

The `seealso` Package (v1.2)

⇒ 中文版

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

2017/03/23

1 Introduction

When preparing index, the macros `\see` and `\seealso` defined by L^AT_EX 2_ε'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 L^AT_EX 2_ε 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 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 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* bar, foobaz

`\seenopage` The macros `\seenopage` and `\seealsonopage` save the original definitions of
`\seealsonopage` `\see` 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\langle macro \rangle{\langle seelist \rangle}{\langle name \rangle}
```

where `\langle macro \rangle` is the macro name, `\langle seelist \rangle` is the new see list, and `\langle name \rangle` is the
name of the see list to be output (like `\seename`). Using `\DeclareSeealsoMacro`,
two macros will be defined, the one is `\langle macro \rangle`, the other is `\langle macro \rangle name`. If
there has no `\langle macro \rangle name` predefined, it is defined to be `\langle name \rangle`, 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[\langle seelists \rangle]{\langle kv-options \rangle}
```

where `\langle seelists \rangle` is a comma separated list, each item of which is a see list to be
setup. `\langle seelists \rangle` 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. `\langle kv-options \rangle` is the options in
key-value syntax.

All possible output styles are summarized in table 1.

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>

表 1: Output styles used by `\seealsosetup`

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

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

Besides the output styles, `\seealsosetup` accepts `enditem` and `enditem+` options, which is a list of token that determines whether to output the see list at the end of line, if `activecr` option is true. The default value of the list includes some common macros in the `index` environment: `\indexspace`, `\item`, `\subitem`, `\subsubitem`, and `\end`. For example, if we have a special index which supports 4th level item `\subsubsubitem` and a special space `\myindexspace`, we can use

```
\seealsosetup{enditem+={\subsubsubitem,\myindexspace}}
```

to add the new macros to the list.

`\SeealsoGobble`

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

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

3 Known Issues

Here are some known problems and their solutions:

- Since the output format of page numbers of `\seepage`, `\seealso` 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 `\seealso`. If it

is impossible, we can set special page number format to identify them, or we can use `\SeealsoGobble` to disable page numbers.

刘海洋

leoliu.pku@gmail.com

2017/03/23

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
```

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` `\seealsopage` `\see` 与 `\seealso` 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 bar
```

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 bar, foobaz
```

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

```
\DeclareSeealsoMacro\langle macro\rangle{\langle seelist\rangle}{\langle name\rangle}
```

其中 `\langle macro\rangle` 是新的宏名称，`\langle seelist\rangle` 是新的参见列表名，`\langle name\rangle` 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 `\langle macro\rangle`，一个是对应的名称 `\langle macro\rangle name`。`\langle macro\rangle name` 如果事先没有定义，则定义其值为 `\langle name\rangle`，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` 命令用于设置参见列表的输出格式。其语法为：

```
\seealsosetup[\langle seelists\rangle]{\langle kv-options\rangle}
```

其中，可选参数 `\langle seelists\rangle` 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 `\langle seelists\rangle` 或设为空，则设置默认的输出格式。而选项 `\langle kv-options\rangle` 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}}
```

除了输出格式，`\seealsosetup` 还接受 `enditem` 和 `enditem+` 选项，用于设置 `seealso` 宏包在 `activecr` 选项设置时，进入行尾之后遇到什么记号时决定输出参见

列表。通常情况下并不需要设置该选项, `enditem` 选项的默认值已经包括了在标准的 `theindex` 环境中会出现的一些宏: `\indexspace`、`\item`、`\subitem`、`\subsubitem`、`\end`。使用 `enditem+` 选项则可以向列表中增加新的记号。例如, 如果某种特殊的索引支持 4 级项 `\subsubsubitem` 和特殊的间隔 `\myindexspace`, 就可以使用

```
\seealsosetup{enditem+={\subsubsubitem,\myindexspace}}
```

向列表中添加新的记号。

`\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{,}}
```

6 已知问题

以下是一些已知的问题和解决方案:

- 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。

7 Implementation / 代码实现

7.1 准备工作

引入相关编程工具。

`etoolbox` 是基本宏工具。原有的 `\ifinlist` 和 `\ifinlistcs` 不能用于带花括号的文字的查找，会限制我们的使用。为此按 `etoolbox` 手册的建议，补充了 `\seealso@ifstrinlist` 和 `\seealso@ifstrinlistcs` 两个宏来在列表中搜索字符串。它们会比 `\ifinlist` 与 `\ifinlistcs` 性能略差一点。

```
1 \RequirePackage{etoolbox}
2 % {<listmacro>}{<string>}{<>true>}{<>false>}
3 \long\def\seealso@ifinlist@#1#2#3#4{%
4   \def\next{#4}%
5   \def\do##1{%
6     \ifstrequal{##1}{#2}
7       {\def\next{#3}\listbreak}
8     }%
9   \dolistloop{#1}%
10  \next}
11 % {<string>}{<listmacro>}{<>true>}{<>false>}
12 \protected\long\def\seealso@ifinlist#1#2{%
13   \seealso@ifinlist@{#2}{#1}}
14 % {<string>}{<listcsname>}{<>true>}{<>false>}
15 \protected\long\def\seealso@ifinlistcs#1#2{%
16   \expandafter\seealso@ifinlist@\csname #2\endcsname{#1}}
```

`kvoptions` 用于处理宏包选项。

```
17 \RequirePackage{kvoptions}
18 \SetupKeyvalOptions{
19   family=seealso@opt,
20   prefix=seealso@,
21   setkeys=\kvsetkeys}
```

声明宏包选项。

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

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

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

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

执行选项。

```
24 \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`。

```
25 \def\seealso@charlet#1{%
26   \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`。

```
27 \let\seealso@macrolist\empty
```

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

```
28 \def\seealso@clearlist#1{%
29   \global\cslet{seealso@#1@list}\empty}
30 \AtBeginDocument{\forlistloop\seealso@clearlist\seealso@macrolist}
```

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

```
31 \newcommand\SeealsoPrintList{%
32   \forlistloop\seealso@printlist\seealso@macrolist
33   \forlistloop\seealso@clearlist\seealso@macrolist}
```

`\ifseealso@gobblefirstlistsep` 测试是否忽略第一个列表分隔符（因为前面使用了 `\SeealsoGobble`）。

```
34 \newif\ifseealso@gobblefirstlistsep
```

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

```
35 \newif\ifseealso@firstitem
```

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

```
36 \def\seealso@printlist#1{%
37   \ifcsemtyp{seealso@#1@list}
38   {}
39   {\ifseealso@gobblefirstlistsep
40     \seealso@gobblefirstlistsepfalse
41     \else
42     \csuse{seealso@#1@listsep}%
43     \fi
44     \csuse{seealso@#1@nameformat}{\csuse{#1name}}%
45     \seealso@firstitemtrue
46     \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。

```
47 \def\seealso@listitem#1#2{%
48   \ifseealso@firstitem
49   \seealso@firstitemfalse
50   \else
51   \csuse{seealso@#1@itemsep}%
52   \fi
53   \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。

```
54 \newcommand\DeclareSeealsoMacro[3]{%
```

首先定义参见命令 #1 本身。

```
55 \newcommand#1[2]{%
56   \seealso@setactivecr
57   \seealso@ifinlistcs{##1}{\seealso@#2@list}
58   {}
59   {\listcsgadd{\seealso@#2@list}{##1}}%
60   \csuse{\seealso@#2@pageformat}{##2}}%
```

将参见命令加入列表。

```
61 \listadd\seealso@macrolist{#2}%
```

定义 name 选项，用来设置 $\langle macro \rangle$ name。如 \backslash seename 和 \backslash alsoname。

```
62 \define@key{\seealso@#2}{name}{%
63   \csdef{#2name}{##1}}%
```

如果事先没有定义，则定义 $\langle macro \rangle$ name 为 #3。

```
64 \ifcsdef{#2name}
65   {}
66   {\kvsetkeys{\seealso@#2}{name=#3}}%
```

定义无参选项。

```
67 \def\do##1{%
68   \define@key{\seealso@#2}{##1}{%
69     \csdef{\seealso@#2@##1}{#####1}}%
70   \kvsetkeys{\seealso@#2}{##1=\csuse{\seealso@##1}}%
71   \docsvlist{listsep,itemsep}%
```

定义有一个参数的选项。

```
72 \def\do##1{%
73   \define@key{\seealso@#2}{##1}{%
74     \csdef{\seealso@#2@##1}{#####1{#####1}}%
75     \kvsetkeys{\seealso@#2}{##1=\csuse{\seealso@##1}{#####1}}%
76     \docsvlist{nameformat,itemformat,pageformat}}
77 \onlypreamble{\DeclareSeealsoMacro}
```

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

```
78 \newif\ifseealso@iscractive
```

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

```
79 \def\seealso@setactivecr{%
80   \ifseealso@activecr
81     \unless\ifseealso@iscractive
82       \catcode`\^M=\active
83       \seealso@charlet\^M\seealso@cr
84       \seealso@iscractivetrue
85     \fi
86   \fi}
```

\backslash seealso@cr 是在 activecr 选项下，换行符的定义。它检测后面的记号是否在 \backslash seealso@enditemlist 中，以判断是否处于折行状态，决定是否输出列表。

```
87 \def\seealso@cr{%
88   \futurelet\next\seealso@cr@aux}
```

`\seealso@cr@aux` 是实际完成判断折行与输出的辅助过程。如果折行，则补全折行造成的空白；否则取消激活换行符为宏，并输出参见列表。

```
89 \def\seealso@cr@aux{%
90   \seealso@testwrap
91   \ifseealso@wrap
92     \space
93   \else
94     \catcode\^^M=5
95     \seealso@iscractivefalse
96     \SeealsoPrintList
97   \fi}
```

`\ifseealso@wrap` 判断当前是否在折行的行尾。

```
98 \newif\ifseealso@wrap
```

`\seealso@testwrap` 检查由 `\seealso@cr` 获取的下一记号 `\next` 是否在列表 `\seealso@enditemlist` 中，确定当前是否在折行行尾。

```
99 \def\seealso@testwrap{%
100   \seealso@wraptrue
101   \forlistloop\seealso@testwrap@aux\seealso@enditemlist}
102 \def\seealso@testwrap@aux#1{%
103   \ifx#1\next
104     \seealso@wrapfalse
105     \expandafter\listbreak
106   \fi}
```

`\seealso@enditemlist` 表示索引项结束的记号列表。用于判断是否处于折行状态。

```
107 \let\seealso@enditemlist\empty
```

`\seealso@enditemlistadd` 向结束记号列表中增加一项。

```
108 \def\seealso@enditemlistadd#1{%
109   \seealso@ifinlist{#1}\seealso@enditemlist
110   {}
111   {\listadd\seealso@enditemlist{#1}}}
```

下面定义结束记号列表的在 `seealso` 族下的键 `enditem` 与 `enditem+`，允许用户自定义此列表。这里要求列表总包含 `\seealso@cr`。

```
112 \define@key{seealso}{enditem}{%
113   \let\seealso@enditemlist\empty
114   \listadd\seealso@enditemlist{\seealso@cr}%
115   \forcsvlist\seealso@enditemlistadd{#1}}
116 \define@key{seealso}{enditem+}{%
117   \forcsvlist\seealso@enditemlistadd{#1}}
```

结束记号列表的默认值，除了 `\seealso@cr` 外，还包括通常在 `theindex` 等环境中出现的其他一些宏。

```
118 \kvsetkeys{seealso}{%
119   enditem={\indexspace,\item,\subitem,\subsubitem,\end}}
```

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。
120 \def\seealso@listsep{,\space}

\seealso@itemsep 参见列表项之间的分隔符。
121 \def\seealso@itemsep{,\space}

\seealso@nameformat 参见名“see also”的输出格式。
122 \def\seealso@nameformat#1{\emph{#1}\space}

\seealso@itemformat 参见列表项的输出格式。
123 \def\seealso@itemformat#1{#1}

\seealso@pageformat 参见页码的输出格式。
124 \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 125 \def\seealso@define@key#1{%
126   \define@key{seealso}{#1}{%
127     \csdef{seealso@#1}{##1}}
128 \def\seealso@define@keyarg#1{%
129   \define@key{seealso}{#1}{%
130     \csdef{seealso@#1}###1{##1}}}

为每个输出格式定义键。
131 \forcsvlist\seealso@define@key{listsep,itemsep}
132 \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。
    这里检查是否定义了 hyperref 的 \hyperindexformat（表示打开了索引的超链接功能），如果有 \hyperindexformat，则需要调整参数次序以解决 \hyperindexformat
与 \see 等命令嵌套的问题。
133 \AtBeginDocument{%
134   \ifundef\hyperindexformat{%
135     \let\SeealsoGobble\seealso@gobble
136   }{%
137     \def\SeealsoGobble#1{\seealso@swap{\seealso@gobble{#1}}}%
138     \def\seealso@swap#1#2#3#4{#2#3#4#1}%
139   }}

\seealso@gobble 实际的 \SeealsoGobble 命令，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。
140 \def\seealso@gobble#1{%
141   \@ifnextchar#1%
142     {\seealso@gobbleignorespaces}

```

如果 `\SeealsoGobble` 后面即将输出，即为换行符或 `\SeealsoPrintList`，则决定后面不再输出第一个参见列表的 `listsep`。

```

143   {\ifseealso@activecr
144     \ifx\@let@token\seealso@cr
145     \seealso@gobblefirstlistseptrue
146     \fi
147   \else
148     \ifx\@let@token\SeealsoPrintList
149     \seealso@gobblefirstlistseptrue
150     \fi
151   \fi}}
```

`\seealso@gobbleignorespaces` 吞掉分隔符参数 #1（通常是一个逗号）并忽略后面的空格。

为了处理折行时会遇到的换行符宏 `\seealso@cr`，在发现后面的参数是 `\seealso@cr` 时就递归调用自身把 `\seealso@cr` 也清除掉。此时就好像直接把两行连接起来，不再调用 `\seealso@cr` 的复杂判断流程。

```

152 \def\seealso@gobbleignorespaces#1{%
153   \@ifnextchar\seealso@cr{\seealso@gobbleignorespaces}{\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 设置默认的输出版式；否则对可选参数的每一项，设置参数 #2 的格式。为了避免因嵌套定义造成使用时参数不得不使用双重 ##，这里作了特别的展开处理。

```

154 \newcommand\seealsosetup[2] [] {%
155   \ifstrempy{#1}{%
156     \kvsetkeys{seealso}{#2}%
157   }{%
158     \edef\do##1{%
159       \noexpand\kvsetkeys{seealso@##1}{\unexpanded{#2}}}%
160     \docsvlist{#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`

```

161 \let\seenopage\see
162 \let\seealsoonopage\seealso
```

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

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

```

163 \DeclareSeealsoMacro\seepage{see}{see}
164 \DeclareSeealsoMacro\seealsoopage{also}{see also}
```

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

```

\seealso 165 \ifseealso@override
166   \def\see{\seepage}
167   \def\seealso{\seealsoopage}
```

168 \fi

版本历史

v0.1	listsep 为空。	14
General: 初始版本。	处理 hyperref 兼容性。	14
v1.0	\seealso@cr: 处理折行问题。	12
\DeclareSeealsoMacro: 新增以定义新的参见命令。	\seealso@gobbleignorespaces: 正确处理 \SeealsoGobble 在自动折行下的行为。	15
\SeealsoGobble: 支持吞掉分隔符的页码格式, 即不显示页码。	\seealso@setup: 正确处理有参数的格式。	15
\seealso@setup: 新增格式设置。		
General: 功能较完备的版本。	v1.2	
v1.1	General: 允许 \see 等命令参数中带花括号。	10
\SeealsoGobble: 不再要求设置		

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	\ifseealso@override . 165	\seealso@enditemlist 101,
\^ 82, 83, 94	\ifseealso@wrap . . . 91, <u>98</u>	<u>107</u> , 109, 111, 113, 114
\~ 26	\ifstrequal 6	\seealso@enditemlistadd 108, 115, 117
A	\indexspace 119	\seealso@firstitemfalse 49
activecr (option) . 2, 7, <u>23</u>	\item 119	\seealso@firstitemtrue 45
C	L	\seealso@gobble 135, 137, <u>140</u>
\csname 16	\long 3, 12, 15	\seealso@gobblefirstlistsepfalse 40
D	O	\seealso@gobblefirstlistseptrue 145, 149
\DeclareSeealsoMacro . . . 3, 8, <u>54</u> , 163, 164	options:	\seealso@gobbleignorespaces 142, <u>152</u>
	activecr 2, 7	\seealso@ifinlist 12, 109
	override 2, 7	\seealso@ifinlist@ 3, 13, 16
	override (option) . 2, 7, <u>22</u>	\seealso@ifinlistcs 15, 57
E	P	\seealso@iscractivefalse 95
\end 119	\protected 12, 15	\seealso@iscractivetrue 84
\endcsname 16	S	\seealso@itemformat . <u>123</u>
H	\see 3, 7, 161, <u>165</u>	\seealso@itemsep <u>121</u>
\hyperindexformat . . . 134	\seealso . . . 3, 7, 162, <u>165</u>	
I	\seealso@charlet . . 25, 83	
\ifseealso@activecr 80, 143	\seealso@clearlist <u>28</u> , 33	
\ifseealso@firstitem 35, 48	\seealso@cr	
\ifseealso@gobblefirstlistsep 34, 39	\seealso@cr@aux . . . 88, 89	
\ifseealso@iscractive 78, 81	\seealso@define@key 125, 131	
	\seealso@define@keyarg 125, 132	

<code>\seealso@listitem</code> . 46 , 47	<code>\seealso@swap</code> ... 137 , 138	<code>\SeealsoPrintList</code> ...
<code>\seealso@listsep</code> 120	<code>\seealso@testwrap</code> . 90 , 99 3 , 7 , 31 , 96 , 148
<code>\seealso@macrolist</code> ..	<code>\seealso@testwrap@aux</code>	<code>\seealsosetup</code> ... 3 , 8 , 154
... 27 , 30 , 32 , 33 , 61 101 , 102	
<code>\seealso@nameformat</code> . 122	<code>\seealso@wrapfalse</code> .. 104	<code>\seenopage</code> 3 , 7 , 161
<code>\seealso@pageformat</code> . 124	<code>\seealso@wraptrue</code> ... 100	<code>\seepage</code> ... 2 , 7 , 163 , 166
<code>\seealso@printlist</code> 32 , 36	<code>\SeealsoGobble</code> .. 4 , 9 , 133	<code>\subitem</code> 119
<code>\seealso@setactivecr</code>	<code>\seealso@nopage</code> .. 3 , 7 , 161	<code>\subsubitem</code> 119
..... 56 , 79	<code>\seealso@page</code> 2 , 7 , 163 , 167	