

符合 GB/T 7714-2015 标准的 biblatex 参考文献样式 *

胡振震¹

2016-07-01

目 录

示例	3
1 概述	4
1.1 biblatex 宏包的参考文献生成方法	5
1.2 gb7714-2015 样式加载方式和选项说明	6
1.2.1 样式加载方式	6
1.2.2 增加的选项	7
1.2.3 与标准选项的兼容性	8
1.3 样式文件的进一步说明	9
1.3.1 关于参考文献数据源: bib 文件的说明	9
1.3.2 关于参考文献著录样式: bbx 文件的说明	13
1.3.3 关于参考文献标注样式: cbx 文件的说明	13
2 GB/T 7714-2015 标准说明与实现	14
2.1 顺序编码制	14
2.1.1 参考文献表	14
2.1.2 文献标注法	14
2.2 作者年制	15
2.2.1 参考文献表	15
2.2.2 文献标注法	15
2.3 各类文献在 biblatex 中对应的条目和域	15
2.3.1 专著/book	16
2.3.2 标准/standard	16
2.3.3 专著中的析出文献/inbook	17
2.3.4 连续出版物/periodical	17
2.3.5 连续出版物的析出文献/article	17
2.3.6 报纸析出的文献/newspaper	18
2.3.7 专利/patent	18
2.3.8 电子资源/online	19
2.3.9 汇编或论文集/collection	19
2.3.10 汇编或论文集析出中的文献/incollection	19

*This Manual was created with biblatex v3.10, last revised at April 8, 2018;
Style Files (gb7714-2015*.*) have version number: 2018/04/03 v1.0k.

¹Email: hzzmail@163.com

2.3.11	会议录或会议文集/proceedings	20
2.3.12	会议文集中析出的文献/inproceedings	20
2.3.13	学位论文/thesis	20
2.3.14	报告/report	20
2.3.15	手册或档案/manual	21
2.3.16	未出版物/unpublished	21
2.4	多语言文献	21
2.5	标准的其它细节要求	23
2.5.1	数字	24
2.5.2	英文字母	24
2.5.3	标点	24
2.5.4	责任者	24
2.5.5	文献类型标识和载体	25
2.5.6	版次	25
2.5.7	出版项	25
2.5.8	页码	25
2.5.9	访问路径 URL 和 DOI	26
2.5.10	卷和期	26
3	参考文献著录格式测试	26
3.1	GB/T 7714-2015 标准中的参考文献格式示例	26
3.2	测试: 专著 book 和专著中的析出文献 inbook 及标准 standard 文献	30
3.3	测试: 连续出版物 periodical 和连续出版物中的析出文献 article	30
3.4	测试: 电子资源或在线资源 online	30
3.5	测试: 学位论文 thesis、专利文献 patent	31
3.6	测试: 报告 report、手册 manual 和档案、未出版物 unpublished	31
3.7	测试: 会议文集 proceedings 和会议文集中析出的文献 inproceedings 及汇编 collection 和汇编中的析出文献 incollection	31
3.8	测试: 双语文献	31
3.9	测试: gbalign 选项	31
3.10	测试: gbpub 选项	31
3.11	测试: gbnoauthor 选项和 online 条目仅存 url 信息	32
3.12	测试: gbnamefmt 选项	32
3.13	测试: beamer 类	32
3.14	测试: 上标和非上标等不同的标注格式	32
3.15	测试: phdthesis 等条目类型的兼容性	33
3.16	测试: 当责任者等需要判断中英文的信息中存在编组时的处理	34
3.17	测试: 处理参考文献信息中 & 等特殊字符	34
3.18	测试: 处理作者年制 article 中卷信息缺省的标点	34
3.19	测试: 标题中有\LaTeX{} 等名称时的情况	34

4	样式文件设计、实现、应用	35
4.1	基本思路	35
4.2	利用字符流解析或者 xstring 宏包的函数	35
4.3	标点和空格的特点和机制	37
4.4	动态数据修改	39
4.4.1	基本原理和方法	39
4.4.2	数据模型	41
4.4.3	用于关联条目的动态修改的命令封装及关联条目格式	42
4.5	标注 (引用) 标签	43
4.5.1	标签中的标点和间隔符号	44
4.6	日期格式控制	44
4.6.1	文献表条目标签中的日期格式	46
4.7	作者格式控制	46
4.7.1	作者信息输出与格式控制	47
4.8	样式文件设计途径	49
4.8.1	文献表条目内容组织的原理以及问题查找的方法	49
4.9	宏包选项增加	51
4.10	文献表环境和打印输出	53
4.11	biblatex 版本以及样式的兼容性	53
4.12	脚注或旁注文献	58
4.13	参考文献标题	58
4.14	其它需要说明的问题	59
5	总结与致谢	59
6	存在的问题和下一步工作	60
6.1	存在的问题	60
6.2	下一步工作	60
7	更新历史	61

示 例

例 1.	使用 biblatex 宏包的文档编译命令	5
例 2.	中文文献排序时的 biber 选项	5
例 3.	输出引用文献数据时的 biber 选项	5
例 4.	biblatex 参考文献生成最小工作示例	6
例 5.	gb7714-2015 顺序编码制加载方式	7
例 6.	gb7714-2015 作者年制加载方式	7
例 7.	参考文献文本转换为 bib 文件 perl 脚本使用方式	7
例 8.	顺序编码制 cbx 样式命令使用说明	13
例 9.	作者年制 cbx 样式命令使用说明	14

例 10. 专著/book 条目的域格式	16
例 11. 标准/standard 条目的域格式	17
例 12. 专著析出文献/inbook 条目的域格式	17
例 13. 连续出版物/periodical 条目的域格式	17
例 14. 连续出版物析出文献/article 条目的域格式	18
例 15. 报纸析出的文献/newspaper 条目的域格式	18
例 16. 专利文献/patent 条目的域格式	19
例 17. 电子资源/online/electronic/www 条目的域格式	19
例 18. 学位论文/thesis/mastersthesis/phdthesis 条目的域格式	20
例 19. 报告/report/techreport 条目的域格式	21
例 20. 设置 set 条目集用于双语文献动态方法	21
例 21. 设置 set 条目集用于双语文献动态方法	22
例 22. 设置 set 条目集用于双语文献静态方法	22
例 23. 在 bib 文件中正确设置 set 条目集的静态方法	22
例 24. 在 bib 文件中正确设置关联条目的静态方法	23
例 25. 设置关联条目的动态方法	23
例 26. texlive2016 中 biblatex3.4 版 Name 域格式输入参数的修改	46

1 概述

提供了符合 GB/T 7714-2015 信息与文献 参考文献著录规则要求的 biblatex 参考文献样式。分为两种编制方式: 一、顺序编码制; 二、作者年制。主要特点包括:

1. 完成了 GB/T 7714-2015 标准的完整实现, 包括两种编制方式下的各类型参考文献著录格式和标注格式等基本内容, 还包括: 双语文献格式, 带页码的标注格式, 作者年制下仅有年的标注格式和文献按语言集中并自动排序, 起止卷期自动解析, 增加 gbnoauthor 选项控制作者年制责任者缺省的处理, 增加 gbpub 选项控制出版信息缺省时的处理, 增加 gbalign 选项控制顺序编码制文献表的标签对齐方式, 提供右对齐、左对齐和项对齐三种方式。

2. 实现了用户文献数据录入优化, 用户在录入参考文献数据时, 只需要录入文献的实际信息即可, 不需要录入文献标识符和载体标识符, 无需录入 language 或者其它域信息来区分中英文参考文献, 实现中英文自动判断并处理。支持一些特殊或老的条目类型, 比如 standard, newspaper, www, mastersthesis, phdthesis 等。

3. 实现了对 biblatex 不同版本的兼容, 能够应用于 biblatex3.2 以前的老版本, 也能用于 3.3 版姓名处理方式改变后的版本。即可以与 texlive2014/2015/2016/2017 配合使用, 无需升级 biblatex 情况下直接使用 biblatex-gb7714-2015 宏包 (即本样式)。当然 ctex2.9.4 的用户可能要升级一下 biblatex, 因为 ctex 多年没有更新, 其中的 biblatex 版本过低。

4. 测试了样式文件在 book/report/article 文档类以及 beamer 类下的适用性, 结果表明均能满足要求。文档详细介绍了样式文件的使用方法和注意事项, 说明了各条目类型的著录格式及其在 biblatex 中对应信息域的构成, 以及域信息的录入方法, 并严格按照 GB/T 7714-2015 标准测试了各种类型的

文献。

样式文件由如下文件构成:

顺序编码制的 `gb7714-2015.bbx`, `gb7714-2015.cbx` 文件和作者年制的 `gb7714-2015ay.bbx`, `gb7714-2015ay.cbx` 文件。

说明文档源文件包括:

`biblatex-gb7714-2015.tex`, `eg*.tex` 分别是本样式说明文档和测试用例。

脚本文件包括:

`*.bat`和`*.sh`分别是 windows 下和 linux 下的编译脚本。`*.pl`是用来将 gb7714 格式的著录文献文本转换为 bib 文件的 perl 脚本, `*.dat`是用于转换的测试文本文件。

1.1 biblatex 宏包的参考文献生成方法

biblatex 生成参考文献有其优越性。笔者从最初开始学习 latex 时利用 thebibliography 环境生成参考文献, 到后来对格式化参考文献有更多需求, 开始寻求利用参考文献宏包, 再到最后选择了使用 biblatex 宏包 [也由于对 bibtex 语言不熟悉, 偷懒不想学 (L)], 经过大量的实践总结 biblatex 宏包参考文献生成的优点如下:

1. 使用简单。样式随宏包加载, 参考文献源 bib 文件利用 addbibresource 加载, 打印文献表利用 printbibliography 命令可以在文档任意位置打印。参考文献只需三步编译, 第一遍 xelatex, 第二遍 biber, 第三遍 xelatex, 当然如果需要反向超链接, 除设置 backref 选项外, 还需要第四遍 xelatex。例1 给出编译命令, 其中 xelatex 命令也可以用 mkllatex -xelatex 代替, -synctex=-1 选项也可以是-synctex=1。

示例 1. 使用 biblatex 宏包的文档编译命令

代码

```
1 xelatex --synctex=-1 jobname.tex
2 biber jobname
3 xelatex --synctex=-1 jobname.tex
4 xelatex --synctex=-1 jobname.tex
```

2. 划分自由。利用 biblatex 宏包可以在一个文档中生成任意数量的参考文献表, 而不需要用 include 把分档划分成不同的文件来生成分章参考文献。利用 refsection 和 refsegment 可以很方便地划分参考文献节, 甚至还可以嵌套使用。

3. 支持全面。后端处理程序 biber 处理大数据量参考文献毫无压力, 不用担心 bibtex 可能存在的内存不足等局限, 字符编码也支持 utf-8, 完全支持中文的 bibtex 键 (引用关键字)。此外, biber 还具有根据拼音和笔画排序 (例2)、输出引用文献的数据 (例3) 等其它更多功能。

示例 2. 中文文献排序时的 biber 选项

代码

```
1 %按拼音排序, biber命令
2 biber -l zh_pinyin jobname
3 %按笔画排序, biber命令
4 biber -l zh_stroke jobname
```

示例 3. 输出引用文献数据时的 biber 选项

代码

```
1 biber jobname.tex --output-format=bibtex
```

4. 定制方便。biblatex 宏包实现参考文献著录格式使用的是 tex 语言, 相比传统的 bibtex 语言学习更为容易。biblatex 提供了很多不同类型的参考文献标准样式, 查看、参考和引用都很方便, 因此定制起需要的参考文献格式也很便捷。

下面直接给出最小工作示例的 tex 源文档 (例4), 用以简单介绍 biblatex 宏包参考文献生成的一般方法。给出的详细注释, 说明了使用 biblatex 的文档基本结构, 所有基于 biblatex 生成参考文献的文档无论大小万变不离其宗, 当然除 tex 文档外另需准备保存参考文献信息的 bib 文件 (详见1.3.1节)。要进一步了解 biblatex 参考文献生成相关内容可以参考: biblatex 宏包手册或者 LaTeX 文档中文参考文献的 biblatex 解决方案^②。

示例 4. biblatex 参考文献生成最小工作示例

代码

```
1 \documentclass{article}%文档类
2 %引言区开始:
3 %加载ctex宏包, 中文支持
4 \usepackage{ctex}
5 %加载geometry宏包, 定义版面
6 \usepackage[left=20mm,right=20mm,top=25mm, bottom=15mm]{geometry}
7 %加载hyperref宏包, 使用超链接
8 \usepackage[colorlinks=true,pdfstartview=FitH,linkcolor=blue,anchorcolor=violet,citecolor=
   magenta]{hyperref}
9 %加载biblatex宏包, 使用参考文献
10 %其中后端backend使用biber
11 %标注(引用)样式citestyle, 著录样式bibstyle都采用gb7714-2015样式
12 \usepackage[backend=biber,bibstyle=gb7714-2015,%nature,%
13 citestyle=gb7714-2015%,backref=true%
14 ]{biblatex}
15 %biblatex宏包的参考文献数据源加载方式
16 \addbibresource[location=local]{example.bib}
17
18 %正文区开始:
19 \begin{document}
20 %正文内容, 引用参考文献
21 详见文献\cite{Peebles2001-100-100}\parencite{Miroslav2004--}
22 参考文献\cite[见][49页]{蔡敏2006--}\parencite[见][49页]{Miroslav2004--}
23
24 %打印参考文献表
25 \printbibliography[heading=bibliography,title=参考文献]
26 \end{document}
```

1.2 gb7714-2015 样式加载方式和选项说明

1.2.1 样式加载方式

例4中给出了宏包和样式的基本加载方式, gb7714-2015 样式类似。注意, 当著录样式 bibstyle 和标注样式 citestyle 相同时, 两者可以合并为一个选项 style。比如:

^②地址:<https://github.com/hushidong/biblatex-solution-to-latex-bibliography>

示例 5. gb7714-2015 顺序编码制加载方式

代码

```
1 \usepackage[backend=biber,bibstyle=gb7714-2015,citestyle=gb7714-2015]{biblatex}
2 %如果要设置参考文献表序号标签对齐方式的话请设置gbalign选项，如果要取消缺省出版项自动填补的话设置
  gbpub=false，比如：
3 \usepackage[backend=biber,style=gb7714-2015,gbalign=left,gbpub=true]{biblatex}
```

示例 6. gb7714-2015 作者年制加载方式

代码

```
1 \usepackage[backend=biber,style=gb7714-2015ay]{biblatex}
2 %如果要取消缺省出版项自动填补的话设置gbpub=false，如果要使用佚名或NOAUTHOR代替补充缺失的author
  信息可以设置gbnoauthor=true，比如：
3 \usepackage[backend=biber,style=gb7714-2015ay,gbpub=true,gbnoauthor=true]{biblatex}
```

示例 7. 参考文献文本转换为 bib 文件 perl 脚本使用方式

代码

```
1 perl gb7714texttobib.pl in=textfilename out=bibfilename
```

1.2.2 增加的选项

gb7714-2015 样式增加了部分用于控制标签对齐、出版项缺省处理、责任者（作者）缺省处理的选项，使用方式与 biblatex 宏包选项完全相同：

gbalign =right, left, gb7714-2015. default is right
为顺序编码制增加的选项，用于选择参考文献表序号标签的对齐方式。gbalign=right 是默认的右对齐；gbalign=left 是左对齐；gbalign=gb7714-2015 是项对齐方式。对于作者年制该选项无效。测试结果见第3.9节

gbpub =true, false. default is true
为控制出版信息缺失处理增加的选项。默认选择 gbpub=true，自动利用：[出版地不详]，[出版者不详]，[S.l.]，[s.n.] 等填补缺省信息；设置 gbpub=false 时，则取消自动处理，使用标准样式的方式取消相应项的输出。测试结果见第3.10节。

gbnoauthor =true, false. default is false
为作者年制增加的选项，用于控制责任者缺失时的处理。默认选择 gbnoauthor=false，当作者信息缺失时不做处理，使用标准样式的处理方式；设置 gbnoauthor=true 时，则根据 GB/T 7714-2015 的要求进行处理，中文文献使用佚名来代替 author，英文文献用 NOAUTHOR 来代替 author。测试结果见第3.11节。

gbnamefmt =uppercase, lowercase, none, pinyin. default is uppercase
为姓名大小写格式控制增加的选项。默认选择 gbnamefmt=uppercase，大小写要求符合 GB/T 7714-2015 的要求。gbnamefmt=lowercase 时，大小写根据输入信息确定。gbnamefmt=none 时，姓名的格式与 biblatex 标准样式一致。gbnamefmt=pinyin 时，姓名的格式采用一种常用的中文拼音全拼方式，比如对于 Zhao, Yu Xin 或 Yu Xin Zhao 这个姓名拼音格式化为 ZHAO Yu-xin。需要注意当使用 pinyin 选项时，姓名中应给出完整的名而不是名的缩写，否则出来的效果未必好看。测试结果见第3.12节。

gbtype =true, false. default is true
为控制是否输出题名后面的文献类型和载体标识符而增加的选项。默认选择 gbtype=true，根据 GB/T 7714-2015 要求输出标识符，例如“在线的期刊析出文献题名 [J/OL]”。而 gbtype=false 时，不输出标识符，例如“在线的期刊析出文献题名”。

1.2.3 与标准选项的兼容性

绝大部分 biblatex 标准样式选项可与 gb7714-2015 样式联合使用，下面给出一些经过兼容性测试的选项说明。需要注意的是：使用 gb7714-2015 样式时（即 `style=gb7714-2015`），backend 选择应指定为 biber，还有一些选项已经在样式设计中固定，不能修改设置。比如 `sorting`, `maxnames`, `minnames`, `date`, `useprefix`, `giveninits` 等。

url =true, false. default: true

该选项控制是否打印 url 域并获取日期。该选项只影响 url 信息是可选的那些条目类型。而 @online 条目的 url 域总是会打印出来。它是导言区选项，与样式相关，gb7714-2015 样式做了特别支持，可以兼容使用。

uniquelist =true, false, minyear default: minyear

该选项用于作者年制样式，用于正文中引用（标注）标签的作者列表控制（目的是消除歧义）。当 `uniquelist=true` 时，自动利用扩展作者姓名列表长度的方式消除 labelname 列表的歧义；当 `=false` 时则禁用扩展，标签仅使用一个作者，消除歧义通过跟在年份后面的字母实现；默认使用 `minyear`，即当被截短的作者姓名列表存在歧义时，只有当年份相同，才会扩展列表长度以消除歧义。

注意当使用 `uniquelist=false` 后标签只有一个作者，但文中可能有同姓作者的文献，这时根据 `unique-name` 选项的设置，biblatex 会使用姓名的其它部分比如名来消除歧义，但如果想强制要求仅用姓作为文中的标注标签，那么可以设置 `uniqueusername=false`，但此时标注是可能存在歧义的。

hyperref =true, false, auto. default: auto

是否将引用和后向引用转化为可点击的超链接。这是宏包的载入时选项，与样式无关，自然可以使用。

backref =true, false. default: false

是否在文献中打印出反向引用。这是宏包的载入时选项，与样式无关，自然可以使用。

refsection =none, part, chapter, section, subsection. default: none

该选项自动在文档分段处（例如一章或一节）开始一个新的参考文献分节。是宏包的载入时选项，与样式无关，自然可以使用。需要注意与 `titlesec` 宏包联用时，该选项会失效。

refsegment =none, part, chapter, section, subsection. default: none

类似于 `refsection` 选项，但开始的是一个新的参考文献片段。

citereset =none, part, chapter, section, subsection. default: none

该选项在文档分段处（例如一章或一节）自动执行 `citereset` 命令。

labeldate =year, short, long, terse, comp, ymd, edtf. default: year

类似于 `date` 选项，但控制的是由 `DeclareLabeldate` 选择的日期域的格式。

其它 = 下面还有很多选项，有些是宏包载入时选项，与样式无关，一般可以使用，但笔者没有做测试，各位朋友可以测试使用。选项的作用可以参考 biblatex 使用手册，以及笔者和 Wenbo 翻译的中文版。

- `related=true, false`. default: true
- `defernumbers=true, false` default: false
- `sortcites=true, false` default: false
- `maxitems=integer` default: 3
- `minitems=integer` default: 1

- autocite=plain, inline, footnote, superscript, ...
- autopunct=true, false default: true
- language=autobib, autocite, auto, language default: autobib
- clearlang=true, false default: true
- autolang=none, hyphen, other, other*, langname default: none
- block=none, space, par, nbpar, ragged default: none
- notetype=foot+end, footonly, endonly default: foot+end
- indexing=true, false, cite, bib default: false
- backrefstyle=none, three, two, two+, three+, all+ default: three
- backrefsetstyle=setonly, memonly, setormem, setandmem, memandset, setplusmem default: setonly
- loadfiles=true, false default: false
- abbreviate=true, false default: true
- julian=true, false default: false
- punctfont=true, false default: false
- arxiv=abs, ps, pdf, format default: abs
- mincrossrefs=integer default: 2
- minxrefs=integer default: 2
- isbn=true, false default: true
- doi=true, false default: true
- eprint=true, false default: true

1.3 样式文件的进一步说明

1.3.1 关于参考文献数据源: bib 文件的说明

参考文献数据以 bibtex 格式保存在 bib 文件中。生成参考文献除 tex 源文档外, 还需创建参考文献数据源文件即 bib 文件。bib 文件数据源准备完成后, 在加载 biblatex 宏包时, 使用 addbibresource 命令将其加载进 tex 源文档中。**注意: 数据源可以加载多个, 比如多个章节的参考文献放在不同的 bib 文件中, 那么全部加载进来即可。**

bib 文件中的参考文献信息是以条目形式组织, 一篇文献创建一条记录即一个参考文献条目, 一个条目由若干个数据域构成。文献的各部分信息应录入到条目的对应数据域中。GB/T 7714-2015 标准中的文献类型与本样式中条目类型对应关系如表1所示, 各类条目具体的著录格式详见2.3节。

组成各个条目的不同数据域 (字段) 保存有参考文献的各部分信息, 比如作者、标题、出版项、日期等等, 这些在标准中称为著录项目, 其对应关系如表2所示。需要注意的是, note 域在本样式中也做特殊用途, 即在 book 类型的 note 域中输入 standard 表示标准, 在 article 类型的 note 域中输入 news 表示报纸, 当然也可以不做特殊用, 而只是表示杂项信息, 因为标准和报纸可以用 standard 和 newspaper 类型表示, 尽管这两种类型不是 biblatex 原生的条目类型。而 usera 域用于表示参考文献类型和载体标识符, 一般不用出现在 bib 文件中, 而由样式文件自动处理得到, 这也为了使 bib 文件更纯粹, 也为了兼容不同的样式, 因为如果在 bib 文件中给出 usera 域, 如果某一样式需要使用 usera

表 1. 常见条目类型

GB/T 7714-2015 标准中的参考文献类型	biblatex 中的条目类型
专著	book
标准	standard/book or inbook with field note=standard
专著中的析出文献	inbook
连续出版物	periodical
连续出版物的析出文献	article
报纸析出的文献	newspaper/article with field note=standard
专利	patent
电子资源	online/www/electronic
会议录或会议文集	proceedings
会议文集中析出的文献	inproceedings/conference
汇编或论文集	collection
汇编或论文集析出中的文献	incollection
学位论文	thesis/mastersthesis/phdthesis
报告	report/techreport
手册或档案	manual
未出版物	unpublished

域且用途不同，那么就会有兼容性问题。当然不用手动输入这一信息也可以为用户减负，因为用户可以直接使用从网络导出参考文献信息而不用再额外添加一个参考文献类型和载体标识符。本宏包自动判断语言，而不使用一个表示语言的域比如 `language` 来标记文献的语言类型也处于同样的考虑，当然还有一篇文献中存在多语言的情况是无法用一个 `language` 标识的，比如一本英文著作被翻译为中文，那么原作者是英文，而译者则是英文，这时标记 `language` 就存在困难。

各个数据域的录入应符合 bib 文件规范。需要注意：有时直接从网络获取的参考文献信息中可能带有一些特殊字符比如 `%`, `&` 等，这些字符在 `tex` 中通常需要做转义处理，本样式中对像 `title`, `journal` 等常见的域中出现的特殊字符已经做了转义，但是一些不常见的域比如 `abstract` 等没有考虑，所以用户需要手动处理，例如把 `%` 改为 `\%`，否则可能导致出错。下面详细介绍本样式中使用的域及其数据录入方式：

author 在 biblatex 中 `author` 域属于 `name` 数据类型，输入数据时，各姓名间用 `and` 连接，当姓名过多省略时，用 `others` 代替。

单个姓名，对于中文作者直接输入中文姓名即可。比如：于潇 and 刘义 and 柴跃廷 and others
对于英文作者，单个姓名有两种 biblatex 可以解析的输入方式：

① `prefix lastname, suffix, firstname middlename`

② `firstname middlename lastname or firstname prefix lastname`

对于需要输入前后缀的姓名只能采用第一种方式，比如：DES MARAIS, Jr., D J and H STRAUSS and SUMMONS, R. E. and others

这里的第一个姓名输入为前缀，姓，后缀，名，中间名。第二个姓名输入为名，姓。第三个姓名输入为姓，名，中间名。

需要强调：对于第二种输入方式，姓名各个组成部分最好首字母是大写的，首字母非大写可能导致

表 2. 常见信息域

GB/T 7714-2015 中的文献著录项目	biblatex 中的域
责任者	author
题名	title
译者	translator
版本 (主要用于出版物)	edition
版本 (主要用于软件和手册)	version
出版地	location/address
出版者	publisher
出版者 (大学和研究所)	institution/school
出版者 (会议主办方、手册和电子资源出品方)	organization
日期	date
日期 (不可解析的日期)	year
页码	pages
析出文献来源的责任者	bookauthor
析出文献来源的标题	booktitle
连续出版物题名 (期刊名)	journal/journaltitle
期刊的卷	volume
期刊的期/专利号等	number
获取和访问路径	url
引用日期	urldate
数字对象标识符	doi
杂项	note
文献类型载体标识符	usera

解析出错，比如姓名只有两个组成部分 `firstname` 和 `lastname`，如果 `firstname` 小写的话，有可能会解析为 `prefix lastname`。对于第一种输入方式，则至少 `lastname` 需要首字母大写，否则有可能将 `lastname` 解析成 `prefix`。其中 `lastname` 也称 `famillyname`，`firstname middlename` 两者共称 `givenname`

对于机构作者，不需要解析，直接输入机构名，英文的各个机构名用 `{}` 包起来，比如：

中国企业投资协会 and 台湾并购与私募股权协会 and 汇盈国际投资集团

`{International Federation of Library Association and Institutions}`

title 直接输入需要打印的内容，`subtitle` 或 `titleaddon` 域类似

usera 不用输入，自动处理

translator 与 `author` 域类似，只是输入的是译者

edition 直接输入整数，或者需要打印的内容

location 直接输入需要打印的地址内容，而 `address` 域在 `biblatex` 中作为 `location` 别名，表示相同的内容。

publisher 直接输入需要打印的出版者内容，`institution`，`organization` 域类似

date 日期可以格式化输入，格式化输入 `biblatex` 会自动解析，如果无法解析会忽略该域。格式化的输入方式是：

年-月-日/年-月-日

比如:2001-05-06/2001-08-01

其中第一个年-月-日会解析并存储到 year, month, day 域中, 第二个会解析并存储到 endyear, endmonth, endday 域中。更多细节参考 biblatex 手册的 Table 8: Date Interface。

year year 域的输入与 date 域类似, 为了兼容一些老的 bib 文件, 把 year 直接用 map 转换成 date, 所以在本样式的使用中输入 year 域与 date 域相同。

但 year 与 date 存在一定的差异, 即 year 可以处理仅有年的信息或者需要原样打印的内容。比如: 1881(清光绪七年)。

这一信息如果放在 date 中会被自动忽略, 但放到 year 域中, 本样式会先将其拷贝到 date 中进行解析, 无法解析的话, date 域忽略, 但 year 信息仍然存在, 并原样打印。

pages 可以格式化输入或输入需要打印的内容。格式化输入时, 页码用整数, 当有范围时, 用短横线 (使用多个短横线也没有问题) 隔开。比如:59-60。当无法解析时, 输入内容被认为是需要完整打印的内容。

urldate urldate 域与 date 域类似, 只是解析时, 存储到 urlday, urlmonth, urlyear, urlendday, urlendmonth, urlendyear 域中。

url 直接输入需要打印的网址内容

doi 直接输入需要打印的 DOI 内容

note 在本样式中 note 域有特殊功能, 当其内容为 standard 或 news 时, 判断条目类型为标准和报纸析出的文献。

bookauthor 用于析出文献时, 作为析出文献来源文献的作者, 其输入方式与 author 相同。

booktitle 用于析出文献时, 作为析出文献来源文献的题名, 其输入方式与 title 相同。booktitleaddon 域输入方式也相同。

volume 连续出版物的卷, 格式化输入用整数, 当有范围时中间用短横线连接, 比如:1-4。当无法解析时, 输入内容被认为是需要完整打印的内容。

number 连续出版物的期或报纸的版次, 输入与 volume 类似。或者是专利等的号时, 直接输入需要打印的内容。

journal 用于连续出版物析出文献, 表示连续出版物的题名, 直接输入需要打印的内容。journaltitle, journalsubtitle 域类似处理。

version 用于 report 和 manual 的版本信息, 直接输入需要打印的内容。

除了上述输入内容要求外, GB/T 7714-2015 还有对数字、字母大小写等有一些格式要求, 这些细节需要注意, 请参考:

- 数字:[2.5.1节](#)
- 字母大小写:[2.5.2节](#)
- 卷和期:[2.5.10节](#)
- 版次:[2.5.6节](#)
- 出版项:[2.5.7节](#)
- 页码:[2.5.8节](#)

1.3.2 关于参考文献著录样式: bbx 文件的说明

参考文献著录样式也称著录表样式或著录格式, 主要分两类: 顺序编码制和编码年制。顺序编码制样式中各条参考文献条目以数字序号按引用先后顺序组织。著录格式中序号格式见2.1.1节, 各类型文献条目的著录格式见2.3节, 参考文献条目中各信息域及其录入方式见1.3.1节。

作者年制样式中各条参考文献条目以作者-年为标签以一定的顺序排列。作者年制的著录格式与顺序编码制基本相同 (除了把年份提到了作者后面作为文献条目内的标签)。数据源 bib 文件中各条目的数据录入与顺序编码制完全一致。

注意: 因为作者年制有分文种文献集中的要求, 所以修改 nyt 排序格式, 增加 userb 作为 name 前的排序域, 默认情况下, 本样式文件将作者为中文的文献的 userb 域设置成 cn, 英文的设置成 en。这一设置过程, 在 biber 处理时自动完成。当出现问题或者有更多文种分集且有特殊顺序时, 可以为 bib 文件中给相应文种文献的 userb 域手动设置适合排序的字符串。比如: 中文文献设置为 cn, 英文文献设置为 en, 法文文献设置为 fr, 那么排序中, 相应的中文文献排在最前面, 英文文献在中间, 法文文献最后, 因为升序情况下字母顺序是 c 然后 e 然后 f。

1.3.3 关于参考文献标注样式: cbx 文件的说明

正文中的参考文献标注样式也称引用样式或引用标签样式, 也分两类: 顺序编码制和作者年制。顺序编码制的标注样式大体使用标准标注样式 numeric, 仅对 cite 等部分命令做了修改。为满足 GB/T 7714-2015 第 10.1.3 节的要求, 增加了 pagescite 命令。为方便在同一文章中同时使用上标和非上标的标注方式, 仅将 cite 命令修改为上标模式, 而 parencite 保留标准样式提供的非上标模式。这些命令使用方式如下:

示例 8. 顺序编码制 cbx 样式命令使用说明

代码

```
1 不带页码的引用(上标, 方括号包围):
2   \cite{Peebles2001-100-100}
3 不带页码的引用(非上标, 方括号包围):
4   \parencite{Miroslav2004--}
5 带页码的引用:
6   \cite[见][49页]{蔡敏2006--} \parencite[见][49页]{Miroslav2004--}
7   \pagescite{Peebles2001-100-100} \pagescite[][201-301]{Peebles2001-100-100}
8 在页脚中引用和打印文献表:
9   \footnote{在脚注中引用\footcite{赵学功2001--}} \footfullcite{赵学功2001--}
```

其中, 当不指定页码时, pagescite 命令默认调用参考文献的页码数据进行输出, 如果需要指定页码, 那么需要在第二个 [] 内输入页码内容。

注意: 对于多个文献一起的压缩形式, 指定页码只会应用最后一个参考文献的页码, 这是不正确的, 但这种情况其实本不应出现, 因为指定页码本来就需要具体化指某一文献。使用时请尽可能使用 pagescite{key1}pagescite{key2} 这种形式而不是 pagescite{key1,key2}。

作者年制的标注样式情况类似。对 cite, parencite 进行了修改, 将引用标签用圆括号括起来。为满足 GB/T 7714-2015 第 10.2.4 节的要求, 增加了 pagescite 命令。为满足 GB/T 7714-2015 第 10.2.1 节

的要求,增加了 yearpagescite 命令用于处理文中已有作者信息只需要年份和页码的情况 (为兼容考虑,顺序编码制也给出该命令,但作用与 pagescite 命令相同)。使用方式如下:

示例 9. 作者年制 cbx 样式命令使用说明

代码

```
1 不带页码的引用:
2   \cite{Peebles2001-100-100} \parencite{Miroslav2004--}
3 带页码的引用:
4   \cite[见][49页]{蔡敏2006--} \parencite[见][49页]{Miroslav2004--}
5   \pagescite{Peebles2001-100-100} \pagescite[][201-301]{Peebles2001-100-100}
6 作者年制文中已有作者只需要年份和页码的情况,使用命令yearpagescite,比如:
7   见赵耀东\yearpagescite[][205]{赵耀东1998--}和Simon\yearpagescite[][15]{Simon2001--}。
8
9 需要注意,\yearpagescite命令在输出年份的同时也输出页码,但有时有些参考文献格式不要求页码,此时可
   以用\yearcite命令,或者也可利用cite*和citeyear命令手动实现(如果要用超链接目前可以使用cite*),
   比如:
10  见赵耀东\yearcite{赵耀东1998--}
11  见赵耀东(\cite*{赵耀东1998--})
12  见赵耀东(\citeyear{赵耀东1998--})
13 在页脚中引用和打印文献表:
14  \footnote{在脚注中引用\footcite{赵学功2001--}} \footfullcite{赵学功2001--}
```

各引用命令的效果见3.2, 3.14节。

2 GB/T 7714-2015 标准说明与实现

2.1 顺序编码制

2.1.1 参考文献表

GB/T 7714-2015 规定采用顺序编码制组织参考文献时,各篇文献应按正文部分标注的序号依次列出。具体参考 GB/T 7714-2015 第 9.1 节。

2.1.2 文献标注法

标注则根据在正文中引用的先后顺序连续编码,将序号置于方括号内。

同一处引用多篇文献,各篇序号间用逗号隔开,遇连续序号,起讫序号用短横线连接。

多次引用同一著者的同一文献时,可在序号的方括号外著录该文献引文页码,这一要求与引用(标注)样式无关,需要作者在写文档时使用相应的引用命令并在需要时输入页码信息。针对这一要求,在 cite 等常用命令基础上,新定义了一个引用命令 pagescite,其使用方式详见第1.3.3节。其标注效果见第3.2节。标注样式更详细要求参考 GB/T 7714-2015 第 10.1 节。

如果顺序编码制采用脚注方式,则序号由计算机自动生成圈码。多次引用同一著者的同一文献时,若采用脚注方式应重复著录参考文献(这里可以理解为,采用该方式,同一文献的不同页码的引文就相当于一篇新的引文),只是在参考文献列表中可以简化(当然不进行简化,简单复制后录入对于 latex 的参考文献处理其实更方便)。事实上对于顺序编码用脚注方式,GB/T 7714-2015 并没有详细说明脚注方式到底是什么?从举例看只是序号用类似于脚注的标签,那么对于参考文献样式来说,与非脚注

方式的差别仅在于引用和参考文献条目的序号标签的差别，如此是容易通过修改样式文件得到的，但实际使用中除非有特殊要求，否则使用意义不大，因此本样式没有实现该功能。

2.2 作者年制

2.2.1 参考文献表

GB/T 7714-2015 规定采用作者年制组织时，各篇文献首先按文种组织，可分为中文，日文，西文，俄文和其他文种等部分；然后按照著者字顺和出版年排列。中文文献可以按著者汉语拼音字顺排序，也可按笔画顺序排列。具体参考 GB/T 7714-2015 第 9.2 节。

2.2.2 文献标注法

各篇文献的标注内容由著者姓 (lastname) 和出版年构成，并置于 () 内。对于使用汉字的语言来说，整个姓名都是 lastname 所以标注的是全名。机构团体名也整体标注。

若正文中已有著者姓名，则 () 内只标注出版年，这一点样式文件无法判断，只能是文档作者自身把握，当然样式文件提供了标签只有年份、附加年份和页码信息的引用命令 `yearpagescite`，方便文档写作者使用，使用方法详见第1.3.3节。

引用多个著者的文献时，对西文只需标注第一著者的姓 (而在参考文献列表中的作者按最大数量三个处理，这与顺序编码制一致，参考 GB/T 7714-2015 第 8.1.2 节)，其后附 “et al.”，对于中文著者，标注第一著者的姓名，其后附 “等”。姓名与 “et al.” “等” 间留适当空隙。

注意到在 GB/T 7714-2015 第 10.2.1 节给出的例子中作者姓的大小写格式与参考文献表中的要求是不同的，这说明标注中的作者姓名是由写文档的作者来决定的，因此本样式文件原样输出 bib 源文件中作者姓的大小格式。

引用同一著者同一年出版的多篇文献时，出版年后应采用小写字母 a,b,c 等区别。

多次引用同一著者的同一文献，在正常标注外，需在 () 外以角标形式著录引文页码，这一问题样式文件无法判断，只能提供可以形成该格式的引用命令，供文档作者使用，因此提供 `pagescite` 命令，使用方法详见第1.3.3节。

标注要求具体参考 GB/T 7714-2015 第 10.2 节。

注意：一般情况下，当文献作者缺省时，作者年制就没有作者可以用，因此文献题名用来生成标签，这样会导致文献表中文献题名后的文献类型标识/文献载体标识消失 (这是因为题名用于生产标签后，题名域会被清除，自然也就不输出题名相关的信息了，见 “[Information and documentation-the Dublin core metadata element set](#)”)。但是可以用佚名替代缺省作者的方式避免这个问题，即可以使用样式文件提供的选项 `gbnoauthor=true`，一旦设置该选项为 true，则缺省的作者会根据文献语种填充为佚名或 NOAUTHOR。默认情况下，不进行这种处理，即相当于设置选项 `gbnoauthor=false`。而顺序编码制因为标签是数字序号，所以不存在这个问题。

2.3 各类文献在 biblatex 中对应的条目和域

gb7714-2015.bbx 是按照 GB/T 7714-2015 要求实现的 biblatex 参考文献著录样式文件。

根据 GB/T 7714-2015 要求并结合 biblatex 的条目类型和数据域，对各类参考文献做如下考虑：

2.3.1 专著/book

条目类型 1 专著对应的 biblatex 的 entrytype 为:book, 文献类型标识用 M 表示。

其著录格式为 (参考 GB/T 7714-2015 第 4.1 节):

主要责任者. 题名: 其他题名信息 [文献类型标识/文献载体标识]. 其他责任者. 版本项. 出版地: 出版者, 出版年: 引文页码 [引用日期]. 获取和访问路径. 数字对象唯一标识符.

其对应的 biblatex 数据域为:

示例 10. 专著/book 条目的域格式

代码

```
1 author.title[usera].translator.edition.location:publisher,date或year:pages[urldate].url.doi
```

其中标题相关的附加信息除了可以直接在 title 域中录入外, 还可以在 subtitle 或 titleaddon 域中添加, 后面出现的 booktitle, journaltitle, 也有类似情况, 可以在 booktitleaddon 或者 journalsubtitle 中附加信息。其中出版地用 location 域表示, 也可以用传统的 address 表示, biblatex 将 address 作为 location 的别名处理, 使用两者中的任何一个都可以表示出版地信息。**特别强调:usera 域不用录入, 该域内容由 bbx 样式文件根据条目类型自动处理得到。**

注意: 由于 biblatex 不支持 standard 条目类型, 所以“标准”类型可以用 book 或 inbook 替代, 但使用 note 域等于 standard 作为一个区分, 当 note 域数据存在且内容等于 standard 时, 就将其作为“标准”文献进行处理, 其文献类型标识用 S 表示。这里为什么使用 note 域而不是 type 域和 keywords 域, 是因为考虑到 note 域一般情况没有什么特殊意义, 使用它不会导致冲突, 而 type 域在 biblatex 标准样式中没有被 book 和 article 条目类型当作支持的域, 对于支持该域的条目比如 thesis, type 域又有特殊的意义, 是用来区分 master 和 doctor 的, 而 keywords 域倒是可以使用, 只是该域一般很少在 jabref 之类软件的默认域中, 需要进一步设置, 而且可能带来不通用的问题。

2.3.2 标准/standard

“标准”(standard) 作为一种文献条目类型 biblatex 并不支持, 因此可以如上一小节所述, 直接使用 book 或 inbook 类型加 note 域代替。或者使用本节下面给出的方式。

因为在一些 bib 文件中可能存在 standard 类型, 为兼容性考虑, 本样式实现了对 standard 条目类型的支持。著录格式的处理原理与前一节所述相同, 只是利用动态数据将 standard 类型转换为 book/inbook 类型。因此在 bib 文件中可以直接使用 standard 类型, 但要注意使用其它样式时可能存在移植障碍, 因为其它样式可能不支持 standard 类型, 为样式移植性考虑最好使用上一节的方式。

条目类型 2 标准对应的 biblatex 的 entrytype 为:standard。文献类型标识用 S 表示。

其著录格式为 (与 book 和 inbook 类型类似, 其中圆括号内是与 inbook 类似时存在的内容, 同时当出版地和出版者不存在时直接忽略, 这是与 book 和 inbook 不同的地方。):

主要责任者. 文献题名 [文献类型标识/文献载体标识]. 其他责任者 (//所在文献集主要责任者. 文献集题名: 其他题名信息). 版本项. 出版地: 出版者, 出版年: 文献的页码 [引用日期]. 获取和访问路径. 数字对象唯一标识符.

其对应的 biblatex 数据域为:

代码

```
1 author.title[usera] (//bookauthor.booktitle).edition.location:publisher,date或year:pages[urldate].url.doi
```

2.3.3 专著中的析出文献/inbook

条目类型 3 专著中的析出文献对应的 biblatex 的 entrytype 为:inbook。文献类型标识用 M 表示。

其著录格式为 (参考 GB/T 7714-2015 第 4.2 节):

析出文献主要责任者. 析出文献题名 [文献类型标识/文献载体标识]. 析出文献其他责任者//专著主要责任者. 专著题名: 其他题名信息. 版本项. 出版地: 出版者, 出版年: 析出文献的页码 [引用日期]. 获取和访问路径. 数字对象唯一标识符.

其对应的 biblatex 数据域为:

示例 12. 专著析出文献/inbook 条目的域格式

代码

```
1 author.title[usera]//bookauthor.booktitle.edition.location:publisher,date或year:pages[urldate].url.doi
```

2.3.4 连续出版物/periodical

条目类型 4 连续出版物对应的 biblatex 的 entrytype 为:periodical。文献类型标识用 J 表示。

其著录格式为 (参考 GB/T 7714-2015 第 4.3 节):

主要责任者. 题名: 其他题名信息 [文献类型标识/文献载体标识]. 年, 卷 (期)-年, 卷 (期). 出版地: 出版者, 出版年 [引用日期]. 获取和访问路径. 数字对象唯一标识符.

其对应的 biblatex 数据域为:

示例 13. 连续出版物/periodical 条目的域格式

代码

```
1 author/editor.title[usera].year或date,volume(number)-endyear, endvolume(endnumber).location:institution,date或year[urldate].url.doi
```

其中连续出版物的出版者用 institution 表示。需要注意: 因为连续出版物可能用到两个日期, 两个卷, 两个期, 所以录入数据时需要特别处理。不需要录入 endyear 等信息, 只需要在到 year 或 date 域录入日期, 由 biber 自动解析, 两个日期之间用/分隔。而卷和期由于可能有合订模式, 合订卷期之间用/分隔 (参考 GB/T 7714-2015 第 8.8.3 节), 因此如果需要解析有起止范围的卷和期, 录入到 volume 和 number 域的信息中起止值之间应用-分隔。

2.3.5 连续出版物的析出文献/article

条目类型 5 连续出版物的析出文献对应的 biblatex 的 entrytype 为:article。文献类型标识用 J 表示。

其著录格式为 (参考 GB/T 7714-2015 第 4.4 节):

析出文献主要责任者. 析出文献题名 [文献类型标识/文献载体标识]. 连续出版物题名: 其他题名信息, 年, 卷 (期): 页码 [引用日期]. 获取和访问路径. 数字对象唯一标识符.

其对应的 biblatex 数据域为:

示例 14. 连续出版物析出文献/article 条目的域格式

代码

```
1 author.title[usera].journaltitle或journal,year,volume(number):pages[urldate].url.doi
```

需要注意: 由于 biblatex 不支持 newspaper 条目类型, 所以条目类型报纸析出的文献用 article 表示, 但使用 note 域等于 news 作为一个区分, 当 note 域数据存在且内容等于 news 时, 就将其作为报纸的析出文献进行处理。报纸文献类型标识用 N 表示, 报纸的版次用 number 域描述。

2.3.6 报纸析出的文献/newspaper

biblatex 没有将报纸的析出文献 (newspaper) 作为一种文献条目类型, 因此可以如上一小节所述, 直接利用 article 类型加 note 域代替。或者使用本节下面给出的方式。

因为报纸析出的文献的文献标识码比较特殊不是 J 而是 N, 所以可以把报纸析出的文献独立出来进行处理。为方便使用考虑, 本样式增加了对一个全新的类型 newspaper 的支持, 这种支持通过类似于 standard 类型的方式实现, 没有对数据模型进行改动或增加, 而完全利用动态数据修改将 newspaper 类型转换为 article 类型。因此在 bib 文件中可以直接使用 newspaper 类型, 但要注意使用其它样式时可能存在移植障碍, 因为其它样式可能不支持 newspaper 类型, 为样式移植性考虑最好使用上一节的方式。

条目类型 6 报纸析出的文献对应一个新的 entrytype 为:newspaper。文献类型标识用 N 表示。

其著录格式为 (类似于 article):

析出文献主要责任者. 析出文献题名 [文献类型标识/文献载体标识]. 报纸题名: 其他题名信息, 日期 (版号)[引用日期]. 获取和访问路径. 数字对象唯一标识符.

其对应的 biblatex 数据域为:

示例 15. 报纸析出的文献/newspaper 条目的域格式

代码

```
1 author.title[usera].journaltitle或journal,date(number)[urldate].url.doi
```

2.3.7 专利/patent

条目类型 7 专利文献对应的 biblatex 的 entrytype 为:patent。文献类型标识用 P 表示。

其著录格式为 (参考 GB/T 7714-2015 第 4.5 节):

专利申请者或所有者. 专利题名: 专利号 [文献类型标识/文献载体标识]. 公告日期或公开日期 [引用日期]. 获取和访问路径. 数字对象唯一标识符.

其对应的 biblatex 数据域为:

```
1 author.title:number[usera].date或year[urldate].url.doi
```

2.3.8 电子资源/online

条目类型 8 电子资源对应的 biblatex 的 entrytype 为:online 或 electronic 或者 www。文献类型标识用 EB 表示。(注意:biblatex 将 electronic 或 www 作为 online 条目类型的别名,对于标准样式来说这两者出现在 bib 文件中等同于 online,但这种等同标准样式是在驱动层进行处理的,而 gb7714 样式还需要处理文献类型标识,本样式文件做了进一步支持。因此 bib 文件中也可以直接使用 electronic 和 www)

其著录格式为 (参考 GB/T 7714-2015 第 4.6 节):

主要责任者. 题名: 其他题名信息 [文献类型标识/文献载体标识]. 出版地: 出版者, 出版年: 引文页码 (更新或修改日期)[引用日期]. 获取和访问路径. 数字对象唯一标识符.

其对应的 biblatex 数据域为:

示例 17. 电子资源/online/electronic/www 条目的域格式

代码

```
1 author.title[usera].organization,date或year:pages(date/enddate/eventdate)[urldate].url.doi
```

注意: 尽管 GB/T 7714-2015 中给出的著录格式包含出版地和出版者,但通常情况下具有出版地和出版者的文献会归类到其它条目类型中,至于存在的 url 信息,只要标识文献载体即可,即一般情况下 (出版地: 出版者, 出版年: 引文页码) 这些信息很少出现在 online[EB] 条目中。因此默认情况下,gb7714-2015 样式只处理出现 organization 中的出版项信息,此外用 date 表示更新或修改日期,urldate 表示引用 (访问) 日期。如果出现复杂情况,更新或修改日期还可以利用 enddate/eventdate 表示。

以上是 GB/T 7714-2015 直接给出著录格式的条目类型,还有一些类型并没有给出具体格式,但在例子中也有所体现,本样式文件根据这些例子,给出了著录格式。

2.3.9 汇编或论文集/collection

条目类型 9 汇编文献对应的 biblatex 的 entrytype 为:collection。文献类型标识用 G 表示。

其著录格式为 采用与 book 一致的格式。

2.3.10 汇编或论文集析出中的文献/incollection

条目类型 10 汇编中的析出文献对应的 biblatex 的 entrytype 为:incollection。文献类型标识用 G 表示。

其著录格式为 采用与 inbook 一致的格式。

2.3.11 会议录或会议文集/proceedings

条目类型 11 会议文集的 biblatex 的 entrytype 为:proceedings。文献类型标识用 C 表示。

其著录格式为 采用与 book 类似的格式。

2.3.12 会议文集中析出的文献/inproceedings

条目类型 12 会议文集中析出的文献对应的 biblatex 的 entrytype 为:inproceedings。文献类型标识用 C 表示。(注意:biblatex 将 conference 作为 inproceedings 条目类型的别名,对于标准样式来说 conference 出现在 bib 文件中等同于 inproceedings,但这种等同,标准样式是在驱动层进行处理的,而 gb7714-2015 样式还需要处理文献类型标识,本样式文件做了进一步支持。因此 bib 文件中也可以使用 conference。)

其著录格式为 采用与 inbook 类似的格式。

2.3.13 学位论文/thesis

条目类型 13 学位论文对应的 biblatex 的 entrytype 为:thesis。文献类型标识用 D 表示。(注意:biblatex 将 mastersthesis 或 phdthesis 作为 thesis 条目类型的别名,对于标准样式来说这两者出现在 bib 文件中基本等同于 thesis,但却会增加 type 信息。但这种等同,标准样式是在驱动层进行处理的,而 gb7714-2015 样式还需要处理文献类型标识并且不需要 type 信息,本样式文件做了进一步支持。因此 bib 文件中也可以使用 mastersthesis 和 phdthesis)。

其著录格式为 由 biblatex 的标准 thesis 格式修改得到。

主要责任者. 题名: 其他题名信息 [文献类型标识/文献载体标识]. 其他责任者. 出版地: 出版者, 出版年: 引文页码 [引用日期]. 获取和访问路径. 数字对象唯一标识符.

其对应的 biblatex 数据域为:

示例 18. 学位论文/thesis/mastersthesis/phdthesis 条目的域格式

```
1 author.title[usera].translator.location:institution,date或year:pages[urldate].url.doi
```

代码

注意: thesis 和后面 report 及 manual 的出版者用 institution 域表示,体现的是机构而不是一般的出版社。

2.3.14 报告/report

条目类型 14 报告对应的 biblatex 的 entrytype 为: report。文献类型标识用 R 表示。(注意:biblatex 将 techreport 作为 report 条目类型的别名,对于标准样式,techreport 出现在 bib 文件中等同于 report,但这种等同标准样式是在驱动层处理的,而 gb7714-2015 样式还需要处理文献类型标识,本样式文件做了进一步支持。因此 bib 文件中也能直接使用 techreport 类型。)

其著录格式为 (由 biblatex 的标准 report 格式修改得到,注意当出版地和出版者不存在时忽略这两项)

主要责任者. 题名: 其他题名信息 [文献类型标识/文献载体标识]. 其他责任者. 类型. 号码. 版本项. 出版地: 出版者, 出版年: 引文页码 [引用日期]. 获取和访问路径. 数字对象唯一标识符.

其对应的 biblatex 数据域为:

示例 19. 报告/report/techreport 条目的域格式

```
代码
1 author.title[usera].translator.type number.version.location:institution,date 或year:pages[
  urldate].url.doi
```

注意: 因为有的报告文献可能存在类型和报告号信息, 比如 AIAA 9076 或 AD 730029 等, 所以著录格式需要有所体现, 而这两个数据体现在 type 和 number 两个域中, 或者在 version 域中体现也可, 而对于标题中的出现的报告号, 可以直接在标题或子标题或者附加标题中体现。需要注意: report 和后面 manual 的版本信息放在 version 域中, 而不是 book 等条目的 edition 域中。

2.3.15 手册或档案/manual

条目类型 15 手册和档案采用一种格式, 对应的 biblatex 的 entrytype 为:manual。文献类型标识用 A 表示。

其著录格式为 直接采用 report 的格式, 而不是标准样式中的 manual 格式, 这种方式下, 当没有出版地和出版者时, 完全省略, 因为 GB/T 7714-2015 并没有明确这种情况怎么处理。

2.3.16 未出版物/unpublished

条目类型 16 未出版物, 对应的 biblatex 的 entrytype 为:unpublished。文献类型标识用 Z 表示。

其著录格式为 也直接采用 report 格式处理。

2.4 多语言文献

某些期刊对于参考文献有双语文献要求, 那么可以通过条目集类型 (set)/或者条目关联 (related) 来解决。具体要求见 GB/T 7714-2015 第 6.1 节。

利用条目集类型满足双语文献要求 使用条目集类型 (set) 时, 有静态和动态两种方法: 动态方法使用更为方便, 在写文档时候, 直接利用 defbibentryset 设置双语文献的 set, 然后引用 set 的 bibtex 键。比如:

示例 20. 设置 set 条目集用于双语文献动态方法

```
代码
1 \defbibentryset{bilangyi2013}{易仕和2013--,Yi2013--}
2 专著, 双语文献引用\cite{bilangyi2013}
```

得到的参考文献打印结果见3.2节的参考文献表。需要注意的是: 因为 set 条目类型除了子条目关键词信息外, 并无其他信息, 因此它的标注标签通常会空。这个问题目前的解决方法是设置一个指定格式和内容中间无空格无英文逗号的关键词, 用它来作为标签, 这个问题与 biblatex 版本有关, v3.7

及之前版本没有这个问题，因为 `set` 中还带有第一个子条目的信息，但 v3.8 及以上版本就不行了。比如：

示例 21. 设置 `set` 条目集用于双语文献动态方法

代码

```
1 \defbibentryset{易仕和, 等, 2013}{易仕和2013--,Yi2013--}
2 专著, 双语文献引用\cite{易仕和, 等, 2013}
```

注意到“易仕和, 等, 2013”中的逗号是中文全角逗号，这样使得“易仕和, 等, 2013”以一个整体字符串当作关键字，而不会被分开解析。但是这种解决方案中的中文全角逗号与其它标签的英文逗号的差异使得该问题并没有完美解决。对于 v3.8 及以上版本，可以利用后面介绍的关联 (related) 方法来解决，或者也可以用静态方法手动添加标签来解决。

静态方法是在 `bib` 源文件中给出条目集 (`set`) 并使用 `biber` 后端进行解析，条目的域信息采用如下方法定义：

示例 22. 设置 `set` 条目集用于双语文献静态方法

代码

```
1 @Set{set1,
2 entryset = {key1,key2,key3},
3 }
4 %如果要达到上例动态设置set一样的结果，在bib文件中静态设置set条目如下：
5 @Set{bilangyi2013,
6 entryset = {易仕和2013--,Yi2013--},
7 }
```

需要注意，使用静态条目集时，如果仅采用上述这般简单设置，中文排序会出现问题，条目集会出现在文献表末尾，这是因为条目集没有设置 `userb` 域用于排序，而通常的条目都是利用动态数据修改设置了 `userb` 域，在 `biber` 运行中因为要解析文献集，所以无法对 `userb` 域进行处理。但动态方法没有这一问题，因为其解析过程直接会利用第一个子条目的排序信息。静态方法的这个问题可以在 `set` 条目中手动设置 `userb` 域来修正。此外，对于 v3.8 以上版本的 `biblatex`，无论动态方法还是静态方法，条目集不复制第一个子条目信息，因此引用时也无法生成标注标签，这就是前面介绍动态方法时讨论过的问题，该问题也可以通过在 `set` 条目中手动设置 `label` 域来解决。比如：

示例 23. 在 `bib` 文件中正确设置 `set` 条目集的静态方法

代码

```
1 %在bib文件中静态设置set条目如下，其中：
2 %手动设置userb域用于解决排序问题
3 %手动设置label域用于解决标注标签问题
4 @Set{bilangyi2013,
5 entryset = {易仕和2013--,Yi2013--},
6 label={易仕和, 等, 2013},
7 userb={cn}
8 }
```

还要注意的是，动态 `set` 条目集等价于使用 `nocite` 命令，因此只要定义了动态条目集的文献都会出现在文献表中，因此如果不引用相应的文献，那么就不应该对其定义动态条目集。

利用条目关联满足双语文献要求 除上述给出的条目集方案外,关联条目方法则是另一种可行方案,该方案的讨论可以见“Again about the set label for authoryear style”^③。该方案同样也有静态和动态两种方法,静态就是修改 bib 文件内容,动态则是在源文档中做设置。

静态方法很简单,bib 文件中条目设置如例24所示,它能解决双语同时显示的问题,也能解决排序和标注标签问题,唯一的问题在于修改了 bib 文件后当不需要双语文献时,它还需要改回来,这会带来不便,因此可以考虑下面的动态方法,但要注意动态方法需要利用多个 DeclareStyleSourcemap,因此该方法只适用于 biblatex3.7 及以上版本。

示例 24. 在 bib 文件中正确设置关联条目的静态方法

代码

```
1 %在bib文件中静态设置条目如下, 其中:
2 %易仕和2013--条目中增加了related域用于关联其对应的英文条目Yi2013--
3 @Book{易仕和2013--,
4   Title = {超声速和高超声速喷管设计},
5   Address = {北京},
6   Author = {易仕和 and 赵玉新 and 何霖 and 张敏莉},
7   Publisher = {国防工业出版社},
8   Year = {2013}
9   related = {Yi2013--}
10 }
11 @Book{Yi2013--,
12   Title = {Supersonic and hypersonic nozzle design},
13   Address = {BeiJing},
14   Author = {Yi, S H and Zhao, Y X and He, L and Zhang, M L},
15   Publisher = {National Defense Industry Press},
16   Year = {2013}
17 }
```

动态方法利用动态数据修改自动添加 related 域,避免对 bib 文件做直接修改。本样式中对该过程进行了封装,定义一个新的命令 defdoublelangentry, 例如:

示例 25. 设置关联条目的动态方法

代码

```
1 \defdoublelangentry{易仕和2013--}{Yi2013--}
```

使用该命令后,可以引用主条目“易仕和 2013--”生成双语文献。但要注意由于 DeclareStyleSourcemap 命令只能在导言区中使用,因此 defdoublelangentry 命令也只能出现在导言区中,这也是相比条目集动态方法的唯一遗憾。实现的具体细节见4.4.3节。

2.5 标准的其它细节要求

除了第2.3节针对不同条目类型的著录格式要求外,GB/T 7714-2015 还有一些细节要求比如文字、符号等需要满足,可以采用如下方法:

^③<https://github.com/plk/biblatex/issues/681>

2.5.1 数字

方法 1 用户给 bib 源文件录入数字时, 应按照 GB/T 7714-2015 第 6.2 节要求用阿拉伯数字表示。

2.5.2 英文字母

方法 2 为了符合西文文献责任者的字母大小写习惯, 本 bbx 样式文件, 通过判断是否存在 *first name* 来确定是否是个人作者, 当存在 *first name* 时认为是个人作者, 不存在则是机构作者, 当是个人作者时 *lastname* 按 GB/T 7714-2015 要求全大写, 是机构作者则仅大写首字母。所以对于仅有 *lastname* 的个人作者, 填入信息英文姓的字母请全用大写。个人著者的格式要求参考 GB/T 7714-2015 第 6.3 节。

用户给 bib 源文件录入出版项、西文期刊名缩写以及西文文献的字母时, 应按照 GB/T 7714-2015 第 6.4 节, 第 6.5 节, 6.6 节要求, 使用符合要求的习惯用法和大小写方式。本样式文件使用原样打印的方式进行处理。

实际上, 对于英文大小写问题, GB/T 7714-2015 除了责任者的大写要求外, 其它的要求相对比较模糊, 所以一般不同的期刊会有各自不同的要求, 比如不能使用缩写, 而 GB/T 7714-2015 规定可参照 ISO 4 的要求等。从作者的经验看, 一般国内的期刊对于字母大小写通常要求: 责任者 (全部大写); 题名 (句首字母大写其它全部小写); 期刊名会议名 (单词首字母大写); 出版项和其它 (单词首字母大写)。所以在录入 bib 文件时可以按照这种常见方式来输入可以减少后期的修改。

2.5.3 标点

方法 3 用户给 bib 源文件录入引文信息时不需要考虑标点符号问题, 只需录入各数据域的信息即可。

本样式文件实现了 GB/T 7714-2015 第 7 部分所给出的著录用符号要求。

2.5.4 责任者

方法 4 用户给 bib 源文件录入引文的责任者信息时, 当责任者为多级机关团体时, 用户填入 *author* 数据信息时, 应按照 GB/T 7714-2015 第 8.1.4 节要求, 用英文句点. 号分隔。

当责任者是个人的英文名, 且具有名、姓、前缀和后缀, 应按照第 1.3.1 节给出姓名录入方式处理才能正确解析, 比如: von Peebles, Jr., P. Z., 其中 von 为姓前的前缀, Jr. 为姓后的后缀, P. Z. 为缩写名 (包括 *first name* 和 *middle name*)。

本样式文件实现了 GB/T 7714-2015 第 8.1 节要求的责任者样式, 能自动判断责任者是中文还是英文, 并分别处理。并且设置了全局选项 `useprefix=true`, 可以使用前缀。

2.5.5 文献类型标识和载体

方法 5 用户在给 bib 源文件录入引文题名信息时，不需要给出文献类型标识/文献载体标识。同一责任者的合订题名，应用户根据 GB/T 7714-2015 第 8.2.1 节的要求，在多个题名间用英文分号分隔，并整体录入到 title 数据域中。而分卷号，卷次，册次等信息时，除了专利号用 number 域录入外，其它可以直接在 title 数据域或者 subtitle/titleaddon 等数据域中给出。

本样式文件实现了符合 GB/T 7714-2015 第 8.2 节要求的格式，能根据条目类型选择文献类型标识/文献载体标识，自动录入到自定义域 usera 中，并在各类参考文献条目驱动中直接使用。各不同类型文献的文献类型标识/文献载体标识，参考 GB/T 7714-2015 表 B.1 和 B.2。

2.5.6 版次

方法 6 用户在给 bib 源文件录入版次信息时，只要录入版次的整数数字或者录入需要打印的字符串。

本样式文件实现了 GB/T 7714-2015 第 8.3 节要求的格式，对于一般的版式格式，根据 edition/version 域输入的整数进行处理，其它特殊的版本说明，比如新 1 版，明刻本等直接在 edition 域录入后原样打印。

2.5.7 出版项

方法 7 用户在给 bib 源文件录入出版项信息时，当出版日期有其它形式的纪年时，将其置于公元纪年后面的 () 内，并整体录入到 year 数据域中，比如：1845(清同治四年)。而引用日期应录入到 urldate 数据域。当除了出版日期外还有修改/更新日期等时，可在 year 或 date 数据域录入第二个日期，并用 / 符号与前一个出版日期隔开。而专利的公告日期和其它条目类型的出版年应录入到 date 域中。

本样式文件实现了 GB/T 7714-2015 第 8.4 节要求的格式。当出版地和出版者缺省时，中英文自动区分处理。对于用 / 符号隔开的两个日期，biblatex 后端 biber 能自动解析，后一个日期数据自动解析到 endyear 等域可作为修改日期等在样式文件中使用。

2.5.8 页码

方法 8 用户在给 bib 源文件录入页码信息时，可以在 pages 域中根据需要录入可解析的页码(即用整数表示页码，起讫页码用-分隔)，比如：81-86。也可以直接录入需要打印的信息，比如：序 2-3 等。

本样式实现了 GB/T 7714-2015 第 8.5, 8.8.2 节的要求，对于能解析的页码自动解析，对于不能解析的页码则原样输出。

2.5.9 访问路径 URL 和 DOI

方法 9 用户在给 bib 源文件录入获取和访问路径、数字对象唯一标识符信息时, 将访问路径录入到 url 域中, 数字对象唯一标识符录入到 doi 域中即可。

本样式文件实现了 GB/T 7714-2015 第 8.6,8.7 节要求的格式。

2.5.10 卷和期

方法 10 用户在给 bib 源文件录入卷、期等信息时, 如 1.3.1 节中所述, 合期的期号用/间隔, 比如 9/10, 填入 number 域, 报纸的版次也填入 number 域。

本样式文件实现了 GB/T 7714-2015 第 8.8 节要求的析出文献相关格式。

3 参考文献著录格式测试

3.1 GB/T 7714-2015 标准中的参考文献格式示例

普通图书 (book) (张伯伟, 2002) (师伏堂日记: 第 4 册 2009) (胡承正, 等, 2010) (美国妇产科医师学会, 2010) (康熙字典: 已集上: 水部 1962) (汪昂, 1881(清光绪七年)) (蒋有绪, 等, 1998) (中国企业投资协会, 等, 2013) (罗斯基, 2009) (库恩, 2012) (候文顺, 2010) (CRAWFPRD, et al., 1995) (International Federation of Library Association and Institutions, 1977) (O'BRIEN, 1994) (Kinchy, 2012) (Praetzelis, 2011)

论文集 (collection)、会议录 (proceedings) (中国职工教育研究会, 1985) (中国社会科学院台湾史研究中心, 2012) (雷光春, 2012) (陈志勇, 2011) (Babu, et al., 2014)

报告 (report) (中华人民共和国国务院办公厅, 2013) (汤万金, 等, 2013) (Calkin, et al., 2011) (U.S. Department of Transportation Federal Highway Administration, 1990) (World Health Organization, 1970)

学位论文 (thesis) (马欢, 2011) (吴云芳, 2003) (CALMS, 1965)

专利文献 (patent) (张凯军, 2012) (河北绿洲生态环境科技有限公司, 2001) (KOSEKI, et al., 2002)

标准文献 (book,inbook,note=standard) (全国信息文献标准化技术委员会, 2010) (全国广播电视标准化技术委员会, 2007) (国家环境保护局科技标准司, 1996) (*Information and documentation-the Dublin core metadata element set:ISO 15836:2009* 2013)

专著中析出的文献 (inbook) (“卷 39 乞致任第一” 1988) (白书农, 1998) (汪学军, 2002) (国家标准局信息分类编码研究所, 1988) (“宋史卷三: 本纪第三” 1977) (楼梦麟, 等, 2011) (BUSECK, et al., 1980) (FOURNEY, 1971)

期刊中析出的文献 (article) (杨洪升, 2013) (李炳穆, 2000) (于潇, 等, 2012) (陈建军, 2010) (DES MARAIS, et al., 1992) (Saito, et al., 2006) (Walls, et al., 2013) (Franz, et al., 2013) (Park, et al., 2010)

报纸中析出的文献 (article,note=news) (丁文祥, 2000) (张田勤, 2000) (傅刚, 等, 2000) (刘裕国, 等, 2013)

电子资源 (online) (萧钰, 2001) (李强, 2012) (Commonwealth libraries bureau of library development.Pennsylvania department of education office, 2013) (*Dublin Core metadata element set: version*

【A1-普通图书-著录格式示例】

- 中国企业投资协会, 台湾并购与私募股权协会, 汇盈国际投资集团, 2013. 投资台湾: 大陆企业赴台投资指南[M]. 北京: 九州出版社.
- 候文顺, 2010. 高分子物理: 高分子材料分析、选择与改性[M/OL]. 北京: 化学工业出版社: 119 [2012-11-27]. <http://apabi.lib.pku.edu.cn>.
- 师伏堂日记: 第 4 册 2009. 北京: 北京图书馆出版社: 155.
- 库恩, 2012. 科学革命的结构: 第 4 版[M]. 金吾伦, 胡新和, 译. 2 版. 北京: 北京大学出版社.
- 康熙字典: 已集上: 水部 1962. 同文书局影印本. 北京: 中华书局: 50.
- 张伯伟, 2002. 全唐五代诗格会考[M]. 南京: 江苏古籍出版社: 288.
- 汪昂, 1881(清光绪七年). 增订本草备要: 四卷[M]. 刻本. 上海: 老二酉堂.
- 罗斯基, 2009. 战前中国经济的增长[M]. 唐巧天, 毛立坤, 姜修宪, 译. 杭州: 浙江大学出版社.
- 美国妇产科医师学会, 2010. 新生儿脑病和脑性瘫痪发病机制与病理生理[M]. 段涛, 杨慧霞, 译. 北京: 人民卫生出版社: 38-39.
- 胡承正, 周详, 缪灵, 2010. 理论物理概论: 上[M]. 武汉: 武汉大学出版社: 112.
- 蒋有绪, 郭泉水, 马娟, 等, 1998. 中国森林群落分类及其群落学特征 and X_{La}T_EX[M]. 北京: 科学出版社.
- CRAWFPRD W, GORMAN M, 1995. Future libraries: dreams, maddness, & reality[M]. Chicago: American Library Association.
- International Federation of Library Association and Institutions, 1977. Names of persons: national usages for entry in catalogues[M]. 3rd ed. London: IFLA International Office for UBC.
- KINCHY A, 2012. Seeds, Sciences, and struggle : the global politics of transgenic crops[M/OL]. Cambridge, Mass.: MIT Press: 50 [2013-07-14]. <http://lib.myilibrary.com>.
- O'BRIEN J A, 1994. Introduction to information systems[M]. 7th ed. Burr Ridge, Ill.: Irwin.
- PRAETZELLIS A, 2011. Death by theory: a tale of mystery and archaeological theory[M/OL]. Rev. ed. [S.l.]: Rowman & Littlefield Publishing Group, Inc.: 13 [2012-07-26]. <http://lib.myilibrary.com>.

【A2-论文集、会议录-著录格式示例】

- 中国社会科学院台湾史研究中心, 2012. 台湾光复六十五周年暨抗战史实学术研讨会论文集[C]. 北京: 九州出版社.
- 中国职工教育研究会, 1985. 职工教育研究论文集[G]. 北京: 人民教育出版社.
- 汪学军, 2002. 中国农业转基因生物研发进展与安全管理[C]// 国家环境保护总局生物安全管理办公室. 中国国家生物安全框架实施国际合作项目研讨会论文集. 北京: 中国环境科学出版社: 22-25.
- 陈志勇, 2011. 中国财税文化价值研究: “中国财税文化国际学术研讨会”论文集[C/OL]. 北京: 经济科学出版社, [2013-10-14]. <http://apabi.lib.pku.edu.cn>.
- 雷光春, 2012. 综合湿地管理: 综合湿地管理国际研讨会论文集[C]. 北京: 海洋出版社.

BABU B V, NAGAR A K, DEEP K, et al., 2014. Proceedings of the second international conference on soft computing for problem solving, December 28-30[C]. New Delhi: Springer.

FOURNEY M E, 1971. Advances in holographic photoelasticity[C]//Symposium on Applications of Holography in Mechanics, August 23-25, 1971, University of Southern California, Los Angeles, California. New York: ASME: 17-38.

【A3-报告-著录格式示例】

中华人民共和国国务院新闻办公室, 2013. 国防白皮书: 中国武装力量的多样化运用[R/OL]. [2014-06-11]. <http://www.mod.gov.cn>.

汤万金, 杨跃翔, 刘文, 等, 2013. 人体安全重要技术标准研制最终报告:7178999X[R/OL]. [2014-06-24]. <http://www.nstrs.org.cn>.

CALKIN D, AGER A, THOMPSON M, 2011. A comparative risk assessment framework for wildland fire management: the 2010 cohesive strategy Science report:RMRS-GTR-262[R]: 8-9.

U.S. Department of Transportation Federal Highway Administration, 1990. Guidelines for handling excavated acid-producing materials, PB 91-194001[R]. Springfield: U.S. Department of Commerce National Information Service.

World Health Organization, 1970. Factors regulating the immune response: report of WHO Scientific Group[R]. Geneva: WHO.

【A4-学位论文-著录格式示例】

吴云芳, 2003. 面向中文信息处理的现代汉语并列结构研究[D/OL]. 北京: 北京大学, [2013-10-14]. <http://thesis.lib.pku.edu.cn>.

马欢, 2011. 人类活动影响下海河流域典型区水循环变化分析[D/OL]. 北京: 清华大学: 27 [2013-10-14]. <http://www.cnki.net>.

CALMS R B, 1965. Infrared spectroscopic studies on solid oxygen[D]. Berkeley: Univ. of California.

【A5-专利-著录格式示例】

张凯军, 2012. 轨道火车及高速轨道火车紧急安全制动辅助装置: 201220158825.2[P]. 2012-04-05.

河北绿洲生态环境科技有限公司, 2001. 一种荒漠化地区生态植被综合培育种植方法: 01129210.5[P/OL]. 2001-10-24 [2002-05-28]. <http://211.152.9.47/sipoasp/zlijs/hyjs-yx-new.asp?recid=01129210.5&leixin>.

KOSEKI A, MOMOSE H, KAWAHITO M, et al., 2002. Compiler: US828402[P/OL]. 2002-05-25 [2002-05-28]. <http://FF&p=1&u=metahtml/PT0/search-bool.html&r=5&f=G&l=50&col=AND&d=PG01&s1=IBM.AS.&OS=AN/IBM&RS=AN/IBM>.

【A6-标准文献-著录格式示例】

全国信息文献标准化技术委员会, 2010. 文献著录: 第 4 部分 非书资料:GB/T3792.4-2009[S]. 北京: 中国标准出版社: 3.

全国广播电视标准化技术委员会, 2007. 广播电视音像资料编目规范: 第 2 部分 广播资料:GY/T202.2-2007[S]. 北京: 国家广播电影电视局广播电视规划院: 1.

国家标准局信息分类编码研究所, 1988. 世界各国和地区名称代码 GB/T 2659-1986[S]// 全国文献工作标准化技术委员会. 文献工作国家标准汇编: 3. 北京: 中国标准出版社: 59-92.

国家环境保护局科技标准司, 1996. 土壤环境质量标准:GB 15616-1995[S/OL]. 北京: 中国标准出版社: 2-3 [2013-10-14]. <http://wenku.baidu.com>.

Information and documentation-the Dublin core metadata element set:ISO 15836:2009 2013. [2013-03-24]. <http://www.iso.org>.

注意: 这里最后一篇标准文献没有责任者, 所以作者年制默认情况下无法给出文献标识符。

【A7-专著中析出的文献-著录格式示例】

卷 39 乞致任第一 1988. // 苏魏公文集: 下册. 北京: 中华书局: 590.

宋史卷三: 本纪第三 1977. // 宋史: 第 1 册. 北京: 中华书局: 49.

楼梦麟, 杨燕, 2011. 汶川地震基岩地震动特征分析[M/OL]// 同济大学土木工程防灾国家重点实验室. 汶川地震震害研究. 上海: 同济大学出版社: 11-12 [2013-05-09]. <http://apabi.lib.pku.edu.cn>.

白书农, 1998. 植物开花研究[M]// 李承森. 植物科学进展. 北京: 高等教育出版社: 146-163.

BUSECK P R, NORD G L Jr., VEBLEN D R, 1980. Subsolidus phenomena in pyroxenes[M]// PREWITT C. Pyroxense. Washington, D.C.: Mineralogical Society of America: 117-211.

【A8-期刊中析出的文献-著录格式示例】

于潇, 刘义, 柴跃廷, 等, 2012. 互联网药品可信交易环境中主体资质审核备案模式[J]. 清华大学学报 (自然科学版), 52(11): 1518-1523.

李炳穆, 2000. 理想的图书管理员和信息专家的素养与形象[J]. 图书情报工作(2): 5-8.

杨洪升, 2013. 四库馆私家抄校书考略[J]. 文献(1): 56-75.

陈建军, 2010. 从数字地球到智慧地球[J/OL]. 国图资源导刊, 7(10): 93 [2013-03-20]. <http://d.g.wanfangdata.com.cn>. DOI: 10.3969/j.issn.1672-5603.2010.10.038.

DES MARAIS D J, STRAUSS H, SUMMONS R E, et al., 1992. Carbon isotope evidence for the stepwise oxidation of the Proterozoic environment[J]. Nature, 359: 605-609.

FRANZ A, DANIELEWICZ M A, WONG D M, et al., 2013. Phenotypic screening with oleaginous microalgae reveals modulators of lipid productivity[J/OL]. ACS chemical biology, 8: 1053-1062 [2014-06-26]. <http://pubs.acs.org>.

PARK J R, TOSAKA Y, 2010. Metadata quality Control in Digital repositories and collections: criteria, semantics, and mechanisms[J/OL]. Cataloging & classification quarterly, 48(8): 696-715 [2013-09-05]. <http://www.tandfonline.com>.

SAITO M, MIYAZAKI K, 2006. Jadeite-bearing metagabbro in serpentinite mélange of the “kurosegawa belt” in Izumi Town, Yatsushiro city, kumamoto prefecture, central kyushu[J]. Bulletin of geological survey of japan, 57(5/6): 169-176.

WALLS S C, BARICHIVICH W J, BROWN M E, 2013. Drought, deluge and declines: the impact of precipitation extremes on amphibians in a changing climate[J/OL]. *Biology*, 2(1): 399-418 [2013-11-04]. <http://www.mdpi.com>.

【A9-报纸析出的文献-著录格式示例】

丁文祥, 2000. 数字革命与竞争国际化[N]. *中国青年报*(15).

傅刚, 赵承, 李佳路, 2000. 大风沙过后的思考[N/OL]. *北京青年报*(14) [2005-07-12]. <http://www.bjyouth.com.cn/Bqb/20000412/GB/4216%5ED0412B1401.htm>.

刘裕国, 杨柳, 张洋, 等, 2013. 雾霾来袭, 如何突围[N/OL]. *人民日报* [2013-11-06]. <http://paper.people.com.cn>.

张田勤, 2000. 犯罪 DNA 库与生命伦理学计划[N]. *大众科技报*(7).

【A10-电子资源-著录格式示例】

李强, 2012. 化解医患矛盾需釜底抽薪[EB/OL]. (2012-05-03) [2013-03-25]. <http://wenku.baidu.com>.

萧钰, 2001. 出版业信息化迈入快车道[EB/OL]. (2001-12-19) [2002-04-15]. <http://www.creader.com/news/20011219/200112190019.html>.

Commonwealth libraries bureau of library development.Pennsylvania department of education office, 2013. Pennsylvania library laws[EB/OL]. [2013-03-24]. <http://www.racc.edu>.

Dublin Core metadata element set: version 1.1 2012. (2012-06-14) [2014-06-11]. <http://dublincore.org>.

3.2 测试: 专著 book 和专著中的析出文献 inbook 及标准 standard 文献

专著 book 和专著中的析出文献 inbook 及标准 standard 文献测试, 参见:

顺序编码制: [egbooks.tex](#), [egbooks.pdf](#)。

作者年制: [egbooksay.tex](#), [egbooksay.pdf](#)。

3.3 测试: 连续出版物 periodical 和连续出版物中的析出文献 article

连续出版物 periodical 和连续出版物中的析出文献 article 文献测试, 参见:

顺序编码制: [egjournal.tex](#), [egjournal.pdf](#)。

作者年制: [egjournalay.tex](#), [egjournalay.pdf](#)。

3.4 测试: 电子资源或在线资源 online

在线资源 online 文献测试, 参见:

顺序编码制: [egonline.tex](#), [egonline.pdf](#)。

作者年制: [egonlineay.tex](#), [egonlineay.pdf](#)。

注意: 对于作者年制, 这里有 4 篇文献都是 noauthor, 有两篇有年份可以轻易分开, 还有两篇没有年份存在歧义, 所以在标注中用了 [n.d.] 加 a 和 b 分开, 但在参考文献表中, 各个版本的 biblatex 表现是不同的, 其中 3.4 版因为进行 `newbibmacro*{date+extrayear}` 的定义时候, 首先判断 `iffieldundef{\thefield{datelabelsource}year}`, 当不存在 `datelabelsource`

的值 +year 的域时, 就不再添加了。如果需要加 extrayear 也可以修改出来, 但其实并无必要。这与标注中用的 newbibmacro*{cite:labelyear+extrayear}(在 authoryear.cbx 文件中) 的定义是不一样的。更多的内容详见 4.6.1 节。

3.5 测试: 学位论文 thesis、专利文献 patent

学位论文 thesis 文献专利文献 patent 文献测试, 参见:

顺序编码制: [egthesispatent.tex](#), [egthesispatent.pdf](#)。

作者年制: [egthesispatentay.tex](#), [egthesispatentay.pdf](#)。

注意: 专利文献 {刘加林 1993-} 的 location 定义了中国, GB/T 7714-2015 中其实并不需要该域, 但这里并没有去掉, 先放着也许以后标准修改后可能用的着。如果要去掉, 那么将 bib 文件中该条目的 location 去掉就行了。

3.6 测试: 报告 report、手册 manual 和档案、未出版物 unpublished

报告 report、手册 manual 和档案、未出版物 unpublished 文献测试, 参见:

顺序编码制: [egreport.tex](#), [egreport.pdf](#)。

作者年制: [egreportay.tex](#), [egreportay.pdf](#)。

3.7 测试: 会议文集 proceedings 和会议文集中析出的文献 inproceedings 及汇编 collection 和汇编中的析出文献 incollection

proceedings 和 inproceedings 及 collection 和 incollection 文献测试, 参见:

顺序编码制: [egprcdorinprcd.tex](#), [egprcdorinprcd.pdf](#)。

作者年制: [egprcdorinprcday.tex](#), [egprcdorinprcday.pdf](#)。

3.8 测试: 双语文献

双语文献的两种实现方法, 一是基于 set 的动态方法, 二是基于 related 的动态方法, 参见: [egaligngb7714-2015ay.tex](#), [egaligngb7714-2015ay.pdf](#)。

3.9 测试: gbalign 选项

gbalign 选项主要控制顺序编码制序号标签三种对齐方式, 分别是:

右对齐, 参见: [egalignright.tex](#), [egalignright.pdf](#)。

左对齐, 参见: [egalignleft.tex](#), [egalignleft.pdf](#)。

项对齐, 参见: [egaligngb7714-2015.tex](#), [egaligngb7714-2015.pdf](#)。

3.10 测试: gbpub 选项

gbpub 选项测试参见:

顺序编码制 (作者年制类似):

[eggbpubtrue.tex](#), [eggbpubtrue.pdf](#)。

[eggbpubfalse.tex](#), [eggbpubfalse.pdf](#)。

3.11 测试: gbnoauthor 选项和 online 条目仅存 url 信息

online 条目信息严重缺失问题测试参见:

[eggbnoauthortrue.tex](#), [eggbnoauthortrue.pdf](#).

[eggbnoauthorfalse.tex](#), [eggbnoauthorfalse.pdf](#).

gbnoauthor 选项测试参见:

[eggbnoauthortrueay.tex](#), [eggbnoauthortrueay.pdf](#).

[eggbnoauthorfalseay.tex](#), [eggbnoauthorfalseay.pdf](#).

3.12 测试: gbnamefmt 选项

姓名默认情况下字母大写的, 即 `gbnamefmt=uppercase`, 设置 `gbnamefmt=lowercase` 可以不处理保持原始的大小写, 设置 `gbnamefmt=none` 则使用 biblatex 标准样式的处理方式, 参见:

`gbnamefmt=lowercase` 情况: [eggbnamefmta.tex](#), [eggbnamefmta.pdf](#).

`gbnamefmt=none` 情况: [eggbnamefmtb.tex](#), [eggbnamefmtb.pdf](#).

3.13 测试: beamer 类

beamer 类中的参考文献测试, 参见:

顺序编码制: [egbeamer.tex](#), [egbeamer.pdf](#).

作者年制: [egbeameray.tex](#), [egbeameray.pdf](#).

3.14 测试: 上标和非上标等不同的标注格式

不带页码的引用 (顺序编码制上标, 方括号包围; 作者年制行内, 括号包围):

- 见文献 (von Peebles, 2001)。

不带页码的引用 (顺序编码制非上标, 方括号包围; 作者年制行内, 括号包围):

- 在文献 (Miroslav, et al., 2004) 中。

带页码的引用 (标准命令, 默认样式; 增加命令, GB/T 7714-2015 样式):

- (见 蔡敏, 等, 2006, 49 页)(见 Miroslav, et al., 2004, 49 页)。
- (von Peebles, 2001)¹⁰⁰(von Peebles, 2001)²⁰¹⁻³⁰¹

已有作者只需要年份和页码的情况 (命令 `yearpagescite`), 作者年制 (上标), 顺序编码制 (与 `pagescite` 作用相同); 已有作者只需要年份的情况 (命令 `yearcite`):

- 见赵耀东 (1998)²⁰⁵ 和 Simon(2001)¹⁵ 的文献。
- 见赵耀东 (1998)
- 见赵耀东 (1998)
- 见赵耀东 (1998)

【gb7714-2015 顺序编码制上标和非上标标注测试】

MIROSLAV D L, DEJAN V T, BRIAN L E, 2004. 信号处理滤波器设计-基于 matlab 和 mathematica 的设计方法[M]. 朱义胜, 董辉, 译. 北京: 电子工业出版社.

蔡敏, 徐慧慧, 黄柄强, 2006. UML 基础和 Rose 建模教程[M]. 北京: 人民邮电出版社.

- 赵耀东, 1998. 新时代的工业工程师[M/OL]. 台北: 天下文化出版社. [1998-09-26]. [http://www.ie.nthu.edu.tw/info/ie.newie.htm\(Big5\)](http://www.ie.nthu.edu.tw/info/ie.newie.htm(Big5)).
- SIMON M K, OMURA J K, SCHOLTZ R A, et al., 2001. Spread Spectrum Communications Handbook[M]. New York City: McGraw-Hill, Inc.
- Von PEEBLES P Z Jr., 2001. Probability, random variable, and random signal Principles and L^AT_EX[M]. 4th ed. New York: McGraw-Hill: 100.

3.15 测试: phdthesis 等条目类型的兼容性

测试某些 bib 文件给出的 mastersthesis, phdthesis, www, electronic, standard, techreport, conference 条目类型, 测试本样式增加的 newspaper 类型。

- newspaper:(张田勤, 2000)
- standard:(全国文献工作标准化委员会第七分委员会, 1986; 国家标准局信息分类编码研究所, 1988)
- conference:(Li, et al., 2004)
- www:(萧钰, 2001)
- electronic:(omg Inc., 2003)
- techreport:(Humphrey, et al., 1971)
- mastersthesis:(张志祥, 1998)
- phdthesis:(张若凌, 2004)

【兼容 phdthesis 等条目类型】

全国文献工作标准化委员会第七分委员会, 1986. GB/T 5795-1986 中国标准书号[S]. 北京: 中国标准出版社.

国家标准局信息分类编码研究所, 1988. 世界各国和地区名称代码 GB/T 2659-1986[S]// 全国文献工作标准化技术委员会. 文献工作国家标准汇编: 3. 北京: 中国标准出版社: 59-92.

张志祥, 1998. 间断动力系统的随机扰动及其在守恒律方程中的应用[D]. 北京: 北京大学数学学院.

张田勤, 2000. 犯罪 DNA 库与生命伦理学计划[N]. 大众科技报(7).

张若凌, 2004. 高超声速非平衡流动的计算和实验研究[D]. 绵阳: 中国空气动力研究与发展中心.

萧钰, 2001. 出版业信息化迈入快车道[EB/OL]. (2001-12-19) [2002-04-15]. <http://www.creader.com/news/20011219/200112190019.html>.

HUMPHREY R L, WAGNER J L, 1971. The sonic throat method and real gas one-dimensional flow: an application to air and nitrogen[R]. AD 730029.

LI Y, MOLISCH A F, ZHANG J, 2004. Practical approaches to channel estimation and interference suppression for OFDM based UWB communications[C]// Proc. of the IEEE 6th Circ. and Syst., Symp. on Emerging Technologies: Frontiers of Mobile and Wireless Communication. Vol. 1. [S.l.]: [s.n.]: 21-24.

Omg Inc., 2003. Unified Modeling Language: Superstructure[EB/OL]. (2003). <http://www.omg.org>.

3.16 测试: 当责任者等需要判断中英文的信息中存在编组时的处理

当责任者等需要判断中英文的信息中存在编组时的处理 (International Federation of Library Association and Institutions, et al., 1977; International Federation of Library Association and Institutions, 1977) (Ben Hadj Alaya-Feki A., et al., 2008; 中国企业投资协会, 等, 2014; 中国企业投资协会, 等, 2015)

【中英文判断信息中存在编组的测试】

中国企业投资协会, 台湾并购与私募股权协会, 汇盈国际投资集团, 等, 2014. 投资台湾: 大陆企业赴台投资指南[M]. 北京: 九州出版社.

中国企业投资协会, 台湾并购与私募股权协会, 汇盈国际投资集团, 等, 2015. 投资台湾: 大陆企业赴台投资指南[M]. 北京: 九州出版社.

Ben Hadj Alaya-Feki A., SAYRAC B, JEMAA S B, et al., 2008. Interference cartography for hierarchical dynamic spectrum access[C]// Proc. IEEE Symp. on New Frontiers in Dynamic Spectrum Access Networks (DySPAN). Chicago, IL: [s.n.]: 1-5.

International Federation of Library Association and Institutions, 1977. Names of persons: national usages for entry in catalogues[M]. 3rd ed. London: IFLA International Office for UBC.

International Federation of Library Association and Institutions, FCC, FBI, et al., 1977. Names of persons: national usages for entry in catalogues[M]. 3rd ed. London: IFLA International Office for UBC.

3.17 测试: 处理参考文献信息中 & 等特殊字符

文献中 (Rodriguez, et al., n.d.) 的 booktitle 域中含有%, &, # 符号, 样式文件自动处理使其符合 tex 代码规则。

【处理参考文献信息中 & 等特殊字符】

RODRIGUEZ J, MARQUES P, RADWAN A, et al., n.d. Cognitive Radio and Cooperative Strategies for Power Saving in Multi-Standard Wireless Devices[C]// Future % Network & Mobile # Summit 2010. Florence, Italy: [s.n.].

3.18 测试: 处理作者年制 article 中卷信息缺省的标点

文献 (刘彻东, 1998)(亚洲地质图编目组, 1978) (高光明, 1998)

【author-year style:article without volume】

亚洲地质图编目组, 1978. 亚洲地层与地质历史概述[J]. 地质学报, 3: 194-208.

刘彻东, 1998. 中国的青年刊物: 个性特色为本[J]. 中国出版(5): 38-39.

高光明, 1998. 信号情报接收机的发展现状及趋势[J]. 电讯技术, 38(2): 60-65.

3.19 测试: 标题中有\LaTeX{} 等名称时的情况

文献 (von Peebles, 2001)(赵凯华, 等, 1995)(蒋有绪, 等, 1998)

【title with \LaTeX{}】

蒋有绪, 郭泉水, 马娟, 等, 1998. 中国森林群落分类及其群落学特征 and X_qLaTeX[M]. 北京: 科学出版社.

赵凯华, 罗蔚茵, 1995. 新概念物理教程[M]. 北京: 高等教育出版社.

Von PEEBLES P Z Jr., 2001. Probability, random variable, and random signal Principles and LaTeX[M]. 4th ed. New York: McGraw-Hill: 100.

4 样式文件设计、实现、应用

biblatex 相当完善和强大, 笔者开发样式文件过程中仅使用了 biblatex 提供的一小部分功能, 更多功能其实都没有涉及到。可以说, biblatex 作为参考文献问题的一个完整解决方案是名副其实的。笔者在设计样式文件时应用 biblatex 宏包功能的思路 and 实现总结如下:

4.1 基本思路

考虑到我国引用参考文献通常是中英文混合的情况, 因此不考虑针对中文的本地化处理, 而是在英文本地化的基础 (英文的本地化字符串设置文档是 english.lbx) 上添加一些中文要求的本地化字符串来使用。而为了区分使用中英文的字符串, 对参考文献条目中数据域内容进行中英文判断, 若是中文则使用中文字符串, 若是英文则使用英文字符串。

这里的中英文判断主要是在 tex 中的判断, 利用对域中的信息进行检测, 当信息的第一个字符是 CJK 字符时, 判断该域的信息是中文的, 否则是英文的。

在作者年制中不同语种文献分集时也有一个中英文判断, 主要是利用动态数据修改的方法。利用 perl 正则表达式判断作者和标题域中是否有中文信息 (可以用排除法即匹配不是英文数字标点的字符通常是中文字符, 当然最直接的是用 unicode 的中文字符范围匹配), 有的话设置 cn, 否则为 en。事实上这种判断可以进一步扩展, 利用强大的 perl 正则表达式, 对所有的关键域进行判断, 并在一些域中设置标识, 用于 tex 输出时进行判断, 而避免直接使用上述 CJK 字符判断函数。

4.2 利用字符流解析或者 xstring 宏包的函数

1. cjk 字符判断

```
1 %定义判断中文字符的函数, 用于判断作者等信息是否由中文字符构成
2 %2E00-2E7F 追加标点
3 %2E80-2EFF cjk部首补充
4 %2FF0-2FFF 表意文字描述符
5 %3000-303F cjk符号和标点
6 %3300-33FF cjk兼容
7 %3400-4DBF cjk统一表意符号扩展
8 %4E00-9FBF cjk统一表意符号
9 %中文范围4E00-9FA5
10 \providetoggle{ifCJKforgbt}
11 \def\testCJKfirstchar#1#2{%
12 \ifnumgreater{`#1}{`"2E7F}{\toggletrue{ifCJKforgbt}}{\togglefalse{ifCJKforgbt}}%
13 }%
14
```

```

15 %利用edef展开或xstring抽取第一个字符判断
16 %现在采用xstring方法，避免抽取的是编组符号
17 \def\testCJKfirst#1{%
18 %\edef\tempa{#1}%可以替换下面两句，但失去忽略多余编组功能
19 \exploregroups%
20 \StrChar{#1}{1}[\tempa]%
21 \expandafter\testCJKfirstchar\tempa&}

```

2. 卷期范围解析

```

1 %用于解析连续出版物，2个卷期的情况
2 %\def\multivolumeparsetoparts#1/#2{\def\multivolumefirst{#1}\def\multivolumessecond{#2}}
3 %\def\multinumberparsetoparts#1/#2{\def\multinumberfirst{#1}\def\multinumbersecond{#2}}
4 %范围起止间隔符号还是用-，而不是与date相同的/，因为有合期期刊的问题，需要用到/符号
5 \newcommand{\multivolparser}[1]{%
6   \IfSubStr{#1}{-}%
7     {\StrBefore{#1}{-}[\multivolfirst]\StrBehind{#1}{-}[\multivolsecond]]%
8     {\def\multivolfirst{#1}\def\multivolsecond{}}}%
9 }
10
11 \newcommand{\multinumberparser}[1]{%
12   \IfSubStr{#1}{-}%
13     {\StrBefore{#1}{-}[\multinumberfirst]\StrBehind{#1}{-}[\multinumbersecond]]%
14     {\def\multinumberfirst{#1}\def\multinumbersecond{}}}%
15 }
16
17 %这里利用李志奇的范围数据解析函数，修改后用来处理卷volume的范围。
18 %他的这个函数是通用的，只能在使用数据前使用，如果多次使用前一次得到的数据就会被冲掉
19 %这里的范围起止判断符号还是用-，而不是与date相同的/，因为有合期期刊的问题，需要用到/符号
20 %\def\gbt@parse@range#1{%
21 % \edef\gbt@tmpa{#1}%
22 % \expandafter\gbt@parse@rangei\gbt@tmpa-\@empty}
23 %\def\gbt@parse@rangei#1-#2\@empty{%
24 % \global\def\gbt@range@first{#1}%
25 % \def\gbt@tmpa{#2}%
26 % \ifx\gbt@tmpa\@empty%
27 % \global\def\gbt@range@second{}%
28 % \else%
29 % \gbt@parse@rangeii#2\@empty%
30 % \fi}
31 %\def\gbt@parse@rangeii#1-#2\@empty{%
32 % \def\gbt@tmpa{#2}%
33 % \ifx\gbt@tmpa\@empty%
34 % \global\def\gbt@range@second{#1}%
35 % \else%
36 % \gbt@parse@rangeii#2\@empty%
37 % \fi}
38
39 %这里写的number的解析函数，与上面的卷解析函数是类似的。
40 %\def\multinumberparser#1{%

```

```

41 % \edef\gbnumbertmp{#1}%
42 % \expandafter\multinumberparsetoparts\gbnumbertmp-\@empty}
43 %\def\multinumberparsetoparts#1-#2\@empty{%
44 % \def\multinumberfirst{#1}%
45 % \def\gbnumbertmp{#2}%
46 % \ifx\gbnumbertmp\@empty%
47 % \def\multinumbersecond{}%
48 % \else%
49 % \multinumberparsetopartsb#2\@empty%
50 % \fi}
51 %\def\multinumberparsetopartsb#1-#2\@empty{%
52 % \def\gbnumbertmp{#2}%
53 % \ifx\gbnumbertmp\@empty%
54 % \def\multinumbersecond{#1}%
55 % \else%
56 % \multinumberparsetopartsb#2\@empty%
57 % \fi}

```

4.3 标点和空格的特点和机制

1. 标点符号设置比如:renewrobustcmd*{\bibinitperiod}}{} 和 renewcommand*{\revsdnamepunct}}{} 等用于全局修改标点。

2. 如 isdot/adddot, adddot 将原样 (逐字, 如实) 句点转换为缩写点在有利用 printtext 输入原样字符的时候很重要比如 [s.n.] 中, 比如 title 末尾。

3. 利用 newunit 输出的标点, 需要后面遇到 printfield 等命令有内容才输出标点, 这是标点的异步处理机制。

4. setunit* 的作用是前面 printtext 等没有输出时候就不输出, 见 4.11.7.3 节的应用以及出版项缺省时的代码处理。

5. 标点和空格的调试方法: 有时引入多余的空格和标点后很难快速处理, 那么需要调试, 调试的第一步是注释掉多余的代码, 实现无多余空格或标点情况 (一种快速方法是直接在一个 printtext 输出各相关域), 然后一步步恢复原来的代码, 直到找到多余空格或标点产生的原因, 最后根据原因作出修改。

6. 多余空格的引入, 可能来自于如下方面 (下面是已经遇到的, 可能还有更多)

- 行末空格引入, 解决方法: 把相关的代码行结尾用% 符号注释。
- mkbibparens 等命令引入, 解决方法: 利用 printtext{()}, printtext{}} 来代替
- nopunct 等命令引入, 这种引入的空格使用 unspace 命令还消除不了, 解决方法: 避免使用 nopunct 命令, 而在标点设置时多做判断, 只有有内容时才设置标点, 而不是过分依赖 biblatex 提供的标点异步处理机制。

7. 短横线相关的字符, 参考: unicode 表

- 0x2d:hyphen :-, 直接输入键盘短横线得到-, 即为连字符
- 0x2010:hyphen :-
- 0x2011:non breaking hyphen :-

- 0x2012:figure dash :
- 0x2013:en dash :-, 直接输入两个键盘短横线得到--
- 0x2014:em dash :—, 中文全角的一段破折号线, 可以用三个短横线得到---
- 0x2212:mathematical minus :—

biblatex 中页码范围中的间隔符是\bihrangedash, 默认则是 en dash。可以通过设置如下命令做修改:

```
1 \DefineBibliographyExtras{english}{\renewcommand*{\bihrangedash}{-}}% 将页码间隔符替换为hyphen连
   字符
```

而日期成分之间的间隔符则是\bibdatesep, 默认是 hyphen, 即连字符。

8. 使用 printtext 等命令确保异步处理机制不被破坏, 典型的例子为, 专利的公告日期 (对于 3.7 以上版本) 使用了 biblatex 的内容宏, 但还需要将其用 printtext 包围起来, 比如:

```
1 %
2 % 专利的公告日期、或报纸的日期的输出宏
3 % 20160701, v1.0, 新增加
4 % 20180405, 为biblatexv3.10版本, 出现多出点bug做处理, Hu Zhenzhen
5 %
6 % 原理方法:加上printtext避免破坏异步标点机制
7 %
8 \newbibmacro*{newsdate}{%%
9 \printtext{\blx@isodate{}}}%
10 }
```

9. beamer 中标题后面的标点情况, beamer 对 bibmacro{title} 做了 patch 以实现特殊的效果。比如:

```
1 \pretocmd{\abx@macro@title}
2 {\ifcsundef{abx@name@labelname}{\let\bbx@tempa\labelnamepunct}%
3 \bbx@tempa\newblock\unspace\usebeamercolor[fg]{bibliography entry title}}{}
4 \apptocmd{\abx@macro@title}
5 {\ifcsundef{abx@field@title}{\ifpunct}{\midsentence\newunitpunct}}
6 \newblock\unspace\usebeamercolor[fg]{bibliography entry note}}{}{}
```

该 patch 详见 beamerbaselocalstructure.sty 文件。需要注意abx@field@title, 在 biblatex 中是有 newbibmacro{title} 产生的。注意到在 apptocmd 做的 patch 中, midsentence 命令隐藏了其前面的标点并重新做标点最终, 因此其后的 newunitpunct 命令必将输出标点, 因此无论如何标题后面必然会出现一个 newunitpunct 控制的标点, 这个标点一般情况下是可以用的, 但是对于有些文献类型比如 inbook, 析出文献标题后面没有标点只有//, 因此需要将该标点去掉, 所以需要反处理该 patch。通过测试表明, 无论是在 title 前再做 pretocmd 或者在 title 后再做 apptocmd 都会产生问题, 多个的 apptocmd 的机制大体是前一个做完后, 后一个在其基础上处理, 而 apptocmd 主要做的是封装, 因此前一个产生的效果无法消除。但通过思考发现, 使用 patchcmd 可以对前面的效果产生影响, 它不是封装而是一种替换, 所以可以将前面封装的效果去掉。比如:

```
1 %
2 % 文献标题后的标点问题
3 % 20180405, v1.0k, 为texlive2017以上版本中的beamer兼容性做的处理, Hu Zhenzhen
4 %
```



```

5 % 原理方法:如下代码处理texlive2017以上版本中, beamer中文献的标题后出现两个点的情况:
6 % texlive2017以上的beamer中对macro{title}做了patch, 正常情况下不会出现两个点的情况, 但由于
7 % 之前为了处理texlive2015, 2016下的title格式添加了adddot, 导致出现两个点的情况, 而且也影响
8 % 析出文献的//符号的输出, 因此再次对macro{title}做patch消除beamer中做apptocmd时添加的\newunitpunct
9 \ifboolexpr{%
10     test{\iftoggle{iftlfive}}
11     or
12     test{\iftoggle{iftlsix}}
13 }{%
14     {%texlive 2017对应iftlseven以上版本
15     \DeclareFieldFormat{title}{#1}%
16     \DeclareFieldFormat[article,patent,thesis,unpublished]{title}{#1}
17     \AtBeginDocument{%
18     \@ifclassloaded{beamer}
19         {\patchcmd{\abx@macro@title}{\newunitpunct}{\newunitpunct}{\newunitpunct}}
20     }

```

4.4 动态数据修改

利用 biber 在处理数据源时的动态处理, 可以处理一些数据, 比如设置一些域的值, 用于进一步的判断和应用。

4.4.1 基本原理和方法

biber 动态数据处理, 详见 biblatex 说明文档 Dynamic Modification of Data 一节和 bbx 文件中 DeclareSourcemap 的内容。

1. map 的作用, 对条目逐条进行处理

2. final 作用, 当不成立 map 终止。

3. append 的作用, 给域添加信息

4. origfieldvalue 来源

5. overwrite 选项有无的作用, 无 overwrite 时, 只要域原有信息, 那么就不再 map, 有 overwrite 时, 则进行覆盖。注意: 使用 append 的时候也需要 overwrite 选项。

6. biblatex3.4 以后的版本可以使用 foreach 选项, 而 3.0 版只能一个域一个域的处理, 意味着每一个域处理都要写一个 map 步。

7. date 域仅作为解析用, 不出现在 bbl 文件中, 所以在 bbx 内部进行日期判断的时候要注意, 不使用 date 域而要用由其解析出来的 year 等域进行判断, 而且当 date 域的内容不符合解析格式要求时, 自动忽略掉, 那么信息有可能丢失, 因此当有不符合解析格式的日期信息时应放到 year 域中。

8. 动态数据处理中利用正则表达式可以用来处理特殊字符, 对于参考文献信息中一些特殊字符比如 & 等, 除了利用 jabref 软件的 biblatex 可以自动转换外, 利用动态数据修改也可以做一定的处理, 比如对一些容易出现这种字符的域进行处理, 使其内容符合 tex 源文件书写规则, 即在特殊字符前加上斜杠。注意利用正则表达式处理%, # 时直接在 regexp 中写字符没有问题, 但是对于 & 字符就存在问题, 所以对于该字符使用十六进制表示方法\x26。比如:

```

1 %for texlive >2016

```

```

2 \DeclareStyleSourceMap{
3   \maps[datatype=bibtex]{
4     \map[overwrite, foreach={title,booktitle,journaltitle,journal,publisher,address,location,
5       institution,organization}]{
6       \step[fieldsource=\regex{$MAPLOOP}, match=\regex{([^\])\#}, replace=\regex{$1\\#}]
7     }
8     \map[overwrite, foreach={title,booktitle,journaltitle,journal,publisher,address,location,
9       institution,organization}]{
10      \step[fieldsource=\regex{$MAPLOOP}, match=\regex{([^\])\%}, replace=\regex{$1\\%\}]
11    }
12    \map[overwrite, foreach={title,booktitle,journaltitle,journal,publisher,%
13      address,location,institution,organization}]{
14      \step[fieldsource=\regex{$MAPLOOP}, match=\regex{([^\])\x26}, replace=\regex{$1\\x
15        26}]
16    }
17  }
18 }
19
20 %for texlive 2015
21 \DeclareStyleSourceMap{
22   \maps[datatype=bibtex]{
23     \map[overwrite]{%title,booktitle,journaltitle,journal,publisher,address,location,institution
24       ,organization
25     }
26     \step[fieldsource={title}, match=\regex{([^\])\x26}, replace=\regex{$1\\x26}]
27   }
28   \map[overwrite]{
29     \step[fieldsource={booktitle}, match=\regex{([^\])\x26}, replace=\regex{$1\\x26}]
30   }
31   \map[overwrite]{
32     \step[fieldsource={journaltitle}, match=\regex{([^\])\x26}, replace=\regex{$1\\x26}]
33   }
34   \map[overwrite]{
35     \step[fieldsource={journal}, match=\regex{([^\])\x26}, replace=\regex{$1\\x26}]
36   }
37   \map[overwrite]{
38     \step[fieldsource={publisher}, match=\regex{([^\])\x26}, replace=\regex{$1\\x26}]
39   }
40   \map[overwrite]{
41     \step[fieldsource={address}, match=\regex{([^\])\x26}, replace=\regex{$1\\x26}]
42   }
43   \map[overwrite]{
44     \step[fieldsource={location}, match=\regex{([^\])\x26}, replace=\regex{$1\\x26}]
45   }
46   \map[overwrite]{
47     \step[fieldsource={institution}, match=\regex{([^\])\x26}, replace=\regex{$1\\x26}]
48   }
49   \map[overwrite]{
50     \step[fieldsource={booktitle}, match=\regex{([^\])\#}, replace=\regex{$1\\#\}]
51   }
52 }

```

```

47         \map[overwrite]{
48         \step[fieldsource={booktitle}, match=\regexp{([^\[])\%}, replace=\regexp{$1\\%\%}]
49     }
50 }
51 }

```

4.4.2 数据模型

下面介绍 biblatex 数据模型与动态数据处理的关系。

biblatex 提供了数据模型定义功能，这在真正需要定义一些新的条目类型和域时是用得到的。但实际上，从本样式实践的经验看，为了 biblatex 的兼容性考虑，没有必要使用数据模型定义功能去费力地定义新的条目类型，反而直接利用动态数据处理来将 bib 中的一些类型转换成 biblatex 支持的类型会更方便，其中要做一定的区分可以利用一些域的信息进行判断，比如 note 域等于 news 或 standard。

1. biblatex 默认做的别名处理其实是用 DeclareDriverSourcemap 做的，这是驱动层的映射。在本样式中，因为 usera 域是在用户层的映射中做的，所以 biblatex 做的默认别名处理还不够，还需要在用户层首先处理 usera 域。而因为 DeclareDriverSourcemap 中会增加一些不需要的信息，比如 thesis 类型中的 typer 域在 GB/T7714 标准中是不需要的，这可以利用 DeclareSourcemap, DeclarestyleSourcemap(注意 texlive2016 的 biblatex3.4 仅支持 1 个 styleSourcemap, 而 biblatex3.7 则已经支持多个 styleSourcemap) 先进行映射，这样就避开了 DeclareDriverSourcemap。或者在增加 DeclareDriverSourcemap, 将标准样式中的默认 DeclareDriverSourcemap 增加的信息去掉 (详见 gb7714-2015ay.bbx 中的处理和说明)。

2. 数据模型在 blx-dm.DEF 中，可以看到 standard 有条目类型定义，但域和约束没有定义，而 biblatex 标准样式默认做的别名处理中的其它一些条目比如 electronic 等是有域和约束定义的。

3. 标准样式 standard.BBX 中的 DeclareBibliographyAlias{*}{misc} 将一些数据模型中没有定义的条目直接用 misc 条目的驱动处理，这不是驱动层的数据源映射，只是利用其驱动进行输出。有一个问题是 DeclareBibliographyAlias{newspaper}{article} 是否可以实现？还是要定义数据模型？从实践结果看是不需要定义数据模型的，并且驱动的别名处理等价于驱动层映射。比如：做了用户层映射后：

```

1 \DeclareSourcemap{
2   \maps[datatype=bibtex]{%增加文献标识符如[M],[J]等，下面首先生成这一信息
3     \map{
4       \pertype{newspaper}%增加一个新闻报纸的类型newspaper
5       \step[fieldset=usera, fieldvalue={N}]%因为没有专门的驱动，这句的目的是定义一个usera域，
6         方便映射为article 后判断
7       \step[fieldset=note, fieldvalue=news]
8     }
9   }

```

做别名处理：

```

1 \DeclareBibliographyAlias{newspaper}{article}%定义驱动别名，以替代驱动层映射

```

能达到驱动层或样式层中条目类型转换所实现的效果。

```

1 \DeclareStyleSourcemap{

```

```

2      \maps[datatype=bibtex]{
3          \map{%尝试未定义数据模型的newspaper类型映射为article, newspaper完全是针对gb7714的新类
              型, 在biblatex 中完全没有定义
              %但从实践看, 并没有什么影响, 映射过来就可以了。
              \step[tources=newspaper, typetarget=article, final]
4          }
5      }
6  }
7
8  }

```

4. 从实践看, 数据模型中完全没有定义的条目类型 newspaper 也可以进行数据源映射。

4.4.3 用于关联条目的动态修改的命令封装及关联条目格式

实现动态的条目关联, 可以利用动态数据修改的方法。要对多个条目实现关联, 那么需要多次数据映射, biblatex3.7 及以上版本可以存在多个 DeclareStyleSourcemap, 因此做关联的封装命令就是基于此:

```

1  \newcommand{\defdoublelangentry}[2]{%
2  \edef\entrykeya{#1}
3  \edef\entrykeyb{#2}
4  \DeclareStyleSourcemap{
5      \maps[datatype=bibtex]{
6          \map{
7              \step[fieldsource=entrykey, match=\entrykeya, final]
8              \step[fieldset=related, fieldvalue=\entrykeyb]
9          }
10     }
11 }
12 }

```

命令 defdoublelangentry 有两个输入参数, 第一个是主条目, 第二个是与主条目关联的条目的列表。多次使用该命令, 即可实现多次数据映射, 当然因为 DeclareStyleSourcemap 只能存在于导言区, 因此 defdoublelangentry 也只能存在于导言区。

关联条目之间用命令 relateddelim 分隔, 但实际上因为 related 宏输出时, 不知道是何原因考虑的是主条目的关联条目之间的分隔, 这在双语文献中是有问题的, 需要在主条目后也进行分隔, 因此对该宏修改如下:

```

1  \renewbibmacro*{related}{%
2  \ifboolexpr{ test {\iffieldundef{related}} or test {\ifrelatedloop} }
3  {}
4  {\usebibmacro{begrelated}%
5   \def\bbx@tempa{}%
6   \setcounter{bbx:relatedtotal}{0}%
7   \def\do##1{%
8       \entrydata{##1}%
9       \ifrelatedloop
10      {}
11      {\stepcounter{bbx:relatedtotal}%
12       \gappto{\bbx@tempa}{##1,}}}%
13   \docsvfield{related}%
14   \restorefield{related}{\bbx@tempa}%

```

```

15 \ifnumgreater{\value{bbx:relatedtotal}}{0}
16   {\listcsxadd{bbx:relatedloop}{\strfield{entrykey}}}%
17   \iffieldundef{clonesourcekey}
18     {}
19     {\listcsxadd{bbx:relatedloop}{\strfield{clonesourcekey}}}%
20   \setcounter{bbx:relatedcount}{0}%
21   \def\do{%
22     \stepcounter{bbx:relatedcount}%
23     \ifnumgreater{\value{bbx:relatedcount}}{0}%为使主条目与关联条目做分隔，这里做修改，从1改为0
24     {\iffcsundef{relateddelim}\strfield{relatedtype}}
25     {\printtext{\relateddelim}}
26     {\printtext{\csuse{relateddelim}\strfield{relatedtype}}}}
27   {}}%
28 \iffibmacroundef{related:\strfield{relatedtype}}
29   {\appto{\do}{\usebibmacro{related:default}}}
30   {\appto{\do}{\usebibmacro*{related:\strfield{relatedtype}}}}}%
31 \iffieldformatundef{related:\strfield{relatedtype}}
32   {\def\bbx@tempa{related}}
33   {\def\bbx@tempa{related:\strfield{relatedtype}}}%
34 \iffieldformatundef{relatedstring:\strfield{relatedtype}}
35   {\def\bbx@tempb{relatedstring:default}}
36   {\def\bbx@tempb{relatedstring:\strfield{relatedtype}}}%
37 \printtext[\bbx@tempa]{%
38   \usebibmacro{begrelatedloop}%
39   \iffieldundef{relatedstring}
40     {\ifboolexpr{
41       test {\ifnumgreater{\value{bbx:relatedtotal}}{1}}
42       and
43       test {\iffibxstring{\thefield{relatedtype}s}}
44     }
45     {\printtext[\bbx@tempb]{%
46       \bibstring[\mkrelatedstring]{\thefield{relatedtype}s}}
47     {\iffieldbibstring{relatedtype}
48       {\printtext[\bbx@tempb]{%
49         \bibstring[\mkrelatedstring]{\thefield{relatedtype}}}}
50       {}}}
51     {\iffieldbibstring{relatedstring}
52       {\printtext[\bbx@tempb]{%
53         \bibstring[\mkrelatedstring]{\thefield{relatedstring}}}}
54       {\printfield[\bbx@tempb]{relatedstring}}}%
55     \docsvfield{related}%
56     \usebibmacro{endrelatedloop}}}%
57 {}%
58 \usebibmacro{endrelated}}

```

4.5 标注 (引用) 标签

引用标签的生成机制，即责任者截短时的歧义消除问题，见说明文档 4.11.4 Name Disambiguation 节。

1. uniquename 的作用，用于姓名间的歧义消除。

2. uniquelist 的作用，用于姓名列表间的歧义消除。

关于这两个选项的参数设置，可以在样式中设置，也可以在宏包加载时设置，**注意宏包加载时的设置会覆盖样式中的设置。**

还要注意，说明文档中 uniquelist 的默认选项是 false，这是错误的，从实践看是 true，从 biblatex.sty 文件的代码看也是如此，比如：

```
1 \DeclareBibliographyOption[boolean]{uniquelist}[true]{%
2   \ifcsdef{blx@opt@uniquelist@#1}
3     {\letcs\blx@uniquelist{blx@opt@uniquelist@#1}}
4     {\blx@err@invopt{uniquelist=#1}{}}}
5 \DeclareTypeOption[boolean]{uniquelist}[true]{%
6   \ifcsdef{blx@opt@uniquelist@#1}
7     {\letcs\blx@uniquelist{blx@opt@uniquelist@#1}}
8     {\blx@err@invopt{uniquelist=#1}{}}}
9 \DeclareEntryOption[boolean]{uniquelist}[true]{%
10  \ifcsdef{blx@opt@uniquelist@#1}
11    {\letcs\blx@uniquelist{blx@opt@uniquelist@#1}}
12    {\blx@err@invopt{uniquelist=#1}{}}}
13 \def\blx@opt@uniquelist@false{0}
14 \def\blx@opt@uniquelist@true{1}
15 \def\blx@opt@uniquelist@minyear{2}
```

无论选项是否给出，上面定义选项中的代码都会自动执行，当选项不给出，则自动调用默认的参数（因为代码中直接使用了输入参数，当没有输入参数，自然就会使用默认参数）来进行处理，显然这里是 true。

4.5.1 标签中的标点和间隔符号

文献表中的标签和正文中的标注标签都有标点和间隔符号问题，但两者可能存在一定的差异，不同版本的 biblatex 中也可能不同，但新的版本中通常是可以用一个相同的钩子来实现的，比如作者年制中作者和年份之间的标点：

```
1 \renewcommand*{\nameyeardelim}{\addcomma\space}
```

4.6 日期格式控制

随着 biblatex 的更新，日期格式控制更趋完善。加上 biblatex3.8a 中 biber 解析出日期中月份和天数都不带引导的 0，比如 6 月直接解析为 6，而不像以前那样解析为 06，所以需要特别的注意。因此对于新版本，有些日期比如 online 的发布日期，访问日期等，采用专门的格式 edtf 控制，而老版本则仍然使用原来定义的输出宏。比如：

```
1 \defversion{3.7}{date}{
2   \DeclareFieldFormat{urldate}{#1}
3   \renewbibmacro*{urldate}{%
4     \addspace\printtext{[]}\printurldate\printtext{[]}}%能用高层命令+选项尽量用命令(比如这里的\
      printurldate)，而不用\blx@edtfdate这种更底层的命令
5
6   \newbibmacro*{newsdate}{%新增加一个新闻日期
```



```

7 \blx@edtfdate{}{}%
8 }
9
10 \newbibmacro*{modifydate}{%新增加一个带括号的日期，用于表示电子资源的更新和修改日期，而公告日期则按日
    期格式
11 \iffielddundef{year}{%
12 \iffielddundef{endyear}{\iffielddundef{eventyear}{}\printtext{(\printeventdate\printtext
    {})}%
13 {\printtext{(\printenddate\printtext{)}}}%
14 }{\iffieldequalstr{year}{}\%因为year存在，但为空
15 }\printtext{(\blx@edtfdate{}{}\printtext{)}}}%
16 }%
17 }%
18 }
19
20 \defversion{3.4}{date}{
21 \renewbibmacro*{urldate}{%
22 \addspace\printtext{[]\printfield{urlyear}%
23 \iffielddundef{urlmonth}{}\bibrangedash\printfield{urlmonth}}%
24 \iffielddundef{urlday}{}\bibrangedash\printfield{urlday}\printtext{[]}}
25
26 \newbibmacro*{newsdate}{%%新增加一个新闻日期
27 \iffielddundef{year}{}\printfield{year}%
28 \iffielddundef{month}{}\bibrangedash\printtext{\thefield{month}}%
29 \iffielddundef{day}{}\bibrangedash\printfield{day}}}%
30 }
31
32 \newbibmacro*{modifydate}{%新增加一个带括号的日期，用于表示电子资源的更新和修改日期，而公告日期则按日
    期格式
33 \iffieldequalstr{year}{}\%替换\iffielddundef{year}，因为year总是存在，但为空
34 \iffielddundef{endyear}{%
35 \iffielddundef{eventyear}{}\printtext{\mkbibparens{\printtext{\printfield{eventyear}}}%
36 \iffielddundef{eventmonth}{}\bibrangedash\thefield{eventmonth}}%
37 \iffielddundef{eventday}{}\bibrangedash\printfield{eventday}}}%
38 }%
39 }{%
40 \iffielddundef{endyear}{}\printtext{\mkbibparens{\printtext{\printfield{endyear}}}%
41 \iffielddundef{endmonth}{}\bibrangedash\thefield{endmonth}}%
42 \iffielddundef{endday}{}\bibrangedash\printfield{endday}}}%
43 }%
44 }%
45 }%
46 {%
47 \iffielddundef{year}{}\printtext{\mkbibparens{\printtext{\printfield{year}}}%
48 \iffielddundef{month}{}\bibrangedash\thefield{month}}%
49 \iffielddundef{day}{}\bibrangedash\printfield{day}}}%
50 }%
51 }%
52 }%

```

```

53 }
54
55 \iftoggle{iftlfive}%%根据texlive/biblatex版本选择
56   {\switchversion{3.4}{date}}%
57   {\iftoggle{iftlsix}%
58     {\switchversion{3.4}{date}}%
59     {\switchversion{3.7}{date}}%
60   }

```

4.6.1 文献表条目标签中的日期格式

主要用于 authoryear 样式，在 authoryear 样式中有一个 mergedate 选项，使用该选项可以进行更多的标签日期格式控制。其中主要是重定义了 date+extradate(3.8 版，以前的版本叫 date+extrayear)。

其中有两点需要注意：

1. printdateextra, printlabeldateextra, 这样的命令在 biblatex.sty 中是找不到的，因为 biblatex 实际定义的如下的命令：

```

1 {\protected\def\blx@imc@printlabeldate{\csuse{mkdaterange#1}{label}}
2   \protected\def\blx@imc@printlabeldateextra{\csuse{mkdaterange#1extra}{label}}}

```

然后由如下命令，解析为 printdateextra, printlabeldateextra:

```

1 \blx@regimcs{\printlabeldate \printlabeltime \printlabeldateextra \stripzeros \forcezerosy \
   forcezerosmdt \mkyearzeros \mkmonthzeros \mkdayzeros \mktimezeros}

```

biblatex 很多命令都有类似的方式。

2. 日期的数据来源由 DeclareLabeldate 定义选择。日期的具体格式，则由宏包选项控制。比如 labeldate=ymd 来控制。(3.5 版以上，以前的是 datelabel 选项)

4.7 作者格式控制

★【注意】：texlive2015 中的 biblatex 版本是 3.0，texlive2016 中 biblatex 的版本是 3.4，新版本对于名字域打印有了较大变化（即不同版本的 biblatex 对于 DeclareNameFormat 的输入参数处理有所不同，如例26所示），所以需做相应的修改，为此在 biblatex 中首先进行版本判断，然后根据版本不同分别进行设置。

示例 26. texlive2016 中 biblatex3.4 版 Name 域格式输入参数的修改

代码

```

1   for biblatex version 3.0
2   #1 The last names.
3   #2 The last names, given as initials.
4   #3 The first names.
5   #4 The first names, given as initials.
6   #5 The name prefixes,
7   #6 The name prefixes, given as initials.
8   #7 The name affixes,
9   #8 The name affixes, given as initials.
10  for biblatex version 3.4
11  \namepartfamily
12  \namepartfamilyi

```

```

13 \namepartgiven
14 \namepartgiveni
15 \namepartprefix
16 \namepartprefixi
17 \namepartsuffix
18 \namepartsuffixi

```

样式文件中的处理见4.7.1节。

4.7.1 作者信息输出与格式控制

作者信息的输出最高一层是在驱动中:

```
1 \usebibmacro{author/editor+others/translator+others}
```

其中当存在作者时, 由如下宏输出:

```
1 \renewbibmacro*{author}
```

其中作者列表又由如下命令输出:

```
1 \printnames{author}
```

printnames 这种命令的输出格式通常由相应输出域的域格式控制, 这里则是 author 域控制:

```

1 \DeclareNameAlias{author}{sortname}%for author year style
2
3 \DeclareNameAlias{author}{default}%for numeric style

```

其中 sortname, default 域格式又是:

```

1 \DeclareNameAlias{sortname}{family-given/given-family}
2
3 \DeclareNameAlias{author}{default}
4 \DeclareNameAlias{default}{given-family}

```

其中的关键是 given-family 和 family-given 两个域格式使用 name:family-given 和 name:family-given 宏进行输出。真实的作者姓名格式在这两个宏中, 因此做一定的处理。注意其中的输入参数为: namepartfamily, namepartgiveni, namepartprefix, namepartsuffix (使用名的首字母缩写) 或 namepartfamily, namepartgiven, namepartprefix, namepartsuffix (不使用缩写)。

```

1 %biblatex3.3版后(比如texlive2016中的3.4版)的使用方式
2 \renewbibmacro*{name:given-family}[4]{%利用family-given定义given-family
3   \ifuseprefix
4     {\usebibmacro{name:delim}{##3##1}%
5     \usebibmacro{name:hook}{##3##1}%
6     \ifdefvoid{##3}{}%
7     \ifcapital
8       {\mkbibnameprefix{\MakeCapital{##3}}\isdot}
9       {\mkbibnameprefix{##3}\isdot}%
10    \ifprefchar{}{\bibnamedelimc}}%
11    \ifdefvoid{##2}{\mkbibnamefamily{\MakeCapital{##1}}}{\mkbibnamefamily{\MakeUppercase{##1}}}\
      isdot
12    \ifdefvoid{##2}{%{\revsdsnamepunct\bibnamedelimd\mkbibnamegiven{\MakeUppercase{##2}}\isdot}%\
      MakeCapital

```

```

13 \ifdefvoid{##4}{-}{\bibnamedelimd\mkbibnamesuffix{##4}\isdot}}
14 {\usebibmacro{name:delim}{##1}%
15 \usebibmacro{name:hook}{##1}%
16 \ifdefvoid{##2}{\mkbibnamefamily{\MakeCapital{##1}}}{\mkbibnamefamily{\MakeUppercase{##1}}}\
    isdot
17 %
18 \ifboolexpe{%
19 test {\ifdefvoid{##2}}
20 and
21 test {\ifdefvoid{##3}}}
22 {}
23 {\revsnamepunct}%
24 \ifdefvoid{##2}{-}{\bibnamedelimd\mkbibnamegiven{\MakeUppercase{##2}\isdot}%\MakeCapital
25 \ifdefvoid{##3}{-}{\bibnamedelimd\mkbibnameprefix{##3}\isdot}
26 \ifdefvoid{##4}{-}{\bibnamedelimd\mkbibnamesuffix{##4}\isdot}}
27 }
28 \renewbibmacro*{name:family-given}[4]{%
29 \usebibmacro{name:given-family}{##1}{##2}{##3}{##4}}
30
31 %biblatex3.3版前(比如texlive2015中的3.0版)的使用方式
32 \renewbibmacro*{name:last-first}[4]{%
33 \ifuseprefix
34 {\usebibmacro{name:delim}{##3##1}%
35 \usebibmacro{name:hook}{##3##1}%
36 \ifblank{##3}{-}{%
37 \ifcapital
38 {\mkbibnameprefix{\MakeCapital{##3}}\isdot}
39 {\mkbibnameprefix{##3}\isdot}%
40 \ifpunctmark{'}{-}{\bibnamedelimc}}}%
41 %\mkbibnamelast{##1}\isdot
42 \ifblank{##2}{\MakeCapital{##1}}{\mkbibnamelast{\MakeUppercase{##1}}\isdot%\MakeUppercase %\
    mkbibnamelast{\MakeUppercase{##1}} %\MakeSentenceCase
43 %注意上一句\MakeCapital后面如果再跟一个{}包含##1, 则没有效果, 可能是包在里面少了一层展开
44 %因为机构名通常包括在{}内, 所以要多展开一次才行, 所以这里去掉##1 外面的{}
45 %\mkbibnamelast{\MakeUppercase{##1}}\isdot
46 %\ifblank{##4}{-}{\bibnamedelimd\mkbibnameaffix{##4}\isdot}%这句放到后面
47 \ifblank{##2}{-}{\revsnamepunct\bibnamedelimd\mkbibnamefirst{\MakeUppercase{##2}}\isdot}%
48 \ifblank{##4}{-}{\addcomma\addspace\bibnamedelimd\mkbibnameaffix{##4}\isdot}}
49 {\usebibmacro{name:delim}{##1}%
50 \usebibmacro{name:hook}{##1}%
51 %\mkbibnamelast{##1}\isdot %3.9.1 Generic Commands and Hooks, 对姓重新处理, 如下句: %\
    mkbibnamelast{\MakeUppercase{##1}}
52 \ifblank{##2}{\MakeCapital{##1}}{\mkbibnamelast{\MakeUppercase{##1}}\isdot %大写, 参考4.6.4
    Miscellaneous Commands, \MakeUppercase %\MakeSentenceCase
53 %\mkbibnamelast{\MakeUppercase{##1}}\isdot
54 %\ifblank{##4}{-}{\bibnamedelimd\mkbibnameaffix{##4}\isdot}%这句放到后面
55 \ifblank{##2##3}{-}{\revsnamepunct}%
56 \ifblank{##2}{-}{\bibnamedelimd\mkbibnamefirst{\MakeUppercase{##2}}\isdot}%
57 \ifblank{##3}{-}{\bibnamedelimd\mkbibnameprefix{##3}\isdot}%

```

```

58      \addcomma\addspace%
59      \ifblank{##4}{-}{\addcomma\addspace\bibnamedelimd\mkbibnameaffix{##4}\isdot}%
60    }
61  }
62
63  \renewbibmacro*{name:first-last}[4]{%
64  \usebibmacro{name:last-first}{##1}{##2}{##3}{##4}}

```

4.8 样式文件设计途径

biblatex 的核心-参考文献样式文件的设计，更详细内容见 [LaTeX 文档中文参考文献的 biblatex 解决方案](#)。

1. driver 的作用: 条目驱动，修改一些顺序，略去一些输出和标点
2. micro 的作用: 输出宏，修改需要的输出内容
3. format 的作用: 域打印格式，修改一些斜体，强调样式等。
4. command 的作用: 一些设置和命令
5. biblatex.sty/def 和各 bbx/cbx 文件中的代码

4.8.1 文献表条目内容组织的原理以及问题查找的方法

biblatex 是利用 tex 语句控制文献内容组织，因此我们可以坚定一点就是，对于文献内容组织一定不是神秘的，是可以分析清楚的，当出现问题的时候，通过一步步的分析是能解决好的。

如前所述，根据宏包设置，使用 biber 解析完参考文献信息后，就需要利用样式中的各种定义来输出文献表。我们从某一条参考文献来看问题。某一类型的参考文献输出由该类型的 driver 控制，driver 中控制各种 bibmacro 来组织内容输出，bibmacro 利用各种 fieldformat，command 来控制域的格式和标点，当出现一些特殊的问题时，我们可以顺着这样的思路来查找。

下面我们举例来说明本样式的作者年制中，当作者缺省时，利用 title 判断中英文出错的问题。这个问题本质上由于前面说过的，当作者缺省时，把 title 作为标签后，会把 title 域清除，因此再利用 title 域信息判断中英文，自然无法正常工作。我们来看看这是怎么一个过程：

这是一个作者缺省的 book 条目，在 gb7714-2015ay 样式不使用佚名方式的情况下报错，主要是 testCJKfirst 函数报错。这个函数主要用在作者，出版项等这些信息输出时的中英文判断。因为作者缺省，因此作者输出时就没有判断，而出错只能是在出版项中。我们把 book 的 driver 中 macro 输出一项一项注释掉并测试发现，当作者输出相关的 macro 存在时存在错误，注释掉则没有问题。于是深入来分析：

在 book 类的 driver 简化为如下三项输出：

```

1 \DeclareBibliographyDriver{book}{%
2   \usebibmacro{bibindex}%
3   \usebibmacro{begentry}%
4   \usebibmacro{author/editor+others/translator+others}%
5   \setunit{\labelnamepunct}\newblock
6   \usebibmacro{maintitle+title}%
7   \newunit
8   \usebibmacro{publisher+location+date}%
9   \newunit\newblock

```

```
10 \usebibmacro{finentry}}
```

其中 `\usebibmacro{author/editor+others/translator+others}` 导致出错, 该 macro 的思路是当存在 author 时输出 author, 没有时判断是否存在 editor, 存在则输出, 不存在则判断是否存在 translator, 并进一步判断输出:

```
1 \newbibmacro*{author/editor+others/translator+others}{%
2   \ifboolexpr{
3     test \ifuseauthor
4     and
5     not test {\ifnameundef{author}}
6   }
7   {\usebibmacro{author}}
8   {\ifboolexpr{
9     test \ifuseeditor
10    and
11    not test {\ifnameundef{editor}}
12  }
13   {\usebibmacro{editor+others}}
14   {\usebibmacro{translator+others}}}}
```

显然这个例子中不存在 author 和 editor, 因此宏转到 `\macro{translator+others}`, 我们来看这个宏:

```
1 \newbibmacro*{translator+others}{%
2   \ifboolexpr{
3     test \ifusetranslator
4     and
5     not test {\ifnameundef{translator}}
6   }
7   {\printnames{translator}%
8     \setunit{\printdelim{translatortypedelim}}%
9     \usebibmacro{translator+othersstrg}%
10    \clearname{translator}}
11  {}}
```

这个宏来自 `biblatex.def`, 显然这个宏表明当 translator 不存在则不做任何事情, 这种情况下是不会出错的, 于是我们再找, 发现这个宏在 `authoryear.bbx` 中重定义了。

```
1 \renewbibmacro*{translator+others}{%
2   \usebibmacro{bbx:translator}{translator+othersstrg}}
3
4 \newbibmacro*{bbx:translator}[1]{%
5   \ifboolexpr{
6     test \ifusetranslator
7     and
8     not test {\ifnameundef{translator}}
9   }
10  {...code for existed translator...}%
11  {\global\undef\bbx@lasthash
12   \usebibmacro{labeltitle}%
13   \setunit*{\printdelim{nonameyeardelim}}}%
}
```



```
14 \usebibmacro{date+extradate}}
```

显然当 translator 不存在时, 使用了 labeltitle 宏, 再来看这个宏:

```
1 \newbibmacro*{labeltitle}{%
2   \iffieldundef{label}
3     {\iffieldundef{shorttitle}
4       {\printfield{title}%
5         \clearfield{title}}
6       {\printfield{title}{shorttitle}}}
7   {\printfield{label}}}
```

从中可以看到, 该宏把 title 域作为标签进行输出, 并在输出后清除了 title 域, 自然, 在完成了整个 author/editor+others/translator+others 宏后, title 域就不存在了, 因此在后面的 publisher+location+date 宏中使用 title 域来进行中英文判断是必然会出错的, 找到了问题所在, 那么就有相应的方法, 比如在动态数据修改时, 把 title 信息存储到 userd 域中, 然后 publisher+location+date 宏利用该域的信息进行判断这样就不会出现问题了。

4.9 宏包选项增加

biblatex 的宏包选项增加

1. 在 gb7714-2015.bbx 文件中有四种常见的判断方式包括: 版本的判断, 宏包选项的判断, 宏包选项控制的切换标记的判断, 一般的域的是否定义或与字符串比较是否相同的判断。

第一种判断, 是直接根据已有的信息比如 `\abx@version` 进行处理, 该出的处理结果是直接展开的, 可以用于后面的任意定义。

第二种判断, 是根据给出的宏包选项值或者不给出选项进行处理, 例如定义的宏包选项 `gbalign`, 这一选项是 string 类型。在选项的执行代码中直接给出每一个选项值对应要执行的命令。比如:

```
1 \DeclareBibliographyOption[string]{gbalign}[right]{%
2   \ifstrequal{##1}{left}{\setalignleft}{}
3   \ifstrequal{##1}{gb7714-2015}{\setaligngbstyle}{}
4 }
```

这种情况下, 特别要注意 `DeclareBibliographyOption` 选项定义命令中给出的默认值 `right` 是在加载该选项但不给出值时的默认值, 而不是将选项设置为该默认值。因此当加载时不给出 `gbalign` 选项时, 不执行任何的语句, 即不执行 `setalignleft`, 也不执行 `setaligngbstyle`(因为不使用这两个命令的情况下项标签是右对齐的, 这样也就是 `gbalign` 选项默认是右对齐的, 只有设置宏包选项为 `left` 或 `gb7714-2015` 时才会因为 `setalignleft` 和 `setaligngbstyle` 命令二改变)。如果给出选项 `gbalign`, 相当于 `gbalign=` 默认值, 如果给出选项 `gbalign=` 指定值, 那么值就是指定值, 然后并执行 `DeclareBibliographyOption` 的定义代码, 根据值进行处理。

第三种判断, 根据宏包选项控制对切换标记进行设置, 然后再代码中应用切换标记判断。比如 `gbpub` 选项, 这一选项类似于 `standard.bbx` 中定义的 `url` 等选项。

```
1 \newtoggle{bbx:gbpub}
2 \DeclareBibliographyOption[boolean]{gbpub}[true]{%
3   %\settoggle{bbx:gbpub}{#1} %或采用下面这一句
4   \ifstrequal{##1}{false}{\togglefalse{bbx:gbpub}}{\toggletrue{bbx:gbpub}}}
```

默认可以利用切换标记 `bbx:gbpub` 进行任何的定义。但如果展开的层次是与这里定义的宏包选项同一层次的，那么切换标记会根据默认值是 `false` 而展开。如果是下一层的展开（就是不需要在加载时展开的），那么会根据展开时切换标记 `bbx:gbpub` 的值进行展开。

注意到在上述这一定义后面，使用了选项默认执行命令：

```
1 \ExecuteBibliographyOptions{gbpub}
```

这是宏包选项定义中的代码会立刻被执行，即 `bbx:gbpub` 的值设置为 `true`。`ExecuteBibliographyOptions` 命令是用来执行选项的，如果不给出这一命令，且宏包加载选择中没有给出选项设置，那么 `toggle` 只有新建 `toggle` 时的默认定义。还要注意这里的切换标记设置是立刻进行的，如果这里的定义不是设置切换标记，而是直接调用命令，比如上面的 `gbalign` 的情况，那么就需要该被调用命令是有定义的，即定义至少与宏包选项定义在同一展开层次，甚至先于宏包选项定义。

还需要注意，如果给出了命令

```
1 \ExecuteBibliographyOptions{gbpub}%等价于下一条命令
2 \ExecuteBibliographyOptions{gbpub=true}
3 \ExecuteBibliographyOptions{gbpub=false}
```

等价于在宏包加载时给出选项。但是宏包加载选项设置可以覆盖这个命令的设置。

如果在宏包加载时给出选项 `gbpub=false`，那么 `bbx:gbpub` 的值设置为 `false`。如果在宏包加载时给出选项 `gbpub`，`gbpub=true`，那么 `bbx:gbpub` 的值设置为 `true`。

第四种判断，利用 `iffielddundef` 和 `iffieldequalstr` 进行判断即可。

2. 总结起来，宏包选项的问题需要注意宏包选项的展开顺序。宏包选项的定义如果不使用 `ExecuteBibliographyOptions` 命令或者不在加载时设置，那么定义的代码是不执行的。因此如果其他代码中需要使用该宏包选项定义代码中的信息是无法正确获取的。而当使用 `ExecuteBibliographyOptions` 命令或者加载选项时，那么宏包选项代码会立即展开，其中需要使用的宏则需要有定义，即先于或与宏包选项定义处于同一展开层。

```
1 %\newtoggle{bbx:gbpub}
2 %\DeclareBibliographyOption[boolean]{gbpub}[true]{%
3 % \settoggle{bbx:gbpub}{#1}}
4 %这种机制的标识判断，能用于usemacro使用，域格式定义中，但无法用来判断后定义macro
5 %定义宏和使用宏是两个不同的展开层级，而可以在定义macro中进行判断
6 %\ExecuteBibliographyOptions{gbpub}
7 %但是使用这句默认设置可以用来定义macro
8
9 %\DeclareBibliographyOption{gbpub}[true]{%应使用这种直接的机制
10 %\ifstrequal{#1}{false}{\pubaddmacronewdefine}{\pubaddmacroredefine}}
11
12 %特别注意DeclareBibliographyOption，而给出的默认值是在加载时没有选项没有给出值时的默认值。
13 %当加载时不给出gbpub选项时，不执行ifstrequal判断语句，无任何加载。
14 %如果给出选项gb，或者gb=true，那么ifstrequal判断仍然是假，执行的默认的加载\pubaddmacroredefine
15 %如果给出选项false，那么ifstrequal判断是真，执行的就是非默认的加载\pubaddmacronewdefine
16 %\ExecuteBibliographyOptions{gbpub}%这句报错是因为\pubaddmacroredefine没有定义，如果把定义放到前面那么应该是可以的。
17
18 %还是采用另一种方式可以较好的进行默认选项的处理
19 \newtoggle{bbx:gbpub}
```

```

20 \DeclareBibliographyOption[boolean]{gbpub}[true]{%
21 %\settoggle{bbx:gbpub}{#1} %或采用下面这一句
22 \ifstrequal{#1}{false}{\togglefalse{bbx:gbpub}}{\toggletrue{bbx:gbpub}}}
23 \ExecuteBibliographyOptions{gbpub}

```

4.10 文献表环境和打印输出

biblatex 宏包的参考文献表的打印

1. 根据 biblatex.sty 文件中的 `\blx@bibliography` 命令可以知道, 定义的参考文献表的环境的开始代码在命令中 `\blx@bibliography` 中, 结束代码在 `\blx@endbibliography` 中, 循环项代码在 `\blx@bibitem` 中的 `\blx@item@{\blx@theenv}` 命令中。参考文献循环打印依靠 `\blx@listloop` 命令实现。

2. 从 `\blx@listloop` 可以知道, 各条参考文献表的指引信息是混合在一起的并且以 | 字符为分隔。将混合的参考文献指引信息解析成单独的指引信息, 并交给 `\blx@bibitem` 命令处理。

3. `\blx@bibitem` 首先开始一个编组, 然后根据单独一条参考文献的指引信息, 获取数据, 执行判断, 利用参考文献表环境定义的循环项代码对参考文献内容设置段落格式, 最后结束编组。在定义 `gbalign=gb7714-2015` 样式的段落格式时, 采用了 `parshape` 或者 `hangindent` 命令来实现, 所以其中增加了一个 `par` 命令以使其生效。

4.11 biblatex 版本以及样式的兼容性

样式首先根据 biblatex 版本信息进行版本判断, 然后做针对性的兼容处理。biblatex 在 texlive 中的版本对应关系是: texlive2015 对应 biblatex3.0; texlive2016 对应 biblatex3.4, 从 3.2 版开始修改了姓名机制 (如 4.7 节所述); texlive2017 对应 biblatex3.7; 2017 年底又出了 biblatex3.8a 到 3.10 版。

1. biblatex 版本判断

原来是使用一个简单的字符流处理来解析版本信息, 但容易出现问題。于是换成利用 xstring 宏包的命令来提取版本中的数字进行判断:

```

1 %版本判断
2 \providetoggle{iftexlivesix}%用于处理biblatex3.3开始改变的新的姓名机制
3 \providetoggle{iftexliveseven}%用于处理biblatex3.7的兼容性
4 \providetoggle{iftexliveeight}%用于处理biblatex3.8a的兼容性
5 %\def\numparserta#1.#2a\relax{#1}%注意relax的重要性
6 %\def\numparsertb#1.#2a\relax{#2}
7 %\def\numinteger{\expandafter\numparserta\abx@version\relax}
8 %\def\numdigital{\expandafter\numparsertb\abx@version\relax}
9 \StrChar{\abx@version}{3}{\numdigital}
10 %根据biblatex更新历史, 当版本大于3.2开始, 就用设置\toggletrue{iftexlivesix}
11 \ifnumcomp{\numdigital}{>}{2}{\toggletrue{iftexlivesix}}{\togglefalse{iftexlivesix}}
12 \ifnumcomp{\numdigital}{>}{6}{\toggletrue{iftexliveseven}}{\togglefalse{iftexliveseven}}
13 \ifnumcomp{\numdigital}{>}{7}{\toggletrue{iftexliveeight}}{\togglefalse{iftexliveeight}}

```

2. biblatex 版本的一些兼容性处理

宏包选项:

```

1 firstinits=true,%for texlive2014,2015;bibtex <=3.2
2 giveninits=true,%for texlive2016,2017;bibtex >=3.3

```

```

3  labeldate=true,%for texlive2014,2015,2016,2017;bibtex <3.8
4  labeldateparts=true,%for bibtex >=3.8a

```

命令:

```

1  \printdateextralabel,%for texlive2014,2015,2016;bibtex <=3.4
2  \printlabeldateextra,%for texlive2017;bibtex >=3.7
3  \DeclareSortingScheme %for texlive2014,2015,2016,2017;bibtex <3.8
4  \DeclareSortingTemplate %for bibtex >=3.8a

```

3. 版本与兼容性的最终处理方法

经过最终设计，下面给出的另一版本兼容性处理方式取代前面给出的处理方式，便于以后的扩展。这种方式自然兼容最新的版本，但老的版本的代码处理必须要到位，即因为 bibtex 版本升级增加了新的老版本时，对应该老版本的选择必须要实现，否则可能漏掉这些版本的处理。

```

1  %版本判断，当版本继续更新时，增加一个新的toggle用以处理新的旧版，最新版本永远用iftlatest
2  \providetoggle{iftfive}%用于处理bibtex3.2之前的版本
3  \providetoggle{iftlsix}%用于处理bibtex3.3开始改变的新的姓名机制后版本
4  \providetoggle{iftseven}%用于处理bibtex3.7的兼容性
5  \providetoggle{iftleight}%用于处理bibtex3.8到3.9的兼容性
6  \providetoggle{iftlatest}%用于最新版bibtex，目前是bibtex3.10，20180117
7  \StrBefore{\abx@version}{.}[\numinteger]%以点区分整数和小数
8  \StrBehind{\abx@version}{.}[\numdigital]%小数部分可能带字母所以需将其去掉，见下一行代码
9  \IfInteger{\numdigital}{\StrGobbleRight{\numdigital}{1}[\numdigital]}
10 \togglefalse{iftfive}
11 \togglefalse{iftlsix}
12 \togglefalse{iftseven}
13 \togglefalse{iftleight}
14 \toggletrue{iftlatest}
15 \ifnumcomp{\numinteger}={}{3}{
16   \ifnumcomp{\numdigital}{<}{10}{\ifnumcomp{\numdigital}{>}{7}{\toggletrue{iftleight}\togglefalse
17     {iftlatest}}}{}}}%3.8 版本用iftleight=true表示
18   \ifnumcomp{\numdigital}{=}{7}{\toggletrue{iftseven}\togglefalse{iftlatest}}}%3.7 版本用
19     iftseven=true表示
20   \ifnumcomp{\numdigital}{<}{7}{\ifnumcomp{\numdigital}{>}{2}{\toggletrue{iftlsix}\togglefalse{
21     iftlatest}}}{}}}%3.3-3.6 版本用iftlsix=true表示
22   \ifnumcomp{\numdigital}{<}{3}{\toggletrue{iftfive}\togglefalse{iftlatest}}}%3.0-3.2 版本用
23     iftfive=true表示
24 }{\ifnumcomp{\numinteger}{>}{3}{
25   \blx@warning@noline{%
26     biblatex version is >= 4.x.\MessageBreak
27     if errors raised, Please contact biblatex-gb7714-2015 pkg author.}%
28 }{\toggletrue{iftfive}\togglefalse{iftlatest}}}%2.x版本统一用iftfive=true表示
29   \blx@warning@noline{%
30     biblatex version is <= 2.x.\MessageBreak
31     if errors raised, Please contact biblatex-gb7714-2015 pkg author.}%
32 }}
33 \newcommand\defversion[2]{\csdef{codeversion#1#2}}%定义不同版本的命令
34 \newcommand\switchversion[2]{\csuse{codeversion#1#2}}%使用不同版本的命令
35 \defversion{3.0}{opt}{

```

```

33 %biblatex3.3版前(比如texlive2015中的3.0版)的使用方式
34
35 %增加姓名大小写格式处理选项
36 \DeclareBibliographyOption{gbnamefmt}[uppercase]{%
37     %\ifstrequal{##1}{gbuppercase}{-}{-}%
38     \ifstrequal{##1}{lowercase}{\execgblowercase}{-}%
39     \ifstrequal{##1}{none}{\execnamedefault}{-}
40     \ifstrequal{##1}{pinyin}{\execnamepinyin}{-}
41 }
42 %\ExecuteBibliographyOptions{gbnamefmt}
43
44 %增加标签对齐选项, right是默认的右对齐, left是左对齐, gb7714-2015无效, 仍然为右对齐模式, 因为舍弃了
    list类环境后出错。
45 \DeclareBibliographyOption{gbalign}[right]{%texlive2015中的3.0版中的DeclareBibliographyOption选项没
    有类型说明
46     \ifstrequal{##1}{right}{-}{-}
47     \ifstrequal{##1}{left}{\setalignleft}{-}
48     \ifstrequal{##1}{gb7714-2015}{-}{-}
49 }
50
51 %增加一个出版项自动处理控制选项, 当true时使用出版者不详等信息补充缺失的出版信息。
52 \newtoggle{bbx:gbpub}
53 \DeclareBibliographyOption{gbpub}[true]{%
54     %\settoggle{bbx:gbpub}{#1} %或采用下面这一句
55     \ifstrequal{##1}{false}{\togglefalse{bbx:gbpub}}{\toggletrue{bbx:gbpub}}
56 \ExecuteBibliographyOptions{gbpub}
57
58 %增加一个处理佚名或noauthor的控制选项
59 %因为在顺序编码制中, 不需要使用, 这里也增加, 为了与作者年制的兼容性考虑。
60 \DeclareBibliographyOption{gbnoauthor}[true]{-}%
61
62 \ExecuteBibliographyOptions{
63     sorting=none,
64     useprefix=true, %名字的信息包括前缀
65     firstinits=true, %名字有缩写, 参考3.1.2.3 Internal
66     %giveninits=true,
67     date = year, %日期仅写到年
68     maxnames = 3, %设置名字最大数量
69     minnames = 3 %设置缩减后的名字最小数量
70     %uniquename = init
71 }
72 }
73
74 \defversion{3.4}{opt}{
75     % Alter settings that carry through from biblatex
76     %biblatex3.3版后(比如texlive2016中的3.4版)的使用方式
77
78     %增加姓名大小写格式处理选项
79     \DeclareBibliographyOption[string]{gbnamefmt}[uppercase]{%

```

```

80     %\ifstrequal{##1}{gbuppercase}{-}{-}%
81     \ifstrequal{##1}{lowercase}{\execgblowercase}{-}%
82     \ifstrequal{##1}{none}{\execnamedefault}{-}%
83     \ifstrequal{##1}{pinyin}{\execnamepinyin}{-}%
84 }
85 %\ExecuteBibliographyOptions{gbnamefmt}
86
87 %增加标签对齐选项, right是默认的右对齐, left是左对齐, gb7714-2015是项对齐方式
88 \DeclareBibliographyOption[string]{gbalign}[right]{%
89     \ifstrequal{##1}{left}{\setalignleft}{-}%
90     \ifstrequal{##1}{gb7714-2015}{\setaligngbstyle}{-}%
91 }
92
93 %增加一个出版项自动处理控制选项, 当true时使用出版者不详等信息补充缺失的出版信息。
94 \newtoggle{bbx:gbpub}
95 \DeclareBibliographyOption[boolean]{gbpub}[true]{%
96     %\settoggle{bbx:gbpub}{#1} %或采用下面这一句
97     \ifstrequal{##1}{false}{\togglefalse{bbx:gbpub}}{\toggletrue{bbx:gbpub}}%
98 \ExecuteBibliographyOptions{gbpub}
99
100 %增加一个处理佚名或noauthor的控制选项
101 %因为在顺序编码制中, 不需要使用, 这里也增加, 为了与作者年制的兼容性考虑。
102 \DeclareBibliographyOption[boolean]{gbnoauthor}[true]{-}%
103
104 \ExecuteBibliographyOptions{
105     sorting=none,
106     useprefix=true, %名字的信息包括前缀
107     %firstinits=true, %名字有缩写, 参考3.1.2.3 Internal
108     giveninits=true,
109     date = year, %日期仅写到年
110     maxnames = 3, %设置名字最大数量
111     minnames = 3 %设置缩减后的名字最小数量
112     %uniquename = init
113 }
114 }
115
116 \defversion{3.7}{opt}{%
117     %biblatex3.7版后(比如texlive2017中的3.7版)的使用方式
118
119 %增加姓名大小写格式处理选项
120 \DeclareBibliographyOption[string]{gbnamefmt}[uppercase]{%
121     %\ifstrequal{##1}{gbuppercase}{-}{-}%
122     \ifstrequal{##1}{lowercase}{\execgblowercase}{-}%
123     \ifstrequal{##1}{none}{\execnamedefault}{-}%
124     \ifstrequal{##1}{pinyin}{\execnamepinyin}{-}%
125 }
126 %\ExecuteBibliographyOptions{gbnamefmt}
127
128 %增加标签对齐选项, right是默认的右对齐, left是左对齐, gb7714-2015是项对齐方式

```



```

129 \DeclareBibliographyOption[string]{gbalign}[right]{%
130     \ifstrequal{##1}{left}{\setalignleft}{}
131     \ifstrequal{##1}{gb7714-2015}{\setaligngbstyle}{}
132 }
133
134 %增加一个出版项自动处理控制选项，当true时使用出版者不详等信息补充缺失的出版信息。
135 \newtoggle{bbx:gbpub}
136 \DeclareBibliographyOption[boolean]{gbpub}[true]{%
137     %\settoggle{bbx:gbpub}{#1} %或采用下面这一句
138     \ifstrequal{##1}{false}{\togglefalse{bbx:gbpub}}{\toggletrue{bbx:gbpub}}
139 \ExecuteBibliographyOptions{gbpub}
140
141 %增加一个处理佚名或noauthor的控制选项
142 %因为在顺序编码制中，不需要使用，这里也增加，为了与作者年制的兼容性考虑。
143 \DeclareBibliographyOption[boolean]{gbnoauthor}[true]{}%
144
145 \ExecuteBibliographyOptions{
146     sorting=none,
147     useprefix=true, %名字的信息包括前缀
148     %firstinits=true, %名字有缩写，参考3.1.2.3 Internal
149     giveninits=true,
150     date = year, %日期仅写到年
151     urldate =edtf, %iso8601
152     eventdate =edtf,
153     maxnames = 3 , %设置名字最大数量
154     minnames = 3 %设置缩减后的名字最小数量
155     %uniquename = init
156 }
157 }
158
159 \defversion{3.10}{opt}{
160     %biblatex3.10版后的使用方式
161
162     %增加姓名大小写格式处理选项
163     \DeclareBibliographyOption[string]{gbnamefmt}[uppercase]{%
164         %\ifstrequal{##1}{gbuppercase}{}{}%
165         \ifstrequal{##1}{lowercase}{\execgblowercase}{}%
166         \ifstrequal{##1}{none}{\execnamedefault}{}
167         \ifstrequal{##1}{pinyin}{\execnamepinyin}{}
168     }
169     %\ExecuteBibliographyOptions{gbnamefmt}
170
171     %增加标签对齐选项，right是默认的右对齐，left是左对齐，gb7714-2015是项对齐方式
172     \DeclareBibliographyOption[string]{gbalign}[right]{%
173         \ifstrequal{##1}{left}{\setalignleft}{}
174         \ifstrequal{##1}{gb7714-2015}{\setaligngbstyle}{}
175     }
176
177     %增加一个出版项自动处理控制选项，当true时使用出版者不详等信息补充缺失的出版信息。

```

```

178 \newtoggle{bbx:gbpub}
179 \DeclareBibliographyOption[boolean]{gbpub}[true]{%
180     %\settoggle{bbx:gbpub}{#1} %或采用下面这一句
181     \ifstrequal{##1}{false}{\togglefalse{bbx:gbpub}}{\toggletrue{bbx:gbpub}}}
182 \ExecuteBibliographyOptions{gbpub}
183
184 %增加一个处理佚名或noauthor的控制选项
185 %因为在顺序编码制中，不需要使用，这里也增加，为了与作者年制的兼容性考虑。
186 \DeclareBibliographyOption[boolean]{gbnoauthor}[true]{}%
187
188 \ExecuteBibliographyOptions{
189     sorting=none,
190     useprefix=true, %名字的信息包括前缀
191     %firstinits=true, %名字有缩写，参考3.1.2.3 Internal
192     giveninits=true,
193     date = year, %日期仅写到年
194     urldate =iso, %iso8601, edtf
195     eventdate =iso,
196     maxnames = 3 , %设置名字最大数量
197     minnames = 3 %设置缩减后的名字最小数量
198     %uniquename = init
199 }
200 }
201
202 \iftoggle{ifftlfive}{\switchversion{3.0}{opt}}{}%biblatex<=3.2
203 \iftoggle{ifftlsix}{\switchversion{3.4}{opt}}{}%3.3<=biblatex<=3.6
204 \iftoggle{ifftlseven}{\switchversion{3.7}{opt}}{}%biblatex=3.7
205 \iftoggle{ifftleight}{\switchversion{3.7}{opt}}{}%3.8<=biblatex<=3.9
206 \iftoggle{ifftlatest}{\switchversion{3.10}{opt}}{}%biblatex最新3.10

```

4.12 脚注或旁注文献

标注命令自定义，在脚注中引用文献，或者把引文放在脚注中或者旁注中

1. 标注命令定义可以使用 `DeclareCiteCommand` 命令，设置上标，包围符号，标注前后内容等。针对 GB/T7714-2015 中关于引用标注的特殊要求，增加了一些方便实现要求的命令比如 `pagescite`、`yearpagescite` 等。

2. 在 footnote 的脚注文本中使用 `footcite` 命令

3. 在正文中使用 `footfullcite` 命令，见 4.11.6 Mixing Programming Interfaces

4.13 参考文献标题

参考文献标题修改并将标题加入目录中

1. 利用 `defbibheading{bibliography}[\bibname]{}` 重定义标题

2. 使用 `bibliography` 外的其它标题样式比如 `bibintoc` 等

3. 利用 `titlesec` 等宏包进行修改标题样式，利用其局部性可以调整生成多种样式。注意：利用 `titlesec` 后，直接在 `defbibheading{bibliography}[\bibname]{}` 中放置居中命令 `centering` 无效。

4.14 其它需要说明的问题

下面的问题想到哪写到哪，没有特别的顺序：

1. 因为采用 xelatex 编译，所以样式文件直接采用 UTF-8 编码，没有考虑 GBK 编码。

2. ★【Most Important】【注意】：当在顺序编码和作者年制的切换，或者 biblatex 版本切换时，如果使用出错，可先清理一下辅助文件，清理完后，重新编译即可。

3. 需要注意：当 bibtex 键中含有中文的时候，texlive2015 中的 biblatex3.0 版的对参考文献条目的超链接会出现问题，而 texlive2016 中的 biblatex3.4 或以后的版本则没有问题。

4. GB/T 7714-2015 中的作者年制要求参考文献按文种集合，且中文在前英文在后。主要通过定义 `DeclareSortingScheme{nyt}`(biblatex3.7 以前版本) 或 `DeclareSortingTemplate{nyt}`(biblatex3.8 以后版本)，利用 `userb` 域排序实现。默认情况下，样式能基本正确的区分中文和英文文献并排序。当如果出现错误的情况，用户可以手动修改 bib 源文件，将 `userb` 域设置成合适的字符串，用于排序，详见 1.3.2 节的说明。

5. 关于出版地和出版者同时缺省的情况，GB/T 7714-2015 中没有给出明确的说明，但英文给出了一个例子 (见 GB/T 7714-2015 附录 A.3) 而中文没有，英文的样式是 `[S.l. : s.n.]`，这种形式本样式文件中没有给出，而直接用两者分开的形式，`[S.l.] : [s.n.]`，事实上这里作者认为没有必要把 `s.l.` 和 `s.n.` 合起来，不仅与缺省两者之一的情况不统一，样式处理起来也增加不必要的麻烦。

6. 目前符合 GB/T7714-2005 或 GB/T7714-2015 参考文献著录规则的 biblatex 样式有好几个实现，除本样式外，还有李志奇 (icetea)^④ 和沈周 (szsdk)^⑤ 分别提供的样式文件，效果是类似的。此外，Casper Ti. Vector 提供的 biblatex 样式 `caspervector` 也是不错的中文参考文献样式^⑥。感谢各位作者的分享！

7. 本文档根据 GB/T 7714-2015 提供的参考文献表著录格式示例做了测试和验证，详见第 3.1 节。测试系统环境为：

- windows7x86+texlive 2014，采用 xelatex 编译；
- windows7x64+texlive 2015，采用 xelatex 编译；
- 虚拟机 xp+texlive 2016，采用 xelatex 编译；
- Deepin linux-x64v15.3+texlive 2016，采用 xelatex 编译。
- windows7x64+texlive 2017，采用 xelatex 编译；

5 总结与致谢

通过对 GB/T 7714-2015 标准的分析，对 biblatex 的学习和理解，在 biblatex 标准标准样式基础上，设计完成了符合 GB/T 7714-2015 标准的 biblatex 参考文献样式。从测试实践看，基本能够满足使用要求，用户可以放心使用。遇到问题时，除了可以查看本文档说明外，也可以看样式文件代码，其中给出了详细注释，如果遇到无法解决的问题，请邮件联系作者。

^④<http://bbs.ctex.org/forum.php?mod=viewthread&tid=74474>

^⑤<http://bbs.ctex.org/forum.php?mod=viewthread&tid=152561&extra=page%3D1>

^⑥<https://gitlab.com/CasperVector/biblatex-caspervector>

最后要感谢如下各位师长和朋友，正是在各位的帮助建议下，本样式不断升级逐渐完善。包括：moewew(biblatex 现在的维护者之一，给了不少有益的建议)、李志奇(基于 biblatex 的符合 GBT7714-2005 的中文文献生成工具作者，笔者以前使用该工具，其代码对于设计本样式很有启发)、LeoLiu(刘海洋，CJK 字符判断函数^⑦对本宏包非常有帮助)、chinatex(china tex 版主，给了很多建议和帮助，并且一起合作)、Sheng wenbo(biblatex 用户手册合作译者，LaTeX2e 插图指南第三版译者，我们一起翻译的过程相互激励相互促进)、zepinglee(gbt7714-2015 bst 样式作者，给了很多建议和讨论)、Harry Chen(ctex 维护者，给了不少好的建议)、liubenyan(关于项目组织给出了很好的建议)、刘小涛(讨论了关于 zotero 的使用并提出了建议)、ghicigi(讨论了 GB 中作者年制标注标签的一些问题)、秀文工作组、leipility、qingkuan、湘厦人、秋平、任蒲军、fredericky123、qiuzhu、chaoxiaosu、Old Jack、Wu Nailong、Yibai Zhang、wayne508、钟乙源、Xiaodong Yao、dsycircle、rpjshu、zjsdut、谢澜涛、Zutian Luo、海阔天空、zzqzyx、程晨、xmtangjun、蔡伟等等。当然还有更多这里没有列全的朋友们的热心帮忙，在此一并表示感谢！

6 存在的问题和下一步工作

6.1 存在的问题

1. shorthand 的问题没有遇到，其应用可能需要进一步理解。
2. 当专著同时存在作者和编者的时候，GB/T 7714-2015 没有明确的规定，所以目前样式文件中以 biblatex 标准样式的方式处理，这种处理因为与本地化相关，直接应用可能不好看的，也许需要修改。
3. 因为 GB/T 7714-2015 中给出了一些著录格式，如果把这些著录格式作为一个严格标准，那么条目中只能出现其中规定的域，而往往在 bib 文件中可能存在一些另外的信息比如 chapter 等，而且从标准样式修改的驱动中也仍然带有这些域的处理，如果为了标准化规范化考虑，可以去掉国标中没有提到的域的信息，可能使得内容更为标准，这可以通过修改增加数据模型，数据源动态修改，驱动修改(驱动中目前存在较多的似乎用不到的域，而且意义不是非常明确，这个等到 biblatex 说明文档中文版完成后结合它全面的进行梳理)三条路子做到，需要的话，可以在下一步实现(2017-0226)。

6.2 下一步工作

1. 到 1.0i 版为止，除了基本功能需求外，进一步完善了：GB7714 风格的文献表标签项对齐设计，编组内信息的中英文判断，特殊或老的 bibtex 条目类型支持，改善空格设计以满足断行要求，支持了宏包选项(url 等)应用，增加了宏包选项用于 GB7714 风格实现控制(gbpub 等)，重新设计了版本兼容方式，以后的版本中将更容易兼容 biblatex 的升级。剩下的问题可能是一些文献具有特殊信息或者特殊情况时带来的适应性问题，这需要经过大量的测试来发现问题。各位朋友如果发现什么问题，请邮件联系，作者会非常感谢！

2. biblatex 宏包的说明文档中文版，已经由 Shen wenbo 和我基本完成，下一步是完善，校对，以及增加新版的内容。如果有朋友觉得这个事情有意义，愿意一起来完成这个事情，非常欢迎，请 email 联系。

^⑦<http://bbs.ctex.org/forum.php?mod=viewthread&tid=152663&extra=page%3D3>

7 更新历史

2018-04-03 date of update: 2018-04-03 to version v1.0k

1. 修正了 gbnamfmt 中的一些小错误。

correct some flaws for gbnamfmt option.

2. 修改了代码用于兼容 3.11 版本

change the separator before related block for v3.11.

3. 页码范围的间隔符从 en dash 改为 hyphen

change the pages range separator from en dash to hyphen.

4. 修正了 v3.7 以上版本中专利文献中公告日期后多出点的问题, 该 bug 是由于输入公告日期没有使用 printtext 导致异步标点机制破坏所致。

correct a bug of newdate in patent for biblatex >v3.7, which added an additional dot before urldate caused by broken asynchronous punctuation .

5. 修正了 texlive2017 以上版本中 beamer 类中标题后面多出点的问题, 由于 beamer 会对 bibmacro{title} 做 patch 导致其输出不同于普通文档类, 该 bug 是由于 beamer 升级后 patch 的内容发生变化导致。

correct a bug of punctuation after title with beamer for >texlive2017, the bug is caused by the update of beamer.

2018-04-03 update to version 1.0k

1. 修正了 gbnamfmt 中的一些小错误。

2. 修改了代码用于兼容 3.11 版本

3. 页码范围的间隔符从 en dash 改为 hyphen

4. 修正了 v3.7 以上版本中专利文献中公告日期后多出点的问题, 该 bug 是由于输入公告日期没有使用 printtext 导致异步标点机制破坏所致。

5. 修正了 texlive2017 以上版本中 beamer 类中标题后面多出点的问题, 由于 beamer 会对 bibmacro{title} 做 patch 导致其输出不同于普通文档类, 该 bug 是由于 beamer 升级后 patch 的内容发生变化导致。

2018-01-20 update to version 1.0j

1. 增加 gbtype 选项用于控制是否输出题名后的标识符, 见1.2.2 节。

2. 进一步修改了版本判断机制, 以使最新版本的兼容性更强, 见4.11节。

3. 根据刘小涛的需求和建议, 增加了 gbnamfmt 选项用于控制姓名的大小写和输出格式, 同时根据 zotero 从 cnki 识别输出中文文献姓名中带逗号的情况做了兼容性处理, 见3.12节。

4. 为统一样式增加的选项将原来的 align 选项修改为 gbalign, 见1.2.2节。

5. 通过正确使用 nameyear delim 相关命令, 修正了作者年制, 标注和著录表中的中作者与年份之间的标点符号, 见4.5.1节。

6. 根据 ghiclj 的建议, 增加了 yearcite 命令以满足, 作者年制中作者已经给出仅需要年份信息而不需要页码信息的情况, 说明和测试见3.2, 3.14节。

2017-11-21 update to version 1.0i

1. 因为 biblatex 版本升级, 3.8 及以上版的 set 类型不再复制第一个子条目的信息, 因此增加使用关联条目的解决方案, 详见 2.4, 4.4.3 节。

2. 修正了一个 liubenyan 发现的 bug。当标题中含有 `\LaTeX{}` 这样的宏时, cjk 判断函数出错。解决如下:

这个问题是这样的,

因为在 cjk 判断函数中, 使用了 `xstring` 的 `StrChar` 函数来抽取字符, 但这个函数默认情况下需要其参数完全展开。因为 `\LaTeX{}` 宏比较复杂, 展开时会出现问题。设置该函数不展开或展开一次, 都可以解决判断出错的问题。比如:

```
\expandarg
%
\StrChar{english}{1}[\tempa]%
\tempa

\StrChar{中文}{1}[\tempa]%
\tempa

\StrChar{english \LaTeX{ } abc}{1}[\tempa]%
\tempa
```

但解决的是直接给出文本的情况, 在 biblatex 使用中需要用 `\thefield` 取出文本, 显然 `\thefield` 不止展开一次, 因此不展开或者展开一次, 都会出现问题, 所以无解。只能从另外一个角度出发。

考虑到动态数据修改时, 也可以利用正则表达式抽取数据, 因此利用它来将 `title` 信息的第一个非特殊符号字符抽取出来, 放到 `userd` 中用于 cjk 判断, 这样就避开了 `\LaTeX{}` 展开的问题。

3. 针对 biblatex 3.8a 的更新做了兼容性处理, 主要是修改版本判断和处理机制, 替换新的宏包选项, 替换新的排序格式命令。详见第 4.11 节

4. 重写了范围解析函数, 详见 4.2 节

2017-04-11 update to version 1.0h

1. texlive2017 中 biblatex 3.7 对于 authoryear 样式中的 date+extrayear 宏有一定的修改, 从原来 texlive2016 中的命令 `printdateextralabel` 转换到了 `printlabeldateextra`。因此做一个修改。

```
%设置因为mergedate默认为true时的情况, 来自\bbx@opt@mergedate@compact
%去掉作者后面包围年份的圆括号
%这里因为biblatex版本的不同, 做不同的处理, texlive2017版的在iftexliveseven=true中设置
%2016等低版本的在iftexliveseven=false中设置
%重设date+extrayear格式, 源来自authoryear.BBX
\iftoggle{iftexliveseven}{%
  \renewbibmacro*{date+extrayear}{%
    \iffieldundef{labelyear}
    {}
    {\printtext{[parens]}%这里把括号去掉
      \iffieldsequal{year}{labelyear}
      {\printlabeldateextra}%
      {\printfield{labelyear}%
        \printfield{extrayear}}}}}%
}{%}
```



```

16 \renewbibmacro*{date+extrayear}{%
17   \iffieldundef{\thefield{datelabelsource}year}
18   {}
19   {\printtext{%[parens]}%这里把括号去掉
20     \iffieldsequal{year}{\thefield{datelabelsource}year}
21     {\printdateextralabel}%
22     {\printfield{labelyear}%
23       \printfield{extrayear}}}}}%
24 }

```

当然其实这还有更简单的解决方法就是:

```
1 \let\printdateextralabel=\printlabeldateextra
```

这个方式似乎有点问题。

2. 根据同学 (zjsdut@163.com) 发现的问题, 修改一个 bug, 感谢。当 online 类型仅有 url 信息时, url 前面多了一个点。这是 modifydate 宏设计中 printtext 位置导致标点异步处理机制失效所产生现象。因此作出修改, newbibmacro*{modifydate} 宏详见4.6节。

3. 增加一个选项 gbnoauthor。当给出选项 gbnoauthor=true 时, 作者年制中当作者缺省时, 使用佚名或 noauthor 代替, 即将佚名或 noauthor 作为作者处理。默认情况下 gbnoauthor=true 不处理, 即当无作者进行处理, 选择增加的代码详见4.11节。同时也修改了中英文排序判断和佚名代替的机制。

```

1 \map{%因为无法进行cjk字符判断, 所以用反的思路, 判断没有英文字符, 没有空格, 没有逗号等字符情况下
2   %认为是中文的, 将文献userb设置成cn, 用于排序
3   \step[fieldsource=author,match=\regex{[~a-zA-Z\s\.\,\'\{\}\-\:0-9]},final]
4   \step[fieldset=userb,fieldvalue={cn}]
5   }
6 \map{%因为无法进行cjk字符判断, 所以用反的思路, 判断没有英文字符, 没有空格, 没有逗号等字符情况下
7   %认为是中文的, 将文献userb设置成cn, 用于排序
8   \step[fieldsource=title,match=\regex{[~a-zA-Z\s\.\,\'\{\}\-\:0-9]},final]
9   \step[fieldset=userb,fieldvalue={cn}]
10  }%如果没有作者和标题, 那么剩下的最可能有意义的只有网址了, 而网址通常是英文的, 因此不用再进一步对
    其它域进行判断了。
11 \map{%将没有设置的userb设置成en, 即认为不是中文的就是英文的。
12   \step[fieldset=userb,fieldvalue={en}]
13   }

1 \def\dealnoauthor{
2 \DeclareStyleSourcemap{
3 \maps[datatype=bibtex]{
4   \map{%默认情况下将空缺作者设置为佚名
5     \step[fieldsource=userb,match={cn},final]
6     \step[fieldset=author, fieldvalue={佚名}]
7   }
8   \map{%默认情况下将空缺作者设置为佚名
9     \step[fieldsource=userb,match={en},final]
10    \step[fieldset=author, fieldvalue={NOAUTHOR}]
11  }
12 }
13 }}

```

下面是已经取消的以前的处理方式:

```

1 \map{%默认情况下将空缺作者设置为佚名
2 \step[fieldset=author, fieldvalue={佚名}]
3 }
4 \map[overwrite]{%当标题带有英文字符, 且作者为佚名的情况, 设置作者为noauthor,
5 %也有一些特殊情况可能处理不到, 比如标题中英混合, 作者又却是
6 \step[fieldsource=title,match=\regexp{[a-zA-Z]},final]
7 \step[fieldsource=author,match=\regexp{佚名}, replace={NOAUTHOR}]
8 }
9 \map{%因为无法进行cjk字符判断, 所以用反的思路, 判断没有英文字符, 没有空格, 没有逗号等字符情况下
10 %认为是中文的, 将文献userb设置成cn, 用于排序
11 \step[fieldsource=author,match=\regexp{[^a-zA-Z\s\.\,\'\{\}\-\]}},final]
12 \step[fieldset=userb,fieldvalue={cn}]
13 }
14 \map{%将没有设置的userb设置成en, 即认为不是中文的就是英文的。
15 \step[fieldset=userb,fieldvalue={en}]
16 }

```

4. 修改多语言参考文献间的分割符号, 即将 `par` 改为 `newline`, 避免采用 `gb7714-2015` 的项对齐方式时, 不同语言的参考文献间的分段导致没有缩进。(测试结果见:3.9节的项对齐方式)

```

1 %\renewcommand*{\entrysetpunct}{\addot\par\nobreak}
2 \renewcommand*{\entrysetpunct}{\addot\newline\nobreak}

```

5. Zeping Lee 兄发现了一个小问题, 感谢, 一直没有注意到这个问题。这里做出修改: 主要是作者年制中, 期刊析出的文献中, 当卷信息不存在时, 期刊名和期是连在一起的, 而不是中间有个逗号, 例如 `GB/T 7714-2015` 中第 10.2.4 节中的“刘彻东条目”。主要修改如下 (结果测试见:3.18节):

```

1 %调整期刊名的格式, 来源于standard.bbx
2 \renewbibmacro*{journal+issuetitle}{%
3 \usebibmacro{journal}%
4 %\setunit*{\addspace}%
5 %\setunit*{\addcomma\addspace}%修改为增加一个逗号
6 \iffieldundef{series}%
7 {}%
8 {\newunit%
9 \printfield{series}%
10 \setunit{\addspace}}%
11 %\usebibmacro{volume+number+eid}%
12 %\setunit{\addspace}%
13 \usebibmacro{issue+date}%
14 %\setunit{\addcolon\space}%
15 %换成逗号和空格
16 \usebibmacro{issue}%
17 \iffieldundef{volume}{\setunit{\addcomma\space}}%
18 \usebibmacro{volume+number+eid}%把卷期放到年份后面
19 %\newunit
20 }
21
22 %调整journal, 首先判断子标题, 然后在设置标点。避免直接设置标点, 当volume不存在是需要使用\nopunct去标
    点进而引入不必要的空格
23 \renewbibmacro*{journal}{%

```

```

24 \iffielddundef{journaltitle}%
25 {}%
26 {\printtext[journaltitle]{%
27 \printfield[titlecase]{journaltitle}%
28 \iffielddundef{journalsubtitle}{\setunit{\subtitlepunct}%
29 \printfield[titlecase]{journalsubtitle}}}}
30
31 %调整issue+date, 原在authoryear.BBX中\bbx@opt@mergedate@compact中定义
32 %当issue存在时, 才设置newunit。避免直接设置标点, 当volume不存在是需要使用\nopunct 去标点进而引入不必要的空格
33 \renewbibmacro*{issue+date}{%
34 \iffielddundef{issue}%
35 {}%
36 {\printtext[parens]{\printfield{issue}}\newunit}}
37
38 %增加一个number带括号的格式, 避免使用mkbibparens而引入一个不必要的空格
39 \DeclareFieldFormat{addnumflag}{%
40 \nobreak\printtext{(\}\nobreak #1\nobreak\printtext{)}}
41
42 %调整期刊卷和期的格式, 来自standard.bbx
43 \renewbibmacro*{volume+number+eid}{%
44 \printfield{volume}%
45 %\setunit*{\adddot}%去掉点号
46 %\printfield{number}%
47 \iffielddundef{number}{\printfield[addnumflag]{number}}%
48 %\iffielddundef{number}{\printtext{\mkbibparens{\printfield{number}}}}% 增加一个圆括号
49 \iffielddundef{eid}{%
50 \setunit{\addcomma\space}%
51 \printfield{eid}}

```

6. wayne508 同学提出了一个需求, 就是不希望使用出版项缺省时的默认处理, 即不使用 [出版地不详], [出版者不详], [S.l.], [s.n.] 等填充, 因此增加了一个宏包选项 gbpub, 当等于 false 时, 去掉自动处理, 使用 biblatex 的标准处理方式。增加选项代码见4.11节, 下面给出的是具体的处理代码:

```

1 %出版社和地址的处理
2 \newbibmacro*{location+institution+date}{%
3 \iftoggle{bbx:gbpub}%
4 {\testCJKfirst{\thefield{title}}
5 \iflistundef{location}{\iftoggle{ifCJKforgbt}{\printtext{[出版地不详]}\printtext{[S.l.\adddot]}}
6 {\printlist{location}}}%
7 % \iflistundef{institution}
8 % {\setunit*{\addcomma\space}}
9 % {\setunit*{\addcolon\space}}%
10 % \printlist{institution}%
11 % \setunit*{\addcomma\space}%
12 \addcolon\addspace%
13 \iflistundef{institution}{%
14 \iftoggle{ifCJKforgbt}{\printtext{[出版者不详]}\printtext{\mkbibbrackets{s.n.}}}% \bibstring{
    nopublisher}%[s.n.\adddot]

```

```

15 {\printlist{institution}}}%
16 \setunit{\addcomma\addspace}%
17   %\usebibmacro{date}%
18   \printfield{year}%
19   \bibrangedash%
20   \iffielddundef{endyear}{ }\{\printfield{endyear}}}%
21   \newunit}%
22 { \printlist{location}%
23   \iflistundef{institution}%
24     {\setunit*{\addcomma\space}}
25     {\setunit*{\addcolon\space}}}%
26   \printlist{institution}%
27   \setunit*{\addcomma\space}%
28   \usebibmacro{date}%
29   \newunit}%
30 }
31
32 \renewbibmacro*{publisher+location+date}{%
33 \iftoggle{bbx:gbpub}%
34 {\testCJKfirst{\thefield{title}}}%
35 %\testifnoteeqstd%
36 \iflistundef{location}{%\addot
37 \iffieldequalstr{note}{standard}{ }\{\%从gbt7714-2015标准低19页看到，标准存在出版项时输出，没有时完全省
   略。
38 %\iftoggle{ifnoteeqstandard}{ }\{\%}替换为上一句，尽可能用biblatex提供的函数
39 \iftoggle{ifCJKforgbt}{\printtext{[出版地不详]}\addcolon\addspace}{\printtext{[S.1.\addot]}\
   addcolon\addspace}%
40 }}% \bibstring{noaddress}
41 {\printlist{location}\addcolon\addspace}%
42 %\addcolon\addspace%
43 \iflistundef{publisher}{%
44 \iffieldequalstr{note}{standard}{ }\{\%
45 %\iftoggle{ifnoteeqstandard}{ }\{\%}替换为上一句，尽可能用biblatex提供的函数
46 \iftoggle{ifCJKforgbt}{\printtext{[出版者不详]}\setunit{\addot\addspace}\setunit*{\addcomma\
   addspace}}%
47 {\printtext{\mkbibbrackets{s.n.}}\setunit{\addot\addspace}\setunit*{\addcomma\addspace}}%
48 }}%
49 {\printlist{publisher}\setunit*{\addcomma\addspace}}%
50 %\addcomma\addspace%
51 \usebibmacro{date}%
52 %\newunit %去掉这个标点
53 }%
54 {\printlist{location}%
55   \iflistundef{publisher}
56     {\setunit*{\addcomma\space}}
57     {\setunit*{\addcolon\space}}}%
58   \printlist{publisher}%
59   \setunit*{\addcomma\space}%
60   \usebibmacro{date}%

```

```

61     %\newunit
62   }%
63 }

```

下面这种处理方式是有问题的，即默认处理情况不是希望的 true 的情况，但如果给出宏包加载选项时没有问题的。因此采用上一种方式。

```

1   %\newtoggle{bbx:gbpub}
2   %\DeclareBibliographyOption[boolean]{gbpub}[true]{%
3   % \settoggle{bbx:gbpub}{#1}}
4   %这种机制的标识判断，能用于usemacro使用，域格式定义中，但无法用来定义macro
5   %定义宏和使用宏是两个不同的展开层级
6   %\ExecuteBibliographyOptions{gbpub}
7   %但是使用这句默认设置可以用来定义macro
8   \DeclareBibliographyOption[bool]{gbpub}[true]{%应使用这种直接的机制
9   \ifstrequal{#1}{true}{\pubaddmacroredefine}{}}
10
11  %出版社和地址的处理
12  %新增一个样式用于输出连续出版物的地址，单位，时间
13  %类似\newbibmacro*{publisher+location+date}
14  \newbibmacro*{location+institution+date}{%
15    \printlist{location}%
16    \iflistundef{institution}
17      {\setunit*{\addcomma\space}}
18      {\setunit*{\addcolon\space}}%
19    \printlist{institution}%
20    \setunit*{\addcomma\space}%
21    \usebibmacro{date}%
22    \newunit}
23  %当没有出版社地址时，直接判断title的信息是否是中文，若为中文，则写出版地不详，否则用英文的字符表示。
24  %事实上title对于每个文献来说是必须的，所以用它判断是最快的，而且一般标题和出版社的语言是一样的。
25  \def\pubaddmacroredefine{%
26  \renewbibmacro*{publisher+location+date}{%
27  \testCJKfirst{\thefield{title}}}%
28  %\testifnoteeqstd%
29  \iflistundef{location}{%\adddot
30  \iffieldequalstr{note}{standard}}{%从gbt7714-2015标准低19页看到，标准存在出版项时输出，没有时完全省
    略。
31  %\iftoggle{ifnoteeqstandard}{%}%替换为上一句，尽可能用biblatex提供的函数
32  \iftoggle{ifCJKforgbt}{\printtext{[出版地不详]}\addcolon\addspace}{\printtext{[S.l.\adddot]}\
    addcolon\addspace}%
33  }% \bibstring{noaddress}
34  {\printlist{location}\addcolon\addspace}%
35  %\addcolon\addspace%
36  \iflistundef{publisher}{%
37  \iffieldequalstr{note}{standard}}{%
38  %\iftoggle{ifnoteeqstandard}{%}%替换为上一句，尽可能用biblatex提供的函数
39  \iftoggle{ifCJKforgbt}{\printtext{[出版者不详]}\setunit{\adddot\addspace}\setunit*{\addcomma\
    addspace}}%
40  {\printtext{\mkbibbrackets{s.n.}}\setunit{\adddot\addspace}\setunit*{\addcomma\addspace}}%
41  }%

```

```

42   {\printlist{publisher}\setunit*{\addcomma\addspace}}%
43   %\addcomma\addspace%
44   \usebibmacro{date}%
45   %\newunit %去掉这个标点
46   }
47   %新增一个样式用于输出连续出版物的地址，单位，时间
48   %类似与上面的\newbibmacro*{publisher+location+date}
49   \newbibmacro*{location+institution+date}{%
50   \testCJKfirst{\thefield{title}}
51   \iflistundef{location}{\iftoggle{ifCJKforgbt}{\printtext{[出版地不详]}\printtext{[S.l.\addot]}}
52   %
53   {\printlist{location}}}%
54   % \iflistundef{institution}
55   % {\setunit*{\addcomma\space}}
56   % {\setunit*{\addcolon\space}}%
57   % \printlist{institution}%
58   % \setunit*{\addcomma\space}%
59   \addcolon\addspace%
60   \iflistundef{institution}{%
61   \iftoggle{ifCJKforgbt}{\printtext{[出版者不详]}\printtext{\mkbibbrackets{s.n.}}}% \bibstring{
62   nopublisher}%[s.n.\addot]
63   {\printlist{institution}}}%
64   \setunit{\addcomma\addspace}%
65   %\usebibmacro{date}%
66   \printfield{year}%
67   \bibbrangedash%
68   \iffieldundef{endyear}{\printfield{endyear}}%
69   \newunit}
70   }

```

2017-02-26 update to version 1.0g

1. 进一步增加兼容性，支持条目类型比如 MASTERSTHESIS, PHDTHESIS, www, electronic, standard, techreport, conference 等，支持本样式增加的 newspaper 类型。因此在 bib 文件中可以直接使用这些条目类型。具体的测试详见3.17节。

为了实现兼容，主要从三个方面进行修改，包括用户层数据源映射，样式层的数据源映射，驱动。关于数据源映射和数据模型的原理详见4节。

因为 biblatex 提供的一些类型的别名的处理是在驱动层数据源映射时处理，所以要实现完全的兼容，还需要在用户层或者样式层进一步处理，首先是标识符的问题。因为以前做的标识符处理时在用户层映射中，所以这里仍然如此：

```

1   \map{
2     \pertype{newspaper}%增加一个新闻报纸的类型newspaper
3     \step[fieldset=usera, fieldvalue={N}]%因为没有专门的驱动，这句的目的是定义一个usera 域，方便映
4     射为article 后判断
5     \step[fieldset=note, fieldvalue=news]
6   }
7   \map{
8     \pertype{standard}%兼容老的standard类型

```



```

8      \step[fieldset=usera, fieldvalue={S}]%因为没有专门的驱动，这句的目的是定义一个usera 域，方便映
      射为book和inbook后判断
9      \step[fieldset=note, fieldvalue=standard]
10     }
11     \map{
12         \pertype{inproceedings}
13         \pertype{conference}%兼容老的conference类型
14         \step[fieldset=usera, fieldvalue={C}]
15         \step[fieldsource=institution] %有时会把version和edition 混淆，这里处理后就没有这个问题，可以
      直接用version
16         \step[fieldset=publisher, origfieldval]
17         \step[fieldsource=editor] %
18         \step[fieldset=bookauthor, origfieldval]
19     }
20     \map{
21         \pertype{report}
22         \pertype{techreport}%techreport类型
23         \step[fieldset=usera, fieldvalue={R}]
24     }
25     \map{
26         \pertype{thesis}
27         \pertype{mastersthesis}%兼容老的mastersthesis和phdthesis 类型
28         \pertype{phdthesis}
29         \step[fieldset=usera, fieldvalue={D}]
30     }
31     \map{
32         \pertype{online}
33         \pertype{electronic}%兼容老的electronic类型
34         \pertype{www}%兼容老的www类型
35         \step[fieldset=usera, fieldvalue={EB}]
36     }

```

其次，因为 biblatex 标准样式在处理条目别名是在驱动层的映射中，这里面引入了一些对于 gb7714 样式来说不需要的信息，比如 type 信息，因此需要将其去掉，所以在样式层映射中进行处理。因为 standard 条目可能用 book 也可能用 inbook 驱动输出，所以转换过程就需要有选择。这里有两种方式可以处理，一是用域是否存在进行判断（比如 booktitle 域），然后分别转换为 book 类型和 inbook 类型，二是直接都转换成 inbook 类型，然后对 inbook 驱动进行修改，因为 inbook 驱动与 book 驱动的差异仅在于所析出源文献那一块，所以，在驱动中用 booktitle 域进行判断，如果该域不存在，那么去掉这一块的处理，inbook 驱动可以等价于 book 驱动，但是这种方式中处理标识符后面的标点可能存在问题，biblatex 中处理标点的机制有很多好处，但是当样式作者在修改域格式是引入一些诸如 [] 之类符号时处理时比较麻烦的。这里采用第一种方式。样式层映射为：

```

1     \DeclareStyleSourcemap{
2         \maps[datatype=bibtex]{
3             % \map{%尝试未定义数据模型的newspaper类型映射为article，newspaper完全是针对gb7714的新类型，在biblatex
              中完全没有定义
4             % %但从实践看，并没有什么影响，映射过来就可以了。这一段可以用下面的驱动层别名映射替代，所以这里注释掉用
              下面的\DeclareBibliographyAlias命令
5             % \step[typesource=newspaper, typetarget=article, final]
6             % }
7             \map{%尝试未定义数据模型的standard类型映射为book，standard类型在blx-dm中有出现，但仅定义了类

```

```

    型，域和约束等都没有定义
8      \step[fieldsource=booktitle,final]%当存在booktitle域是映射为inbook
9      \step[typesource=standard, typetarget=inbook, final]
10     }
11     \map{%尝试未定义数据模型的standard类型映射为book, standard类型在blx-dm中有出现, 但仅定义了类
        型，域和约束等都没有定义
12     \step[typesource=standard, typetarget=book, final]%当不存在booktitle 域是映射为book
13     }
14     \map{%先于标准样式的driver层映射, 将其先映射过来, 并取消type设置
15     \step[typesource=mastersthesis, typetarget=thesis, final]
16     %\step[fieldset=type, fieldvalue=mathesis]
17     }
18     \map{%先于标准样式的driver层映射, 将其先映射过来, 并取消type设置
19     \step[typesource=phdthesis, typetarget=thesis, final]
20     %\step[fieldset=type, fieldvalue=phdthesis]
21     }
22     \map{%先于标准样式的driver层映射, 将其先映射过来, 并取消type设置
23     \step[typesource=techreport, typetarget=report, final]
24     %\step[fieldset=type, fieldvalue=techreport]
25     }
26   }
27 }
28 \DeclareBibliographyAlias{newspaper}{article}%定义驱动别名, 尝试以替代驱动层映射, 实践表明是可行的。

```

2. 在 online 类型中, 公告日期改为首选用 date 实现, 然后用 enddate, 当没有 date 和 enddate 时则用 eventdate 输出。代码详见4.6节。

3. 为方便 bib 文件生成, 构建可以从 gb7714-2015 格式的参考文献表文本转 bib 文件的 perl 程序, 利用它可以批量解析参考文献信息并转换为 bib 数据源文件。详见:gb7714texttobib.pl, 测试文件见:gb7714texteg.dat。

4. 在输出标识符的 usera 域格式中考虑标准样式的 url 选项, 以便实现对是否打印 url 和 urldate 的控制。这个需求是 Wenbo Sheng 兄提出的, 这里做出修改。

```

1      \DeclareFieldFormat{gbtypeflag}{%
2      \iftoggle{bbx:url}{\iffieldundef{url}%当存在url时, 增加一个OL标识符
3      {\nobreak\printtext{[]\nobreak#1\nobreak\printtext{[]}}%
4      {\nobreak\printtext{[]\nobreak#1\nobreak\printtext{\texttt{/}OL}}}%
5      }{\nobreak\printtext{[]\nobreak#1\nobreak\printtext{[]}}}}
6
7      \DeclareFieldFormat{gbtypeflagN}{%用于报纸newspaper
8      \iftoggle{bbx:url}{\iffieldundef{url}%当存在url时, 增加一个OL标识符
9      {\nobreak\printtext{[]\nobreak N\printtext{[]}\nobreak}%
10     {\nobreak\printtext{[]\nobreak N\printtext{\texttt{/}OL}}\nobreak}%
11     }{\nobreak\printtext{[]\nobreak N\printtext{[]}\nobreak}}
12
13     \DeclareFieldFormat{gbtypeflagS}{%用于标准standard
14     \iftoggle{bbx:url}{\iffieldundef{url}%当存在url时, 增加一个OL标识符
15     {\nobreak\printtext{[]\nobreak S\printtext{[]}\nobreak}%
16     {\nobreak\printtext{[]\nobreak S\printtext{\texttt{/}OL}}\nobreak}%
17     }{\nobreak\printtext{[]\nobreak S\printtext{[]}\nobreak}}

```

5. 在一些条目类型如 inbook 等的标识符后面 (如 [M]/) 加入一个不可断行短空格, 使紧跟其后的单词能

正确断行，当然也可以增加一个可断行的短空格 `addthinspace`，方便直接在`//`后面断行。

```
1 \usebibmacro{title}%
2 %\nopunct
3 %\iffielddundef{booktitle}{\adddot\addspace}{%兼容standard时，如果standard没有booktitle的应转换为book
   类，因为都转成inbook 类，所以这里做如下处理
4 \printtext{\texttt{//}\addnbthinspace}%\texttt{//}
5 \usebibmacro{bybookauthor}%
```

6. 对参考文献的一些域中存在的一些特殊字符比如 `&`，`%`，`#` 等进行处理，方法是利用动态数据修改。同时因为 `texlive2015/texlive2016` 中 `biblatex` 版本的不同分别进行处理。这个需求是湘厦人提出的，这里做出修改，详见4.4.1节。

2016-12-31 update to version 1.0f

1. 利用 `biblatex` 提供的 `iffieldequalstr` 函数替换用于判断 `note` 域值等于 `new` 或 `standard` 的函数，比如：

```
1 %case 1:
2 %\providetoggle{ifnoteeqstandard}%判断是否是标准
3 %\def\testifnoteeqstd{%
4 %\iffielddundef{note}{\togglefalse{ifnoteeqstandard}}%判断一下，是否是标准
5 % {\def\comparetmp{standard}\edef\comparetmpb{\thefield{note}}}%
6 % \ifx\comparetmp\comparetmpb%
7 % \toggletrue{ifnoteeqstandard}%
8 % \else%
9 % \togglefalse{ifnoteeqstandard}%
10 % \fi}}
11 %\iffieldequalstr{note}{standard}{true}{false} %可以利用这一函数代替
12
13 %case 2:
14 \iffieldequalstr{note}{standard}{\printfield[gbtypeflags]{usera}}%判断是否为标准
15 {\iffieldequalstr{note}{news}{\printfield[gbtypeflag]{usera}}}% 判
   断是否为报纸
16 {\printfield[gbtypeflag]{usera}}}% 其它
17 % \def\comparetmp{news}\def\comparetmpa{standard}\edef\comparetmpb{\thefield{note}}%
18 % \ifx\comparetmp\comparetmpb%判断是否为报纸
19 % \printfield[gbtypeflag]{usera}%
20 % \else%
21 % \ifx\comparetmpa\comparetmpb%判断是否为标准
22 % \printfield[gbtypeflags]{usera}%
23 % \else%
24 % \printfield[gbtypeflag]{usera}%
25 % \fi%
26 % \fi%
27
28 %case 3:
29 \iffieldequalstr{note}{news}{\usebibmacro{newsdate}}%判断是否为报纸
30 {\usebibmacro{date}}%
31 % \def\comparetmp{news}\edef\comparetmpb{\thefield{note}}%
32 % \ifx\comparetmp\comparetmpb%
33 % \usebibmacro{newsdate}%
```

```

34 % \else%
35 % \usebibmacro{date}%
36 % \fi
37
38 %case 4:
39 \renewbibmacro*{publisher+location+date}{%
40 \testCJKfirst{\thefield{title}}%
41 %\testifnoteeqstd%
42 \iflistundef{location}{%\adddot
43 \iffieldequalstr{note}{standard}{}%从gbt7714-2015标准低19页看到，标准存在出版项时输出，没有时完全省略。
44 %\iftoggle{ifnoteeqstandard}{}%替换为上一句，尽可能用biblatex提供的函数
45 \iftoggle{ifCJKforgbt}{\printtext{[出版地不详]}\addcolon\addspace}{\printtext{[S.1.\adddot]}\addcolon\addspace}}% \bibstring{noaddress}
46 {\printlist{location}\addcolon\addspace}%
47 %\addcolon\addspace%
48 \iflistundef{publisher}{%
49 \iffieldequalstr{note}{standard}{}%
50 %\iftoggle{ifnoteeqstandard}{}%替换为上一句，尽可能用biblatex提供的函数
51 \iftoggle{ifCJKforgbt}{\printtext{[出版者不详]}\setunit{\adddot\addspace}\setunit*{\addcomma\addspace}}%
52 {\printtext{\mkbibbrackets{s.n.}}\setunit{\adddot\addspace}\setunit*{\addcomma\addspace}}}%
53 {\printlist{publisher}\setunit*{\addcomma\addspace}}%
54 %\addcomma\addspace%
55 \usebibmacro{date}%
56 %\newunit %去掉这个标点
57 }

```

2. 之前 1.0e 版增加 gbalgn 选项的时候，没有测试对 texlive2015 的兼容性，所以导致一些错误。因为 texlive2015 的 biblatex3.0 版本的 DeclareBibliographyOption 命令定义选项时不像 texlive2016 的 biblatex3.4 版的是带类型说明的。所以做出一定的处理，把该命令分两个版本进行设置。同时需要注意新定义的参考文献表环境在 texlive2015 中的 biblatex3.0 中无效且出错，所以直接去掉，因此文献表的标签的项对齐效果在 texlive2015 中的 biblatex3.0 版中无法实现。代码详见4.11节。

3. 之前 1.0e 版解决编组符号包围的责任者的中英文判断问题的时候，没有测试对 texlive2015 的兼容性，所以导致一些错误。因为使用了 xstring 宏包的功能，但 texlive2015 的 biblatex3.0 版本不默认加载 xstring 宏包，所以在修改样式文件，在其中加载一下该宏包。

```

1 \RequirePackage{xstring}%为兼容texlive2015的biblatex3.0不加载xstring包的问题

```

2016-12-07 update to version 1.0e

1. 应海阔天空和 xmtangjun 等朋友的要求，在同一文献中可以使用上标或非上标的标注方式，修改顺序编码制的标注样式文件，去掉 parencite 命令的上标模式，恢复非上标方式。这样可以在同一篇文章中使用 cite 命令标注上标，而 parencite 命令标注非上标。而作者年制没有这一问题，不做修改。具体的效果见第3.14节的内容。

2. 给宏包增加了一个选项 gbalgn，用于控制顺序编码制的参考文献表的标签对齐方式，默认是 right 即右对齐，可以设置 left 即左对齐，也可以设置 gb7714-2015，即以各条参考文献自身为基准对齐实现对齐。效果见第3.9节的内容。增加一个选项，真正实现起来并不复杂，但在未明白其运行机制之前尝试了好长时间，显得很麻烦，具体的机制见第4节。修改代码如下：

```

1 \DeclareBibliographyOption[string]{gbalign}[right]{%

```

```

2 \ifstrequal{#1}{left}{\setalignleft}{}
3 \ifstrequal{#1}{gb7714-2015}{\setaligngbstyle}{}
4 }
5
6 %修改序号标签格式为左对齐
7 \def\setalignleft{\DeclareFieldFormat{labelnumberwidth}{\mkbibbrackets{##1}\hfill}}
8 %修改序号标签格式为以各条参考文献为基础进行对齐的方式
9 \def\setaligngbstyle{%
10 \def\blx@bibitem##1{%
11   \blx@ifdata{##1}
12   {\beginingroup
13     \blx@getdata{##1}%
14     \blx@bibcheck
15     \iftoggle{blx@skipentry}{}{%
16       \blx@setdefaultrefcontext{##1}%
17       \global\let\blx@noitem\@empty
18       \blx@setoptions@type\abx@field@entrytype
19       \blx@setoptions@entry
20       \blx@thelabelnumber
21       \addtocounter{instcount}\@ne
22       \blx@initsep
23       \blx@namesep
24       \csuse{blx@item@blx@theenv}\relax
25 % \blx@initsep %移动到上面去, 恢复bibnamesep等的作用机制
26 % \blx@namesep
27       \csuse{blx@hook@bibitem}%
28       \blx@execute
29       \blx@initunit
30       \blx@anchor
31       \blx@beglangbib
32       \bibsentence
33       \blx@pagetracker
34       \blx@driver\abx@field@entrytype
35       \blx@postpunct
36       \blx@endlangbib}%
37   \par\endgroup}%这里增加了一个\par
38   {}}
39 \newcommand{\itemcmd}{%
40 \settowidth{\lengthid}{[\printfield{labelnumber}]}
41 \addtolength{\lengthid}{\biblabelsep}
42 \setlength{\lengthlw}{\textwidth}
43 \addtolength{\lengthlw}{-\lengthid}
44 \addvspace{\bibitemsep}%恢复\bibitemsep的作用
45 %\parshape 2 0em \textwidth \lengthid \lengthlw
46 \hangindent\lengthid
47 [\printfield{labelnumber}]\hspace{\biblabelsep}}
48 \newlength{\lengthid}
49 \newlength{\lengthlw}
50 \defbibenvironment{bibliography}

```

```

51 {\begingroup\setlength{\parindent}{0em}}
52 {\endgroup}
53 {\itemcmd}}

```

3. map 中当有 append 选项时也需要 overwrite 选项, 这不知道是不是 texlive 2016 中 biber 升级后的原因。之前使用 texlive2015 的时候没有问题。所以修改为:

```

1  %*.bbx
2      \map{%将entrykey放入keywords中用于后期的使用
3          \step[fieldsource=entrykey]
4          \step[fieldset=keywords, origfieldval]
5      }
6      \map[overwrite]{%这里还必须有overwrite, 怎么之前会觉得没有问题呢, 可能是之前版本拷错了, 还是之前是
7          texlive2015 变16后biber有了变化(2016-1207修改正确)
8          \step[fieldsource=note, final]%将note域信息复制给keywords, 用于输出时容易区分标准和报纸
9          \step[fieldset=keywords, fieldvalue={}, append]
10         \step[fieldset=keywords, origfieldval, append]
11     }

```

4. 顺序制中, 出版项后没有日期的情况下, 出现逗号这是有问题的, 所以修改为:

```

1  %gb7714-2015.bbx
2  %出版社和地址的处理, 当没有出版社地址时
3  %直接判断title的信息是否是中文, 若为中文, 则写出版地不详, 否则用英文的字符表示。
4  %事实上title对于每个文献来说是必须的, 所以用它判断是最快的, 而且一般标题和出版社的语言是一样的。
5  \newbibmacro*{publisher+location+date}{%
6      \testCJKfirst{\thefield{title}}%
7      \testifnoteeqstd%
8      \iflistundef{location}{%\adddot
9          \iftoggle{ifnoteeqstandard}}{%
10         \iftoggle{ifCJKforgbt}{\printtext{[出版地不详]}\addcolon\addspace}{\printtext{[S.1.\adddot]}\addcolon\addspace}}% \bibstring{noaddress}
11     {\printlist{location}\addcolon\addspace}%
12     %\addcolon\addspace%
13     \iflistundef{publisher}{%
14         \iftoggle{ifnoteeqstandard}}{%
15         \iftoggle{ifCJKforgbt}{\printtext{[出版者不详]}\setunit{\addot\addspace}\setunit*{\addcomma\addspace}}%
16         {\printtext{\mkbibbrackets{s.n.}}\setunit{\addot\addspace}\setunit*{\addcomma\addspace}}}%
17     {\printlist{publisher}\setunit*{\addcomma\addspace}}%在这里修改使用了setunit命令, 与ay.bbx统一。
18     %\addcomma\addspace%
19     \usebibmacro{date}%
20     %\newunit %去掉这个标点
21 }

```

5. 当 urldate 域给出的信息不全时, 比如只有年和月, 而没有日, 那么就需要进行判断, 只输出存在的信息, 因此对 urldate 域格式做修改, 代码详见4.6节。

6. 当责任者等需要判断中英文的信息是用编组符号包含的时候, 原来的 CJK 判断函数会出现问题, 所以利用 xstring 宏包做一定的修改, 修改完成后可以应对信息中存在编组的情况, 详见4.2节。

1. 用于 usera 域的 gbtypeflag 域打印格式, 明明在 aritle/book 类中没有问题, 但在 beamer 中就会出现问
题, 多出一个点了。到现在还没有搞明白怎么会多出点来, printtext 命令明明没有输出点, 不像 S.l. 还有一个点
的输出, 这里只有] 符号, 但就是多了一个点。从最后修改成功看, 这里就是多了一个点, 而且是 literal period,
所以后面的点无法覆盖它, 所以需要先用 adddot 命令将其转换为缩写的点, 而且似乎用 isdot 也不行, 其原因
还得再分析分析。因此做如下修改:

```

1 %域格式修改
2 \DeclareFieldFormat{title}{#1\adddot\addthinspace}%
3 \DeclareFieldFormat{journaltitle}{#1\isdot}
4 \DeclareFieldFormat%将期刊等文献的标题中原来带的引号去掉
5 [article,patent,thesis,unpublished]
6 {title}{#1\adddot\addthinspace}
7 \DeclareFieldFormat%将期刊等文献的标题中原来带的引号去掉
8 [inbook,incollection,inproceedings]
9 {title}{#1\nopunct\unspace}
10
11 %同时移动title的micro的printtext[title]的括号范围:
12 %重设title的输出, 将文献类型标识符输出出去, 原输出来自biblatex.def文件
13 \newbibmacro*{title}{%
14   \ifboolexpr{%
15     test{\iffieldundef{title}}%
16     and%
17     test{\iffieldundef{subtitle}}%
18   }%
19   {%
20     {\printtext[title]{%
21       \printfield[titlecase]{title}%
22       \ifboolexpr{test {\iffieldundef{subtitle}}}%这里增加了对子标题的判断, 解决不判断多一个点的问题
23       {}{\setunit{\subtitlepunct}%
24         \printfield[titlecase]{subtitle}}%
25       \iffieldundef{titleaddon}{}%判断一下titleaddon, 否则直接加可能多一个空格
26       {\setunit{\subtitlepunct}\printfield{titleaddon}}%
27       \iffieldundef{note}{\printfield[gbtypeflag]{usera}}%在标题后直接给出文献标识字母, 判断一下, 是否
           是报纸和标准
28       {\def\comparetmp{news}\def\comparetmpa{standard}\edef\comparetmpb{\thefield{note}}%
29       \ifx\comparetmp\comparetmpb%判断是否为报纸
30       \printfield[gbtypeflag]{usera}%
31       \else%
32         \ifx\comparetmpa\comparetmpb%判断是否为标准
33         \printfield[gbtypeflags]{usera}%
34         \else%
35         \printfield[gbtypeflag]{usera}%
36         \fi%
37       \fi%
38     }}%
39     \iffieldundef{booktitle}{\newunit}{}%当title是析出时, 不要标点
40     %\newunit
41   }%将\printtext[title]的结束编组放到这里来。
42 }%
```

43 }

还需要注意的是如果 `gbtypeflag` 域格式中不直接输出 `[]`，而用 `mkbibbrackets` 也能解决这些问题，但是会因为 `ctex` 对于中英文间空格的默认处理加入空格，所以只能采用上面的方式。

还有 `beamer` 类中很多不同域之间的空格似乎比其它类中更宽，不知道原因，难道是 `beamer` 重新定义了 `\space` 命令？

2. 在参考文献表中加入逐字文本 (原样文本, 如实文本), 也就是直接插入文本信息, 或者用 `printtext` 插入都会导致一些问题, 上面的第 1 点就是典型问题之一, 还比如出版项缺省等问题。在有利用 `printtext` 插入原样文本的时候, 要特别注意在 `driver` 中该命令前后几行的代码后加注释, 否则容易带入空格, 注释后就可以消除。

3. 同样的 `periodical` 条目类型的 `title` 输出也修改了 `printtext[title]` 的结束编组位置。 `journaltitle` 域格式也加了 `isdot`。 `patent` 的 `title` 也修改了 `printtext[title]` 的结束编组位置。

4. 修改了 `location+institution+date` 的 `s.n.` 的处理方式与 `publisher+location+date` 的方式类似。中英文判断也往外放到一层, 与 `publisher+location+date` 一致, 这样就不会出现不判断的问题。

5. 3.3 版以后的 `family-given` 格式的 `given name` 用全大写代替首字母大写。

2016-11-14 update

1. 很早之前思考的利用 `biber` 的动态修改数据功能来进行佚名问题处理是合理的, 因为 `biblatex` 不能在 `tex` 处理过程中添加域的信息, 所以任何要进入域的信息都需要在运行 `biber` 命令之时或者之前处理。利用正则表达式可以完成一定的区分, 尽管可能有一些特殊情况无法涵盖, 但如下的处理可以基本正确的实现功能。如果不需要用佚名的方式, 那么只要把这段处理注释掉就可以了。

```
1 \map{%因为要做佚名处理, 所以这里先对有些只有编者而没有作者的情况进行处理
2     \step[fieldsource=editor]
3     \step[fieldset=author,origfieldval]
4 }
5 \map{%默认情况下将空缺作者设置为佚名
6     \step[fieldset=author, fieldvalue={佚名}]
7 }
8 \map[overwrite]{%当标题带有英文字符, 且作者为佚名的情况, 设置作者为noauthor,
9     %也有一些特殊情况可能处理不到, 比如标题中英混合, 作者又却是
10    \step[fieldsource=title,match=\regexp{[a-zA-Z]},final]
11    \step[fieldsource=author,match=\regexp{佚名}, replace={NOAUTHOR}]
12 }
```

2. 关于文种分集排序的问题, 之前要求用户自己往 `userb` 域填信息, 现在通过如下处理, 可以避免, 也是用的正则表达式判断, 但有些特殊情况可能会有问题, 出现问题的话, 手动在 `bib` 源文件中添加 `userb` 域信息是可以解决的。到这里为止, 在使用本样式文件时, 除了必须要输入的引文的信息外, 其它信息都不需要再输入了, 包括原来就已经处理的 `usera` 域 (用于添加文献类型标识符的), 这里的 `userb` 域用于文种分集排序的, 都不必输入了。

```
1 \map{%将entrykey放入keywords中用于后期的使用
2     \step[fieldsource=entrykey]
3     \step[fieldset=keywords, origfieldval]
4     \step[fieldsource=note,final]%将note域信息复制给keywords, 用于输出时容易区分标准和报纸
5     \step[fieldset=keywords, fieldvalue={},append]
6     \step[fieldset=keywords, origfieldval,append]
7 }
8 \map{%因为无法进行CJK字符判断, 所以用反的思路, 判断没有英文字符, 没有空格, 没有逗号等字符情况下
```

```

9      %认为是中文的，将文献userb设置成cn，用于排序
10     \step[fieldsource=author,match=\regexp{[^\a-zA-Z\s\.\,\'\{\}\]},final]
11     \step[fieldset=userb,fieldvalue={cn}]
12   }
13 \map{%将没有设置的userb设置成en，即认为不是中文的就是英文的。
14     \step[fieldset=userb,fieldvalue={en}]
15   }

```

3. 增加了一个 yearpagescite 命令用于处理：作者年制文中已有作者只需要年份和页码的情况，而顺序制的情况下该命令与 pagescite 命令作用相同。其使用方式如下：

```

1  %使用方式：
2  见赵耀东\yearpagescite[] [205]{赵耀东1998--}和Simon\yearpagescite[] [15]{Simon2001--}的文献。
3
4  %增加的命令：
5  %*2015ay.cbx中：
6  %增加一个命令yearpagescite用于当文中作者已经存在，需要页码和年份的情况。
7  %参考来源，biblatex.DEF,其中\DeclareCiteCommand*{\citeyear}命令
8  \DeclareCiteCommand{\yearpagescite}
9    {\printtext{()}\usebibmacro{prenote}}
10   {\printfield{year}\printfield{extrayear}}
11   {\multicitedelim}
12   {\printtext{()}\textsuperscript{\usebibmacro{postpages}}}}
13 %*2015.cbx中：
14 %增加一个命令yearpagescite为与ay样式兼容，命令等同于\pagescite
15 \DeclareCiteCommand{\yearpagescite}[\mkbibsuperscript]{\mkbibbrackets,仍然用上标
16   {\usebibmacro{cite:init}}%
17   \usebibmacro{prenote}%
18   }
19   {\usebibmacro{cite:index}}%
20   \usebibmacro{cite:comp}}
21   {}
22   {\usebibmacro{cite:dump}}%
23   \usebibmacro{postpages}}

```

4. 在出版者缺省的情况下，当出版者后面没有更多信息的情况下，缺省字符串后面应该有一个点，因此做出修正为：

```

1  %出版社和地址的处理，当没有出版社地址时
2  %直接判断title的信息是否是中文，若为中文，则写出版地不详，否则用英文的字符表示。
3  %事实上title对于每个文献来说是必须的，所以用它判断是最快的，而且一般标题和出版社的语言是一样的。
4  \newbibmacro*{publisher+location+date}{%
5  \iffieldundef{title}{\testCJKfirst{\thefield{title}}}%
6  \testifnoteeqstd%
7  \iflistundef{location}{%\adddot
8  \iftoggle{ifnoteeqstandard}{%
9  \iftoggle{ifCJKforgbt}{\printtext{[出版地不详]}\setunit*{\addcolon\addspace}}{\printtext{[S.1.\adddot
10     ]}\setunit*{\addcolon\addspace}}}% \bibstring{noaddress}
11  \printlist{location}\setunit*{\addcolon\addspace}}%
12  %\addcolon\addspace%
13  \iflistundef{publisher}{%

```

```

13 \iftoggle{ifnoteeqstandard}{\{%
14 \iftoggle{ifCJKforgbt}{\printtext{[出版者不详]}\setunit{\addot\addspace}\setunit*{\addcolon\addspace
    }}
15 \%mkbibbrackets
16 \%{\printtext{[s.n.\addot]}\setunit{\addot\addspace}\setunit*{\addcolon\addspace}}}\%bibstring{
    nopublisher}
17 {\printtext{\mkbibbrackets{s.n.}}\setunit{\addot\addspace}\setunit*{\addcolon\addspace}}}}
18 {\printlist{publisher}\setunit*{\addcolon\addspace}}}%
19 \%addcomma\addspace%
20 \%usebibmacro{date}%
21 \%newunit %去掉这个标点
22 }

```

5. 反向链接, backref 的格式并没有要求, 但考虑到中文环境还是将其格式改一下, 因此修改英文本地化字符串为“引用页”。

```

1 \DefineBibliographyStrings{english}{
2 bibliography = {参考文献},
3 references = {参考文献},
4 %bytranslator= {\addcomma\ 译\addot}, %将trans. by 改成 译
5 bytranslator= {\addcomma\ 译},%\addperiod
6 and = {\addcomma},%将第2和3人名见的and符号改成 逗号, 用\finalnamedelim命令也可以定义, 参见3.9.1节
7 %andothers = {et al.}, %将超过3个人名的省略, et al.改成为 等
8 andotherscn = {等}, %将超过3个人名的省略, et al.改成为 等
9 noaddress = {[S.l.]},
10 nopublisher = {[s.n.]},
11 backrefpage = {引用页:},
12 backrefpages = {引用页:},
13 }

```

6. 在处理姓名相关的问题时, 利用 DeclareNameFormat 的方式控制需要的姓和名的前后顺序, 当 maxbibnames 和 maxcitenames 不一致时, 可能用到 last-first/first-last(biblatex 3.2 以前的版本)/family-given/given-family(3.3 以后版本), 其中第一个姓名和后面姓名的姓和名的前后顺序是不同的。可以直接利用其中的 name:first-last 和 name:last-first 或 name:family-given 和 name:given-family 宏做修改控制具体姓名成分的格式, 而避免重定义 DeclareNameFormat 格式, 详见 4.7.1 节。

7. 作者年制区分文献表和引用中的作者名数量, 引用相关的选项设置需要放到 cbx 文件中, 否则可能失效。同时因为一些特殊情况下, 姓名数量截短为 1 个的引用标签, 可能无法区分文献, 所以默认情况下, biblatex 会增加作者数量用于区分, 这是因为 uniquelist 会自动重设 maxcitenames 和 mincitenames, 因此修改 uniquelist 选项为 minyear, 明确在年份也一样的情况下再利用增加姓名进行区分。*ay.bbx 文件中的选项设置为:

```

1 \ExecuteBibliographyOptions{
2 useprefix = true, %名字的信息包括前缀
3 %firstinits = true, %名字有缩写, 参考3.1.2.3 Internal
4 giveninits = true,
5 date = year, %日期仅写到年
6 %maxnames = 3, %设置名字最大数量
7 %minnames = 3, %设置缩减后的名字最小数量
8 maxbibnames=3, %将文献列表和引用中最大名字数量区分开
9 minbibnames=3,
10 }

```

```

11 %bibtex3.3版前(比如texlive2015中的3.0版)的使用方式
12 \ExecuteBibliographyOptions{
13   useprefix = true, %名字的信息包括前缀
14   firstinits = true, %名字有缩写, 参考3.1.2.3 Internal
15   date = year, %日期仅写到年
16   %maxnames = 3, %设置名字最大数量
17   %minnames = 3, %设置缩减后的名字最小数量
18   maxbibnames=3, %将文献列表和引用中最大名字数量区分开
19   minbibnames=3,
20 }

```

*ay.cbx 文件中的选项设置为:

```

1 \ExecuteBibliographyOptions{
2   %autocite = superscript,
3   %autopunct = true,
4   %sorting = none,
5   maxcitenames=1,
6   mincitenames=1,
7   uniqueness=init,%因为使用了名字缩写选项, 所以需要设置uniqueness=init 而不是full 避免冲突
8   labeldate=true,
9   uniqueness=minyear,
10 }

```

2016-11-11 update

1. 说明文档增加了版本和修改时间信息, 修正了一些错误和不妥的说法, 增加了一些说明比如报纸版次, 报告条目域格式等, 去掉一些不必要的注释, 简化各样式文件内容。

2. 由 Harry Chen 提议, 将 english 本地化文件中的参考文献标题信息改为中文的, 因为本样式多在中文环境下使用, 修改为中文后, printbibliography 命令中不提供 title 信息的情况下, 参考文献列表标题默认为参考文献。感谢 Harry Chen 在 github 上的 commit! 修改如下:

```

1 \DefineBibliographyStrings{english}{
2   bibliography = {参考文献},
3   references = {参考文献},
4   %bytranslator= {\addcomma\ 译\adddot}, %将trans. by 改成 译
5   bytranslator= {\addcomma\ 译},%\addperiod
6   and = {\addcomma},%将第2和3人名见的and符号改成 逗号, 用\finalnamedelim命令也可以定义, 参见3.9.1节
7   %andothers = {et al.}, %将超过3个人名的省略, et al.改成为 等
8   andotherscn = {等}, %将超过3个人名的省略, et al.改成为 等
9   noaddress = {[S.l.]},
10  nopublisher = {[s.n.]}
11 }

```

3. 当作者名只有一个, 但又有 and others 表示多个作者的时候, 标准样式中作者名和 et al. 之间是空格而不是逗号链接, 但 gb7714-2015 要求在等之前用逗号, 所以做出修改如下。

```

1 %判断作者或译者是否中文, 若中文用字符等, 否则用etcl.
2 \newbibmacro*{name:andothers}{%
3   \ifboolexpr{
4     test {\ifnumequal{\value{listcount}}{\value{liststop}}}

```

```

5     and
6     test \ifmorenames
7 }%
8 {%这里做一个判断是在处理author还是translator用于两者是不同语言的情况
9 \ifcurrentname{translator}{\testCJKfirst{\thefield{usere}}}{\testCJKfirst{\thefield{userf}}}%
10 %这句判断如果放到\andothersdelim后面会在等或etc.前增加一个空格，所以放前面
11 \ifnumgreater{\value{liststop}}{1}%
12     {\finalandcomma}%
13     {\finalandcomma}%biblatex作者要区别单作者加等的情况，这里为符合gbt7714-2015第7.2节的要求加上了逗
    号。
14 \andothersdelim\iftoggle{ifCJKforgbt}{\bibstring{andotherscn}}{\bibstring{andothers}}%
15 %\andothersdelim\bibstring{andotherscn}
16 }%
17 {}

```

4. 给 report 和 manual 驱动添加了译者域，这在实际中是用的到的，同时打印 version 域的格式也做了处理，并且修改中文判断函数，增加了注释符以避免带入空格，这个问题在之前体现为版本域前多了一个空格。

```

1 %对version的版本信息做出修改，源来自biblatex.DEF
2 \DeclareFieldFormat{version}{%
3 \testCJKfirst{\thefield{title}}%
4 \ifinteger{#1}%
5 {\iftoggle{ifCJKforgbt}{\printtext{#1版}}}%
6 {\mkbibordedition{#1}~\bibstring{version}}}%
7 {#1\isdot}}

```

5. 把作者年制的参考文献列表和引用中的作者名数量做区分。列表中最大为 3 个，引用中最大为 1 个。(这里还有点问题，进一步修改见 2016-11-14 的更新。)

```

1 %%biblatex3.3版后(比如texlive2016中的3.4版)的使用方式
2 %\ExecuteBibliographyOptions{
3 % useprefix = true, %名字的信息包括前缀
4 % %firstinits = true, %名字有缩写，参考3.1.2.3 Internal
5 % giveninits = true,
6 % date = year, %日期仅写到年
7 % %maxnames = 3 , %设置名字最大数量
8 % %minnames = 3, %设置缩减后的名字最小数量
9 % maxbibnames=3, %将文献列表和引用中最大名字数量区分开
10 % minbibnames=3,
11 % maxcitenames=1,
12 % mincitenames=1
13 % %uniquename = init
14 %}
15 %
16 %%biblatex3.3版前(比如texlive2015中的3.0版)的使用方式
17 %\ExecuteBibliographyOptions{
18 % useprefix = true, %名字的信息包括前缀
19 % firstinits = true, %名字有缩写，参考3.1.2.3 Internal
20 % date = year, %日期仅写到年
21 % %maxnames = 3 , %设置名字最大数量
22 % %minnames = 3, %设置缩减后的名字最小数量

```



```

23 % maxbibnames=3, %将文献列表和引用中最大名字数量区分开
24 % minbibnames=3,
25 % maxcitenames=1,
26 % mincitenames=1
27 % %uniquename = init
28 %}

```

2016-10-22 update

1. 修改版本判断机制，版本 3.3 以后的版本设置判断标签 `iftexlivesix` 为真，采用新的姓名处理机制。修改如下：

```

1 \providetoggle{iftexlivesix}
2 %\def\versionstr{3.4}
3 %\def\versionstra{3.6}
4 %\ifx\abx@version\versionstr
5 %\toggletrue{iftexlivesix}
6 %\else
7 %\ifx\abx@version\versionstra
8 %\toggletrue{iftexlivesix}
9 %\else
10 %\togglefalse{iftexlivesix}
11 %\fi
12 %\fi
13 %改变版本判断机制，根据biblatex更新历史可知，版本3.3开始使用新的姓名处理机制
14 %所以当版本大于3.2开始，就用设置\toggletrue{iftexlivesix}
15 \def\numparserta#1.#2\relax{#1}%注意relax的重要性
16 \def\numparsertb#1.#2\relax{#2}
17 \def\numinteger{\expandafter\numparserta\abx@version\relax}
18 \def\numdigital{\expandafter\numparsertb\abx@version\relax}
19 \ifnumcomp{\numdigital}{>}{2}{\toggletrue{iftexlivesix}}{\togglefalse{iftexlivesix}}

```

2016-10-11 update

1. 真的是需求推动事物发展，秋平同学提出需要把顺序编码制的参考文献序号标签设为左对齐。于是可以做如下修改。需要用的可以把下面这段加进 `gb7714-2015.bbx` 中，不需要的就不用任何处理，左对齐还是右对齐其实还是看个人喜好，我其实觉得右对齐挺好的。

```

1 %修改序号标签格式为左对齐,注意各参考文献内容还是对齐的,
2 %这样就会使得序号标签与参考文献内容的间隔增大，这个问题是没有办法解决的
3 %因为采用list做具有一定宽度的序号标签，\labelwidth只能设置一个，且是最宽的标签的宽度
4 %但总的来说参考文献内容对齐是合理和漂亮的，
5 %而标签则只能对齐一个方向，要么左对齐要么右对齐，看个人选择了。
6 %\DeclareFieldFormat{shorthandwidth}{\mkbibbrackets{#1}} %来自numeric.BBX
7 \DeclareFieldFormat{labelnumberwidth}{\mkbibbrackets{#1}\hfill}

```

2. 测试了老电脑装的 `texlive2014`，没有问题通过。

2016-10-04 update

1. 今天广州的秋平同学使用更新后的 biblatex3.6 版出错。是因为 bbx 文件中的版本判断只有 3.4 和其它，所以应急加了一段对于 3.6 的判断。这个问题以后可能还会出现因为 biblatex 会不断的更新，所以需要设计一个更合理的判断，这个等实现以后再更新。

2. 在说明文档中增加了一些说明，修改了一些错别字。

2016-07-20 update

1. 去掉 texlive2016 和 texlive2015 选项，直接根据 biblatex 宏包的版本进行判断。
 2. 增加了 unpublished 条目类型驱动，并按报告 report 进行处理，但文献标识码用 Z 表示。
-

2016-07-01 update

1. 增加了 pagescite 命令，实现 GB/T7714-2015 对于引用标注中输出页码的特殊格式要求。
 2. 测试了 texlive2015, texlive2016, 发现其中关于名字域格式的差异，并作出修改。增加了两个宏包选项，一个是 texlive2016, 另一个是 texlive2015。使用 texlive2016 版本时，带选项 texlive2016 即可，其它情况带选项 texlive2015
-

2016-06-20 update

1. 利用判断 CJK 字符的函数，判断条目中著者，译者域是否是 CJK 字符，做相应的处理。
 2. 利用范围解析函数，可对卷期等进行解析，并按 GB/T7714-2015 要求输出。
-

2016-05-20 update

基本完成样式文件，实现的功能包括：

1. 实现 GB/T7714-2015 要求的参考文献著录格式。
2. 利用 map 功能使录入参考文献数据时不需要文献类别标识符。
3. 多语言文献的处理方法和条目格式。