项目较多时，可能在 `\see` 或 `\seealso` 等命令所在的项目中产生折行。但由于默认情况下（`activecr` 选项打开）`seealso` 包遇到换行符就会输出参见列表，会造成错误。一种解决方法是将 `line_max` 设置得很大，避免折行；另一种解决方法是设置 `activecr=false`，同时在 Makeindex 的格式文件中显式设置 `delim_t` 为 `\SeealsoPrintList` 来输出参见列表。
- 由于 `\seepage`, `\seealsopage` 的页码显示形式与普通页码相同，所以可能会造成页码重叠的情况。最好尽量避免交叉引用的项目有多种形式的索引。如果不能避免，为解决此问题，可以设置特别的页码输出格式以示区分，或者借用 `\SeealsoGobble` 设置不输出页码。
- 在使用 `hyperref` 功能时，用 `\SeealsoGobble` 禁止显示页码的功能会失效。这是因为 `\see`、`\seealso` 等命令会被 `\hyperindexformat` 包裹。为解决此问题，可以设置 `hyperref` 的选项 `hyperindex=false` 禁用索引的超链接，或者考虑对 `hyperref` 做特殊的补丁。

7 Implementation / 代码实现

7.1 准备工作

引入相关编程工具。

```
1 \RequirePackage{etoolbox}
2 \RequirePackage{kvoptions}
3 \SetupKeyvalOptions{
4   family=seealso@opt,
5   prefix=seealso@}
```

声明宏包选项。

`override` 重定义 `\see` 与 `\seealso` 为有页码的形式。默认打开。

```
6 \DeclareBoolOption[true]{override}
```

`activecr` 使用换行符作为输出 `\see` 等命令的指令。默认打开。

```
7 \DeclareBoolOption[true]{activecr}
```

执行选项。

```
8 \ProcessKeyvalOptions*
```

`\seealso@charlet` 参数 #1 是一个字符或 `\` 加字符的形式, `\seealso@charlet` 将此字符看做活动字符的宏, 使用 `\let` 与后面的内容赋值, 但本身不改变字符的 catcode。

```
9 \def\seealso@charlet#1{%
10   \begingroup\lccode`\~=#1\lowercase{\endgroup\let~}}
```

7.2 参见列表及其实现

`\seealso@macrolist` 列表记录所有独立的类似 `\seepage` 的宏。默认只有 `see` 和 `also` 两组, 对应命令 `\seepage` 和 `\seealsopage`。

```
11 \let\seealso@macrolist\empty
```

`\seealso@clearlists` 清空参见列表, 如 `\seealso@see@list`、`\seealso@also@list`。

```
12 \def\seealso@clearlists{%
13   \def\do##1{\cslet{seealso@##1@list}\empty}%
14   \dolistloop\seealso@macrolist}
15 \AtBeginDocument{\seealso@clearlists}
```

`\SeealsoPrintList` 输出参见列表, 如 `\seealso@see@list`、`\seealso@also@list`。

```
16 \newcommand\SeealsoPrintList{%
17   \forlistloop\seealso@printlist\seealso@macrolist
18   \seealso@clearlists}
```

`\ifseealso@firstitem` 测试是否是在输出参见列表的第一项。

```
19 \newif\ifseealso@firstitem
```

`\seealso@printlist` 输出参见列表 #1。如果列表为空则无操作。

```
20 \def\seealso@printlist#1{%
21   \ifcempty{seealso@#1@list}
```

```

22 {}
23 {\csuse{seealso@#1@listsep}%
24   \csuse{seealso@#1@nameformat}}{\csuse{#1name}}}%
25   \seealso@firstitemtrue
26   \forlistcsloop{\seealso@listitem{#1}}{\seealso@#1@list}}}}

```

`\seealso@listitem` 输出参见列表的一项。如果不是第一项，同时输出分隔符。

```

27 \def\seealso@listitem#1#2{%
28   \ifseealso@firstitem
29     \seealso@firstitemfalse
30   \else
31     \csuse{seealso@#1@itemsep}%
32   \fi
33   \csuse{seealso@#1@itemformat}{#2}}

```

`\DeclareSeealsoMacro` 定义一个新的带页码的参见命令。`#1` 是命令名，`#2` 是该命令使用的参见列表，`#3` 是列表输出时使用的名字。

```

34 \newcommand\DeclareSeealsoMacro[3]{%
35   \newcommand#1[2]{%
36     \seealso@setactivecr
37     \ifinlistcs{##1}{seealso@#2@list}
38     {}
39     {\listcsadd{seealso@#2@list}{##1}}%
40     \csuse{seealso@#2@pageformat}{##2}}%

```

将参见命令加入列表。

```

41   \listadd\seealso@macrolist{#2}%

```

定义 `name` 选项，用来设置 `\<macro>name`。如 `\seename` 和 `\alsoname`。

```

42   \define@key{seealso@#2}{name}{%
43     \csdef{#2name}{##1}}%

```

如果事先没有定义，则定义 `\<macro>name` 为 `#3`。

```

44   \ifcsdef{#2name}
45   {}
46   {\setkeys{seealso@#2}{name=#3}}%

```

定义无参选项。

```

47   \def\do##1{%
48     \define@key{seealso@#2}{##1}{%
49       \csdef{seealso@#2@##1}{#####1}}%
50     \setkeys{seealso@#2}{##1=\csuse{seealso@##1}}}%
51   \docsvlist{listsep,itemsep}%

```

定义有一个参数的选项。

```

52   \def\do##1{%
53     \define@key{seealso@#2}{##1}{%
54       \csdef{seealso@#2@##1}{#####1{#####1}}%
55       \setkeys{seealso@#2}{##1=\csuse{seealso@##1}{#####1}}}%
56   \docsvlist{nameformat,itemformat,pageformat}}
57 \onlypreamble{\DeclareSeealsoMacro}

```

`\ifseealso@iscractive` 判断当前换行符是否已经被激活。

```
58 \newif\ifseealso@iscractive
```

`\seealso@setactivecr` `activecr` 选项为真时，设置换行符为 `\seealso@cr` 并激活。

```
59 \def\seealso@setactivecr{%
60   \ifseealso@activecr
61     \unless\ifseealso@iscractive
62       \catcode`\^^M=\active
63       \seealso@charlet\^^M\seealso@cr
64       \seealso@iscractivetrue
65   \fi
66 }
```

`\seealso@cr` 是在 `activecr` 选项下，换行符的定义。它取消激活换行符为宏，并输出参见列表。

```
67 \def\seealso@cr{%
68   \catcode`\^^M=5
69   \seealso@iscractivefalse
70   \SeealsoPrintList}
```

下面定义参见列表的默认输出格式。以下宏给出的都是全局的默认值，可以单独修改每一项的输出格式。

`\seealso@listsep` 参见列表之前的分隔符。

```
71 \def\seealso@listsep{,\space}
```

`\seealso@itemsep` 参见列表项之间的分隔符。

```
72 \def\seealso@itemsep{,\space}
```

`\seealso@nameformat` 参见名“see also”的输出格式。

```
73 \def\seealso@nameformat#1{\emph{#1}\space}
```

`\seealso@itemformat` 参见列表项的输出格式。

```
74 \def\seealso@itemformat#1{#1}
```

`\seealso@pageformat` 参见页码的输出格式。

```
75 \def\seealso@pageformat#1{#1}
```

`\seealso@define@key` 定义 `seealso` 族的键，用于设置输出格式。

```
\seealso@define@keyarg 76 \def\seealso@define@key#1{%
77   \define@key{seealso}{#1}{%
78     \csdef{seealso@#1}{##1}}
79 \def\seealso@define@keyarg#1{%
80   \define@key{seealso}{#1}{%
81     \csdef{seealso@#1}###1{##1}}}
```

为每个输出格式定义键。

```
82 \forcsvlist\seealso@define@key{listsep,itemsep}
```

```
83 \forcsvlist\seealso@define@keyarg{nameformat,itemformat,pageformat}
```

`\SeealsoGobble` 检查后面的字符是否 #1，如果是则吞掉分隔符 #1 及后面的空格。这个宏可用于设置 `pageformat` 格式，吞掉一个类似 `,\space` 类型的分隔符。

```
84 \def\SeealsoGobble#1{\ifnextchar#1{\seealso@gobbleignorespaces}{}}
```

`\seealso@gobbleignorespaces` 吞掉参数并忽略后面的空格。

```
85 \def\seealso@gobbleignorespaces#1{\ignorespaces}
```

7.3 定义用户接口

`\seealsosetup` 设置输出格式。可选参数是一个逗号列表：如果可选参数为空，用 #2 设置默认的输出格式；否则对可选参数的每一项，设置参数 #2 的格式。使用 `\forcsvlist` 而没有使用 `\docsvlist` 是为了避免因嵌套定义造成使用时参数不得不使用双重 ##。

```
86 \newcommand\seealsosetup[2][]{%
87   \ifstrempy{#1}{%
88     \setkeys{seealso}{#2}%
89   }{%
90     \def\do##1{\setkeys{seealso@##1}{#2}}%
91     \forcsvlist{\do}{#1}}}
```

`\seenopage` 保存旧的 `\see` 与 `\seealso` 命令定义，使用 `override` 选项时可临时使用旧的定义。

```
\seealsoonopage 92 \let\seenopage\see
93 \let\seealsoonopage\seealso
```

`\seepage` 带页码输出的 `\seepage` 与 `\seealsopage`。使用 `override` 选项时可直接使用 `\see` 与 `\seealsopage` `\seealso` 代替 `\seepage` 与 `\seealsopage`。

如果之前没有定义，这里会同时定义 `\seename` 和 `\alsoname`，与 `makeidx` 初始值一致。

```
94 \DeclareSeealsoMacro\seepage{see}{see}
95 \DeclareSeealsoMacro\seealsopage{also}{see also}
```

`\see` 使用 `override` 时，重定义 `\see` 与 `\seealso`。

```
\seealso 96 \ifseealso@override
97   \def\see{\seepage}
98   \def\seealso{\seealsopage}
99 \fi
```

版本历史

v0.1	的参见命令。.....	10
General: 初始版本。.....	1	<code>\SeealsoGobble</code> : 支持吞掉分隔符的页
v1.0	码格式，即不显示页码。.....	12
General: 功能较完备的版本。.....	1	<code>\seealsosetup</code> : 新增格式设置。.....
<code>activecr</code> : 新增选项。.....	9	<code>override</code> : 新增选项。.....
<code>\DeclareSeealsoMacro</code> : 新增以定义新		9

Index / 代码索引

斜体的数字表示对应项说明所在的页码。下划线的数字表示定义所在的代码行号；而直立体的数字表示对应项使用时所在的行号。

Symbols	override (option) . . . 2, <u>6</u> , 6	\seealso@iscractivetrue 64
\^ 62, 63, 68		
\~ 10		
A	S	
activecr (option) . . . 2, 6, <u>7</u>	\see 2, 6, <u>92</u> , <u>96</u>	\seealso@itemformat . . . 74
	\seealso 2, 6, <u>93</u> , <u>96</u>	\seealso@itemsep 72
	\seealso@charlet . . . <u>9</u> , 63	\seealso@listitem . . . 26, <u>27</u>
	\seealso@clearlists	\seealso@listsep 71
		\seealso@macrolist
D		
\DeclareSeealsoMacro 3, 7, <u>34</u> , 94, 95 12, 18 <u>11</u> , 14, 17, 41
	\seealso@cr 63, <u>67</u>	\seealso@nameformat . . . 73
	\seealso@define@key	\seealso@pageformat . . . 75
 76, 82	\seealso@printlist . . . 17, <u>20</u>
I	\seealso@define@keyarg	\seealso@setactivecr
\ifseealso@activecr . . . 60 76, 83 36, <u>59</u>
\ifseealso@firstitem	\seealso@firstitemfalse	\SeealsoGobble . . . 3, 7, <u>84</u>
. 19, 28 29	\seealsoonopage . . . 2, 6, <u>92</u>
\ifseealso@iscractive	\seealso@firstitemtrue	\seealsoopage . . . 2, 6, <u>94</u> , 98
. 58, 61 25	\SeealsoPrintList
\ifseealso@override . . . 96	\seealso@gobbleignorespaces 2, 6, <u>16</u> , 70
 84, <u>85</u>	\seealso@setup 3, 7, <u>86</u>
O	\seealso@iscractivetrue	\seenopage 2, 6, <u>92</u>
options: 69	\seepage 2, 6, <u>94</u> , 97
activecr 2, 6		
override 2, 6		