

# The GOST package

Igor A. Kotelnikov\*

2017/01/12, v.1.2i

## Abstract

GOST is a bundle of BibTeX styles designed to meet State Standards (GOST) on information, librarianship and publishing issued by The Russian Federation and Interstate Committee of former USSR States.

It comprises 16 BibTeX styles to format bibliography in English, Russian and Ukrainian according to GOST 7.0.5-2008 and GOST 7.1-2003. Both 8-bit and Unicode (UTF-8) versions of each BibTeX style, in each case offering a choice of sorted and unsorted. Further, 2 more styles, `gost780` and `gost780s` styles (not conforming to effective standards) are retained for backwards compatibility.

## 1 Introduction

The package was initially developed by Maksym Polyakov. It was later updated by Igor Kotelnikov to the present status and some code was borrowed from `disser` package developed by Stanislav Kruchinin and unpublished work by Artem Petrenkov.

Nowdays, GOST is a bundle of BibTeX styles designed to meet State Standards (GOST) on information, librarianship and publishing issued by Russian Federation and interstate committee of former USSR States.

The System of Standards includes:

**GOST 7.0.5-2008** Bibliographic reference. General requirements and rules of making.

**GOST 7.1 -2003** Bibliographic record. Bibliographic description. General requirements and rules.

**GOST 7.80 -2000** Bibliographic record. Heading. General requirements and rules.

**GOST 7.11 -2004** Bibliographic description and references. Rules for the abbreviation of words and word combinations in foreign European languages.

**GOST 7.83 -2001** Electronic editions. Basic types and imprint.

ect.

Currently, GOST contains 16 BibTeX styles to format bibliography in English, Russian and Ukrainian according to GOST 7.0.5-2008 and GOST 7.1-2003. Both 8-bit and Unicode (UTF-8)

---

\*kia999@mail.ru

versions of each BibTeX style, in each case offering a choice of sorted and unsorted. Further, 2 styles, `gost780` and `gost780s` styles (which do not conform to effective standards) are retained for backwards compatibility.

All styles in the GOST bundle are derived from single master file `gost.dtx` by applying different set of options as shown in the table below.

Style	utf8	strict	modern	eprint	long	sort	natbib
<code>gost780</code>							
<code>gost780s</code>						+	
<code>gost2003</code>		+		+			
<code>gost2003s</code>		+		+		+	
<code>gost2008</code>			+	+			
<code>gost2008n</code>			+	+			+
<code>gost2008l</code>			+	+	+		
<code>gost2008s</code>			+	+		+	
<code>gost2008ns</code>			+	+		+	+
<code>gost2008ls</code>			+	+	+	+	
<code>ugost2003</code>	+	+		+			
<code>ugost2003s</code>	+	+		+		+	
<code>ugost2008</code>	+		+	+			
<code>ugost2008n</code>	+		+	+			+
<code>ugost2008l</code>	+		+	+	+		
<code>ugost2008s</code>	+		+	+		+	
<code>ugost2008ns</code>	+		+	+		+	+
<code>ugost2008ls</code>	+		+	+	+	+	
Style	utf8	strict	modern	eprint	long	sort	natbib

`Gost2008` style is recommended for most applications. It corresponds to the currently effective Standard 7.0.5-2008. Librarians should use the style `gost2003` instead of `gost2008` to compile a library catalog to meet the Standard 7.1-2003. Use of other styles is best explained through the meaning of options used to compile those styles from the master source.

Two styles, `gost780` and `gost780s`, compiled without `modern` and `strict` options, are retained for backward compatibility. They do not conform the Standards 7.0.5-2008 and 7.1-2003 cited above.

The `strict` option provides conformance to the Standard 7.1-2003. The styles compiled with that option bear the name `gost2003` with possible suffixes `s`, `l`, `n` as explained below. These styles are intended primarily for the librarians who compose a library catalog.

The `modern` option meets the Standard 7.0.5-2008 which can be thought off as a relaxed version of the Standards 7.1-2003. The styles compiled with that option bear the name `gost2008` with possible suffixes `s`, `l`, `n`. The `strict` option has precedence over `modern` so that a style compiled with both options will resemble mainly the `gost2003` style rather than `gost2008`.

If the number of authors exceeds 4, modern styles cut the list of authors to at most 4 persons as prescribed by the Standards. Option `long` overrides this rule to provide backward compatibility with the package `disser` by Stanislav Kruchinin. Two styles, `gost2008l` and `gost2008ls`, compiled with the option `long` mimic behavior of the styles `gost705` and `gost705s` from the `disser` package. Major effect of the `long` option is that the list of authors always precedes book or article title no matter how long is it. Modern styles compiled without `long` place long list of authors behind the title. The names of styles compiled with the option `long` has the suffix `l`. Recall that those styles do not conform effective Standards and their use is discouraged.

The `eprint` option enables formatting electronic publications. In particular, it enables `eprint`, `eprinttype`, `eprintclass`, and `doi` fields for a bibliographic entry. The styles generated without the `eprint` option, ignore the these fields. Starting from the version 1.2 of the GOST package, all modern styles are compiled with this option included, and the suffix `e` which designated this option in earlier versions is not appended to the name of style any more.

The `natbib` option provides compatibility with the `natbib` package. The names of styles compiled with the option `natbib` bear the suffix `n`. Currently 4 styles with that option are available for beta testing.

The `sort` option enables sorting bibliographic references by author names and references titles. The names of styles compiled with the option `sort` bear the suffix `s`.

Finally, the `utf8` option produces bibliographic styles in unicode rather than in 8-bit encoding. Names of those styles bear the prefix `u`.

Beyond bibliographic style, GOST bundle contains CS files (codepage and sorting order).

Encoding	CSF	Sorting order
cp866	ruscii.csf	Cyrillic first, Latin
cp1251	cp1251.csf	Cyrillic first, Latin
koi8-u	koi8u.csf	Cyrillic first, Latin
utf8	utf8cyrillic.csf	Cyrillic first, Latin

In addition, BibTeX8 distribution comes with few more CSFs.

Encoding	CSF	Sorting order
cp866	cp866rus.csf	Latin first, Cyrillic

## 1.1 How to use

1. Select bibliography style by adding appropriate `\bibliographystyle` declaration to your source file `<filename>.tex`, e.g.

```
\bibliographystyle{gost2008}
\bibliography{database}
```

2. Add the field `language="ukrainian"` or `language="russian"` to the bibliographic entries in Ukrainian or Russian languages in your database; English is the default language. German, Italian and French are partially supported.
3. To compile list of references from your database use `bibtex8.exe` rather than `bibtex.exe`. Depending on the codepage of your bibliographic database, indicate one of the CS files listed above as option to `bibtex8.exe`. Run LaTeX, then run `bibTeX8` and LaTeX again:

```
latex <filename>.tex
bibtex8 -B -c <csf_file>.csf <filename>.aux
latex <filename>.tex
```

4. For details on preparing bibliographic database see examples in `gost*.pdf`.
5. `ugost*` styles are primarily intended for use with unicode compilers (`xelatex` and `lualatex`). They should be preferred as well when using 8bit compilers (`latex` and `pdflatex`) if source file is in utf8 encoding.
6. Neither `bibtex.exe` nor `bibtex8.exe` provides correct sorting order of unicode text. It means that using `ugost2008s` or `ugost2008ns` may produce unexpected result for documents in utf8 encoding.
7. `Bibtex8` fails to change case of a string if it contains Cyrillic letter in unicode. Therefore `ugost2008*` styles do not change case of titles and other parts of bibliographic record while 8-bit styles do the case change where appropriate.
8. Either `bibtex8` or `Bibtex8` fail to cut Cyrillic names to initials. Therefore `ugost2008*` styles do not modify name of authors.
9. Package `natbib` is required when choosing styles with suffix `n` in their names.

## 1.2 Customization

Every GOST style defines few commands to format some parts of a reference. You can redefine these commands prior to the `\bibliography{<bibtex_style>}` command. Initial definitions are listed below.

```
\providecommand*\url[1]{\small #1}
\providecommand*\BibUrl[1]{\url{#1}}
\providecommand*\BibAnnote[1]{}
\providecommand*\BibEmph[1]{#1}
```

By default, `gost` styles separate logical parts of a bibliography record by a period and cyrdash (`. ---`). It is legitimate to drop that dash by overriding the command `\BibDash` as follows

```
\providecommand*\BibDash{}
```

By default, `\BibDash` is equivalent to the shorthand `"---` defined by the `babel` package with the option `russian`. It prints a so called Cyrillic dash (`\cyrdash`), which is 20% shorter than ordinary LaTeX dash (`---`), and puts unbreakable space before `\cyrdash` so that dash never appears in the beginning of a line.

### 1.3 Where to get

1. [CTAN:biblio/bibtex/contrib/gost](#).
2. [CTAN:pkg/gost](#).

### 1.4 Version history

#### Version 1.2i (2017.01.12)

1. Documentation and examples update (thanks to Leonid Sinev).
2. Restored `@MastersThesis` instead of `@MasterThesis` (thanks to Leonid Sinev).
3. `media="eresource"` is introduced in addition to `media="online"` and `media="text"`; if present, the `media` field is not ignored any more in modern `bst-styles` compiled without the `strict` option.
4. `location` field is introduced as an alias of `address` field.
5. `@DSCISTHESIS` entry renamed to `@DOCTHESIS`.
6. `school` field in `@THESIS` and similar entries is replaced by `institution` to comply with `biblatex-gost` style.

#### Version 1.2h (2016.08.21)

1. Minor changes in documentation.

#### Version 1.2g (2016.07.25)

1. Minor changes in documentation.

#### Version 1.2f (2016.07.12)

1. Support for patent entry added (thanks to Stanislav Kruchinin).
2. `medium` field renamed to `media` field for compatibility with `biblatex`.

#### Version 1.2e (2016.07.07)

1. Hard coded "URL" string replaced with a language sensitive string (thanks to Roman Budnyi).

#### **Version 1.2d (2015.02.18)**

1. jan, feb, etc. macros fixed.
2. New macro format.month.

#### **Version 1.2c (2015.01.10)**

1. `langid` field is added. It has same meaning as `language` which is now obsolete but is still supported for backward compatibility; `langid` has priority over `language`.
2. `eid` field is added. It has priority over `pages`.
3. The ligature "--- has been substituted with `\BibDash` for `.bst` styles compiled without `modern` options (`gost2003.bst` and `gost2003s.bst`). For modern styles this was done in earlier versions.
4. Spacing around `\BibDash` has been improved.
5. `\BibDash` now typesets short em-dash (`\cyrdash`) only for `russian` and `ukrainian` languages. In earlier versions, it produces short em-dash for all languages.  
(This feature was removed since it did not work with all engines.)

#### **Version 1.2a (2012.08.31)**

1. `\cyrdash` is now defined via `\ProvideTextCommand` rather than `\providecommand`.

#### **Version 1.2 (2012.02.22)**

1. Code refactoring. All styles are now generated from single source file.
2. Support for GOST 7.1-2003. The field `medium` is added to reflect type of material. For most entry types `medium` defaults to `text`.
3. Support for `natbib` package.
4. All modern styles are now compiled with the `eprint` option.

#### **Version 1.1 (2012.01.21)**

1. Support for GOST 7.0.5-2008 and GOST 7.1-2003 is provided.
2. `@Online` entry is added to format a reference to electronic resource on Internet.
3. `@MastersThesis` entry is added to format a reference to master's thesis.
4. `@DSciThesis` entry is added to format a reference to doctor of sciences thesis.
5. `Urldate`, `eprint`, `eprintclass`, `eprinttype` fields are added.

### 1.4.1 Older versions

2012.02.22 Support for natbib package.

2012.02.02 Adaptation to GOST 7.0.5, electronic publishing.

2005.08.12 First version uploaded to CTAN.

2003.06.06 First public version.

## 2 Implementation

We need Russian fonts to produce documentation of the code below. Therefore we switch current language to Russian by issuing the command `\selectlanguage{russian}`.

```
1 (*bst)
2 %%
3 %% This bibstyle attempts to format bibliography according to
4 (!modern)%% GOST 7.1-2003 for bibliographic records.
5 (modern)%% GOST 7.0.5-2008 for bibliographic reference.
6 (*natbib)%%
7 %%-----
8 %% This is an author-year citation style bibliography.
9 %% It requires a special package file to function properly
10 %% such as natbib.sty by Patrick W. Daly.
11 %% The form of the \bibitem entries is
12 %% \bibitem[Jones et al.(1990)]{key}...
13 %% \bibitem[Jones et al.(1990)Jones, Baker, and Smith]{key}...
14 %% where the label part (in brackets) consists of the author names,
15 %% as they should appear in the citation, with the year in parentheses following.
16 %% There must be no space before the opening parenthesis!
17 %% A full list of authors may also follow the year.
18 %% In natbib.sty, it is possible to define the type of enclosures that is
19 %% really wanted (brackets or parentheses), but in either case, there must
20 %% be parentheses in the label.
21 %% The \cite command functions as follows:
22 %% \citet{key}           => Jones et al. (1990)
23 %% \citet*{key}          => Jones, Baker, and Smith (1990)
24 %% \cite{key}            => (Jones et al., 1990)
25 %% \cite*{key}           => (Jones, Baker, and Smith, 1990)
26 %% \cite[chap. 2]{key}   => (Jones et al., 1990, chap. 2)
27 %% \cite[e.g.][]{key}    => (e.g. Jones et al., 1990)
28 %% \cite[e.g.][p. 32]{key} => (e.g. Jones et al., p. 32)
29 %% \citeauthor{key}      => Jones et al.
30 %% \citeauthor*{key}     => Jones, Baker, and Smith
31 %% \citeyear{key}       => 1990
32 %%-----
33 (/natbib)
34
```

## 2.1 Fields

Enlist all entry types allowed in a bibliographic database. Most entries are common for many standard bst styles.

```
35 ENTRY
36 { address
37   annote
38   author
39   booktitle
40   bookauthor
41   chapter
42   edition
43   editor
44   compiler
45   howpublished
46   institution
47   journal
48   key
49   %major           % new in v.1.2i, alias for speciality, not implemented
50   majorcode       % new in v.1.2i, alias for specialitycode
51   month
52   note
53   number
54   organization
55   pages
56   eid             % new in v1.2c
57   publisher
58   school          % alias for institution
59   series
60   %speciality     % new in v.1.2i, eqv. to major in biblatex-gost, not implemented yet...
61   specialitycode % new in v.1.2i, alias of number, eqv. to majorcode in biblatex-gost
62   title
63   %medium         % new in v1.2; renamed to media.
64   media           % new in v1.2f
65   type
66   volume
67   year
68   language
69   langid          % new in v1.2c
70   booklanguage

Entries borrowed from biblatex.
71   %date           % not implemented yet...
72   pagetotal
73   url
74   urldate
75   isbn
76   doi
77   eprinttype     % = archivePrefix
78   eprintclass    % = primaryClass
79   eprint
```



```

80 % new in v1.2f:
81 % appear in biblatex:
82 %addendum % not implemented yet...
83 holder % see patent
84 location % new in v.1.2i, alias of address
85 %subtitle % not implemented yet...
86 titleaddon % new in v.1.2i, see @thesis
87 %version % not implemented yet...
88 % Appear in biblatex-gost for @patent entry:
89 authorcountry % country of the patent authors
90 credits % statement of responsibility, other than provided in Biblatex
91 ipc % Code of the International Patent Classification
92 %media % General material designation NOTE: medium in the above
93 requestnumber % Registration number of the application to the patent document
94 publicationdate % Date of publication
95 publication % and information on the official gazette, which published patent
96 prioritydate % Information about the convention priority: the date of filing of the application,
97 prioritynumber % number and
98 prioritycountry % country name of convention priority.
99 requestdate % ??
100 }
101 {}
102 <!natbib> { label }
103 <natbib> { label extra.label sort.label short.list }
104

```

Declare internal variables and constants used in to format references.

```

105 INTEGERS { output.state before.all mid.sentence after.sentence after.block
106 after.dblslash after.slash after.colon after.semicolon }
107

```

init.state.consts

```

108 FUNCTION {init.state.consts}
109 { #0 'before.all :=
110 #1 'mid.sentence :=
111 #2 'after.sentence :=
112 #3 'after.block :=
113 #4 'after.dblslash :=
114 #5 'after.slash :=
115 #6 'after.colon :=
116 #7 'after.semicolon :=
117 }
118
119 STRINGS { s t }
120
121 STRINGS { curlanguage }
122

```

## 2.2 Formatting functions

change.language Declare function to switch language.

```

123 FUNCTION {change.language}
124 { booklanguage empty$
125   { "" }
126   { booklanguage 'curlanguage :=
127     "\selectlanguageifdefined{"
128     curlanguage *
129     "}" *
130   }
131   if$
132 }
133

```

output.nonnull     Declare functions to output various parts of bibliographic record. Function output.nonnull writes to output buffer the last literal in the stack assuming that it is not empty and adds an appropriate punctuation symbol.

```

134 FUNCTION {output.nonnull}
135 {
136   swap$
137   output.state mid.sentence =
138   { ", " * write$ }
139   { output.state after.block =
140     { add.period$ write$
141       " \BibDash " write$
142       newline$
143       "\newblock " write$
144     }
145     { output.state before.all =
146       'write$
147       { output.state after.dblslash =
148         { "~//" * change.language * " " * write$ }
149         { output.state after.slash =
150           { "~/" * write$ }
151           { output.state after.colon =
152             { ":" * write$ }
153             { "~:" * write$ }
154             { output.state after.semicolon =
155               { ";" * write$ }
156               { "~;" * write$ }
157               { add.period$ " " * write$ }
158             }
159           }
160         }
161       }
162     }
163   }
164   if$
165 }
166   if$
167 }
168   if$
169   mid.sentence 'output.state :=

```

```

170     }
171   if$
172 }
173
output      Function output calls output.nonnull is the last literal string in the stack if it is not empty;
output.check otherwise it discards the literal. Function output.check do the same but also warns if the indicated
            field is empty.
174 FUNCTION {output}
175 { duplicate$ empty$
176   'pop$
177   'output.nonnull
178   if$
179 }
180
181 FUNCTION {output.check}
182 { 't :=
183   duplicate$ empty$
184   { pop$ "empty " t * " in " * cite$ * warning$ }
185   'output.nonnull
186   if$
187 }
188
fin.entry   fin.entry finalizes current entry. It writes dot, if no dot is found in stack, and starts new line.
189 FUNCTION {fin.entry}
190 { add.period$
191   write$
192   newline$
193 }
194
new.block   Declare family of functions to put punctuation marks depending of current status of output
            stack. The just check output state and revert it another state if required. Checking output state
            prevents occasional doubling of punctuation marks.
195 FUNCTION {new.block}
196 { output.state before.all =
197   'skip$
198   { after.block 'output.state := }
199   if$
200 }
201
new.dblslash
202 FUNCTION {new.dblslash}
203 { output.state before.all =
204   'skip$
205   { after.dblslash 'output.state := }
206   if$
207 }
208

```

```

new.slash
209 FUNCTION {new.slash}
210 { output.state before.all =
211   'skip$
212   { after.slash 'output.state := }
213   if$
214 }
215

new.colon
216 FUNCTION {new.colon}
217 { output.state before.all =
218   'skip$
219   { after.colon 'output.state := }
220   if$
221 }
222

new.semicolon
223 FUNCTION {new.semicolon}
224 { output.state before.all =
225   'skip$
226   { after.semicolon 'output.state := }
227   if$
228 }
229

new.sentence
230 FUNCTION {new.sentence}
231 { output.state after.block =
232   'skip$
233   { output.state before.all =
234     'skip$
235     { after.sentence 'output.state := }
236     if$
237   }
238   if$
239 }
240

add.blank
241 FUNCTION {add.blank}
242 { " " * before.all 'output.state :=
243 }
244

not      Declare few logical functions.
245 FUNCTION {not}
246 { { #0 }
247   { #1 }
248   if$
249 }
250

```

and

```
251 FUNCTION {and}
252 {   'skip$
253   { pop$ #0 }
254   if$
255 }
256
```

or

```
257 FUNCTION {or}
258 {   { pop$ #1 }
259   'skip$
260   if$
261 }
262
```

`chop.word` The function `chop.word` in `substr len str chop.word` removes given substring `substr` of length `len` from the beginning of the string `str`.

```
263 ⟨*sort | natbib⟩
264 INTEGERS { len }
265
266 FUNCTION {chop.word}
267 { 's :=
268   'len :=
269   s #1 len substring$ =
270     { s len #1 + global.max$ substring$ }
271   's
272   if$
273 }
274 ⟨/sort | natbib⟩
275
```

`non.stop`

```
276 FUNCTION {non.stop}
277 { duplicate$
278   "}" * add.period$
279   #-1 #1 substring$ "." =
280 }
281
```

`new.block.checka`

```
282 FUNCTION {new.block.checka}
283 { empty$
284   'skip$
285   'new.block
286   if$
287 }
288
```

`new.block.checkb`

```
289 FUNCTION {new.block.checkb}
290 { empty$
```

```

291 swap$ empty$
292 and
293   'skip$
294   'new.block
295 if$
296 }
297

```

new.sentence.checka

```

298 FUNCTION {new.sentence.checka}
299 { empty$
300   'skip$
301   'new.sentence
302 if$
303 }
304

```

new.sentence.checkb

```

305 FUNCTION {new.sentence.checkb}
306 { empty$
307 swap$ empty$
308 and
309   'skip$
310   'new.sentence
311 if$
312 }
313

```

new.dblslash.checka For online entry.

```

314 FUNCTION {new.dblslash.checka}
315 { empty$
316   'skip$
317   'new.dblslash
318 if$
319 }
320

```

field.or.null

```

321 FUNCTION {field.or.null}
322 { duplicate$ empty$
323   { pop$ "" }
324   'skip$
325 if$
326 }
327

```

emphasize Declare function to emphasize last string in stack.

```

328 FUNCTION {emphasize}
329 { duplicate$ empty$
330   { pop$ "" }
331   { "\BibEmph{" swap$ * "}" * }
332 if$

```

```
333 }
334
```

`bracify`        New in v.1.2. An idea borrowed from `apsrev4-1.bst`. Declare function to enclose last word by braces even if empty.

```
335 FUNCTION {bracify}
336 { duplicate$ empty$
337   { pop$ "{}" }
338   { "{" swap$ * "}" * }
339   if$
340 }
341
```

`e.square.brackets`    This and the next functions are used to enclose last word by brackets.

```
342 FUNCTION {enclose.square.brackets}
343 {
344   duplicate$ empty$
345   { pop$ "" }
346   { "[" swap$ * "]" * }
347   if$
348 }
349
```

`e.round.brackets`

```
350 FUNCTION {enclose.round.brackets}
351 {
352   duplicate$ empty$
353   { pop$ "" }
354   { "(" swap$ * ")" * }
355   if$
356 }
357
```

`space.word`        `space.word` inserts space before last string in stack.

```
358 FUNCTION {space.word}
359 { " " swap$ * " " * }
360
```

## 2.3 Standard abbreviations

`bbl.edby`    Declare language-sensitive abbreviations.

```
361 FUNCTION {bbl.edby} % { "\bbledby{}" }
362 { curlanguage "english" =
363   {"Ed.\ by"}
364   { curlanguage "ukrainian" =
365     {!utf8}      {"{\cyr\CYRP\cyrii\cyrd\ \cyrr\cyre\cyrd.}" }
366     {utf8}       {"Під ред."}
367     { curlanguage "russian" =
368     {!utf8}      {"{\cyr\CYRP\cyro\cyrd\ \cyrr\cyre\cyrd.}" }
369     {utf8}       {"Под ред."}
370     { curlanguage "german" =
```

```

371         { "ed." }
372         {"language is not defined: " language "edby" * * warning$ "Ed.\ by"}
373     if$}
374 if$}
375 if$}
376 if$}
377

```

bb1.cmplr

```

378 FUNCTION {bb1.cmplr}
379 { curlanguage "english" =
380   { "Compiler"}
381   { curlanguage "german" =
382     { "Hrsg." }
383     { curlanguage "ukrainian" =
384       (!utf8)      {"{\cyr\CYRU\cyrk\cyr1.}" }
385       (utf8)      {"{Укл.}" }
386       { curlanguage "russian" =
387         (!utf8)    {"{\cyr\CYRS\cyro\cyrs\cyrt.}" }
388         (utf8)    {"{Сост.}" }
389         {"language is not defined: " language "cmplr" * * warning$ "Compiler"}
390         if$}
391       if$}
392     if$}
393   if$}
394

```

bb1.edition

```

395 FUNCTION {bb1.edition} % { "\bb1edition{}" }
396 { curlanguage "english" =
397   {"edition"}
398   { curlanguage "ukrainian" =
399     (!utf8)      {"{\cyr\cyrv\cyri\cyrd.}" }
400     (utf8)      {"{Вид.}" }
401     { curlanguage "russian" =
402       (!utf8)    {"{\cyr\cyri\cyrz\cyrd.}" }
403       (utf8)    {"{Изд.}" }
404       { curlanguage "german" =
405         {" {aus.}" } %%% { "Auf1." } ??
406         { curlanguage "italian" =
407           {"edizione"}
408           { curlanguage "french" =
409             {"\'{e}dition"}
410             {"language is not defined: " language "edition" * * warning$ "edition"}
411             if$}
412           if$}
413         if$}
414       if$}
415     if$}
416   if$}
417

```



bbl.vvolume

```
418 FUNCTION {bbl.vvolume} % { "\bblVolume{" }
419 { curlanguage "english" = curlanguage "french" = or curlanguage "italian" = or
420   {"Volume"}
421   { curlanguage "ukrainian" = curlanguage "russian" = or
422     \!utf8 { "\CYRT\cyro\cyrm" }
423     \utf8 { "Том" }
424     { curlanguage "german" =
425       {"{Band}"} %%% { "Volumen" }
426       {"language is not defined: " language "vvolume" * * warning$ "Volume"}
427       if$}
428   if$}
429 if$}
430
```

bbl.vvol

```
431 FUNCTION {bbl.vvol} % { "\bblVol{" }
432 { curlanguage "english" = curlanguage "french" = or curlanguage "italian" = or
433   {"Vol."}
434   { curlanguage "ukrainian" = curlanguage "russian" = or
435     \!utf8 { "\CYRT."}
436     \utf8 { "Т." }
437     { curlanguage "german" =
438       {"{Bd.}"} %%% { "Vol." }
439       {"language is not defined: " language "vvol" * * warning$ "Vol."}
440       if$}
441   if$}
442 if$}
443
```

bbl.iissue

```
444 FUNCTION {bbl.iissue} % { "\bblIssue{" }
445 { curlanguage "english" =
446   {"Issue"}
447   { curlanguage "ukrainian" =
448     \!utf8 { "\CYRV\cyri\cyrp\cyru\cyrs\cyrk"}
449     \utf8 { "Випуск" }
450     { curlanguage "russian" =
451       \!utf8 { "\CYRV\cyrery\cyrp\cyru\cyrs\cyrk"}
452       \utf8 { "Выпуск" }
453       { curlanguage "german" =
454         {"{Heft}"} %%% { "Ausgabe" }
455         {"language is not defined: " language "iissue" * * warning$ "Issue"}
456         if$}
457     if$}
458   if$}
459 if$}
460
```

bbl.iiss

```
461 FUNCTION {bbl.iiss} % { "\bblIss{" }
```

```

462 { curlanguage "english" =
463   {"Iss."}
464   { curlanguage "ukrainian" =
465     \!utf8 {"\CYRV\cyri\cyrp."}
466     \utf8 {"Вип."}
467     { curlanguage "russian" =
468       \!utf8 {"\CYRV\cyrery\cyrp."}
469       \utf8 {"Вып."}
470       { curlanguage "german" =
471         {\H.}}
472         {"language is not defined: " language "iiss" * * warning$ "Iss."}
473         if$}
474     if$}
475   if$}
476 if$}
477

```

bbl.of

```

478 FUNCTION {bbl.of}      % { "\bbl.of{}" }
479 { curlanguage "english" =
480   {"of"}
481   { curlanguage "german" =
482     {"von" }
483     { curlanguage "ukrainian" =
484       \!utf8 {"{\cyr\cyrii\cyrz}" }
485       \utf8 {"{is}" }
486       { curlanguage "russian" =
487         \!utf8 {"{\cyr\cyri\cyrz}" }
488         \utf8 {"{из}" }
489         {"language is not defined: " language "of" * * warning$ "of"}
490         if$}
491       if$}
492     if$}
493 if$}
494

```

bbl.etal

```

495 FUNCTION {bbl.etal}    % { "\bbl.of{}" }
496 { curlanguage "english" =
497   {"et~al."}
498   { curlanguage "german" =
499     {"u.~a." }
500     { curlanguage "ukrainian" =
501       \!utf8 {"{\cyr\cyrt\cyra~\cyrii\cyrn.}" }
502       \utf8 {"{ra~iH.}" }
503       { curlanguage "russian" =
504         \!utf8 {"{\cyr\cyri~\cyrd\cyrr.}" }
505         \utf8 {"{и~др.}" }
506         {"language is not defined: " language "et~al" * * warning$ "et~al."}
507         if$}
508       if$}

```

```

509   if$}
510 if$}
511

```

bb1.and

```

512 <*natbib>
513 FUNCTION {bb1.and} % { "\bbland{}" }
514 { curlanguage "english" =
515   {"and"}
516   { curlanguage "german" =
517     { "und" }
518     { curlanguage "ukrainian" =
519 <!utf8>       {"{\cyrii}"}
520 <utf8>       {"i"}
521     { curlanguage "russian" =
522 <!utf8>       {"{\cyri}"}
523 <utf8>       {"ѡ"}
524     {"language is not defined: " language "and" * * warning$ "and"}
525     if$}
526   if$}
527   if$}
528 if$}
529 </natbib>
530

```

bb1.nnumber

```

531 FUNCTION {bb1.nnumber} % { "\bb1Number{}" }
532 { curlanguage "english" =
533   {"Number"}
534   { curlanguage "ukrainian" = curlanguage "russian" = or
535 <!utf8>     { "\CYRN\cyro\cyrm\cyre\cyrr" }
536 <utf8>     { "{Homep}" }
537   { curlanguage "german" =
538     {"{Heft}"} %% { "Anzahl" }
539     {"language is not defined: " language "nnumber" * * warning$ "Number"}
540     if$}
541   if$}
542 if$}
543

```

bb1.number

```

544 FUNCTION {bb1.number} % { "\bb1number{}" }
545 { curlanguage "english" =
546   {"number"}
547   { curlanguage "ukrainian" = curlanguage "russian" = or
548 <!utf8>     {"{\cyr\cyrn\cyro\cyrm\cyre\cyrr}"}
549 <utf8>     {"{номер}"}
550   { curlanguage "german" =
551     {"{Heft}"} %% { "anzahl" }???
552     {"language is not defined: " language "number" * * warning$ "number"}
553     if$}
554   if$}

```

```
555 if$}
556
```

bb1.nr

```
557 FUNCTION {bb1.nr} % { "\bb1no{}" }
558 { curlanguage "english" =
559   {"no."}
560   { curlanguage "italian" =
561     { "no" }
562     { curlanguage "ukrainian" = curlanguage "russian" = or
563     (!utf8) { "{\cyr\textnumero}" }
564     (utf8) { "{№}" }
565     { curlanguage "german" =
566       {"{H.}" } %% { "an." }
567       {"language is not defined: " language "nr" * * warning$ "no."}
568     if$}
569   if$}
570 if$}
571 if$}
572
```

bb1.nnr

```
573 FUNCTION {bb1.nnr} % { "\bb1no{}" }
574 { curlanguage "english" =
575   {"No."}
576   { curlanguage "ukrainian" = curlanguage "russian" = or
577   (!utf8) { "{\cyr\textnumero}" }
578   (utf8) { "{№}" }
579   { curlanguage "german" =
580     {"{H.}" } %% { "an." }
581     {"language is not defined: " language "nnr" * * warning$ "No."}
582   if$}
583 if$}
584 if$}
585
```

bb1.in

```
586 FUNCTION {bb1.in} % { "\bb1in{}" }
587 { curlanguage "english" = curlanguage "german" = or
588   {"in"}
589   { curlanguage "ukrainian" = curlanguage "russian" = or
590   (!utf8) { "{\cyr\cyrv}" }
591   (utf8) { "{В}" }
592   {"language is not defined: " language "in" * * warning$ "in"}
593   if$}
594 if$}
595
```

bb1.iin

```
596 FUNCTION {bb1.iin} % { "\bb1in{}" }
597 { curlanguage "english" = curlanguage "german" = or
598   {"In"}

```

```

599 { curlanguage "ukrainian" = curlanguage "russian" = or
600 <!utf8>      { "\CYRV" }
601 <utf8>      { "{B}" }
602 {"language is not defined: " language "iin" * * warning$ "In"}
603 if$}
604 if$}
605

```

bbl.pages

```

606 FUNCTION {bbl.pages} % { "\bblpp." }
607 { curlanguage "english" = curlanguage "french" = or curlanguage "italian" = or
608   {"p."} %%% {"pp."}
609   { curlanguage "ukrainian" = curlanguage "russian" = or
610     <!utf8>      {"{\cyr\cyrs.}" }
611     <utf8>      {"{c.}" }
612     { curlanguage "german" =
613       {"S."} %%% {"s." }
614       {"language is not defined: " language "pages" * * warning$ "p."}
615       if$}
616     if$}
617 if$}
618

```

bbl.page

```

619 FUNCTION {bbl.page} % { "\bblp." }
620 { curlanguage "english" = curlanguage "french" = or curlanguage "italian" = or
621   {"p."}
622   { curlanguage "ukrainian" = curlanguage "russian" = or
623     <!utf8>      {"{\cyr\cyrs.}" }
624     <utf8>      {"{c.}" }
625     { curlanguage "german" =
626       {"S."} %%% {"s." }
627       {"language is not defined: " language "page" * * warning$ "p."}
628       if$}
629     if$}
630 if$}
631

```

bbl.ppages

```

632 FUNCTION {bbl.ppages}% { "\bblPp." }
633 { curlanguage "english" = curlanguage "french" = or curlanguage "italian" = or
634   {"P."} %%% {"Pp." }
635   { curlanguage "ukrainian" = curlanguage "russian" = or
636     <!utf8>      {"{\cyr\CYRS.}" }
637     <utf8>      {"{C.}" }
638     { curlanguage "german" =
639       {"S."}
640       {"language is not defined: " language "ppages" * * warning$ "P."}
641       if$}
642     if$}
643 if$}
644

```

bb1.ppage

```
645 FUNCTION {bb1.ppage} % { "\bb1P." }
646 { curlanguage "english" = curlanguage "french" = or curlanguage "italian" = or
647   {"P."}
648   { curlanguage "ukrainian" = curlanguage "russian" = or
649   (!utf8)   {"{\cyr\CYRS.}" }
650   (utf8)   {"{C.}" }
651     { curlanguage "german" =
652     {"S."}
653     {"language is not defined: " language "ppage" * * warning$ "P."}
654     if$}
655   if$}
656 if$}
657
```

bb1.url Next function was added in version 2016.07.07.

```
658 FUNCTION {bb1.url}
659 { curlanguage "english" =
660   {"Access mode"}
661   { curlanguage "ukrainian" =
662   (!utf8)   {"{\CYRR\cyre\cyrrh\cyri\cyrm\ \cyrd\cyro\cyrs\cyrt\cyru\cyrp\cyru}" }
663   (utf8)   {"{Режим доступу}" }
664   { curlanguage "russian" =
665   (!utf8)   {"{\CYRR\cyre\cyrrh\cyri\cyrm\ \cyrd\cyro\cyrs\cyrt\cyru\cyrp\cyra}" }
666   (utf8)   {"{Режим доступа}" }
667   { curlanguage "german" =
668   {"{online; abgerufen}" }
669   {"language is not defined: " language "urldate" * * warning$ "online; accessed" }
670   if$}
671   if$}
672   if$}
673 if$}
```

bb1.urldate Next function was added in version 2012.01.15.

```
674 FUNCTION {bb1.urldate}
675 { curlanguage "english" =
676   {"online; accessed"}
677   { curlanguage "ukrainian" =
678   (!utf8)   {"{\cyrd\cyra\cyrt\cyra\ \cyrz\cyrv\cyre\cyrr\cyrn\cyre\cyrn\cyrn\cyrya}" }
679   (utf8)   {"{дата звернення}" }
680   { curlanguage "russian" =
681   (!utf8)   {"{\cyrd\cyra\cyrt\cyra\ \cyro\cyrb\cyrr\cyra\cyrrshch\cyre\cyrn\cyri\cyrya}" }
682   (utf8)   {"{дата обращения}" }
683   { curlanguage "german" =
684   {"{online; abgerufen}" }
685   {"language is not defined: " language "urldate" * * warning$ "online; accessed" }
686   if$}
687   if$}
688   if$}
689 if$}
690
```

bb1.techreport

```
691 FUNCTION {bb1.techreport} % rename to bb1.techreport
692 { curlanguage "english" =
693   { "Rep." }
694   { curlanguage "german" =
695     { "Bericht" }
696     { curlanguage "russian" =
697 <!utf8>       { "{\cyr\CYRO\cyrt\cyrch\cyre\cyrt}" }
698 <utf8>       { "{Отчет}" }
699     { "language is not defined: " language "techrep" * * warning$ "Rep." }
700     if$}
701   if$}
702 if$}
703
```

bb1.mathesis

```
704 FUNCTION {bb1.mathesis}
705 { curlanguage "english" =
706   { "Master's thesis" }
707   { curlanguage "german" =
708     { "Diss.~Mag." }
709     { curlanguage "russian" =
710 <!utf8>       { "{\cyr\CYRK\cyrv\cyra\cyrl\cyri\cyrf\cyri\cyrk\cyra\cyrc\cyri"
711 <!utf8>       " \cyro\cyrn\cyrn\cyra\cyrya\ \cyrr\cyra\cyrb\cyro\cyrt\cyra\ " *
712 <!utf8>       "\cyrm\cyra\cyrg\cyri\cyrs\cyrt\cyrr\cyra}" * }
713 <utf8>       { "{Квалификационная работа магистра}" }
714     { "language is not defined: " language "mthesis" * * warning$ "Master's thesis" }
715     if$}
716   if$}
717 if$}
718
```

bb1.phdthesis

```
719 FUNCTION {bb1.phdthesis}
720 { curlanguage "english" =
721   { "Ph.\,D. thesis" }
722   { curlanguage "german" =
723     { "Diss.~Ph.\,D." }
724     { curlanguage "russian" =
725 <!utf8>       { "{\cyr\cyrd\cyri\cyrs.\ \ldots\ \cyrk\cyra\cyrn\cyrd. "
726 <!utf8>       "\cyrn\cyra\cyru\cyrk}" * }
727 <utf8>       { "{дис.\ \ldots\ канд. наук}" }
728     { curlanguage "french" =
729       { "Th\ '{e}se de doctorat" }
730     { "language is not defined: " language "phdthesis" * * warning$ "Ph.\,D. thesis" }
731     if$}
732   if$}
733 if$}
734 if$}
735
```

bb1.docthis

```

736 FUNCTION {bbl.docthesi}
737 { curlanguage "english" =
738   { "Dr.\,Sci. dissertation" }
739   { curlanguage "german" =
740     { "Diss.~Dr." }
741     { curlanguage "russian" =
742       (!utf8)      { "{\cyr\cyrd\cyri\cyrs.\ \ldots\ \cyrd-\cyrr\cyra\ "
743         (!utf8)      { "\cyrn\cyra\cyru\cyrk}" * }
744       (utf8)      { "{дис.\ \ldots\ д-ра наук}" }
745       { "language is not defined: " language "docthesi" * * warning$ "Dr.\,Sci. dissertation" }
746       if$}
747     if$}
748 if$}
749

```

bbl.thesis.type{

```

750 FUNCTION {bbl.thesis.type}
751 { type "mathesis" =
752   { bbl.mathesis }
753   { type "phdthesis" =
754     { bbl.phdthesis }
755     { type "docthesi" =
756       { bbl.docthesi }
757       %%{ "!!!" type * "t" change.case$ }
758     (!utf8)      { type "t" change.case$ }
759     (utf8)      { type }
760     if$}
761   if$}
762 if$}
763

```

bbl.nnoaddress

```

764 FUNCTION {bbl.nnoaddress}
765 { curlanguage "english" =
766   { "S.\ l." }
767   { curlanguage "russian" =
768     (!utf8)      { "{\cyr\CYRB.\ \cyrm.}" }
769     (utf8)      { "{Б.\ м.}" }
770     { "language is not defined: " language "nnoaddress" * * warning$ "S.\ l." }
771     if$}
772 if$}
773

```

bbl.nopublisher

```

774 FUNCTION {bbl.nopublisher}
775 { curlanguage "english" =
776   { "s.\ n." }
777   { curlanguage "russian" =
778     (!utf8)      { "{\cyr\cyrb.\ \cyri.}" }
779     (utf8)      { "{б.\ н.}" }
780     { "language is not defined: " language "nopublisher" * * warning$ "s.\ n." }
781     if$}

```



```
782 if$}
783
```

bb1.nnopublisher

```
784 FUNCTION {bb1.nnopublisher}
785 { curlanguage "english" =
786   { "S.\ n." }
787   { curlanguage "russian" =
788     <!utf8>   { "{\cyr\CYRB.\ \cyri.}" }
789     <utf8>   { "{Б.\ И.}" }
790     { "language is not defined: " language "nnopublisher" * * warning$ "S.\ n." }
791   if$}
792 if$}
793
```

bb1.media.text

```
794 FUNCTION {bb1.media.text}
795 { curlanguage "english" =
796   { "Text" }
797   { curlanguage "russian" = curlanguage "ukrainian" = or
798     <!utf8>   { "{\cyr\CYRT\cyre\cyrk\cyrs\cyrt}" }
799     <utf8>   { "{Текст}" }
800     { "language is not defined: " language "media" * * warning$ "Text" }
801   if$}
802 if$}
803
```

bb1.media.eresource

```
804 FUNCTION {bb1.media.eresource}
805 { curlanguage "english" =
806   { "Electronic resource" }
807   { curlanguage "russian" =
808     <!utf8>   { "{\cyr\CYREREV\cyrl\cyre\cyrk\cyrt\cyrr\cyro\cyrn\cyrn\cyrery\cyrishrt\ "
809     <!utf8>   { "\cyrr\cyre\cyrs\cyru\cyrr\cyrs}" * }
810     <utf8>   { "{Электронный ресурс}" }
811     { curlanguage "ukrainian" =
812     <!utf8>   { "{\cyr\CYRE\cyrl\cyre\cyrk\cyrt\cyrr\cyro\cyrn\cyrn\cyri\cyrishrt\ "
813     <!utf8>   { "\cyrr\cyre\cyrs\cyru\cyrr\cyrs}" * }
814     <utf8>   { "{Електронний ресурс}" }
815     { "language is not defined: " language "media" * * warning$ "Electronic resource" }
816   if$}
817   if$}
818 if$}
819
```

bb1.media.online

```
820 FUNCTION {bb1.media.online}
821 { curlanguage "english" =
822   { "Electronic resource online" }
823   { curlanguage "russian" =
824     <!utf8>   { "{\cyr\CYREREV\cyrl\cyre\cyrk\cyrt\cyrr\cyro\cyrn\cyrn\cyrery\cyrishrt\ "
825     <!utf8>   { "\cyrr\cyre\cyrs\cyru\cyrr\cyrs\ \cyro\cyrn\cyrl\cyra\cyrishrt\cyrn}" * }

```

```

826 (utf8)      { "{Электронный ресурс онлайн}" }
827 { curlanguage "ukrainian" =
828 (!utf8)      { "{\cyr\CYRE\cyrl\cyre\cyrk\cyrt\cyrr\cyro\cyrn\cyri\cyrishrt\ "
829 (!utf8)      { "\cyr\cyre\cyrs\cyru\cyrr\cyrs\ \cyro\cyrn\cyrl\cyra\cyrishrt\cyr}" * }
830 (utf8)      { "{Електронний ресурс онлайн}" }
831 { "language is not defined: " language "media" * * warning$ "Electronic resource" }
832     if$}
833 if$}
834 if$}
835

```

bb1.chief

```

836 FUNCTION {bb1.chief}
837 { curlanguage "english" =
838   { "chief" }
839   { curlanguage "russian" =
840 (!utf8)      { "\cyrr\cyru\cyrk." }
841 (utf8)      { "{рук.}" }
842   { curlanguage "ukrainian" =
843 (!utf8)      { "\cyrr\cyru\cyrk." }
844 (utf8)      { "{рук.}" }
845   { "language is not defined: " language "chief" * * warning$ "chief" }
846     if$}
847 if$}
848 if$}
849

```

bb1.executor

```

850 FUNCTION {bb1.executor}
851 { curlanguage "english" =
852   { "Executor" }
853   { curlanguage "russian" =
854 (!utf8)      { "{\cyr\cyri\cyrs\cyrp\cyro\cyrl\cyrn.}" }
855 (utf8)      { "{исполн.}" }
856   { curlanguage "ukrainian" =
857 (!utf8)      { "{\cyr\cyrv\cyri\cyrk\cyro\cyrn\cyra\cyrv\cyre\cyrc\cyrsfts\sn}" }
858 (utf8)      { "{виконавець}" }
859   { "language is not defined: " language "executor" * * warning$ "executor" }
860     if$}
861 if$}
862 if$}
863

```

bb1.media

```

864 %FUNCTION {bb1.media}
865 %{ type$ "online" =
866 %   { bb1.media.eresource }
867 %   { bb1.media.text }
868 %if$}
869 FUNCTION {bb1.media}
870 { media "online" =
871   { bb1.media.online }

```

```

872 { media "eresource" =
873   { bbl.media.eresource }
874   { bbl.media.text }
875   if$}
876 if$}
877

```

bbl.req

```

878 FUNCTION {bbl.req}
879 {
880   curlanguage "english" =
881     { "req." }
882     { curlanguage "german" =
883       { "ang." }
884       { curlanguage "russian" =
885 (!utf8)   { "{\cyr\cyrz\cyra\cyrya\cyrv\cyr1.}" }
886 (utf8)    { "{заявл.}" }
887         { "language is not defined: " language "req" * * warning$ "req" }
888         if$
889         }
890     if$
891     }
892   if$
893 }
894

```

bbl.publ

```

895 FUNCTION {bbl.publ}
896 {
897   curlanguage "english" =
898     { "publ." }
899     { curlanguage "german" =
900       { "ausg." }
901       { curlanguage "russian" =
902 (!utf8)   { "{\cyr\cyro\cyrp\cyru\cyrb\cyr1.}" }
903 (utf8)    { "{опубл.}" }
904         { "language is not defined: " language "publication" * * warning$ "publication" }
905         if$
906         }
907     if$
908     }
909   if$
910 }
911

```

bbl.priority

```

912 FUNCTION {bbl.priority}
913 {
914   curlanguage "english" =
915     { "priority" }
916     { curlanguage "german" =
917       { "Prioritat" }

```

```

918     { curlanguage "russian" =
919 <!utf8)         { "{\cyr\cyrp\cyrr\cyri\cyro\cyrr\cyri\cyrt\cyre\cyrt}" }
920 <utf8)         { "{\приоритет}" }
921     { "language is not defined: " language "priority" * * warning$ "priority" }
922     if$
923     }
924 if$
925 }
926 if$
927 }
928

```

## 2.4 Formatting functions

Declare functions to format separate elements of bibliographic reference.

```

929 INTEGERS { nameptr namesleft numnames }
930
931

```

`format.names` Function `format.names` has 2 version. First is for bibliographic records rather than for bibliographic references. It is used for `.bst` styles compiled without the `modern` option. It format every name as ‘LastName, F. S.’. Historically, this version was used first for earlier styles included into GOST bundle.

### Important note

Neither `bibtex` nor `bibtex8` can handle unicoded text without troubles. In particular, they fail to reduce a Cyrillic name to initials. Therefore we avoid using `f.` primitive when option `utf8` is in effect.

```

932 <!*modern)
933 FUNCTION {format.names}
934 {
935 <!utf8) #1 "{vv~}{ll}{~jj}{,~f.}" format.name$
936 <utf8) #1 "{vv~}{ll}{~jj}{,~ff}" format.name$
937 }
938 </!modern)

```

Second version drops comma from output so that every name is formatted as ‘LastName F. S.’. It also substitutes 4th and following names by localized term ‘et al’.

```

939 <*modern)
940 FUNCTION {format.names}
941 {
942   's :=
943   #1 'nameptr :=
944   s num.names$ 'numnames :=
945   numnames 'namesleft :=
946   { namesleft #0 > }
947   { s nameptr
948 <!utf8)       "{vv~}{ll}{~jj}{~f.}" format.name$ 't :=
949 <utf8)       "{vv~}{ll}{~jj}{~ff}" format.name$ 't :=
950   nameptr #1 >
951     { nameptr #4 =

```

```

952     numnames #4 > and
953     { "others" 't :=
954       #1 'namesleft := }
955     'skip$
956     if$
957     namesleft #1 >
958     { ", " * t * }
959     { t "others" =
960     t "~others" =
961     or
962     (!strict)           { " " * bbl.etal * }
963     (strict)           { " " * bbl.etal enclose.square.brackets *}
964     { ", " * t * }
965     if$
966     }
967     if$
968     }
969     't
970     if$
971     nameptr #1 + 'nameptr :=
972     namesleft #1 - 'namesleft :=
973     }
974     while$
975 }
976 </modern>
977

```

format.lab.names Declare function to go to optional argument of \bibitem in the styles generated with the option natbib.

```

978 (*natbib)
979 FUNCTION {format.lab.names}
980 { 's :=
981   language empty$
982   { "english" 'curlanguage := }
983   { language 'curlanguage := }
984   if$
985   s #1 "{vv~}{ll}" format.name$
986   s num.names$ duplicate$
987   #2 >
988   %{ pop$ " et~al." * }
989   { pop$ " " bbl.etal * * }
990   { #2 <
991     'skip$
992     { s #2 "{ff }{vv }{ll}{ jj}" format.name$ "others" =
993       %{ " et~al." * }
994       { " " bbl.etal * * }
995       %{ " and " * s #2 "{vv~}{ll}" format.name$ * }
996       { " " bbl.and " " * * * s #2 "{v~}{ll}" format.name$ * }
997     if$
998     }
999   if$

```

```

1000   }
1001   if$
1002 }
1003 </natbib>
1004

```

`format.names.rev` Declare function to format names for authors/bookauthors list after title and etc. Note that `format.names.rev` cuts list of names to at most 4 persons. We do not cut names to initials in this list.

```

1005 FUNCTION {format.names.rev}
1006 {
1007   's :=
1008   #1 'nameptr :=
1009   s num.names$ 'numnames :=
1010   numnames 'namesleft :=
1011   { namesleft #0 > }
1012   { s nameptr
1013     <!utf8>      %"{f.}{~vv}{~ll}{, jj}" format.name$ 't :=
1014     <!utf8>      "{ff}{~vv}{~ll}{, jj}" format.name$ 't :=
1015     <utf8>      "{ff}{~vv}{~ll}{, jj}" format.name$ 't :=
1016     nameptr #1 >
1017     { nameptr #4 =
1018       numnames #4 > and
1019       { "others" 't :=
1020         #1 'namesleft := }
1021       'skip$
1022       if$
1023       namesleft #1 >
1024       { ", " * t * }
1025       { t "others" =
1026         t "~others" =
1027         or
1028         <istrict>      { " " * bbl.etal * }
1029         <strict>      { " " * bbl.etal enclose.square.brackets * }
1030         { ", " * t * }
1031         if$
1032       }
1033       if$
1034     }
1035     't
1036     if$
1037     nameptr #1 + 'nameptr :=
1038     namesleft #1 - 'namesleft :=
1039   }
1040   while$
1041 }
1042

```

`format.key` Function to substitute empty field (usually, author name) with the key field.

```

1043 <natbib>
1044 FUNCTION {format.key}

```

```

1045 { empty$
1046   { key field.or.null }
1047   { "" }
1048   if$
1049 }
1050 </natbib>
1051

```

format.authors

```

1052 FUNCTION {format.authors}
1053 { author empty$
1054   { "" }
1055   { author format.names emphasize}
1056   if$
1057 }
1058

```

author.key.label

```

1059 (*natbib)
1060 FUNCTION {author.key.label}
1061 { author empty$
1062   { key empty$
1063     { cite$ #1 #3 substring$ }
1064     'key
1065     if$
1066   }
1067   { author format.lab.names }
1068   if$
1069 }
1070
1071 FUNCTION {author.editor.key.label}
1072 { author empty$
1073   { editor empty$
1074     { key empty$
1075       { cite$ #1 #3 substring$ }
1076       %'key          %% causes lost of year
1077       { "{}" key * } %% Bug in bibtex8 ??
1078       if$
1079     }
1080     { editor format.lab.names }
1081     if$
1082   }
1083   { author format.lab.names }
1084   if$
1085 }
1086
1087 FUNCTION {author.key.organization.label}
1088 { author empty$
1089   { key empty$
1090     { organization empty$
1091       { cite$ #1 #3 substring$ }

```

```

1092         { "The " #4 organization chop.word #3 text.prefix$ }
1093     if$
1094 }
1095     'key
1096     if$
1097 }
1098 { author format.lab.names }
1099 if$
1100 }
1101
1102 FUNCTION {editor.key.organization.label}
1103 { editor empty$
1104     { key empty$
1105         { organization empty$
1106             { cite$ #1 #3 substring$ }
1107             { "The " #4 organization chop.word #3 text.prefix$ }
1108             if$
1109         }
1110         'key
1111         if$
1112     }
1113     { editor format.lab.names }
1114     if$
1115 }
1116
1117 FUNCTION {calc.short.authors}
1118 { type$ "book" =
1119     type$ "inbook" =
1120     or
1121     'author.editor.key.label
1122     { type$ "proceedings" =
1123         'editor.key.organization.label
1124         { type$ "manual" =
1125             'author.key.organization.label
1126             'author.key.label
1127             if$
1128         }
1129         if$
1130     }
1131     if$
1132     'short.list :=
1133 }
1134
1135 FUNCTION {calc.label}
1136 { calc.short.authors
1137     short.list
1138     "("
1139     *
1140     year duplicate$ empty$
1141     short.list key field.or.null = or

```



```

1142     { pop$ "" }
1143     'skip$
1144   if$
1145   *
1146   'label :=
1147 }
1148
1149 </natbib>
1150

```

ormat.bookauthors This function is used only once, in bookauthor.before, and the latter is used only in inbook entry.

```

1151 FUNCTION {format.bookauthors}
1152 { bookauthor empty$
1153   { "" }
1154   { bookauthor format.names}
1155   if$
1156 }
1157

```

ormat.authors.after

```

1158 FUNCTION {format.authors.after}
1159 { author empty$
1160   { "" }
1161   { author format.names.rev}
1162   if$
1163 }
1164

```

bookauthors.after

```

1165 FUNCTION {format.bookauthors.after}
1166 { bookauthor empty$
1167   { "" }
1168   { bookauthor format.names.rev}% always cuts to 4 persons
1169   if$
1170 }
1171

```

ormat.editors.after

```

1172 FUNCTION {format.editors.after}
1173 { editor empty$
1174   { "" }
1175   { bbl.edby "\ " * editor format.names.rev * }
1176   if$
1177 }
1178

```

ormat.chief.after

```

1179 FUNCTION {format.chief.after}
1180 { editor empty$
1181   { "" }
1182   { bbl.chief "\ " * editor format.names.rev * }

```

```

1183 if$
1184 }
1185

```

at.executor.after

```

1186 FUNCTION {format.executor.after}
1187 { author empty$
1188   { "" }
1189   { bbl.executor ": " * author format.names.rev * }
1190 if$
1191 }
1192

```

at.compiler.after

```

1193 FUNCTION {format.compiler.after}
1194 { compiler empty$
1195   { "" }
1196   { bbl.cmplr "\ " * compiler format.names.rev * }
1197 if$
1198 }
1199

```

**format.title** **Important note** Neither bibtex nor bibtex8 can handle unicoded text without troubles. In particular, bibtex8 fails to change case of a string if it contains Cyrillic letter. Therefore we avoid using `change.case$` when option `utf8` is in effect.

```

1200 FUNCTION {format.title}
1201 { title empty$
1202   { "" }
1203 <utf8> { title "t" change.case$ }
1204 <utf8> { title }
1205 if$
1206 }
1207

```

**format.month** New in version 1.2d. This macro reads month field and translate English names of months to Russian if current language is Russian.

```

1208 FUNCTION {format.month}
1209 { month empty$
1210   { "" }
1211   { curlanguage "russian" =
1212     { month "Jan." =
1213       <utf8> { "\CYRYA\cyrn\cyrv." }
1214       <utf8> { "ЯНВ." }
1215     { month "Feb." =
1216       <utf8> { "\CYRF\cyre\cyrv\cyrr." }
1217       <utf8> { "ФЕВ." }
1218     { month "Mar." =
1219       <utf8> { "\CYRM\cyra\cyrr\cyrt" }
1220       <utf8> { "МАРТ" }
1221     { month "Apr." =
1222       <utf8> { "\CYRA\cyrp\cyrr." }
1223       <utf8> { "АПР." }

```



`address.or.location` New function in version 1.2i. Returns `address` value if not empty; otherwise returns `location` value even if the latter is empty.

```
1272 %FUNCTION {address.or.location}
1273 %{
1274 %   address empty$
1275 %   { location empty$
1276 %     'skip$
1277 %   { location }
1278 %   if$ }
1279 %   { address }
1280 %   if$
1281 %}
1282 FUNCTION {address.or.location}
1283 {
1284   address empty$
1285   { location }
1286   { address }
1287   if$
1288 }
1289
```

`specialitycode.or.number` New function in version 1.2i. Returns `specialitycode` value if not empty; otherwise returns `number` value even if the latter is empty.

```
1290 FUNCTION {specialitycode.or.number}
1291 {
1292   specialitycode empty$
1293   { number }
1294   { specialitycode }
1295   if$
1296 }
1297
```

`institution.or.school` New function in version 1.2i. Returns `institution` value if not empty; otherwise returns `school` value even if the latter is empty.

```
1298 FUNCTION {institution.or.school}
1299 {
1300   institution empty$
1301   { school }
1302   { institution }
1303   if$
1304 }
```

`address.publisher`

```
1305 (*!strict)
1306 %FUNCTION {output.address.publisher}
1307 %{
1308 %   address empty$
1309 %     'skip$
1310 %   { address output
1311 %     publisher empty$
1312 %     'skip$
```

```

1313 %      { new.colon }
1314 %      if$
1315 %    }
1316 %  if$
1317 %  publisher output
1318 %}
1319 FUNCTION {output.address.publisher}
1320 {
1321   address empty$ location empty$ and
1322   'skip$
1323   { address.or.location output
1324     publisher empty$
1325     'skip$
1326     { new.colon }
1327     if$
1328   }
1329   if$
1330   publisher output
1331 }
1332 </!strict>
1333 <!*strict>
1334 FUNCTION {output.address.publisher}
1335 {
1336   address empty$
1337   {
1338     bbl.nnoaddress
1339     publisher empty$
1340     { "~: " * bbl.nopublisher * enclose.square.brackets }
1341     { enclose.square.brackets "~: " * publisher * }
1342     if$
1343   }
1344   {
1345     address output
1346     new.colon
1347     publisher empty$
1348     { bbl.nopublisher enclose.square.brackets }
1349     { publisher }
1350     if$
1351   }
1352   if$
1353   output
1354 }
1355
1356 </strict>
1357

```

ress.publisher.date Otput.address.publisher.date is used in old styles. New styles use output.address.publisher.

```

1358 <*(!(modern | strict))
1359 FUNCTION {output.address.publisher.date}
1360 {
1361   output.address.publisher

```

```

1362 format.date output
1363 }
1364 </!(modern | strict)>
1365

```

*output.bibitem*

```

1366 <!*natbib>
1367 FUNCTION {output.bibitem}
1368 { newline$
1369   "\bibitem" write$
1370   cite$ bracify write$
1371   newline$
1372   langid empty$
1373     { language empty$
1374       { "english" 'curlanguage := }
1375       { language 'curlanguage := }
1376       if$
1377     }
1378     { langid 'curlanguage := }
1379   if$
1380   "\selectlanguageifdefined" curlanguage bracify * write$
1381   newline$
1382   ""
1383   before.all 'output.state :=
1384 }
1385
1386 </!natbib>

```

*format.full.names* In case of *natbib* option, we need *make.full.names* to compose *output.bibitem*, and the  
*author.full* latter in its turn requires some more functions.

```

editor.full 1387 <!*natbib>
author.editor.full 1388 FUNCTION {format.full.names}
make.full.names 1389 { 's :=
output.bibitem 1390   language empty$
1391     { "english" 'curlanguage := }
1392     { language 'curlanguage := }
1393   if$
1394   #1 'nameptr :=
1395   s num.names$ 'numnames :=
1396   numnames 'namesleft :=
1397     { namesleft #0 > }
1398     { s nameptr
1399       "{vv~}{ll}" format.name$ 't :=
1400       nameptr #1 >
1401       {
1402         namesleft #1 >
1403         { ", " * t * }
1404         {
1405           numnames #2 >
1406           curlanguage "english" =
1407           and

```

```

1408         { "," * }
1409         'skip$
1410         if$
1411         t "others" =
1412         %t "~others" =
1413         %or
1414         %{ " et~al." * }
1415         { " " bbl.etal * * }
1416         %{ " and " * t * }
1417         { " " bbl.and " " * * * t * }
1418         if$
1419         }
1420         if$
1421         }
1422         't
1423         if$
1424         nameptr #1 + 'nameptr :=
1425         namesleft #1 - 'namesleft :=
1426         }
1427         while$
1428 }
1429
1430 FUNCTION {author.full}
1431 { author empty$
1432   { "" }
1433   { author format.full.names }
1434   if$
1435 }
1436
1437 FUNCTION {editor.full}
1438 { editor empty$
1439   { "" }
1440   { editor format.full.names }
1441   if$
1442 }
1443
1444 FUNCTION {author.editor.full}
1445 { author empty$
1446   { editor empty$
1447     { "" }
1448     { editor format.full.names }
1449     if$
1450   }
1451   { author format.full.names }
1452   if$
1453 }
1454
1455 FUNCTION {make.full.names}
1456 { type$ "book" =
1457   type$ "inbook" =

```

```

1458 or
1459   'author.editor.full
1460   { type$ "proceedings" =
1461     'editor.full
1462     'author.full
1463   if$
1464   }
1465 if$
1466 }
1467
1468 % =====
1469 FUNCTION {output.bibitem}
1470 { newline$
1471   "\bibitem[" write$
1472   label write$
1473   ")" make.full.names duplicate$ short.list =
1474     { pop$ }
1475     { * }
1476   if$
1477   "]" * write$
1478   cite$ write$
1479   "]" write$
1480 %% language empty$
1481 %%   { "english" 'curlanguage := }
1482 %%   { language 'curlanguage := }
1483 %% if$
1484 langid empty$
1485   { language empty$
1486     { "english" 'curlanguage := }
1487     { language 'curlanguage := }
1488     if$
1489   }
1490   { langid 'curlanguage := }
1491 if$
1492 "\selectlanguageifdefined" curlanguage braceify * write$
1493 newline$
1494 ""
1495 before.all 'output.state :=
1496 }
1497 % =====
1498 %FUNCTION {output.bibitem}
1499 %{ newline$
1500 %   "\bibitem" write$
1501 %% author.key.label
1502 %% year parenthesify *
1503 %% "; lbl:" label * *
1504 %% "; mfn:" make.full.names * *
1505 % label
1506 % make.full.names *
1507 % bracketise write$

```



```

1508 % cite$ bracyfy write$
1509 % newline$
1510 % language empty$
1511 % { "english" 'curlanguage := }
1512 % {language 'curlanguage := }
1513 % if$
1514 % "\selectlanguageifdefined" curlanguage bracyfy * write$
1515 % newline$
1516 % ""
1517 % before.all 'output.state :=
1518 %}
1519 % =====
1520 </natbib>
1521

```

n.dashify

```

1522 FUNCTION {n.dashify}
1523 { 't :=
1524   ""
1525   { t empty$ not }
1526   { t #1 #1 substring$ "-" =
1527     { t #1 #2 substring$ "--" = not
1528       { "--" *
1529         t #2 global.max$ substring$ 't :=
1530       }
1531       { { t #1 #1 substring$ "-" = }
1532         { "-" *
1533           t #2 global.max$ substring$ 't :=
1534         }
1535         while$
1536       }
1537     }
1538   }
1539   { t #1 #1 substring$ *
1540     t #2 global.max$ substring$ 't :=
1541   }
1542   if$
1543 }
1544 while$
1545 }
1546

```

word.in

```

1547 FUNCTION {word.in}
1548 { bbl.iin
1549   " " * }
1550

```

format.btitle

```

1551 FUNCTION {format.btitle}
1552 { title
1553 }

```

1554

.or.space.connect

```
1555 FUNCTION {tie.or.space.connect}
1556 { duplicate$ text.length$ #3 <
1557   { "~" }
1558   { " " }
1559   if$
1560   swap$ * *
1561 }
1562
```

tie.connect Declare function to insert unbreakable space before last word in stack.

```
1563 FUNCTION {tie.connect}
1564 {"~"
1565   swap$ * *
1566 }
1567
1568
```

either.or.chec

```
1569 FUNCTION {either.or.check}
1570 { empty$
1571   'pop$
1572   { "can't use both " swap$ * " fields in " * cite$ * warning$ }
1573   if$
1574 }
1575
```

format.bvolume

```
1576 FUNCTION {format.bvolume}
1577 { volume empty$
1578   { "" }
1579   { bbl.vvol volume tie.connect
1580     series empty$
1581     'skip$
1582     { bbl.of space.word * series emphasize * }
1583     if$
1584     "volume and number" number either.or.check
1585   }
1586   if$
1587 }
1588
```

format.number.series

```
1589 FUNCTION {format.number.series}
1590 { volume empty$
1591   { number empty$
1592     { series field.or.null }
1593     { series empty$
1594       { "there's a number but no series in " cite$ * warning$
1595         bbl.nnr }
1596     }
1597   }
1598 }
```

```

1596         {
1597             %new.dblslash
1598             new.sentence
1599             series
1600             bbl.nr
1601             tie.or.space.connect}
1602         if$
1603             number tie.or.space.connect
1604     }
1605     if$
1606     }
1607     { "" }
1608     if$
1609 }
1610

```

is.num

```

1611 FUNCTION {is.num}
1612 { chr.to.int$
1613   duplicate$ "0" chr.to.int$ < not
1614   swap$ "9" chr.to.int$ > not and
1615 }
1616

```

extract.num

```

1617 FUNCTION {extract.num}
1618 { duplicate$ 't :=
1619   "" 's :=
1620   { t empty$ not }
1621   { t #1 #1 substring$
1622     t #2 global.max$ substring$ 't :=
1623     duplicate$ is.num
1624     { s swap$ * 's := }
1625     { pop$ "" 't := }
1626   if$
1627 }
1628 while$
1629 s empty$
1630 'skip$
1631 { pop$ s }
1632 if$
1633 }
1634
1635 (*debug)

```

eng.ord

```

1636 FUNCTION {eng.ord}
1637 { duplicate$ "1" swap$ *
1638   #-2 #1 substring$ "1" =
1639   { bbl.th * }
1640   { duplicate$ #-1 #1 substring$
1641     duplicate$ "1" =

```

```

1642     { pop$ bbl.st * }
1643     { duplicate$ "2" =
1644       { pop$ bbl.nd * }
1645       { "3" =
1646         { bbl.rd * }
1647         { bbl.th * }
1648         if$
1649       }
1650       if$
1651     }
1652   if$
1653 }
1654 if$
1655 }
1656 </debug>
1657

```

convert.edition

```

1658 FUNCTION {convert.edition}
1659 { edition
1660 % edition extract.num "1" change.case$ 's :=
1661 % s "first" = s "1" = or
1662 %   { bbl.first 't := }
1663 %   { s "second" = s "2" = or
1664 %     { bbl.second 't := }
1665 %     { s "third" = s "3" = or
1666 %       { bbl.third 't := }
1667 %       { s "fourth" = s "4" = or
1668 %         { bbl.fourth 't := }
1669 %         { s "fifth" = s "5" = or
1670 %           { bbl.fifth 't := }
1671 %           { s #1 #1 substring$ is.num
1672 %             { s eng.ord 't := }
1673 %             { edition 't := }
1674 %           if$
1675 %         }
1676 %       if$
1677 %     }
1678 %   if$
1679 % }
1680 % if$
1681 % }
1682 % if$
1683 % }
1684 % if$
1685 % t
1686 }
1687

```

format.edition

```

1688 FUNCTION {format.edition}

```

```

1689 { edition empty$
1690   { "" }
1691   { output.state mid.sentence =
1692   (!utf8)   { convert.edition "l" change.case$ " " * bbl.edition * }
1693   (!utf8)   { convert.edition "t" change.case$ " " * bbl.edition * }
1694   (utf8)    { convert.edition " " * bbl.edition * }
1695   (utf8)    { convert.edition " " * bbl.edition * }
1696     if$
1697   }
1698   if$
1699 }
1700
1701 INTEGERS { multiresult }
1702

```

multi.page.check

```

1703 FUNCTION {multi.page.check}
1704 { 't :=
1705   #0 'multiresult :=
1706   { multiresult not
1707     t empty$ not
1708     and
1709   }
1710   { t #1 #1 substring$
1711     duplicate$ "-" =
1712     swap$ duplicate$ "," =
1713     swap$ "+" =
1714     or or
1715     { #1 'multiresult := }
1716     { t #2 global.max$ substring$ 't := }
1717   if$
1718   }
1719   while$
1720   multiresult
1721 }
1722

```

format.pages

```

1723 %%FUNCTION {format.pages}
1724 %%{ pages empty$
1725 %%  { "" }
1726 %%  { pages multi.page.check
1727 %%    { bbl.ppages pages n.dashify tie.connect }
1728 %%    { bbl.ppage pages tie.connect }
1729 %%    if$
1730 %%  }
1731 %% if$
1732 %%}
1733 FUNCTION {format.pages}
1734 { eid empty$
1735   {

```

```

1736     pages empty$
1737     { "" }
1738     { pages multi.page.check
1739         { bbl.ppages pages n.dashify tie.connect }
1740         { bbl.ppage pages tie.connect }
1741         if$
1742     }
1743     if$
1744 }
1745 { eid multi.page.check
1746     { bbl.ppages eid n.dashify tie.connect }
1747     { bbl.ppage eid tie.connect }
1748     if$
1749 }
1750 if$
1751 }
1752

```

format.pages.page

```

1753 %%FUNCTION {format.pages.page}
1754 %%{ pages empty$
1755 %%    { pagetotal empty$
1756 %%        { "" }
1757 %%        { pagetotal bbl.pages tie.connect }
1758 %%        if$}
1759 %%    { format.pages}
1760 %%    if$
1761 %%}
1762 FUNCTION {format.pages.page}
1763 { eid empty$
1764     { pages empty$
1765         { pagetotal empty$
1766             { "" }
1767             { pagetotal bbl.pages tie.connect }
1768             if$
1769         }
1770         { format.pages}
1771         if$
1772     }
1773     { format.pages }
1774     if$
1775 }
1776

```

mat.vol.num.pages

```

1777 FUNCTION {format.vol.num.pages}
1778 { volume field.or.null
1779     number empty$
1780     'skip$
1781     {
1782         ", no." number tie.or.space.connect *

```

```

1783     volume empty$
1784     { "there's a number but no volume in " cite$ * warning$ }
1785     'skip$
1786     if$
1787   }
1788   if$
1789   pages empty$
1790     'skip$
1791     { duplicate$ empty$
1792       { pop$ format.pages }
1793       { ": " * pages n.dashify * }
1794     if$
1795   }
1796   if$
1797 }
1798

```

format.volume

```

1799 FUNCTION {format.volume}
1800 { volume empty$
1801   { "" }
1802   { bbl.vvol volume tie.or.space.connect }
1803 if$
1804 }
1805

```

format.number

```

1806 FUNCTION {format.number}
1807 { number empty$
1808   { "" }
1809   { bbl.nr number tie.or.space.connect }
1810 if$
1811 }
1812
1813 (*debug)

```

mat.chapter.pages

```

1814 FUNCTION {format.chapter.pages}
1815 { chapter empty$
1816   'format.pages
1817   { type empty$
1818     { bbl.chapter }
1819     { type "1" change.case$ }
1820   if$
1821     chapter tie.or.space.connect
1822     pages empty$
1823     'skip$
1824     { ", " * format.pages * }
1825   if$
1826   }
1827 if$
1828 }

```

```
1829 </debug>
1830
```

empty.misc.check

```
1831 FUNCTION {empty.misc.check}
1832 { author empty$ title empty$ howpublished empty$
1833   month empty$ year empty$ note empty$
1834   and and and and and
1835   key empty$ not and
1836   { "all relevant fields are empty in " cite$ * warning$ }
1837   'skip$
1838   if$
1839 }
1840
```

ormat.thesis.type

```
1841 %FUNCTION {format.thesis.type}
1842 %{ type empty$
1843 %   'skip$
1844 %   { pop$
1845 %<!utf8>   bbl.thesis.type "t" change.case$
1846 %<utf8>   bbl.thesis.type
1847 %   }
1848 %   if$
1849 %}
1850 FUNCTION {format.thesis.type}
1851 { type empty$
1852   'skip$
1853   { pop$
1854     bbl.thesis.type
1855   }
1856   if$
1857 }
```

chrep.type.number Function to format report type and number.

```
1858 %FUNCTION {format.techrep.type.number}
1859 %{ type empty$
1860 %   { bbl.techreport }
1861 %   'type
1862 %   if$
1863 %   number empty$
1864 %<!utf8>   { "t" change.case$ }
1865 %<utf8>   { "" }
1866 %   { number tie.or.space.connect }
1867 %   if$
1868 %}
1869
1870 FUNCTION {format.techreport.type}
1871 { type empty$
1872   { bbl.techreport }
1873   'type
1874   if$
```



```
1875 }
1876
```

`author.before` Declare the function `author.before` to format list of authors in heading of a bibliographic record. If the number of authors is 4 or larger, some styles skip the list of authors in the beginning of the bibliographic record, while other styles always print that list. So, we need two version of `author.before`.

First version is used if `.bst` style is compiled without option `long`. It skips authors if their number is greater than or equal to 4 or if the author field is empty. Note that GOST requires for a long list of authors to be reduced. Hence this first version is preferable. Note also that `format.names` cuts list of names to 4 person at most in case if `modern` option is used. and `format.authors` just emphasizes `format.names`.

```
1877 <!*long>
1878 FUNCTION {author.before}
1879 {
1880   author empty$
1881   'skip$
1882   {author num.names$ #4 <
1883     {format.authors output
1884       new.sentence}
1885     'skip$
1886   if$}
1887 if$
1888 }
1889 </!long>
```

Second version is used if `.bst` style is compiled with the option `long`. It skips only if the author field is empty.

```
1890 <*long>
1891 FUNCTION {author.before}
1892 {
1893   author empty$
1894   'skip$
1895   { format.authors output
1896     new.sentence
1897   }
1898 if$
1899 }
1900 </long>
1901
```

`bookauthor.before` There are also 2 version of the function `bookauthor.before`. Not used anymore!

```
1902 %%<!*long>
1903 %%FUNCTION {bookauthor.before}
1904 %%{
1905 %%  bookauthor empty$
1906 %%  'skip$
1907 %%  {bookauthor num.names$ #4 <
1908 %%    {format.bookauthors output
1909 %%      new.sentence}
1910 %%    'skip$
```

```

1911 %%    if$}
1912 %%  if$
1913 %%}
1914 %%</!long>
1915 %%<*long>
1916 %%FUNCTION {bookauthor.before}
1917 %%{
1918 %%  bookauthor empty$
1919 %%    'skip$
1920 %%    {format.bookauthors output
1921 %%      new.sentence
1922 %%    }
1923 %%  if$
1924 %%}
1925 %%</long>
1926

```

`author.after`      Functions `author.after` and `bookauthor.after` also have by 2 versions. They are used to write authors list after the title followed by a slash. In modern styles, compiled with option `modern`, the list of authors is always cut to at most 4 persons. The cut is performed first by `format.names.rev`, which is called by `format.authors.after`. For old styles, `author.after` just outputs formatted string whereas for new style it skips the string if the number of authors exceeds 3 (and author list is not printed before the title).

```

1927 <!*modern>
1928 FUNCTION {author.after}
1929 {
1930   author empty$
1931   'skip$
1932   {format.authors.after output
1933     new.semicolon }
1934   if$
1935 }
1936 </!modern>
1937 <*modern>
1938 <*!long>
1939 FUNCTION {author.after}
1940 {
1941   author empty$
1942   'skip$
1943   {author num.names$ #3 >
1944     {format.authors.after output
1945       new.semicolon }
1946     'skip$
1947     if$}
1948   if$
1949 }
1950 </!long>
1951 <*long>
1952 FUNCTION {author.after} { }
1953 </long>

```

```
1954 </modern>
1955
```

`bookauthor.after` This function is used only in `inbook` entry. It always cuts list to 4 persons since `format.bookauthors.after` does that.

```
1956 FUNCTION {bookauthor.after}
1957 {
1958   bookauthor empty$
1959     'skip$
1960     {format.bookauthors.after output
1961       new.semicolon }
1962   if$
1963 }
1964
```

`organization.after`

```
1965 FUNCTION {editor.organization.after}
1966 {
1967   compiler empty$
1968     {}
1969     { format.compiler.after output
1970       new.semicolon
1971     }
1972   if$
1973   editor empty$
1974     {}
1975     { format.editors.after output
1976       new.semicolon
1977     }
1978   if$
1979   organization empty$
1980     {}
1981     {organization output
1982       new.semicolon
1983     }
1984   if$
1985 }
1986
```

`format.url`

```
1987 FUNCTION {format.url}
1988 { url empty$
1989   { "" }
1990   {
1991     (!(modern|strict)) "\BibUrl{ " url * "}" *
1992     <modern|strict> bbl.url ": \BibUrl{" * url * "}" *
1993     urldate empty$
1994     { "" }
1995     { " (" bbl.urldate * ": " * urldate * ")" * }
1996     if$ *
1997   }

```

```

1998  if$
1999  }
2000

```

output.url

```

2001 FUNCTION {output.url}
2002 {
2003  url empty$
2004    'skip$
2005    { format.url output }
2006  if$
2007 }
2008

```

format.annotate

```

2009 FUNCTION {format.annotate}
2010 { annotate empty$
2011   { "" }
2012 { after.sentence 'output.state :=
2013   "\BibAnnote{" annotate add.period$ * "}" *
2014 }
2015  if$
2016 }
2017

```

format.isbn Do we really need to provide electronic search for ISBN?

```

2018 FUNCTION {format.isbn}
2019 {
2020  isbn empty$
2021    { "" }
2022    { "ISBN:~\href{http://isbndb.com/search-all.html?kw=" isbn *
2023      }{" * isbn * "}" *
2024    }
2025  if$
2026 }
2027

```

add.doi The Digital Object Identifier (DOI) System is for identifying content objects in the digital environment. DOI names are assigned to any entity for use on digital networks. They are used to provide current information, including where they (or information about them) can be found on the Internet. Information about a digital object may change over time, including where to find it, but its DOI name will not change.

Function add.doi embraces last string in stack into hyperlink that links it to specified doi identifier at <http://dx.doi.org/> web-site.

```

2028 (*eprint)
2029 FUNCTION {add.doi}
2030 { duplicate$ empty$
2031   'skip$
2032   { doi empty$
2033     'skip$
2034     { "\href{http://dx.doi.org/" doi * "}" * swap$ * "}" * }

```

```

2035     if$
2036   }
2037   if$
2038 }
2039 </eprint>

```

If .bst style is compiled without eprint option, we just ignore doi field.

```

2040 <!*eprint>
2041 FUNCTION {add.doi} { }
2042 </!*eprint>
2043

```

**add.media** New in version 1.2. Adds media field if strict options is in effect. If the media field is empty add.media prints a value based on the type field. If the type is also empty add.media prints equivalent of the word text in current language.

```

2044 <!*strict>
2045 FUNCTION {add.media}
2046 { duplicate$ empty$
2047   'skip$
2048   { media empty$
2049     'skip$
2050     { " " * bbl.media enclose.square.brackets * }
2051     if$
2052   }
2053   if$
2054 }
2055 </!*strict>
2056 <*strict>
2057 FUNCTION {add.media}
2058 { duplicate$ empty$
2059   'skip$
2060   { media empty$
2061     { " " * bbl.media.text enclose.square.brackets * }
2062     { " " * bbl.media enclose.square.brackets * }
2063     if$
2064   }
2065   if$
2066 }
2067 </strict>
2068

```

## 2.5 Electronic Publishing Information

The biblatex package provides three fields for electronic publishing information: **eprint**, **eprinttype**, and **eprintclass**. The **eprint** field is a verbatim field similar to **doi** which holds the identifier of the item. The **eprinttype** field holds the resource name, i. e., the name of the site or electronic archive. Optional **eprintclass** field is intended for additional information specific to the resource indicated by the **eprinttype** field. This could be a section, a path, classification information, etc. If the **eprinttype** field is available, the standard styles will use it as a literal label. In the following example, they would print “Resource: identifier” rather than the generic “eprint: identifier”:

```
eprint = {identifier},
eprinttype = {Resource},
```

`format.eprint` The electronic identifier of an online publication. This is roughly comparable to a doi but specific to a certain archive, repository, service, or system. Also see `eprinttype` and `eprintclass`.

This function should use `url`. TO BE DONE YET.

```
2069 (*eprint)
2070 %FUNCTION {format.eprint}
2071 %{ eprint empty$
2072 %   { "" }
2073 %   { eprintclass empty$
2074 %     { " \href{http://arxiv.org/abs/" eprint * "}" * "{" * eprint * "}" * }
2075 %     { eprinttype empty$
2076 %       { " \href{http://arxiv.org/abs/" eprint * "}" *
2077 %         {" * eprintclass * "/" * eprint * "}" *
2078 %       }
2079 %       { " \href{http://arxiv.org/abs/" eprint * "}" *
2080 %         {" * eprinttype * ":" * eprintclass * "/" * eprint * "}" *
2081 %       }
2082 %     if$}
2083 %   if$}
2084 %if$}
2085
2086 %FUNCTION {format.eprint}
2087 %{ eprint empty$
2088 %   { "" }
2089 %   { eprinttype empty$
2090 %     { "" }
2091 %     { eprinttype "~: " *}
2092 %   if$
2093 %   eprintclass empty$
2094 %     { }
2095 %     { eprintclass * "/" *}
2096 %   if$
2097 %   eprint *
2098 % }
2099 % if$
2100 % url empty$
2101 %   { }
2102 %   { "\href{" url * "}-{ " * swap$ * "}" *}
2103 % if$
2104 %}
2105
2106 FUNCTION {format.eprint}
2107 { eprint empty$
2108   { "" }
2109   { eprinttype empty$
2110     { "" }
2111     { eprinttype "~: " *}
```

```

2112     if$
2113     eprintclass empty$
2114     { }
2115     { eprintclass * "/" *}
2116     if$
2117     url empty$
2118     { eprint * }
2119     { "\href{" * url * "}-{ " * eprint * "}" *}
2120     if$
2121     }
2122 if$
2123 }
2124
2125 FUNCTION {output.eprint.url}
2126 {
2127     eprint empty$
2128     { format.url output }
2129     { format.eprint output }
2130     if$
2131 }
2132
2133 </eprint>
2134
2135 <!*eprint>
2136 FUNCTION {output.eprint.url}
2137 {
2138     format.url output
2139 }
2140 </!*eprint>
2141

```

Functions added in v1.2f to format patent entry (thanks to Stanislav Kruchinin).

add.number

```

2142 FUNCTION {add.number}
2143 { duplicate$ empty$
2144     { "" }
2145     { bbl.nr swap$ tie.or.space.connect }
2146     if$
2147 }
2148

```

ormat.type.number

```

2149 FUNCTION {format.type.number}
2150 {
2151     type empty$
2152     { "" }
2153     {
2154         number empty$
2155         { "" }
2156         { type number tie.or.space.connect }
2157         if$

```

```

2158     }
2159   if$
2160 }
2161

```

format.requestdate

```

2162 FUNCTION {format.requestdate}
2163 { requestdate empty$
2164   { "" }
2165   { bbl.req requestdate tie.or.space.connect }
2166   if$
2167 }
2168

```

t.publicationdate

```

2169 FUNCTION {format.publicationdate}
2170 { publicationdate empty$
2171   { "" }
2172   { bbl.publ publicationdate tie.or.space.connect }
2173   if$
2174 }
2175

```

format.prioritydate

```

2176 FUNCTION {format.prioritydate}
2177 { prioritydate empty$
2178   { "" }
2179   { bbl.priority prioritydate tie.or.space.connect }
2180   if$
2181 }
2182

```

## 2.6 Entry types

Text below in this section is borrowed from biblatex manual. Not every field listed below is actually supported by GOST styles. So description below should be considered as a goal or a feature request.

The lists below indicate the fields supported by each entry type. Note that the mapping of fields to an entry type is ultimately at the discretion of the bibliography style. The lists below therefore serve two purposes. They indicate the fields supported by the standard styles which ship with this package and they also serve as a model for custom styles. Note that the required fields are not strictly required in all cases. The fields marked as optional are optional in a technical sense. Bibliographical formatting rules usually require more than just the required fields. The standard styles will generally not perform any formal validity checks, but custom styles may do so. Generic fields like abstract and annotation or label and shorthand are not included in the lists below because they are independent of the entry type.

### 2.6.1 Regular Types

**article** An article in a journal, magazine, newspaper, or other periodical which forms a self-contained unit with its own title. The title of the periodical is given in the journaltitle field. If the issue has its own



title in addition to the main title of the periodical, it goes in the issuetitle field. Note that editor and related fields refer to the journal while translator and related fields refer to the article.

Required fields: author, title, journaltitle, year/date.

Optional fields: translator, annotator, commentator, subtitle, titleaddon, editor, editora, editorb, editorc, journalsubtitle, issuetitle, issuesubtitle, language, origlanguage, series, volume, number, eid, issue, month, pages, version, note, issn, addendum, pubstate, doi, eprint, eprintclass, eprinttype, url, urldate.

```
2183 FUNCTION {article}
2184 {
2185   output.bibitem
2186   author.before
2187   (natbib)  author format.key output
2188   format.title add.media "title" output.check
2189   new.slash
2190   author.after
2191   new.dblslash
2192   journal emphasize add.doi "journal" output.check % new in v.2
2193   new.block
2194   format.date "year" output.check
2195   new.block
2196   format.volume output
2197   format.number output
2198   new.block
2199   format.pages.page output
2200   new.block
2201   note output
2202   new.sentence
2203   % format.url output
2204   output.eprint.url
2205   format.annotate output
2206   fin.entry
2207 }
2208
```

**book** A single-volume book with one or more authors where the authors share credit for the work as a whole. In biblatex, this entry type also covers the function of the @inbook type of traditional BibTeX.

Required fields: author, title, year/date.

Optional fields: editor, editora, editorb, editorc, translator, annotator, commentator, introduction, foreword, afterword, subtitle, titleaddon, maintitle, mainsubtitle, maintitleaddon, language, origlanguage, volume, part, edition, volumes, series, number, note, publisher, location, isbn, chapter, pages, pagetotal, addendum, pubstate, doi, eprint, eprintclass, eprinttype, url, urldate.

```
2209 FUNCTION {book}
2210 {
2211   output.bibitem
2212   author.before
2213   (natbib)  author format.key output
2214   format.btitle add.doi add.media "title" output.check
2215   new.slash
```

```

2216 author.after
2217 editor.organization.after
2218 new.sentence
2219 format.number.series output
2220 new.block
2221 format.edition output
2222 new.block
2223 output.address.publisher
2224 format.date "year" output.check
2225 new.block
2226 format.bvolume output
2227 new.block
2228 format.pages.page output
2229 new.block
2230 (eprint) format.isbn output
2231 (eprint) new.block
2232 note output
2233 new.sentence
2234 % format.url output
2235 output.eprint.url
2236 format.annotate output
2237 fin.entry
2238 }
2239

```

**booklet** A book-like work without a formal publisher or sponsoring institution. Use the field `howpublished` to supply publishing information in free format, if applicable. The field type may be useful as well.

Required fields: `author/editor`, `title`, `year/date`.

Optional fields: `subtitle`, `titleaddon`, `language`, `howpublished`, `type`, `note`, `location`, `chapter`, `pages`, `pagetotal`, `addendum`, `pubstate`, `doi`, `eprint`, `eprintclass`, `eprinttype`, `url`, `urldate`.

```

2240 FUNCTION {booklet}
2241 {
2242   output.bibitem
2243   author.before
2244 (natbib) author format.key output
2245   format.title add.doi add.media "title" output.check
2246   new.slash
2247   author.after
2248   editor.organization.after
2249   new.block
2250   howpublished output
2251   address.or.location output
2252   format.date "year" output.check
2253   new.block
2254   note output
2255   new.sentence
2256 % format.url output
2257   output.eprint.url
2258   format.annotate output
2259   fin.entry

```

2260 }  
2261

**inbook** A part of a book which forms a self-contained unit with its own title. Note that the profile of this entry type is different from standard BibTeX.

Required fields: author, title, booktitle, year/date.

Optional fields: bookauthor, editor, editora, editorb, editorc, translator, annotator, commentator, introduction, foreword, afterword, subtitle, titleaddon, maintitle, mainsubtitle, maintitleaddon, booksubtitle, booktitleaddon, language, origlanguage, volume, part, edition, volumes, series, number, note, publisher, location, isbn, chapter, pages, addendum, pubstate, doi, eprint, eprintclass, eprinttype, url, urldate.

```
2262 FUNCTION {inbook}
2263 {
2264   output.bibitem
2265   author.before
2266   (natbib)   author format.key output
2267   format.btitle add.doi add.media "title" output.check
2268   new.slash
2269   author.after
2270   new.dblslash
2271   %   bookauthor.before
2272   booktitle "booktitle" output.check
2273   new.slash
2274   bookauthor.after
2275   editor.organization.after
2276   new.block
2277   format.edition output
2278   new.block
2279   format.number.series output
2280   new.sentence
2281   output.address.publisher
2282   format.date "year" output.check
2283   new.block
2284   format.bvolume output
2285   new.block
2286   format.pages.page output
2287   new.block
2288   (eprint)   format.isbn output
2289   (eprint)   new.block
2290   note output
2291   new.sentence
2292   %   format.url output
2293   output.eprint.url
2294   format.annotate output
2295   fin.entry
2296 }
2297
```

**incollection** A contribution to a collection which forms a self-contained unit with a distinct author and title. The author refers to the title, the editor to the booktitle, i. e., the title of the collection.

Required fields: author, editor, title, booktitle, year/date.

Optional fields: editora, editorb, editorc, translator, annotator, commentator, introduction, foreword, afterword, subtitle, titleaddon, maintitle, mainsubtitle, maintitleaddon, booksubtitle, booktitleaddon, language, origlanguage, volume, part, edition, volumes, series, number, note, publisher, location, isbn, chapter, pages, addendum, pubstate, doi, eprint, eprintclass, eprinttype, url, urldate.

```
2298 FUNCTION {incollection}
2299 {
2300   output.bibitem
2301   author.before
2302   (natbib)   author format.key output
2303   new.sentence
2304   format.title add.doi add.media "title" output.check
2305   new.slash
2306   author.after
2307   new.dblslash
2308   booktitle "booktitle" output.check
2309   new.slash
2310   editor.organization.after
2311   new.block
2312   output.address.publisher
2313   format.date "year" output.check
2314   new.block
2315   format.bvolume output
2316   format.number.series output
2317   new.block
2318   format.pages.page output
2319   new.block
2320   note output
2321   new.sentence
2322   % format.url output
2323   output.eprint.url
2324   format.annotate output
2325   fin.entry
2326 }
2327
```

**proceedings**      A single-volume conference proceedings. This type is very similar to @collection. It supports an optional organization field which holds the sponsoring institution. The editor is omissible.

Required fields: editor, title, year/date.

Optional fields: subtitle, titleaddon, maintitle, mainsubtitle, maintitleaddon, eventtitle, eventdate, venue, language, volume, part, volumes, series, number, note, organization, publisher, location, month, isbn, chapter, pages, pagetotal, addendum, pubstate, doi, eprint, eprintclass, eprinttype, url, urldate.

```
2328 FUNCTION {proceedings}
2329 {
2330   output.bibitem
2331   (natbib)   editor format.key output
2332   format.btitle add.doi add.media "title" output.check
2333   new.slash
2334   editor.organization.after
```

```

2335 new.block
2336 format.bvolume output
2337 format.number.series output
2338 output.address.publisher
2339 format.date "year" output.check
2340 new.block
2341 note output
2342 new.sentence
2343 % format.url output
2344 output.eprint.url
2345 format.annotate output
2346 fin.entry
2347 }
2348

```

**inproceedings** An article in a conference proceedings. This type is similar to @incollection. It supports an optional organization field.

Required fields: author, editor, title, booktitle, year/date.

Optional fields: subtitle, titleaddon, maintitle, mainsubtitle, maintitleaddon, booksubtitle, booktitleaddon, eventtitle, eventdate, venue, language, volume, part, volumes, series, number, note, organization, publisher, location, month, isbn, chapter, pages, addendum, pubstate, doi, eprint, eprintclass, eprinttype, url, urldate.

```

2349 FUNCTION {inproceedings}
2350 { output.bibitem
2351   author.before
2352   (natbib) author format.key output
2353   new.sentence
2354   format.title add.doi add.media "title" output.check
2355   new.slash
2356   author.after
2357   new.dblslash
2358   booktitle "booktitle" output.check
2359   new.slash
2360   editor.organization.after
2361   new.block
2362   format.bvolume output
2363   format.number.series output
2364   new.block
2365 % address empty$
2366 %   { publisher output
2367 %     format.date "year" output.check
2368 %   }
2369 %   { address output.nonnull
2370 %     new.colon
2371 %     publisher output
2372 %     format.date "year" output.check
2373 %   }
2374 % if$
2375 output.address.publisher
2376 format.date "year" output.check

```

```

2377 new.block
2378 format.pages.page output
2379 new.block
2380 note output
2381 new.sentence
2382 % format.url output
2383 output.eprint.url
2384 format.annotate output
2385 fin.entry
2386 }
2387

```

**manual** Technical or other documentation, not necessarily in printed form. The author or editor is omissible.

Required fields: author/editor, title, year/date.

Optional fields: subtitle, titleaddon, language, edition, type, series, number, version, note, organization, publisher, location, isbn, chapter, pages, pagetotal, addendum, pubstate, doi, eprint, eprintclass, eprinttype, url, urldate.

```

2388 FUNCTION {manual}
2389 { output.bibitem
2390   author empty$
2391   { organization empty$
2392     'skip$
2393     { organization output.nonnull
2394       address output
2395     }
2396     if$
2397   }
2398   { format.authors output.nonnull }
2399   if$
2400 (natbib) author format.key output
2401 new.block
2402 format.btitle add.doi add.media "title" output.check
2403 author empty$
2404 { organization empty$
2405   {
2406     address new.block.checka
2407     address output
2408   }
2409   'skip$
2410   if$
2411   }
2412   {
2413     organization address new.block.checkb
2414     organization output
2415     address output
2416   }
2417   if$
2418   format.edition output
2419   format.date "year" output.check
2420 new.block

```

```

2421 note output
2422 new.sentence
2423 % format.url output
2424 output.eprint.url
2425 format.annotate output
2426 fin.entry
2427 }
2428

```

**patent** A patent or patent request. The number or record token is given in the number field. Use the **type** field to specify the type and the **location** field to indicate the scope of the patent, if different from the scope implied by the type. Note that the location field is treated as a key list with this entry type.

Required fields: author, title, number, year/date.

Optional fields: holder, subtitle, titleaddon, type, version, location, note, date, month, year, addendum, pubstate, doi, eprint, eprint class, eprint type, url, urldate.

```

2429 FUNCTION {patent}
2430 {
2431 output.bibitem
2432 % title output.nonnull
2433 title add.media output.nonnull
2434 % add.blank
2435 % media enclose.square.brackets output % S.Kruchinin's version
2436 new.colon
2437 format.type.number output
2438 add.blank
2439 address.or.location output
2440 new.colon
2441 ipc output
2442 new.slash
2443 format.authors.after "author" output.check
2444 add.blank
2445 authorcountry enclose.round.brackets output.nonnull
2446 (natbib) author format.key output
2447 new.semicolon
2448 holder output.nonnull
2449 new.semicolon
2450 credits output.nonnull
2451 new.block
2452 requestnumber add.number output
2453 new.semicolon
2454 format.requestdate output
2455 new.semicolon
2456 format.publicationdate output
2457 publication output
2458 new.semicolon
2459 format.prioritydate output
2460 prioritynumber output
2461 prioritycountry enclose.round.brackets output
2462 new.block

```

2463 `note output`  
 2464 `new.sentence`  
 2465 `output.url`  
 2466 `format.annotate output`  
 2467 `new.block`  
 2468 `pagetotal output`  
 2469 `fin.entry`  
 2470 `}`  
 2471

**misc** A fallback type for entries which do not fit into any other category. Use the field `howpublished` to supply publishing information in free format, if applicable. The field type may be useful as well. `author`, `editor`, and `year` are omissible.

Required fields: `author/editor`, `title`, `year/date`.

2472 `FUNCTION {misc}`  
 2473 `{ output.bibitem`  
 2474 `format.authors output`  
 2475 `(natbib) author format.key output`  
 2476 `title howpublished new.sentence.checkb`  
 2477 `format.title add.media output`  
 2478 `howpublished new.block.checka`  
 2479 `howpublished output`  
 2480 `new.block`  
 2481 `format.date "year" output.check`  
 2482 `new.block`  
 2483 `note output`  
 2484 `new.sentence`  
 2485 `% format.url output`  
 2486 `output.eprint.url`  
 2487 `format.annotate output`  
 2488 `fin.entry`  
 2489 `}`  
 2490

**unpublished** A work with an author and a title which has not been formally published, such as a manuscript or the script of a talk. Use the fields `howpublished` and `note` to supply additional information in free format, if applicable.

Required fields: `author`, `title`, `year/date`.

Optional fields: `subtitle`, `titleaddon`, `language`, `howpublished`, `note`, `location`, `isbn`, `date`, `month`, `year`, `addendum`, `pubstate`, `url`, `urldate`

2491 `FUNCTION {unpublished}`  
 2492 `{ output.bibitem`  
 2493 `author.before`  
 2494 `(natbib) author format.key output`  
 2495 `format.btitle "title" output.check`  
 2496 `new.slash`  
 2497 `author.after`  
 2498 `editor.organization.after`  
 2499 `new.block`  
 2500 `format.date "year" output.check`



```

2501 new.block
2502 note "note" output.check
2503 new.sentence
2504 % format.url output
2505 output.eprint.url
2506 format.annotate output
2507 fin.entry
2508 }
2509

```

**online** An online resource. Author, editor, and year are omissible. This entry type is intended for sources such as web sites which are intrinsically online resources. Note that all entry types support the url field. For example, when adding an article from an online journal, it may be preferable to use the `@article` type and its url field.

Required fields: author/editor, title, year/date, url.

Optional fields: subtitle, titleaddon, language, version, note, organization, date, month, year, addendum, pubstate, urldate.

```

2510 FUNCTION {online}
2511 { output.bibitem
2512   format.authors output
2513   (natbib) author format.key output
2514   title howpublished new.sentence.checkb
2515   format.title add.doi add.media "title" output.check
2516 %   howpublished new.block.checka
2517   howpublished new.dblslash.checka
2518 (!strict) howpublished output
2519 (strict) howpublished enclose.square.brackets output
2520   editor.organization.after
2521   new.sentence
2522   new.block
2523   output.address.publisher
2524   format.date output
2525   new.block
2526 %   format.url output
2527   output.eprint.url
2528   new.sentence
2529   note output
2530   format.annotate output
2531   fin.entry
2532 }
2533

```

**internet** New in version 2012.02.15.

```

www 2534 FUNCTION {internet} {online}
webpage 2535 FUNCTION {www} {online}
ielectronic 2536 FUNCTION {webpage} {online}
2537 FUNCTION {electronic} {online}

```

**thesis** New in version 2012.02.02.

A thesis written for an educational institution to satisfy the requirements for a degree. Use the type field to specify the type of thesis.

Required fields: author, title, type, institution, year/date.

Optional fields: subtitle, titleaddon, language, note, location, month, isbn, chapter, pages, pagetotal, addendum, pubstate, doi, eprint, eprintclass, eprinttype, url, urldate

```
2538 <!(modern | strict)>
2539 FUNCTION {thesis}
2540 { output.bibitem
2541   format.authors "author" output.check
2542 (natbib) author format.key output
2543   new.sentence
2544   format.btitle "title" output.check
2545   new.colon
2546   bbl.phdthesis format.thesis.type output.nonnull
2547   new.slash
2548   %institution "institution" output.check
2549   institution.or.school "institution/school" output.check
2550   new.block
2551   output.address.publisher.date
2552   new.block
2553   format.pages.page output
2554   note output
2555   new.sentence
2556   format.url output
2557   format.annotate output
2558   fin.entry
2559 }
2560 </!(modern | strict)>
2561 <!*modern | strict>
2562 FUNCTION {thesis}
2563 { output.bibitem
2564   format.authors "author" output.check
2565 (natbib) author format.key output
2566   new.sentence
2567   format.btitle add.doi add.media "title" output.check
2568   new.colon
2569   %%type "type" output.check
2570   bbl.phdthesis format.thesis.type output.nonnull
2571   new.colon
2572   %%number output % code of the speciality
2573   specialitycode.or.number output % code of the speciality, new in v.1.2i
2574   new.colon
2575   titleaddon output % date of defence and approvment; new in v.1.2i
2576   new.slash
2577   format.authors.after output
2578   new.semicolon
2579   %institution "institution" output.check
2580   institution.or.school "institution/school" output.check
2581   new.block
2582   output.address.publisher
2583   format.date "year" output.check
2584   new.block
```

```

2585 format.pages.page output
2586 new.block
2587 note output
2588 new.sentence
2589 % format.url output
2590 output.eprint.url
2591 format.annotate output
2592 fin.entry
2593 }
2594 </modern | strict>
2595

```

**report**

New in version 2012.02.02.

A technical report, research report, or white paper published by a university or some other institution. Use the type field to specify the type of report. The sponsoring institution goes in the institution field.

Required fields: author, title, type, institution, year/date.

Optional fields: subtitle, titleaddon, language, number, version, note, location, month, isrn, chapter, pages, pagetotal, addendum, pubstate, doi, eprint, eprintclass, eprinttype, url, urldate.

```

2596 %FUNCTION {report}
2597 %{
2598 % output.bibitem
2599 % author.before
2600 % new.sentence
2601 % format.title add.doi add.media "title" output.check
2602 % new.colon
2603 %% format.techrep.type.number output.nonnull
2604 % type "type" output.check
2605 % new.slash
2606 % author.after
2607 % editor.organization.after
2608 % new.block
2609 % address output
2610 % new.colon
2611 % institution "institution" output.check
2612 % format.date "year" output.check
2613 % new.block
2614 % note output
2615 % new.block % v.2
2616 % format.pages.page output % v.2
2617 % new.sentence
2618 %% format.url output
2619 % output.eprint.url
2620 % format.annotate output
2621 % fin.entry
2622 %}
2623 FUNCTION {report}
2624 {
2625 output.bibitem
2626 % author.before

```

```

2627 % new.sentence
2628 format.title add.doi add.media "title" output.check
2629 (natbib) title format.key output
2630 new.colon
2631 % format.techrep.type.number output.nonnull
2632 % type "type" output.check
2633 % format.report.type.number "type" output.check
2634 type "type" output.check
2635 new.colon
2636 number output
2637 new.slash
2638 %institution "institution" output.check
2639 institution.or.school "institution/school" output.check
2640 new.semicolon
2641 format.chief.after output % from editor field
2642 new.semicolon
2643 format.executor.after output % from author field
2644 new.block
2645 address.or.location output
2646 new.colon
2647 organization output
2648 format.date "year" output.check
2649 new.block % v.2
2650 format.pages.page output % v.2
2651 new.sentence % или new.block ?
2652 output.eprint.url
2653 new.block
2654 note output
2655 format.annotate output
2656 fin.entry
2657 }
2658

```

## 2.6.2 Type Aliases

The entry types listed in this section are provided for backwards compatibility with traditional BibTeX styles. These aliases are resolved by BibTeX as the data is exported. Bibliography styles will see the entry type the alias points to, not the alias name. All unknown entry types are generally exported as @misc.

**phdthesis** Similar to @thesis except that the type field is optional and defaults to the localized term ‘PhD thesis’. You may still use the type field to override that.

```

2659 (*!(modern | strict))
2660 FUNCTION {phdthesis}
2661 { output.bibitem
2662   format.authors "author" output.check
2663   new.sentence
2664   format.btitle "title" output.check
2665   new.colon
2666   bbl.phdthesis format.thesis.type output.nonnull
2667   new.slash

```

```

2668 %institution "institution" output.check
2669 institution.or.school "institution/school" output.check
2670 new.block
2671 output.address.publisher.date
2672 new.block
2673 format.pages.page output
2674 note output
2675 new.sentence
2676 format.url output
2677 format.annotate output
2678 fin.entry
2679 }
2680 <!(modern | strict)>
2681 <*modern | strict>
2682 FUNCTION {phdthesis}
2683 { output.bibitem
2684   format.authors "author" output.check
2685   natbib author format.key output
2686   new.sentence
2687   format.btitle add.doi add.media "title" output.check
2688   new.colon
2689   bbl.phdthesis format.thesis.type output.nonnull
2690   new.colon
2691   %%number output % code of the speciality
2692   specialitycode.or.number output % code of the speciality, new in v.1.2i
2693   new.colon
2694   titleaddon output % date of defence and approval; new in v.1.2i
2695   new.slash
2696   format.authors.after output
2697   new.semicolon
2698   %institution "institution" output.check
2699   institution.or.school "institution/school" output.check
2700   new.block
2701   output.address.publisher
2702   format.date "year" output.check
2703   new.block
2704   format.pages.page output
2705   new.block
2706   note output
2707   new.sentence
2708   % format.url output
2709   output.eprint.url
2710   format.annotate output
2711   fin.entry
2712 }
2713 </modern | strict>
2714

```

**mastersthesis** Similar to **@thesis** except that the type field is optional and defaults to the localized term ‘Master’s thesis’. You may still use the type field to override that.

```

2715 <*(modern | strict)>

```

```

2716 FUNCTION {mastersthesis}
2717 { output.bibitem
2718   format.authors "author" output.check
2719 (natbib) author format.key output
2720   new.sentence
2721   format.btitle "title" output.check
2722   new.colon
2723   bbl.mathesis format.thesis.type output.nonnull
2724   new.slash
2725   %institution "institution" output.check
2726   institution.or.school "institution/school" output.check
2727   new.block
2728   output.address.publisher.date
2729   new.block
2730   format.pages.page output
2731   note output
2732   new.sentence
2733   format.url output
2734   format.annotate output
2735   fin.entry
2736 }
2737 </!(modern | strict)>
2738 < *modern | strict >
2739 FUNCTION {mastersthesis}
2740 { output.bibitem
2741   format.authors "author" output.check
2742 (natbib) author format.key output
2743   new.sentence
2744   format.btitle add.doi add.media "title" output.check
2745   new.colon
2746   bbl.mathesis format.thesis.type output.nonnull
2747   new.colon
2748   %number output % code of the speciality
2749   specialitycode.or.number output % code of the speciality, new in v.1.2i
2750   new.colon
2751   titleaddon output % date of defence and approval; new in v.1.2i
2752   new.slash
2753   format.authors.after output
2754   new.semicolon
2755   %institution "institution" output.check
2756   institution.or.school "institution/school" output.check
2757   new.block
2758   output.address.publisher
2759   format.date "year" output.check
2760   new.block
2761   format.pages.page output
2762   new.block
2763   note output
2764   new.sentence
2765 % format.url output

```

```

2766 output.eprint.url
2767 format.annotate output
2768 fin.entry
2769 }
2770 </modern | strict>
2771

```

`docthesis` Similar to `@thesis` except that the `type` field is optional and defaults to the localized term ‘Doctor’s of sciences thesis’. You may still use the `type` field to override that.

```

2772 <!(modern | strict)>
2773 FUNCTION {docthesis}
2774 { output.bibitem
2775   format.authors "author" output.check
2776 <natbib> author format.key output
2777   new.sentence
2778   format.btitle "title" output.check
2779   new.colon
2780   bbl.docthesis format.thesis.type output.nonnull
2781   new.slash
2782   %institution "institution" output.check
2783   institution.or.school "institution/school" output.check
2784   new.block
2785   output.address.publisher.date
2786   new.block
2787   format.pages.page output
2788   note output
2789   new.sentence
2790   format.url output
2791   format.annotate output
2792   fin.entry
2793 }
2794 </!(modern | strict)>
2795 <*modern | strict>
2796 FUNCTION {docthesis}
2797 { output.bibitem
2798   format.authors "author" output.check
2799 <natbib> author format.key output
2800   new.sentence
2801   format.btitle add.doi add.media "title" output.check
2802   new.colon
2803   bbl.docthesis format.thesis.type output.nonnull
2804   new.colon
2805   %%number output % code of the speciality
2806   specialitycode.or.number output % code of the speciality, new in v.1.2i
2807   new.colon
2808   titleaddon output % date of defence and approval; new in v.1.2i
2809   new.slash
2810   format.authors.after output
2811   new.semicolon
2812   %institution "institution" output.check
2813   institution.or.school "institution/school" output.check

```

```

2814 new.block
2815 output.address.publisher
2816 format.date "year" output.check
2817 new.block
2818 format.pages.page output
2819 new.block
2820 note output
2821 new.sentence
2822 % format.url output
2823 output.eprint.url
2824 format.annotate output
2825 fin.entry
2826 }
2827 </modern | strict>
2828

```

conference

```

2829 FUNCTION {conference} { inproceedings }
2830

```

techreport      TechReport is similar to @report except that the type field is optional and defaults to the localized term 'technical report'. You may still use the type field to override that.

```

2831 %FUNCTION {techreport}
2832 %{
2833 % output.bibitem
2834 % author.before
2835 % new.sentence
2836 % format.title add.doi add.media "title" output.check
2837 % new.colon
2838 % format.techrep.type.number output.nonnull
2839 % new.slash
2840 % author.after
2841 % editor.organization.after
2842 % new.block
2843 % address output
2844 % new.colon
2845 % institution "institution" output.check
2846 % format.date "year" output.check
2847 % new.block
2848 % note output
2849 % new.block                           % v.2
2850 % format.pages.page output % v.2
2851 % new.sentence
2852 %% format.url output
2853 % output.eprint.url
2854 % format.annotate output
2855 % fin.entry
2856 %}
2857
2858 FUNCTION {techreport}
2859 {

```



```

2860 output.bibitem
2861 % author.before
2862 % new.sentence
2863 format.title add.doi add.media "title" output.check
2864 (natbib) title format.key output
2865 new.colon
2866 % format.techrep.type.number output.nonnull
2867 % type "type" output.check
2868 % format.report.type.number "type" output.check
2869 % type output
2870 format.techreport.type output
2871 new.colon
2872 number output
2873 new.slash
2874 %institution "institution" output.check
2875 institution.or.school "institution/school" output.check
2876 new.semicolon
2877 format.chief.after output % from editor field
2878 new.semicolon
2879 format.executor.after output % from author field
2880 new.block
2881 address.or.location output
2882 new.colon
2883 organization output
2884 format.date "year" output.check
2885 new.block % v.2
2886 format.pages.page output % v.2
2887 new.sentence % или new.block ?
2888 output.eprint.url
2889 new.block
2890 note output
2891 format.annotate output
2892 fin.entry
2893 }
2894

```

default.type

```

2895 FUNCTION {default.type} { misc }
2896

```

## 2.7 Month Abbreviations

Borrowed from merlin.mbs of package custom-bib. This is done for backward compatibility with standard .bst styles which are designed for English. The string in the definition of any month macro must coincide with that used in format.month function in the above.

```

2897 MACRO {jan} {"Jan."}
2898 MACRO {feb} {"Feb."}
2899 MACRO {mar} {"Mar."}
2900 MACRO {apr} {"Apr."}
2901 MACRO {may} {"May"}

```

2902 MACRO {jun} {"Jun."}  
 2903 MACRO {jul} {"Jul."}  
 2904 MACRO {aug} {"Aug."}  
 2905 MACRO {sep} {"Sep."}  
 2906 MACRO {oct} {"Oct."}  
 2907 MACRO {nov} {"Nov."}  
 2908 MACRO {dec} {"Dec."}

## 2.8 Journal Abbreviations

### 2.8.1 Physics and astronomy

Borrowed from physjour.mbs of package custom-bib.

2909 MACRO {aa}{"Astron. \& Astrophys."}  
 2910 MACRO {aasup}{"Astron. \& Astrophys. Suppl. Ser."}  
 2911 MACRO {aj} {"Astron. J."}  
 2912 MACRO {aph} {"Acta Phys."}  
 2913 MACRO {advp} {"Adv. Phys."}  
 2914 MACRO {ajp} {"Amer. J. Phys."}  
 2915 MACRO {ajm} {"Amer. J. Math."}  
 2916 MACRO {amsci} {"Amer. Sci."}  
 2917 MACRO {anofd} {"Ann. Fluid Dyn."}  
 2918 MACRO {am} {"Ann. Math."}  
 2919 MACRO {ap} {"Ann. Phys. (NY)"}

2920 MACRO {adp} {"Ann. Phys. (Leipzig)"}

2921 MACRO {ao} {"Appl. Opt."}

2922 MACRO {apl} {"Appl. Phys. Lett."}

2923 MACRO {app} {"Astroparticle Phys."}

2924 MACRO {apj} {"Astrophys. J."}

2925 MACRO {apjsup} {"Astrophys. J. Suppl."}

2926 MACRO {apss} {"Astrophys. Space Sci."}

2927 MACRO {araa} {"Ann. Rev. Astron. Astrophys."}

2928 MACRO {baas} {"Bull. Amer. Astron. Soc."}

2929 MACRO {baps} {"Bull. Amer. Phys. Soc."}

2930 MACRO {cmp} {"Comm. Math. Phys."}

2931 MACRO {cpam} {"Commun. Pure Appl. Math."}

2932 MACRO {cppcf} {"Comm. Plasma Phys. \& Controlled Fusion"}

2933 MACRO {cpc} {"Comp. Phys. Comm."}

2934 MACRO {cqg} {"Class. Quant. Grav."}

2935 MACRO {cra} {"C. R. Acad. Sci. A"}

2936 MACRO {fed} {"Fusion Eng. \& Design"}

2937 MACRO {ft} {"Fusion Tech."}

2938 MACRO {grg} {"Gen. Relativ. Gravit."}

2939 MACRO {ieeens} {"IEEE Trans. Nucl. Sci."}

2940 MACRO {ieeeps} {"IEEE Trans. Plasma Sci."}

2941 MACRO {ijimw} {"Interntl. J. Infrared \& Millimeter Waves"}

2942 MACRO {ip} {"Infrared Phys."}

2943 MACRO {irp} {"Infrared Phys."}

2944 MACRO {jap} {"J. Appl. Phys."}

2945 MACRO {jasa} {"J. Acoust. Soc. America"}

2946 MACRO {jcp} {"J. Comp. Phys."}  
2947 MACRO {jchp} {"J. Chem. Phys."}  
2948 MACRO {jetp} {"Sov. Phys.--JETP"}  
2949 MACRO {jfe} {"J. Fusion Energy"}  
2950 MACRO {jfm} {"J. Fluid Mech."}  
2951 MACRO {jmp} {"J. Math. Phys."}  
2952 MACRO {jne} {"J. Nucl. Energy"}  
2953 MACRO {jnec} {"J. Nucl. Energy, C: Plasma Phys., Accelerators, Thermonucl. Res."}  
2954 MACRO {jnm} {"J. Nucl. Mat."}  
2955 MACRO {jpc} {"J. Phys. Chem."}  
2956 MACRO {jpp} {"J. Plasma Phys."}  
2957 MACRO {jpsj} {"J. Phys. Soc. Japan"}  
2958 MACRO {jsi} {"J. Sci. Instrum."}  
2959 MACRO {jvst} {"J. Vac. Sci. & Tech."}  
2960 MACRO {nat} {"Nature"}  
2961 MACRO {nature} {"Nature"}  
2962 MACRO {nedf} {"Nucl. Eng. \& Design/Fusion"}  
2963 MACRO {nf} {"Nucl. Fusion"}  
2964 MACRO {nim} {"Nucl. Inst. \& Meth."}  
2965 MACRO {nimpr} {"Nucl. Inst. \& Meth. in Phys. Res."}  
2966 MACRO {np} {"Nucl. Phys."}  
2967 MACRO {npb} {"Nucl. Phys. B"}  
2968 MACRO {nt/f} {"Nucl. Tech./Fusion"}  
2969 MACRO {npbpc} {"Nucl. Phys. B (Proc. Suppl.)"}  
2970 MACRO {inc} {"Nuovo Cimento"}  
2971 MACRO {nc} {"Nuovo Cimento"}  
2972 MACRO {pf} {"Phys. Fluids"}  
2973 MACRO {pfa} {"Phys. Fluids A: Fluid Dyn."}  
2974 MACRO {pfb} {"Phys. Fluids B: Plasma Phys."}  
2975 MACRO {pl} {"Phys. Lett."}  
2976 MACRO {pla} {"Phys. Lett. A"}  
2977 MACRO {plb} {"Phys. Lett. B"}  
2978 MACRO {prep} {"Phys. Rep."}  
2979 MACRO {pnas} {"Proc. Nat. Acad. Sci. USA"}  
2980 MACRO {pp} {"Phys. Plasmas"}  
2981 MACRO {pop} {"Phys. Plasmas"}  
2982 MACRO {ppcf} {"Plasma Phys. \& Controlled Fusion"}  
2983 MACRO {phitrs1} {"Philos. Trans. Roy. Soc. London"}  
2984 MACRO {pr1} {"Phys. Rev. Lett."}  
2985 MACRO {pr} {"Phys. Rev."}  
2986 MACRO {physrev} {"Phys. Rev."}  
2987 MACRO {pra} {"Phys. Rev. A"}  
2988 MACRO {prb} {"Phys. Rev. B"}  
2989 MACRO {prc} {"Phys. Rev. C"}  
2990 MACRO {prd} {"Phys. Rev. D"}  
2991 MACRO {pre} {"Phys. Rev. E"}  
2992 MACRO {ps} {"Phys. Scripta"}  
2993 MACRO {procrs1} {"Proc. Roy. Soc. London"}  
2994 MACRO {rmp} {"Rev. Mod. Phys."}  
2995 MACRO {rsi} {"Rev. Sci. Inst."}

2996 MACRO {science} {"Science"}  
 2997 MACRO {sciam} {"Sci. Am."}  
 2998 MACRO {sam} {"Stud. Appl. Math."}  
 2999 MACRO {st} {"Sky and Telesc."}

## 2.8.2 Supplementary Journal Names

Borrowed from supjour.mbs of package custom-bib.

3000 MACRO {cjp} {"Czech. J. Phys."}  
 3001 MACRO {el} {"Europhys. Lett."}  
 3002 MACRO {en} {"Europhys. News"}  
 3003 MACRO {fujitsustj} {"FUJITSU Sci. Tech. J."}  
 3004 MACRO {ieeed} {"IEEE Trans. Electron Devices"}  
 3005 MACRO {ieeedim} {"IEEE Trans. Instrum. Meas."}  
 3006 MACRO {ieeedjqe} {"IEEE J. Quantum Electron."}  
 3007 MACRO {ieeedm} {"IEEE Trans. Magn."}  
 3008 MACRO {ieeedptl} {"IEEE Photonic Technol. Lett."}  
 3009 MACRO {ieeeduffc} {"IEEE Trans. Ultrason., Ferroelect., Freq. Cont."}  
 3010 MACRO {jem} {"J. Electron. Mater."}  
 3011 MACRO {jes} {"J. Electrochem. Soc."}  
 3012 MACRO {jetplett} {"JETP Lett."}  
 3013 MACRO {jjap} {"Japan. J. Appl. Phys."}  
 3014 MACRO {jpha} {"J. Phys. A: Math. Gen."}  
 3015 MACRO {jphb} {"J. Phys. B: At. Mol. Opt. Phys."}  
 3016 MACRO {jphbold} {"J. Phys. B: At. Mol. Phys."}  
 3017 MACRO {jphc} {"J. Phys.: Condens. Matter"}  
 3018 MACRO {jphcold} {"J. Phys. C: Solid State Phys."}  
 3019 MACRO {jphd} {"J. Phys. D: Appl. Phys."}  
 3020 MACRO {jvsta} {"J. Vac. Sci. Technol. A"}  
 3021 MACRO {jvstb} {"J. Vac. Sci. Technol. B"}  
 3022 MACRO {me} {"Microelectron. Eng."}  
 3023 MACRO {necrd} {"NEC Res. {\&} Develop."}  
 3024 MACRO {pa} {"Physica A"}  
 3025 MACRO {pb} {"Physica B"}  
 3026 MACRO {pc} {"Physica C"}  
 3027 MACRO {pd} {"Physica D"}  
 3028 MACRO {procieeed} {"Proc. IEEE"}  
 3029 MACRO {procspie} {"Proc. SPIE"}  
 3030 MACRO {pssa} {"Phys. Stat. Sol. A"}  
 3031 MACRO {pssb} {"Phys. Stat. Sol. B"}  
 3032 MACRO {rpp} {"Rep. Progr. Phys."}  
 3033 MACRO {sm} {"Synthet. Metal"}  
 3034 MACRO {sost} {"Solid State Technol."}  
 3035 MACRO {ss} {"Surf. Sci."}  
 3036 MACRO {ssc} {"Solid State Commun."}  
 3037 MACRO {sst} {"Semicond. Sci. Technol."}  
 3038 MACRO {suplatt} {"Superlatt. Microstr."}  
 3039 MACRO {sust} {"Supercond. Sci. Technol."}  
 3040 MACRO {znat} {"Z. Naturforsch."}

### 2.8.3 Optics

Borrowed from photjour.mbs.

```
3041 MACRO {appopt} {"Appl. Opt."}
3042 MACRO {bell} {"Bell Syst. Tech. J."}
3043 MACRO {ell} {"Electron. Lett."}
3044 MACRO {jasp} {"J. Appl. Spectr."}
3045 MACRO {jqe} {"IEEE J. Quantum Electron."}
3046 MACRO {jlwt} {"J. Lightwave Technol."}
3047 MACRO {jmo} {"J. Mod. Opt."}
3048 MACRO {josa} {"J. Opt. Soc. America"}
3049 MACRO {josaa} {"J. Opt. Soc. Amer.~A"}
3050 MACRO {josab} {"J. Opt. Soc. Amer.~B"}
3051 MACRO {jdp} {"J. Phys. (Paris)"}
3052 MACRO {oc} {"Opt. Commun."}
3053 MACRO {ol} {"Opt. Lett."}
3054 MACRO {os} {"Opt. Spectrosc."}
3055 MACRO {phtl} {"IEEE Photon. Technol. Lett."}
3056 MACRO {pspie} {"Proc. Soc. Photo-Opt. Instrum. Eng."}
3057 MACRO {vr} {"Vision Res."}
3058 MACRO {zph} {"Z. f. Physik"}
3059 MACRO {zphb} {"Z. f. Physik~B"}
3060 MACRO {zphd} {"Z. f. Physik~D"}
```

### 2.8.4 Physics of condensed Matter

```
3061 MACRO {sse} {"Solid-State Electron."}
3062 MACRO {pss} {"Phys. Sol. State"}
3063 MACRO {nl} {"Nano Lett."}
```

### 2.8.5 Soviet and Russian journals

To be extended.

```
3064 MACRO {sjpp} {"Sov. J. Plasma Phys."}
3065 MACRO {spd} {"Sov. Phys.--Doklady"}
3066 MACRO {sptp} {"Sov. Phys.--Tech. Phys."}
3067 MACRO {spu} {"Sov. Phys.--Uspekhi"}
3068  $\langle$ utf8 $\rangle$ MACRO {ufn} {"\CYRU\CYRF\CYRN"}
3069  $\langle$ utf8 $\rangle$ MACRO {ufn} {"\Y\Phi H"}
3070 MACRO {pu} {"Phys.--Uspekhi"}
3071 MACRO {sjot} {"Sov. J. Opt. Technol."}
3072 MACRO {sjqe} {"Sov. J. Quantum Electron."}
3073 MACRO {sleb} {"Sov. Phys.--Leb. Inst. Rep."}
3074 MACRO {stph} {"Sov. Phys.--Techn. Phys."}
3075 MACRO {stph1} {"Sov. Techn. Phys. Lett."}
```

## 2.9 Main cycle

```
3076
3077 READ
3078
```

## 2.10 Sorting

Next chunk of code governs sorting reference list by authors' names and titles.

```
3079 (*sort | natbib)
3080
```

sortify

```
3081 FUNCTION {sortify}
3082 { purify$
3083 (!utf8) "1" change.case$
3084 }
3085 (/sort | natbib)
3086
```

sort.format.names

```
3087 (*sort)
3088 %% =====
3089 %% This version from old Gost package
3090 %%<!*natbib>
3091 FUNCTION {sort.format.names}
3092 { 's :=
3093   #1 'nameptr :=
3094   ""
3095   s num.names$ 'numnames :=
3096   numnames 'namesleft :=
3097     { namesleft #0 > }
3098     { nameptr #1 >
3099       { " " * }
3100       'skip$
3101       if$
3102       s nameptr
3103       "{vv{ } }{ll{ }}{ f{ }}{ jj{ }}"
3104       format.name$ 't :=
3105       nameptr numnames = t "others" = and
3106         { "et al" * }
3107         %{ bbl.etal * }
3108         { t sortify * }
3109       if$
3110       nameptr #1 + 'nameptr :=
3111       namesleft #1 - 'namesleft :=
3112     }
3113   while$
3114 }
3115 %%</!natbib>
3116 %% This version from plainnat.bst
3117 %% It ignores second and subsequent authors but include year.
3118 %%<*natbib>
3119 %FUNCTION {sort.format.names}
3120 %{ 's :=
3121 % #1 'nameptr :=
3122 % ""
```

```

3123 % s num.names$ 'numnames :=
3124 % numnames 'namesleft :=
3125 % { namesleft #0 > }
3126 % {
3127 %     s nameptr "{vv{ } }{ll{ }}{ ff{ }}{ jj{ }}" format.name$ 't :=
3128 %     nameptr #1 >
3129 %     {
3130 %         " " *
3131 %         namesleft #1 = t "others" = and
3132 %         { "zzzzz" * }
3133 %         { numnames #2 > nameptr #2 = and
3134 %             { "zz" * year field.or.null * " " * }
3135 %             'skip$
3136 %             if$
3137 %             t sortify *
3138 %         }
3139 %     }
3140 %     if$
3141 %     { t sortify * }
3142 %     if$
3143 %     nameptr #1 + 'nameptr :=
3144 %     namesleft #1 - 'namesleft :=
3145 % }
3146 % while$
3147 %}
3148 %%</natbib>
3149 %% =====
3150

```

sort.format.title

```

3151 FUNCTION {sort.format.title}
3152 { 't :=
3153   "A " #2
3154   "An " #3
3155   "The " #4 t chop.word % Removes "The " if any
3156   chop.word % Removes "An " if any
3157   chop.word % Removes "A " if any
3158   sortify
3159   #1 global.max$ substring$
3160 }
3161

```

author.sort

```

3162 %% =====
3163 %% This version from old gost package.
3164 %%
3165 (&!natbib)
3166 FUNCTION {author.sort}
3167 { author empty$
3168   { key empty$
3169     { "to sort, need author or key in " cite$ * warning$

```

```

3170     ""
3171     }
3172     { key sortify }
3173     if$
3174     }
3175     {
3176     author num.names$ #4 <
3177     {author sort.format.names }
3178     {title sort.format.title}
3179     if$
3180     }
3181     if$
3182 }
3183 </!natbib)
3184 %% This version from plainnat.bst
3185 (*natbib)
3186 FUNCTION {author.sort}
3187 { author empty$
3188   { key empty$
3189     { "to sort, need author or key in " cite$ * warning$
3190       ""
3191     }
3192     { key sortify }
3193     if$
3194   }
3195   { author sort.format.names }
3196   if$
3197 }
3198 </natbib)
3199 %% =====
3200

```

author.title.sort

```

3201 <(*!natbib|natbib)
3202 FUNCTION {author.title.sort}
3203 { author empty$
3204   { title empty$
3205     { key empty$
3206       { "to sort, need author, title, or key in " cite$ * warning$
3207         ""
3208       }
3209       { key sortify }
3210       if$
3211     }
3212     { title sort.format.title }
3213     if$
3214   }
3215   {
3216     author num.names$ #4 <
3217     {author sort.format.names }
3218     {title sort.format.title}

```



```

3219     if$
3220   }
3221   if$
3222 }
3223 </!natbib|natbib>
3224 < *natbib|natbib>
3225 FUNCTION {author.editor.sort}
3226 { author empty$
3227   { editor empty$
3228     { key empty$
3229       { "to sort, need author, editor, or key in " cite$ * warning$
3230         ""
3231       }
3232       { key sortify }
3233     }
3234   }
3235   { editor sort.format.names }
3236 } if$
3237 }
3238 { author sort.format.names }
3239 if$
3240 }
3241
3242 FUNCTION {author.organization.sort}
3243 { author empty$
3244   { organization empty$
3245     { key empty$
3246       { "to sort, need author, organization, or key in " cite$ * warning$
3247         ""
3248       }
3249       { key sortify }
3250     }
3251   }
3252   { "The " #4 organization chop.word sortify }
3253 } if$
3254 }
3255 { author sort.format.names }
3256 if$
3257 }
3258
3259 FUNCTION {editor.organization.sort}
3260 { editor empty$
3261   { organization empty$
3262     { key empty$
3263       { "to sort, need editor, organization, or key in " cite$ * warning$
3264         ""
3265       }
3266       { key sortify }
3267     }
3268   }

```

```

3269     { "The " #4 organization chop.word sortify }
3270     if$
3271   }
3272   { editor sort.format.names }
3273   if$
3274 }
3275 </natbib | natbib>
3276
3277

```

presort Function to compute sort.key\$. What is the space string "\_\_\_" for?

```

3278 <!*natbib>
3279 FUNCTION {presort}%#1
3280 {
3281   author.title.sort
3282   "   "
3283   *
3284   year field.or.null sortify
3285   *
3286   "   "
3287   *
3288   title field.or.null
3289   sort.format.title
3290   *
3291   #1 entry.max$ substring$
3292   'sort.key$ :=
3293 }
3294 </!natbib>
3295 <!*natbib>
3296 FUNCTION {presort}%#2
3297 { calc.label
3298   label sortify
3299   %author.title.sort
3300   "   "
3301   *
3302   % ===== plainnat.bst =====
3303   % type$ "book" =
3304   % type$ "inbook" =
3305   % or
3306   % 'author.editor.sort
3307   % { type$ "proceedings" =
3308   %   'editor.organization.sort
3309   %   { type$ "manual" =
3310   %     'author.organization.sort
3311   %     'author.sort
3312   %   }
3313   % }
3314   % if$
3315   % }
3316   % if$
3317   author.title.sort

```

```

3318 " "
3319 *
3320 year field.or.null sortify
3321 *
3322 " "
3323 *
3324 %cite$
3325 title field.or.null sort.format.title
3326 *
3327 #1 entry.max$ substring$
3328 'sort.label :=
3329 sort.label *
3330 % =====
3331 #1 entry.max$ substring$
3332 'sort.key$ :=
3333 }
3334 </natbib>
3335 </sort>
3336
3337 <!*sort>
3338 <!*natbib>
3339 INTEGERS { seq.num }
3340
3341 FUNCTION {init.seq}
3342 { #0 'seq.num :=}
3343
3344 EXECUTE {init.seq}
3345
3346 FUNCTION {int.to.fix}
3347 { "00000000" swap$ int.to.str$ *
3348 #-1 #10 substring$
3349 }
3350
3351 FUNCTION {presort}%#3
3352 {
3353 calc.label % computes label
3354 label sortify % initiates sort.label
3355 " "
3356 *
3357 seq.num #1 + 'seq.num := % advance seq.num
3358 seq.num int.to.fix % prepend seq.num with 0s
3359 'sort.label := % set sort.label to seq.num
3360 sort.label * % append seq.num to label
3361 #1 entry.max$ substring$ % cut if too long
3362 'sort.key$ := % set sort.key$
3363 }
3364 </natbib>
3365 </!sort>
3366
3367 <*sort | natbib>

```

```

3368 ITERATE {presort}
3369
3370 SORT
3371
3372 </sort | natbib>
3373

```

## 2.11 Bibliography list

We need to find longest label to put in into the argument of the `thebibliography` environment. In case of `natbib` options we also need to compute extra suffix for the `year` field if there two or more entries for given label (=author/editor/organization) in that year.

Declare global (external) strings used in calculation of the longest label.

```

3374 <!natbib>STRINGS { longest.label }
3375 <natbib>STRINGS { longest.label last.label next.extra }
3376
3377 <!natbib>INTEGERS { number.label longest.label.width }
3378 <natbib>INTEGERS { number.label longest.label.width last.extra.num }
3379

```

`initialize.longest.label` Initialize those string.

```

3380 <!*natbib>
3381 FUNCTION {initialize.longest.label}
3382 { "" 'longest.label :=
3383   #1 'number.label :=
3384   #0 'longest.label.width :=
3385 }
3386 </!*natbib>
3387 <*natbib>
3388 FUNCTION {initialize.longest.label}
3389 { "" 'longest.label :=
3390   #0 int.to.chr$ 'last.label :=
3391   "" 'next.extra :=
3392   #0 'longest.label.width :=
3393   #0 'last.extra.num :=
3394   #0 'number.label :=
3395 }
3396 </natbib>
3397
3398 EXECUTE {initialize.longest.label}
3399

```

`iterate.longest.label` Iterate though the list of entries to compute label.

```

3400 <!*natbib>
3401 FUNCTION {forward.pass}
3402 { number.label int.to.str$ 'label :=
3403   number.label #1 + 'number.label :=
3404   label width$ longest.label.width >
3405   { label 'longest.label :=
3406     label width$ 'longest.label.width :=

```

```

3407     }
3408     'skip$
3409     if$
3410 }
3411  $\langle$ /!natbib $\rangle$ 
3412  $\langle$ *natbib $\rangle$ 
3413 FUNCTION {forward.pass}
3414 { last.label label =
3415   { last.extra.num #1 + 'last.extra.num :=
3416     last.extra.num int.to.chr$ 'extra.label :=
3417   }
3418   { "a" chr.to.int$ 'last.extra.num :=
3419     "" 'extra.label :=
3420     label 'last.label :=
3421   }
3422   if$
3423   number.label #1 + 'number.label :=
3424 }
3425  $\langle$ /natbib $\rangle$ 
3426
3427 ITERATE {forward.pass}
3428

```

reverse.pass Natbib styles require reverse iteration over all entries.

```

3429  $\langle$ *natbib $\rangle$ 
3430 FUNCTION {reverse.pass}
3431 { next.extra "b" =
3432   { "a" 'extra.label := }
3433   'skip$
3434   if$
3435   extra.label 'next.extra :=
3436   extra.label
3437   duplicate$ empty$
3438   'skip$
3439   { "{\natexlab{" swap$ * "}} * }
3440   if$
3441   'extra.label :=
3442   label extra.label * 'label :=
3443 }
3444
3445 REVERSE {reverse.pass}
3446
3447 FUNCTION {bib.sort.order}
3448 { sort.label 'sort.key$ :=
3449 }
3450
3451 ITERATE {bib.sort.order}
3452
3453 SORT
3454  $\langle$ /natbib $\rangle$ 
3455

```

`begin.bib` Within the `thebibliography` environment we define few formatting macros for user to customize how the reference list is formatted.

```

3456 FUNCTION {begin.bib}
3457 { "\begin{thebibliography}{ longest.label * }" * write$ newline$
3458 "\def\selectlanguageifdefined#1{" write$ newline$
3459 "\expandafter\ifx\csname date#1\endcsname\relax" write$ newline$
3460 % "\else\language\csname l@#1\endcsname\fi" write$ newline$
3461 "\else\selectlanguage{#1}\fi" write$ newline$
3462 "\providecommand*\{href}[2]{\small #2}" write$ newline$
3463 "\providecommand*\{url}[1]{\small #1}" write$ newline$
3464 "\providecommand*\{BibUrl}[1]{url{#1}}" write$ newline$
3465 "\providecommand*\{BibAnnote}[1]{}" write$ newline$
3466 "\providecommand*\{BibEmph}[1]{#1}" write$ newline$
3467 (*modern!modern)
3468 %"\ProvideTextCommandDefault{\cyrdash}{---}" write$ newline$
3469 %\DeclareUTFcharacter[\UTFencname]{x2014}{\cyrdash}
3470 %"\let\cyrdash\textemdash" write$ newline$
3471 %"\ProvideTextCommandDefault{\cyrdash}{\hbox to.8em{--\hss--}}" write$ newline$
3472 %"\ProvideTextCommandDefault{\cyrdash}{\textemdash}" write$ newline$
3473 "\ProvideTextCommandDefault{\cyrdash}{\iflanguage{russian}{\hbox to.8em{--\hss--}}{\textemdash}}" write$ n
3474 %"\ProvideTextCommandDefault{\cyrdash}{%" write$ newline$
3475 %"\ \iflanguage{russian}{\hbox to.8em{--\hss--}}{%" write$ newline$
3476 %"\ \iflanguage{ukrainian}{\hbox to.8em{--\hss--}}{\textemdash}}}" write$ newline$
3477 "\providecommand*\{BibDash}{\ifdim\lastskip>Opt\unskip\nobreak\hskip.2em plus 0.1em\fi" write$ newline$
3478 "\cyrdash\hskip.2em plus 0.1em\ignorespaces}" write$ newline$
3479 "\renewcommand{\newblock}{\ignorespaces}" write$ newline$
3480 (/modern!modern)
3481 (natbib) "\providecommand{\natexlab}[1]{#1}" write$ newline$
3482 preamble$ empty$
3483 'skip$
3484 { preamble$ write$ newline$ }
3485 if$
3486 }
3487
3488
3489 EXECUTE {begin.bib}
3490
3491 EXECUTE {init.state.consts}
3492
3493 ITERATE {call.type$}
3494

```

`end.bib`

```

3495 FUNCTION {end.bib}
3496 { newline$
3497 % "\catcode'\/=11" write$ newline$
3498 "\end{thebibliography}" write$ newline$
3499 }
3500
3501 EXECUTE {end.bib}

```

3502

3503  $\langle$ /bst $\rangle$

That's all, Folks!

## Change History

v0.8	General: entry field <code>annotate</code> added . . . . . 8	v1.2c	General: entry field <code>eid</code> added . . . . . 8, 45, 46
	macro <code>\BibAnnote</code> added . . . . . 52, 86		entry field <code>langid</code> added . . . . . 8, 38
	macro <code>\BibEmph</code> added . . . . . 14, 86		fix Gost2003: "--- replaced by macro
	macro <code>\BibUrl</code> added . . . . . 51, 86		<code>\BibDash</code> . . . . . 10, 35
v0.9	General: bug fix in <code>@inproceedings</code> entry	v1.2d	General: <code>\bbljan</code> etc. macros removed . . . 73
	type . . . . . 61	v1.2e	General: <code>bbl.url</code> added to replace URL
v1.1	General: Entry type <code>@online</code> added . . . . . 65		string . . . . . 22
	entry type <code>@online</code> added . . . . . 14	v1.2f	General: entry <code>patent</code> added . . . . . 8
v1.2	General: <code>@report</code> entry . . . . . 67		entry field <code>patent</code> added . . . . . 55
	<code>@thesis</code> entry . . . . . 66	v1.2i	General: <code>@dscithesis</code> entry renamed to
	entry field <code>medium</code> added . . . . . 8		<code>@docthesi</code> . . . . . 71
	entry fields <code>eprint</code> , <code>eprintclass</code> ,		location field added as alias to <code>address</code>
	<code>eprinttype</code> . . . . . 54		field . . . . . 36
	entry fields <code>eprint</code> , <code>eprintclass</code> ,		<code>media</code> field normalised . . . . . 25
	<code>eprinttype</code> added . . . . . 8		number field is now alias to
	fix <code>bbl.urldate</code> for ukrainian (Andrey		<code>specialitycode</code> . . . . . 36
	Shvajkoy) . . . . . 22		<code>school</code> field is now alias to <code>institution</code> 36
v1.2a	General: default for <code>\cyrdash</code> added . . . . . 86		corrected <code>bbl.phdthesis</code> and <code>bbl.docthesi</code>
v1.2b	General: entry field <code>numpages</code> renamed to		to comply GOST . . . . . 23
	<code>pagetotal</code> . . . . . 8, 46		entry <code>@MastersThesis</code> added again (see
			v1.1) . . . . . 69
			typo fix in <code>format.month</code> . . . . . 34

## 3 Index

Numbers written in dark blue refer to the page where the corresponding entry is described; numbers in black roman refer to the code lines where the entry is used.

<b>Symbols</b>	<code>add.number</code> . . . . . 55	<code>author.full</code> . . . . . 38
<code>\&amp;</code> . . . . . 2909, 2910, 2932,	<code>address.or.location</code> . . . . . 36	<code>author.key.label</code> . . . . . 31
2936, 2941, 2959, 2962,	<code>and</code> . . . . . 13	<code>author.sort</code> . . . . . 79
2964, 2965, 2982, 3023	<code>article</code> . . . . . 56	<code>author.title.sort</code> . . . . . 80
<code>add.blank</code> . . . . . 12	<code>author.after</code> . . . . . 50	<code>bbl.and</code> . . . . . 19
<code>add.doi</code> . . . . . 52	<code>author.before</code> . . . . . 49	<code>bbl.chief</code> . . . . . 26
<code>add.media</code> . . . . . 53	<code>author.editor.full</code> . . . . . 38	<code>bbl.cmplr</code> . . . . . 16

bbl.docthis	23	either.or.chec	42	ielectronic	65
bbl.edby	15	emphasize	14	inbook	59
bbl.edition	16	empty.misc.check	48	incollection	59
bbl.etal	18	enclose.round.brackets	15	init.state.consts	9
bbl.executor	26	enclose.square.brackets	15	initialize.longest.label	84
bbl.iin	20	end.bib	86	inproceedings	61
bbl.iissue	17	eng.ord	43	institution.or.school	36
bbl.iiss	17	extract.num	43	internet	65
bbl.in	20	field.or.null	14	is.num	43
bbl.mathesis	23	fin.entry	11	make.full.names	38
bbl.media.eresource	25	format.address.publisher.date		manual	62
bbl.media.online	25		36	mastersthesis	69
bbl.media.text	25	format.address.publisher	36	misc	64
bbl.media	26	format.annotate	52	multi.page.check	45
bbl.nnoaddress	24	format.authors.after	33	n.dashify	41
bbl.nnopublisher	25	format.authors	31	new.block.checka	13
bbl.nnr	20	format.bookauthors.after	33	new.block.checkb	13
bbl.nnumber	19	format.bookauthors	33	new.block	11
bbl.nopublisher	24	format.btitle	41	new.colon	12
bbl.nr	20	format.bvolume	42	new.dblslash.checka	14
bbl.number	19	format.chapter.pages	47	new.dblslash	11
bbl.of	18	format.chief.after	33	new.semicolon	12
bbl.pages	21	format.compiler.after	34	new.sentence.checka	14
bbl.page	21	format.date	35	new.sentence.checkb	14
bbl.phdthesis	23	format.edition	44	new.sentence	12
bbl.ppages	21	format.editors.after	33	new.slash	12
bbl.ppage	22	format.eprint	54	non.stop	13
bbl.priority	27	format.executor.after	34	not	12
bbl.publ	27	format.full.names	38	online	65
bbl.req	27	format.isbn	52	or	13
bbl.techreport	23	format.key	30	output.address.publisher.date	
bbl.thesis.type'	24	format.lab.names	29		37
bbl.urldate	22	format.month	34	output.bibitem	38
bbl.url	22	format.names.rev	30	output.check	11
bbl.vvolume	17	format.names	28	output.nonnull	10
bbl.vvol	17	format.number.series	42	output.url	52
begin.bib	86	format.number	47	output	11
bookauthor.after	51	format.pages.page	46	patent	63
bookauthor.before	49	format.pages	45	phdthesis	68
booklet	58	format.prioritydate	56	presort	82
book	57	format.publicationdate	56	proceedings	60
bracify	15	format.requestdate	56	report	67
change.language	9	format.techrep.type.number		reverse.pass	85
chop.word	13		48	sort.format.names	78
conference	72	format.thesis.type	48	sort.format.title	79
convert.edition	44	format.title	34	sortify	78
default.type	73	format.type.number	55	space.word	15
docthis	71	format.url	51	specialitycode.or.number	36
editor.full	38	format.vol.num.pages	46	techreport	72
editor.organization.after	51	format.volume	47	thesis	65



tie.connect .....	42				
tie.or.space.connect .....	42				
unpublished .....	64				
webpage .....	65				
word.in .....	41				
www .....	65				
\u .....	363, 365, 368, 372,				
	662, 665, 678, 681, 711,				
	725, 727, 742, 744, 766,				
	768, 769, 770, 776, 778,				
	779, 780, 786, 788, 789,				
	790, 808, 812, 824, 825,				
	828, 829, 1175, 1182, 1196				
			<b>B</b>		
		\bbland .....	513		\bblPp .....
		\bbledby .....	361		632
		\bbledition .....	395		\bblpp .....
		\bblIn .....	596		606
		\bblin .....	586		\bblVol .....
		\bblIss .....	461		431
		\bblIssue .....	444		\bblVolume .....
		\bblno .....	557, 573		418
		\bblNumber .....	531		\BibAnnote .....
		\bblnumber .....	544		2013, 3465
		\bblof .....	478, 495		\BibDash .....
		\bblP .....	645		141, 1265, 3477
		\bblp .....	619		\BibEmph .....
					331, 3466
					\bibitem .....
					11,
					12, 13, 1369, 1471, 1500
					\BibUrl .....
					1991, 1992, 3464
					<b>C</b>
					\cyrdash .....
					3468,
					3469, 3470, 3471,
					3472, 3473, 3474, 3478