

The `seealso` Package (v1.1)

⇒ 中文版

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

2014/04/10

1 Introduction

When preparing index, the macros `\see` and `\seealso` defined 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 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 `\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}}
```

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 `theindex` 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 `\<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{,}}
```

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`

3 Known Issues

Here are some known problems and their solutions:

- 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.

刘海洋

leoliu.pku@gmail.com

2014/04/10

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 **\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 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 **\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}}
```

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

```
\seealsoresetup{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：

```
\seealsoresetup[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。

```
1 \RequirePackage{etoolbox}
2 \RequirePackage{kvoptions}
3 \SetupKeyvalOptions{
4   family=seealso@opt,
5   prefix=seealso@,
6   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。

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

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

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

执行选项。

```
9 \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。

```
10 \def\seealso@charlet#1{%
11   \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`。

```
12 \let\seealso@macrolist\empty
```

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

```
13 \def\seealso@clearlist#1{%
14   \global\cslet{seealso@#1@list}\empty}
15 \AtBeginDocument{\forlistloop\seealso@clearlist\seealso@macrolist}
```

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

```
16 \newcommand\SeealsoPrintList{%
17   \forlistloop\seealso@printlist\seealso@macrolist
18   \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`)。

```
19 \newif\ifseealso@gobblefirstlistsep
```

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

```
20 \newif\ifseealso@firstitem
```

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

```
21 \def\seealso@printlist#1{%
22   \ifcempty{seealso@#1@list}
23   {}
24   {\ifseealso@gobblefirstlistsep
25     \seealso@gobblefirstlistsepfalse
26     \else
27     \csuse{seealso@#1@listsep}%
28     \fi
29     \csuse{seealso@#1@nameformat}{\csuse{#1name}}%
30     \seealso@firstitemtrue
31     \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。

```
32 \def\seealso@listitem#1#2{%
33   \ifseealso@firstitem
34   \seealso@firstitemfalse
35   \else
36   \csuse{seealso@#1@itemsep}%
37   \fi
38   \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。

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

首先定义参见命令 #1 本身。

```
40   \newcommand#1[2]{%
41     \seealso@setactivecr
42     \ifinlistcs{##1}{seealso@#2@list}
43     {}
44     {\listcsgadd{seealso@#2@list}{##1}}%
45     \csuse{seealso@#2@pageformat}{##2}}%
```

将参见命令加入列表。

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

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

```
47   \define@key{seealso@#2}{name}{%
48     \csdef{#2name}{##1}}%
```

如果事先没有定义，则定义 `\<macro>name` 为 #3。

```
49   \ifcsdef{#2name}
50   {}
51   {\kvsetkeys{seealso@#2}{name=#3}}%
```

定义无参选项。

```
52   \def\do##1{%
53     \define@key{seealso@#2}{##1}{%
54       \csdef{seealso@#2@##1}{####1}}%
55     \kvsetkeys{seealso@#2}{##1=\csuse{seealso@##1}}%
56     \docsvlist{listsep,itemsep}%
```

定义有一个参数的选项。

```
57 \def\do##1{%
58   \define@key{seealso@#2}{##1}{%
59     \csdef{seealso@#2@##1}#####1{####1}}%
60     \kvsetkeys{seealso@#2}{##1=\csuse{seealso@##1}{####1}}}%
61 \docsvlist{nameformat,itemformat,pageformat}}
62 \@onlypreamble{\DeclareSeealsoMacro}
```

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

```
63 \newif\ifseealso@iscractive
```

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

```
64 \def\seealso@setactivecr{%
65   \ifseealso@activecr
66     \unless\ifseealso@iscractive
67       \catcode\^^M=\active
68       \seealso@charlet\^^M\seealso@cr
69       \seealso@iscractivetrue
70   \fi
71 }
```

`\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 `\seealso@enditemlist` 中，以判断是否处于折行状态，决定是否输出列表。

```
72 \def\seealso@cr{%
73   \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。

```
74 \def\seealso@cr@aux{%
75   \seealso@testwrap
76   \ifseealso@wrap
77     \space
78   \else
79     \catcode\^^M=5
80     \seealso@iscractivefalse
81     \SeealsoPrintList
82   \fi}
```

`\ifseealso@wrap` 判断当前是否在折行的行尾。

```
83 \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。

```
84 \def\seealso@testwrap{%
85   \seealso@wraptrue
86   \forlistloop\seealso@testwrap@aux\seealso@enditemlist}
87 \def\seealso@testwrap@aux#1{%
88   \ifx#1\next
89     \seealso@wrapfalse
90     \expandafter\listbreak
```

```
91 \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。

```
92 \let\seealso@enditemlist\empty
```

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

```
93 \def\seealso@enditemlistadd#1{%
94   \ifinlist{#1}\seealso@enditemlist
95   {}
96   {\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`。

```
97 \define@key{seealso}{enditem}{%
98   \let\seealso@enditemlist\empty
99   \listadd\seealso@enditemlist{\seealso@cr}%
100   \forcsvlist\seealso@enditemlistadd{#1}}
101 \define@key{seealso}{enditem+}{%
102   \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。

```
103 \kvsetkeys{seealso}{%
104   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。

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

`\seealso@itemsep` 参见列表项之间的分隔符。

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

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

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

`\seealso@itemformat` 参见列表项的输出格式。

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

`\seealso@pageformat` 参见页码的输出格式。

```
109 \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 110 \def\seealso@define@key#1{%
111   \define@key{seealso}{#1}{%
112     \csdef{seealso@#1}{##1}}
113 \def\seealso@define@keyarg#1{%
114   \define@key{seealso}{#1}{%
115     \csdef{seealso@#1}{###1{##1}}}
```

为每个输出格式定义键。

```
116 \forcsvlist\seealso@define@key{listsep,itemsep}
117 \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。

```
118 \AtBeginDocument{%
119   \ifundef\hyperindexformat{%
120     \let\SeealsoGobble\seealso@gobble
121   }{%
122     \def\SeealsoGobble#1{\seealso@swap{\seealso@gobble{#1}}}%
123     \def\seealso@swap#1#2#3#4{#2#3#4#1}%
124   }}
```

`\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。

```
125 \def\seealso@gobble#1{%
126   \@ifnextchar#1%
127   {\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`。

```
128   {\ifseealso@activecr
129     \ifx\@let@token\seealso@cr
130       \seealso@gobblefirstlistseptrue
131     \fi
132   \else
133     \ifx\@let@token\SeealsoPrintList
134       \seealso@gobblefirstlistseptrue
135     \fi
136   \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。

```
137 \def\seealso@gobbleignorespaces#1{%
138   \@ifnextchar\seealso@cr{\seealso@gobbleignorespaces}{\ignorespaces}}
```

7.3 定义用户接口

`\seealso@setup` 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。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。

```
139 \newcommand\seealso@setup[2] []{%
140   \ifstrempy{#1}{%
```

```

141 \kvsetkeys{seealso}{#2}%
142 }{%
143 \edef\do##1{%
144 \noexpand\kvsetkeys{seealso@##1}{\unexpanded{#2}}}%
145 \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 146 \let\seenopage\see
147 \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 `\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。

```

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

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

```

\seealso 150 \ifseealso@override
151 \def\see{\seepage}
152 \def\seealso{\seealsopage}
153 \fi
```

版本历史

v0.1		v1.1	
General: 初始版本。	1	<code>\seealso@cr</code> : 处理折行问题。	11
v1.0		<code>\seealso@gobbleignorespaces</code> : 正确	
General: 功能较完备的版本。	1	处理 <code>\SeealsoGobble</code> 在自动折行	
<code>activecr</code> : 新增选项。	9	下的行为。	13
<code>\DeclareSeealsoMacro</code> : 新增以定义新		<code>\SeealsoGobble</code> : 不再要求设置	
的参见命令。	10	<code>listsep</code> 为空。	13
<code>\SeealsoGobble</code> : 支持吞掉分隔符的页		处理 <code>hyperref</code> 兼容性。	13
码格式，即不显示页码。	13	<code>\seealso@setup</code> : 正确处理有参数的格	
<code>\seealso@setup</code> : 新增格式设置。	13	式。	13
<code>override</code> : 新增选项。	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	A	D
<code>\^</code>	67, 68, 79	<code>\DeclareSeealsoMacro</code>
<code>\~</code>	11	<code>activecr</code> (option)
	2, 6, 8	
		3, 7, 39, 148, 149

