

# The GOST package

Igor A. Kotelnikov\*

2016/02/18, v.1.2d

## 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Kotelnikov to the present status and some code was borrowed from `disser` package developed by Stanislav Kruchinin and unpublished work by Artem Petrenkov.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
<code>cp866</code>	<code>ruscii.csf</code>	Cyrillic first, Latin
<code>cp1251</code>	<code>cp1251.csf</code>	Cyrillic first, Latin
<code>koi8-u</code>	<code>koi8u.csf</code>	Cyrillic first, Latin
<code>utf8</code>	<code>utf8cyrillic.csf</code>	Cyrillic first, Latin

In addition, BibTeX8 distribution comes with few more CSFs.

Encoding	CSF	Sorting order
<code>cp866</code>	<code>cp866rus.csf</code>	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 `ruussian`. 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 What's new in version 1.2d (2015.02.18)

1. `jan`, `feb`, e.t.c. macros fixed.
2. New macro `format.month`.

### 1.5 What's new in 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 (`\cyrdahs`) only for `ruussian` 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.)

### 1.6 What's new in version 1.2a (2012.08.31)

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

### 1.7 What's new in 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.

## 1.8 What's new in 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.

## 1.9 Version history

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 7.80-2000 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   annotate
38   author
39   booktitle
40   bookauthor
41   chapter
42   edition
43   editor
44   compiler
45   howpublished
46   institution
47   journal
48   key
49   month
50   note
51   number
52   organization
53   pages
54   eid % new in v1.2c
55   publisher
56   school
57   series
58   title
59   medium % new in v1.2.
60   type
61   volume
62   year
63   language
64   langid % new in v1.2c
65   booklanguage

```

Remaining entries are borrowed from biblatex.

```

66   pagetotal
67   url

```

```

68   urldate
69   isbn
70   doi
71 %   archive
72   eprinttype   % = archivePrefix
73   eprintclass  % = primaryClass
74   eprint
75 }
76 {}
77 <!natbib> { label }
78 <natbib> { label extra.label sort.label short.list }
79

```

Declare internal variables and constants used in to format references.

```

80 INTEGERS { output.state before.all mid.sentence after.sentence after.block
81 after.dblslash after.slash after.column after.semicolumn }
82

```

init.state.consts

```

83 FUNCTION {init.state.consts}
84 { #0 'before.all :=
85   #1 'mid.sentence :=
86   #2 'after.sentence :=
87   #3 'after.block :=
88   #4 'after.dblslash :=
89   #5 'after.slash :=
90   #6 'after.column :=
91   #7 'after.semicolumn :=
92 }
93
94 STRINGS { s t }
95
96 STRINGS { curlanguage }
97

```

## 2.2 Formatting functions

change.language    Declare function to switch language.

```

98 FUNCTION {change.language}
99 { booklanguage empty$
100   { "" }
101   { booklanguage 'curlanguage :=
102     "\selectlanguageifdefined{"
103     curlanguage *
104     "}" *
105   }
106   if$
107 }
108

```

output.nonnull    Declare functions to output various parts of bibliographic record.



```

109 FUNCTION {output.nonnull}
110 {
111   swap$
112   output.state mid.sentence =
113   { ", " * write$ }
114   { output.state after.block =
115     { add.period$ write$
116     <!modern)      " \BibDash " write$ %%" " quote$ "--- " * * write$ % since v1.2c
117     <modern)      " \BibDash " write$
118     newline$
119     "\newblock " write$
120     }
121     { output.state before.all =
122       'write$
123       { output.state after.dblslash =
124         { "~//" * change.language * " " * write$ }
125         { output.state after.slash =
126           { "~/" * write$ }
127         { output.state after.column =
128           <!(strict | modern)) { ":" * write$ }
129           <strict | modern)   { "~: " * write$ }
130         { output.state after.semicolumn =
131           <!(strict | modern)) { ";" * write$ }
132           <strict | modern)   { "~;" * write$ }
133         { add.period$ " " * write$ }
134         if$
135         }
136         if$
137         }
138         if$
139         }
140         if$
141         }
142         if$
143         }
144         if$
145         mid.sentence 'output.state :=
146       }
147     if$
148   }
149 }

```

```

output
output.check 150 FUNCTION {output}
151 { duplicate$ empty$
152   'pop$
153   'output.nonnull
154   if$
155 }
156
157 FUNCTION {output.check}

```

```

158 { 't :=
159   duplicate$ empty$
160   { pop$ "empty " t * " in " * cite$ * warning$ }
161   'output.nonnull
162   if$
163 }
164
fin.entry      fin.entry finalizes current entry. It writes dot, if no dot is found in stack, and starts new line.
165 FUNCTION {fin.entry}
166 { add.period$
167   write$
168   newline$
169 }
170
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.
171 FUNCTION {new.block}
172 { output.state before.all =
173   'skip$
174   { after.block 'output.state := }
175   if$
176 }
177
new.dblslash
178 FUNCTION {new.dblslash}
179 { output.state before.all =
180   'skip$
181   { after.dblslash 'output.state := }
182   if$
183 }
184
new.slash
185 FUNCTION {new.slash}
186 { output.state before.all =
187   'skip$
188   { after.slash 'output.state := }
189   if$
190 }
191
new.column
192 FUNCTION {new.column}
193 { output.state before.all =
194   'skip$
195   { after.column 'output.state := }
196   if$
197 }
198

```

new.semicolumn

```
199 FUNCTION {new.semicolumn}
200 { output.state before.all =
201   'skip$
202   { after.semicolumn 'output.state := }
203   if$
204 }
205
```

new.sentence

```
206 FUNCTION {new.sentence}
207 { output.state after.block =
208   'skip$
209   { output.state before.all =
210     'skip$
211     { after.sentence 'output.state := }
212     if$
213   }
214   if$
215 }
216
```

add.blank

```
217 FUNCTION {add.blank}
218 { " " * before.all 'output.state :=
219 }
220
```

not        Declare few logical functions.

```
221 FUNCTION {not}
222 { { #0 }
223   { #1 }
224   if$
225 }
226
```

and

```
227 FUNCTION {and}
228 { 'skip$
229   { pop$ #0 }
230   if$
231 }
232
```

or

```
233 FUNCTION {or}
234 { { pop$ #1 }
235   'skip$
236   if$
237 }
238
```

`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`.

```
239 ⟨*sort | natbib⟩
240 INTEGERS { len }
241
242 FUNCTION {chop.word}
243 { 's :=
244   'len :=
245   s #1 len substring$ =
246     { s len #1 + global.max$ substring$ }
247   's
248   if$
249 }
250 ⟨/sort | natbib⟩
251
```

`non.stop`

```
252 FUNCTION {non.stop}
253 { duplicate$
254   "}" * add.period$
255   #-1 #1 substring$ "." =
256 }
257
```

`new.block.checka`

```
258 FUNCTION {new.block.checka}
259 { empty$
260   'skip$
261   'new.block
262   if$
263 }
264
```

`new.block.checkb`

```
265 FUNCTION {new.block.checkb}
266 { empty$
267   swap$ empty$
268   and
269   'skip$
270   'new.block
271   if$
272 }
273
```

`w.sentence.checka`

```
274 FUNCTION {new.sentence.checka}
275 { empty$
276   'skip$
277   'new.sentence
278   if$
279 }
280
```

w.sentence.checkb

```
281 FUNCTION {new.sentence.checkb}
282 { empty$
283   swap$ empty$
284   and
285   'skip$
286   'new.sentence
287   if$
288 }
289
```

w.dblslash.checka For online entry.

```
290 FUNCTION {new.dblslash.checka}
291 { empty$
292   'skip$
293   'new.dblslash
294   if$
295 }
296
```

field.or.null

```
297 FUNCTION {field.or.null}
298 { duplicate$ empty$
299   { pop$ "" }
300   'skip$
301   if$
302 }
303
```

emphasize Declare function to emphasize last string in stack.

```
304 FUNCTION {emphasize}
305 { duplicate$ empty$
306   { pop$ "" }
307   { "\BibEmph{" swap$ * "}" * }
308   if$
309 }
310
```

e.square.brackets New in v.1.2. Declare function to enclose last word in square brackets. It do so only for the string option.

```
311 <*/strict>
312 FUNCTION {enclose.square.brackets} { }
313 </!strict>
314 <*/strict>
315 FUNCTION {enclose.square.brackets}
316 { duplicate$ empty$
317   { pop$ "" }
318   { "[" swap$ * "]" * }
319   if$
320 }
321 </strict>
322
```

```

323
324 % \DescribeFunction{bracify}
325 % \DescribeFunction{bracketise}
326 % \DescribeFunction{parenthesify}
327 % New in v.1.2. An idea borrowed from apsrev4-1.bst.
328 % Declare function to enclose last word in braces, square brackets and
329 % parenthesis.
330 % \begin{macrocode}
331 FUNCTION {bracify}
332 { duplicate$ empty$
333   { pop$ "{}" }
334   { "{" swap$ * "}" * }
335   if$
336 }
337 FUNCTION {bracketise}
338 { duplicate$ empty$
339   { pop$ "[]" }
340   { "[" swap$ * "]" * }
341   if$
342 }
343 FUNCTION {parenthesify}
344 { duplicate$ empty$
345   { pop$ "()" }
346   { "(" swap$ * ")" * }
347   if$
348 }
349
350

```

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

```

351 FUNCTION {space.word}
352 { " " swap$ * " " * }
353

```

## 2.3 Standard abbreviations

`bb1.edby`    Declare language-sensitive abbreviations.

```

354 FUNCTION {bb1.edby}    % { "\bbledby{}" }
355 { curlanguage "english" =
356   {"Ed.\ by"}
357   { curlanguage "ukrainian" =
358     (!utf8)        {"{\cyr\CYRP\cyrii\cyrd\ \cyrr\cyre\cyrd.}"}
359     (utf8)        {"Під ред."}
360   { curlanguage "russian" =
361     (!utf8)        {"{\cyr\CYRP\cyro\cyrd\ \cyrr\cyre\cyrd.}"}
362     (utf8)        {"Под ред."}
363   { curlanguage "german" =
364     {"ed." }
365     {"language is not defined: " language "edby" * * warning$ "Ed.\ by"}
366     if$}

```

```

367     if$}
368   if$}
369 if$}
370

```

bb1.cmplr

```

371 FUNCTION {bb1.cmplr}
372 { curlanguage "english" =
373   { "Compiler" }
374   { curlanguage "german" =
375     { "Hrsg." }
376     { curlanguage "ukrainian" =
377       (!utf8)   {"{\cyr\CYRU\cyrk\cyrk\cyrk.}" }
378       (utf8)   {"{Укл.}" }
379       { curlanguage "russian" =
380         (!utf8) {"{\cyr\CYRS\cyro\cyrs\cyrt.}" }
381         (utf8) {"{Сост.}" }
382         {"language is not defined: " language "cmplr" * * warning$ "Compiler"}
383         if$}
384       if$}
385     if$}
386   if$}
387

```

bb1.edition

```

388 FUNCTION {bb1.edition} % { "\bb1edition{" }
389 { curlanguage "english" =
390   {"edition"}
391   { curlanguage "ukrainian" =
392     (!utf8) {"{\cyr\cyrv\cyri\cyrd.}" }
393     (utf8) {"{вид.}" }
394     { curlanguage "russian" =
395       (!utf8) {"{\cyr\cyri\cyrz\cyrd.}" }
396       (utf8) {"{изд.}" }
397       { curlanguage "german" =
398         {" {aus.}" } %%% { "Auf1." } ??
399         { curlanguage "italian" =
400           {"edizione"}
401           { curlanguage "french" =
402             {"\{e}dition"}
403             {"language is not defined: " language "edition" * * warning$ "edition"}
404             if$}
405           if$}
406         if$}
407       if$}
408     if$}
409   if$}
410

```

bb1.vvolume

```

411 FUNCTION {bb1.vvolume} % { "\bb1Volume{" }
412 { curlanguage "english" = curlanguage "french" = or curlanguage "italian" = or

```

```

413 {"Volume"}
414 { curlanguage "ukrainian" = curlanguage "russian" = or
415 <!utf8> {"\CYRT\cyro\cyrm" }
416 <utf8> {"Том" }
417 { curlanguage "german" =
418 {"{Band}"} %%% { "Volumen" }
419 {"language is not defined: " language "vvolume" * * warning$ "Volume"}
420 if$}
421 if$}
422 if$}
423

```

#### bbl.vvol

```

424 FUNCTION {bbl.vvol} % { "\bblVol{" }
425 { curlanguage "english" = curlanguage "french" = or curlanguage "italian" = or
426 {"Vol."}
427 { curlanguage "ukrainian" = curlanguage "russian" = or
428 <!utf8> {"\CYRT."}
429 <utf8> {"Т."}
430 { curlanguage "german" =
431 {"{Bd.}"} %%% { "Vol." }
432 {"language is not defined: " language "vvol" * * warning$ "Vol."}
433 if$}
434 if$}
435 if$}
436

```

#### bbl.iissue

```

437 FUNCTION {bbl.iissue} % { "\bblIssue{" }
438 { curlanguage "english" =
439 {"Issue"}
440 { curlanguage "ukrainian" =
441 <!utf8> {"\CYRV\cyri\cyrp\cyru\cyrs\cyrk"}
442 <utf8> {"Випуск"}
443 { curlanguage "russian" =
444 <!utf8> {"\CYRV\cyrery\cyrp\cyru\cyrs\cyrk"}
445 <utf8> {"Выпуск"}
446 { curlanguage "german" =
447 {"{Heft}"} %%% { "Ausgabe" }
448 {"language is not defined: " language "iissue" * * warning$ "Issue"}
449 if$}
450 if$}
451 if$}
452 if$}
453

```

#### bbl.iiss

```

454 FUNCTION {bbl.iiss} % { "\bblIss{" }
455 { curlanguage "english" =
456 {"Iss."}
457 { curlanguage "ukrainian" =
458 <!utf8> {"\CYRV\cyri\cyrp."}

```



```

459 <utf8>      {"Вип."}
460      { curlanguage "russian" =
461 <!utf8>      {"\CYRV\cyrery\cyrp."}
462 <utf8>      {"Вип."}
463      { curlanguage "german" =
464      {"{H.}" }
465      {"language is not defined: " language "iiss" * * warning$ "Iss."}
466      if$}
467      if$}
468      if$}
469 if$}
470

```

bbl.of

```

471 FUNCTION {bbl.of}      % { "\bblof{}" }
472 { curlanguage "english" =
473   {"of"}
474   { curlanguage "german" =
475     {"von"}
476     { curlanguage "ukrainian" =
477 <!utf8>      {"{\cyr\cyrii\cyrz}" }
478 <utf8>      {"{із}" }
479     { curlanguage "russian" =
480 <!utf8>      {"{\cyr\cyri\cyrz}" }
481 <utf8>      {"{из}" }
482     {"language is not defined: " language "of" * * warning$ "of"}
483     if$}
484     if$}
485     if$}
486 if$}
487

```

bbl.etal

```

488 FUNCTION {bbl.etal}      % { "\bblof{}" }
489 { curlanguage "english" =
490   {"et~al."}
491   { curlanguage "german" =
492     {"u.~a." }
493     { curlanguage "ukrainian" =
494 <!utf8>      {"{\cyr\cyrt\cyra~\cyrii\cyrn.}" }
495 <utf8>      {"{та~иН.}" }
496     { curlanguage "russian" =
497 <!utf8>      {"{\cyr\cyri~\cyrd\cyrr.}" }
498 <utf8>      {"{и~др.}" }
499     {"language is not defined: " language "et~al" * * warning$ "et~al."}
500     if$}
501     if$}
502     if$}
503 if$}
504

```

bbl.and

```

505 (*natbib)
506 FUNCTION {bbl.and} % { "\bbland{}" }
507 { curlanguage "english" =
508   {"and"}
509   { curlanguage "german" =
510     { "und" }
511     { curlanguage "ukrainian" =
512       \!utf8 { "\cyrii"}
513       \utf8 { "и"}
514       { curlanguage "russian" =
515         \!utf8 { "\cyri"}
516         \utf8 { "и"}
517         {"language is not defined: " language "and" * * warning$ "and"}
518         if$}
519       if$}
520     if$}
521 if$}
522 \!natbib)
523

```

bbl.number

```

524 FUNCTION {bbl.number} % { "\bblNumber{}" }
525 { curlanguage "english" =
526   {"Number"}
527   { curlanguage "ukrainian" = curlanguage "russian" = or
528     \!utf8 { "\CYRN\cyro\cyrm\cyre\cyrr" }
529     \utf8 { "\{Homep}" }
530     { curlanguage "german" =
531       {"{Heft}" } %%% { "Anzahl" }
532       {"language is not defined: " language "nnumber" * * warning$ "Number"}
533       if$}
534     if$}
535 if$}
536

```

bbl.number

```

537 FUNCTION {bbl.number} % { "\bblnumber{}" }
538 { curlanguage "english" =
539   {"number"}
540   { curlanguage "ukrainian" = curlanguage "russian" = or
541     \!utf8 { "\cyr\cyrn\cyro\cyrm\cyre\cyrr"}
542     \utf8 { "\{номер}" }
543     { curlanguage "german" =
544       {"{Heft}" } %%% { "anzahl" } ???
545       {"language is not defined: " language "number" * * warning$ "number"}
546       if$}
547     if$}
548 if$}
549

```

bbl.nr

```

550 FUNCTION {bbl.nr} % { "\bblno{}" }

```

```

551 { curlanguage "english" =
552   {"no."}
553   { curlanguage "italian" =
554     { "no" }
555     { curlanguage "ukrainian" = curlanguage "russian" = or
556 <!utf8>       { "{\cyr\textnumero}" }
557 <utf8>       { "{#}" }
558             { curlanguage "german" =
559               {"{H.}" } %%% { "an." }
560               {"language is not defined: " language "nr" * * warning$ "no."}
561             if$}
562     if$}
563   if$}
564 if$}
565

```

bbl.nnr

```

566 FUNCTION {bbl.nnr} % { "\bblno{}" }
567 { curlanguage "english" =
568   {"No."}
569   { curlanguage "ukrainian" = curlanguage "russian" = or
570 <!utf8>       { "{\cyr\textnumero}" }
571 <utf8>       { "{#}" }
572             { curlanguage "german" =
573               {"{H.}" } %%% { "an." }
574               {"language is not defined: " language "nnr" * * warning$ "No."}
575             if$}
576   if$}
577 if$}
578

```

bbl.in

```

579 FUNCTION {bbl.in} % { "\bblin{}" }
580 { curlanguage "english" = curlanguage "german" = or
581   {"in"}
582   { curlanguage "ukrainian" = curlanguage "russian" = or
583 <!utf8>       { "{\cyr\cyrv}" }
584 <utf8>       { "{B}" }
585             {"language is not defined: " language "in" * * warning$ "in"}
586   if$}
587 if$}
588

```

bbl.iin

```

589 FUNCTION {bbl.iin} % { "\bblIn{}" }
590 { curlanguage "english" = curlanguage "german" = or
591   {"In"}
592   { curlanguage "ukrainian" = curlanguage "russian" = or
593 <!utf8>       { "\CYRV" }
594 <utf8>       { "{B}" }
595             {"language is not defined: " language "iin" * * warning$ "In"}
596   if$}

```

```
597 if$}
598
```

#### bb1.pages

```
599 FUNCTION {bb1.pages} % { "\bb1pp." }
600 { curlanguage "english" = curlanguage "french" = or curlanguage "italian" = or
601   {"p."} %%% {"pp."}
602   { curlanguage "ukrainian" = curlanguage "russian" = or
603     \!utf8   {"{\cyr\cyrs.}"}
604     \utf8    {"{c.}"}
605     { curlanguage "german" =
606       {"S."} %%% {"s." }
607       {"language is not defined: " language "pages" * * warning$ "p."}
608     if$}
609   if$}
610 if$}
611
```

#### bb1.page

```
612 FUNCTION {bb1.page} % { "\bb1p."
613 { curlanguage "english" = curlanguage "french" = or curlanguage "italian" = or
614   {"p."}
615   { curlanguage "ukrainian" = curlanguage "russian" = or
616     \!utf8   {"{\cyr\cyrs.}"}
617     \utf8    {"{c.}"}
618     { curlanguage "german" =
619       {"S."} %%% {"s." }
620       {"language is not defined: " language "page" * * warning$ "p."}
621     if$}
622   if$}
623 if$}
624
```

#### bb1.ppages

```
625 FUNCTION {bb1.ppages}% { "\bb1Pp." }
626 { curlanguage "english" = curlanguage "french" = or curlanguage "italian" = or
627   {"P."} %%% {"Pp." }
628   { curlanguage "ukrainian" = curlanguage "russian" = or
629     \!utf8   {"{\cyr\CYRS.}"}
630     \utf8    {"{C.}"}
631     { curlanguage "german" =
632       {"S."}
633       {"language is not defined: " language "ppages" * * warning$ "P."}
634     if$}
635   if$}
636 if$}
637
```

#### bb1.ppage

```
638 FUNCTION {bb1.ppage} % { "\bb1P." }
639 { curlanguage "english" = curlanguage "french" = or curlanguage "italian" = or
640   {"P."}
```

```

641 { curlanguage "ukrainian" = curlanguage "russian" = or
642 <!utf8>      {"{\cyr\CYRS.}" }
643 <utf8>      {"{C.}" }
644 { curlanguage "german" =
645     {"S."}
646     {"language is not defined: " language "ppage" * * warning$ "P."}
647     if$}
648 if$}
649 if$}
650

```

bb1.urldate Next function was added in version 2012.01.15.

```

651 FUNCTION {bb1.urldate}
652 { curlanguage "english" =
653     {"online; accessed"}
654     { curlanguage "ukrainian" =
655 <!utf8>      {"{\cyr\d\cyra\cyrt\cyra\ \cyrz\cyrv\cyre\cyrr\cyrn\cyre\cyrn\cyrn\cyrya}" }
656 <utf8>      {"{дата звернення}" }
657     { curlanguage "russian" =
658 <!utf8>      {"{\cyr\d\cyra\cyrt\cyra\ \cyro\cyrb\cyrr\cyra\cyrshch\cyre\cyrn\cyri\cyrya}" }
659 <utf8>      {"{дата обращения}" }
660     { curlanguage "german" =
661     {"online; abgerufen"} }
662     {"language is not defined: " language "urldate" * * warning$ "online; accessed" }
663     if$}
664     if$}
665     if$}
666 if$}
667

```

bb1.techreport

```

668 FUNCTION {bb1.techreport} % rename to bb1.techreport
669 { curlanguage "english" =
670     {"Rep." }
671     { curlanguage "german" =
672     {"Bericht" }
673     { curlanguage "russian" =
674 <!utf8>      {"{\cyr\CYRO\cyrt\cyrch\cyre\cyrt}" }
675 <utf8>      {"{Отчет}" }
676     {"language is not defined: " language "techrep" * * warning$ "Rep." }
677     if$}
678     if$}
679 if$}
680

```

bb1.mthesis

```

681 FUNCTION {bb1.mthesis}
682 { curlanguage "english" =
683     {"Master's thesis" }
684     { curlanguage "german" =
685     {"Diss.~Mag." }
686     { curlanguage "russian" =

```

```

687 \!utf8 { "\{cyr\CYRK\cyrv\cyra\cyrl\cyri\cyrf\cyri\cyrk\cyra\cyrc\cyri"
688 \!utf8 " \cyro\cyrn\cyrn\cyra\cyrya\ \cyrr\cyra\cyrb\cyro\cyrt\cyra\ " *
689 \!utf8 " \cyrm\cyra\cyrg\cyri\cyrs\cyrt\cyrr\cyra}" * }
690 \utf8 { "\{Квалификационная работа магистра}" }
691 { "language is not defined: " language "mthesis" * * warning$ "Master's thesis" }
692 if$}
693 if$}
694 if$}
695

```

bb1.phdthesis

```

696 FUNCTION {bb1.phdthesis}
697 { curlanguage "english" =
698   { "Ph.\,D. thesis" }
699   { curlanguage "german" =
700     { "Diss.~Ph.\,D." }
701     { curlanguage "russian" =
702 \!utf8 { "\{cyr\CYRD\cyri\cyrs\cyrs\ldots\ \cyrk\cyra\cyrn\cyrd\cyri"
703 \!utf8 " \cyrd\cyra\cyrt\cyra\ \cyrn\cyra\cyru\cyrk}" * }
704 \utf8 { "\{Дисс\ldots\ кандидата наук}" }
705     { curlanguage "french" =
706       { "Th\{e}se de doctorat" }
707       { "language is not defined: " language "phdthesis" * * warning$ "Ph.\,D. thesis" }
708       if$}
709     if$}
710   if$}
711 if$}
712

```

bb1.dscithesis

```

713 FUNCTION {bb1.dscithesis}
714 { curlanguage "english" =
715   { "Dr.\,Sci. dissertation" }
716   { curlanguage "german" =
717     { "Diss.~Dr." }
718     { curlanguage "russian" =
719 \!utf8 { "\{cyr\CYRD\cyri\cyrs\cyrs\ldots\ \cyrd\cyro\cyrk\cyrt\cyro"
720 \!utf8 " \cyrr\cyra\ \cyrn\cyra\cyru\cyrk}" * }
721 \utf8 { "\{Дисс\ldots\ доктора наук}" }
722     { "language is not defined: " language "dscithesis" * * warning$ "Dr.\,Sci. dissertation" }
723     if$}
724   if$}
725 if$}
726

```

bb1.nnoaddress

```

727 FUNCTION {bb1.nnoaddress}
728 { curlanguage "english" =
729   { "S.\ l." }
730   { curlanguage "russian" =
731 \!utf8 { "\{cyr\CYRB.\ \cyrm.}" }
732 \utf8 { "\{Б.\ м.}" }

```

```

733     { "language is not defined: " language "nnoaddress" * * warning$ "S.\ l." }
734   if$}
735 if$}
736

```

bb1.nopublisher

```

737 FUNCTION {bb1.nopublisher}
738 { curlanguage "english" =
739   { "s.\ n." }
740   { curlanguage "russian" =
741     (!utf8)   { "{\cyr\cyrb.\ \cyri.}" }
742     (utf8)   { "{6.\ и.}" }
743     { "language is not defined: " language "nnoaddress" * * warning$ "s.\ n." }
744     if$}
745 if$}
746

```

bb1.nnopublisher

```

747 FUNCTION {bb1.nnopublisher}
748 { curlanguage "english" =
749   { "S.\ n." }
750   { curlanguage "russian" =
751     (!utf8)   { "{\cyr\CYRB.\ \cyri.}" }
752     (utf8)   { "{Б.\ и.}" }
753     { "language is not defined: " language "nnopublisher" * * warning$ "S.\ n." }
754     if$}
755 if$}
756

```

bb1.medium.text

```

757 FUNCTION {bb1.medium.text}
758 { curlanguage "english" =
759   { "Text" }
760   { curlanguage "russian" = curlanguage "ukrainian" = or
761     (!utf8)   { "{\cyr\CYRT\cyre\cyrk\cyrs\cyrt}" }
762     (utf8)   { "{Текст}" }
763     { "language is not defined: " language "medium" * * warning$ "Text" }
764     if$}
765 if$}
766

```

bb1.medium.elres

```

767 FUNCTION {bb1.medium.elres}
768 { curlanguage "english" =
769   { "Electronic resource" }
770   { curlanguage "russian" =
771     (!utf8)   { "{\cyr\CYREREV\cyr\cyre\cyrk\cyrt\cyrr\cyro\cyrn\cyrn\cyrrery\cyrishrt\ \cyrr\cyre\cyrs\cyru\cyrr}" }
772     (utf8)   { "{Электронный ресурс}" }
773     { curlanguage "ukrainian" =
774     (!utf8)   { "{\cyr\CYRE\cyr\cyre\cyrk\cyrt\cyrr\cyro\cyrn\cyrn\cyri\cyrishrt\ \cyrr\cyre\cyrs\cyru\cyrr}" }
775     (utf8)   { "{Електронний ресурс}" }
776     { "language is not defined: " language "medium" * * warning$ "Electronic resource" }

```

```

777     if$}
778   if$}
779 if$}
780
781

```

bbl.chief

```

782 FUNCTION {bbl.chief}
783 { curlanguage "english" =
784   { "chief" }
785   { curlanguage "russian" =
786     <utf8> { "\cyrr\cyru\cyrk." }
787     <utf8> { "{рук.}" }
788     { curlanguage "ukrainian" =
789       <utf8> { "\cyrr\cyru\cyrk." }
790       <utf8> { "{рук.}" }
791       { "language is not defined: " language "chief" * * warning$ "chief" }
792       if$}
793     if$}
794 if$}
795

```

bbl.executor

```

796 FUNCTION {bbl.executor}
797 { curlanguage "english" =
798   { "Executor" }
799   { curlanguage "russian" =
800     <utf8> { "{\cyr\cyri\cyrs\cyrp\cyro\cyr\cyrn.}" }
801     <utf8> { "{исполн.}" }
802     { curlanguage "ukrainian" =
803       <utf8> { "{\cyr\cyrv\cyri\cyrk\cyro\cyrn\cyra\cyrv\cyre\cyrc\cyrsfts}" }
804       <utf8> { "{виконавець}" }
805       { "language is not defined: " language "medium" * * warning$ "Executor" }
806       if$}
807     if$}
808 if$}
809

```

bbl.medium

```

810 FUNCTION {bbl.medium}
811 { type$ "online" =
812   { bbl.medium.elres }
813   { bbl.medium.text }
814 if$}
815

```

## 2.4 Formatting functions

Declare functions to format separate elements of bibliographic reference.

```

816 INTEGERS { nameptr namesleft numnames }
817

```



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.

```
819 (*!modern)
820 FUNCTION {format.names}
821 {
822   \!utf8 #1 "{vv~}{ll}{~jj}{,~f.}" format.name$
823   \utf8 #1 "{vv~}{ll}{~jj}{,~ff}" format.name$
824 }
825 \!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'.

```
826 (*modern)
827 FUNCTION {format.names}
828 {
829   's :=
830   #1 'nameptr :=
831   s num.names$ 'numnames :=
832   numnames 'namesleft :=
833   { namesleft #0 > }
834   { s nameptr
835     \!utf8 "{vv~}{ll}{~jj}{~f.}" format.name$ 't :=
836     \utf8 "{vv~}{ll}{~jj}{~ff}" format.name$ 't :=
837     nameptr #1 >
838     { nameptr #4 =
839       numnames #4 > and
840       { "others" 't :=
841         #1 'namesleft := }
842       'skip$
843       if$
844       namesleft #1 >
845       { ", " * t * }
846       { t "others" =
847         t "~others" =
848         or
849         \!strict { " " * bbl.etal * }
850         \strict { " " * bbl.etal enclose.square.brackets * }
851         { ", " * t * }
852       if$
853     }
854     if$
855   }
```

```

856      't
857      if$
858      nameptr #1 + 'nameptr :=
859      namesleft #1 - 'namesleft :=
860    }
861  while$
862 }
863 </modern>
864

```

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

```

865 <*natbib>
866 FUNCTION {format.lab.names}
867 { 's :=
868   language empty$
869   { "english" 'curlanguage := }
870   { language 'curlanguage := }
871   if$
872   s #1 "{vv~}{ll}" format.name$
873   s num.names$ duplicate$
874   #2 >
875   {% pop$ " et~al." * }
876   { pop$ " " bbl.etal * * }
877   { #2 <
878     'skip$
879     { s #2 "{ff }{vv }{ll}{ jj}" format.name$ "others" =
880       {% " et~al." * }
881       { " " bbl.etal * * }
882       {% " and " * s #2 "{vv~}{ll}" format.name$ * }
883       { " " bbl.and " " * * * s #2 "{vv~}{ll}" format.name$ * }
884       if$
885     }
886     if$
887   }
888   if$
889 }
890 </natbib>
891

```

`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.

```

892 FUNCTION {format.names.rev}
893 {
894   's :=
895   #1 'nameptr :=
896   s num.names$ 'numnames :=
897   numnames 'namesleft :=
898   { namesleft #0 > }
899   { s nameptr

```

```

900 <!utf8>      %"{f.}{~vv}{~ll}{, jj}" format.name$ 't :=
901 <!utf8>      "{ff}{~vv}{~ll}{, jj}" format.name$ 't :=
902 <utf8>       "{ff}{~vv}{~ll}{, jj}" format.name$ 't :=
903     nameptr #1 >
904         { nameptr #4 =
905             numnames #4 > and
906                 { "others" 't :=
907                     #1 'namesleft := }
908                 'skip$
909             if$
910             namesleft #1 >
911                 { ", " * t * }
912                 { t "others" =
913                 t "~others" =
914             or
915 <!strict>         { " " * bbl.etal * }
916 <strict>         { " " * bbl.etal enclose.square.brackets * }
917                 { ", " * t * }
918             if$
919         }
920     if$
921 }
922 't
923 if$
924 nameptr #1 + 'nameptr :=
925 namesleft #1 - 'namesleft :=
926 }
927 while$
928 }
929

```

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

```

930 <natbib>
931 FUNCTION {format.key}
932 { empty$
933   { key field.or.null }
934   { "" }
935   if$
936 }
937 </natbib>
938

```

`format.authors`

```

939 FUNCTION {format.authors}
940 { author empty$
941   { "" }
942   { author format.names emphasize}
943   if$
944 }
945

```

`author.key.label`

```

946 (*natbib)
947 FUNCTION {author.key.label}
948 { author empty$
949   { key empty$
950     { cite$ #1 #3 substring$ }
951     'key
952     if$
953   }
954   { author format.lab.names }
955   if$
956 }
957
958 FUNCTION {author.editor.key.label}
959 { author empty$
960   { editor empty$
961     { key empty$
962       { cite$ #1 #3 substring$ }
963       '%key          %% causes lost of year
964       { "{}" key * } %% Bug in bibtex8 ??
965       if$
966     }
967     { editor format.lab.names }
968     if$
969   }
970   { author format.lab.names }
971   if$
972 }
973
974 FUNCTION {author.key.organization.label}
975 { author empty$
976   { key empty$
977     { organization empty$
978       { cite$ #1 #3 substring$ }
979       { "The " #4 organization chop.word #3 text.prefix$ }
980       if$
981     }
982     'key
983     if$
984   }
985   { author format.lab.names }
986   if$
987 }
988
989 FUNCTION {editor.key.organization.label}
990 { editor empty$
991   { key empty$
992     { organization empty$
993       { cite$ #1 #3 substring$ }
994       { "The " #4 organization chop.word #3 text.prefix$ }
995       if$

```

```

996     }
997     'key
998     if$
999   }
1000 { editor format.lab.names }
1001 if$
1002 }
1003
1004 FUNCTION {calc.short.authors}
1005 { type$ "book" =
1006   type$ "inbook" =
1007   or
1008     'author.editor.key.label
1009     { type$ "proceedings" =
1010       'editor.key.organization.label
1011       { type$ "manual" =
1012         'author.key.organization.label
1013         'author.key.label
1014         if$
1015       }
1016     }
1017   }
1018 if$
1019 'short.list :=
1020 }
1021
1022 FUNCTION {calc.label}
1023 { calc.short.authors
1024   short.list
1025   "("
1026   *
1027   year duplicate$ empty$
1028   short.list key field.or.null = or
1029     { pop$ "" }
1030   'skip$
1031 if$
1032 *
1033 'label :=
1034 }
1035
1036 </natbib>
1037

```

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

```

1038 FUNCTION {format.bookauthors}
1039 { bookauthor empty$
1040   { "" }
1041   { bookauthor format.names}
1042 if$
1043 }

```

```

1044
mat.authors.after
1045 FUNCTION {format.authors.after}
1046 { author empty$
1047   { "" }
1048   { author format.names.rev}
1049   if$
1050 }
1051
bookauthors.after
1052 FUNCTION {format.bookauthors.after}
1053 { bookauthor empty$
1054   { "" }
1055   { bookauthor format.names.rev}% always cuts to 4 persons
1056   if$
1057 }
1058
mat.editors.after
1059 FUNCTION {format.editors.after}
1060 { editor empty$
1061   { "" }
1062   { bbl.edby "\ " * editor format.names.rev * }
1063   if$
1064 }
1065
ormat.chief.after
1066 FUNCTION {format.chief.after}
1067 { editor empty$
1068   { "" }
1069   { bbl.chief "\ " * editor format.names.rev * }
1070   if$
1071 }
1072
mat.executor.after
1073 FUNCTION {format.executor.after}
1074 { author empty$
1075   { "" }
1076   { bbl.executor ": " * author format.names.rev * }
1077   if$
1078 }
1079
mat.compiler.after
1080 FUNCTION {format.compiler.after}
1081 { compiler empty$
1082   { "" }
1083   { bbl.cmplr "\ " * compiler format.names.rev * }
1084   if$

```

```
1085 }
1086
```

`format.title` **Important note** Neither bibtex nor bibtex8 can handle unencoded 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.

```
1087 FUNCTION {format.title}
1088 { title empty$
1089   { "" }
1090 <utf8> { title "t" change.case$ }
1091 <utf8> { title }
1092 if$
1093 }
1094
```

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

```
1095 FUNCTION {format.month}
1096 { month empty$
1097   { "" }
1098   { curlanguage "russian" =
1099     { month "Jan." =
1100 <utf8> { "\CYRYA\cyrn\cyrv." }
1101 <utf8> { "ЯНВ." }
1102     { month "Feb." =
1103 <utf8> { "\CYRF\cyre\cyrv\cyrr." }
1104 <utf8> { "ФЕВ." }
1105     { month "Mar." =
1106 <utf8> { "\CYRM\cyra\cyrr\cyrt" }
1107 <utf8> { "МАРТ" }
1108     { month "Apr." =
1109 <utf8> { "\CYRA\cyrp\cyrr." }
1110 <utf8> { "АПР." }
1111     { month "May" =
1112 <utf8> { "\CYRM\cyra\cyrishrt" }
1113 <utf8> { "МАЙ" }
1114     { month "Jun." =
1115 <utf8> { "\CYRI\cyryu\cyrn\cyrsftsn" }
1116 <utf8> { "ИЮНЬ" }
1117     { month "Jul." =
1118 <utf8> { "\CYRI\cyryu\cyrl\cyrsftsn" }
1119 <utf8> { "ИЮЛЬ" }
1120     { month "Aug." =
1121 <utf8> { "\CYRA\cyrv\cyrg\." }
1122 <utf8> { "АВГ." }
1123     { month "Sep." =
1124 <utf8> { "\CYRS\cyre\cyn\cyrt." }
1125 <utf8> { "СЕНТ." }
1126     { month "Oct." =
1127 <utf8> { "\CYRO\cyrk\cyrt." }
1128 <utf8> { "ОКТ." }

```





```

1176 %FUNCTION {output.address.publisher}
1177 %{
1178 %   address empty$
1179 %   {
1180 %     bbl.nnoaddress
1181 %     publisher empty$
1182 %     { "~: " * bbl.nopublisher * }
1183 %     { }
1184 %     if$
1185 %     enclose.square.brackets
1186 %   }
1187 %   {
1188 %     address output
1189 %     new.column
1190 %     publisher empty$
1191 %     { bbl.nopublisher enclose.square.brackets }
1192 %     { publisher }
1193 %     if$
1194 %   }
1195 % if$
1196 % output
1197 %}
1198 FUNCTION {output.address.publisher}
1199 {
1200   address empty$
1201   {
1202     bbl.nnoaddress
1203     publisher empty$
1204     { "~: " * bbl.nopublisher * enclose.square.brackets }
1205     { enclose.square.brackets "~: " * publisher * }
1206     if$
1207   }
1208   {
1209     address output
1210     new.column
1211     publisher empty$
1212     { bbl.nopublisher enclose.square.brackets }
1213     { publisher }
1214     if$
1215   }
1216   if$
1217   output
1218 }
1219
1220 </strict>
1221

```

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

```

1222 <*(modern | strict)>
1223 FUNCTION {output.address.publisher.date}
1224 {

```

```

1225 output.address.publisher
1226 format.date output
1227 }
1228 </!(modern | strict)>
1229

```

output.bibitem

```

1230 (*!natbib)
1231 %%FUNCTION {output.bibitem}
1232 %%{ newline$
1233 %% "\bibitem" write$
1234 %% cite$ bracing write$
1235 %% newline$
1236 %% language empty$
1237 %% { "english" 'curlanguage := }
1238 %% { language 'curlanguage := }
1239 %% if$
1240 %% "\selectlanguageifdefined" curlanguage bracing * write$
1241 %% newline$
1242 %% ""
1243 %% before.all 'output.state :=
1244 %%}
1245 FUNCTION {output.bibitem}
1246 { newline$
1247 "\bibitem" write$
1248 cite$ bracing write$
1249 newline$
1250 langid empty$
1251 { language empty$
1252 { "english" 'curlanguage := }
1253 { language 'curlanguage := }
1254 if$
1255 }
1256 { langid 'curlanguage := }
1257 if$
1258 "\selectlanguageifdefined" curlanguage bracing * write$
1259 newline$
1260 ""
1261 before.all 'output.state :=
1262 }
1263
1264 </!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 1265 (*natbib)
author.editor.full 1266 FUNCTION {format.full.names}
make.full.names 1267 { 's :=
output.bibitem 1268 language empty$
1269 { "english" 'curlanguage := }
1270 { language 'curlanguage := }

```

```

1271 if$
1272 #1 'nameptr :=
1273 s num.names$ 'numnames :=
1274 numnames 'namesleft :=
1275 { namesleft #0 > }
1276 { s nameptr
1277   "{vv~}{ll}" format.name$ 't :=
1278   nameptr #1 >
1279   {
1280     namesleft #1 >
1281     { ", " * t * }
1282     {
1283       numnames #2 >
1284       curlanguage "english" =
1285       and
1286       { ", " * }
1287       'skip$
1288       if$
1289       t "others" =
1290       %t "~others" =
1291       %or
1292       %{ " et~al." * }
1293       { " " bbl.etal * * }
1294       %{ " and " * t * }
1295       { " " bbl.and " " * * * t * }
1296       if$
1297     }
1298     if$
1299   }
1300   't
1301   if$
1302   nameptr #1 + 'nameptr :=
1303   namesleft #1 - 'namesleft :=
1304   }
1305 while$
1306 }
1307
1308 FUNCTION {author.full}
1309 { author empty$
1310   { "" }
1311   { author format.full.names }
1312   if$
1313 }
1314
1315 FUNCTION {editor.full}
1316 { editor empty$
1317   { "" }
1318   { editor format.full.names }
1319   if$
1320 }

```

```

1321
1322 FUNCTION {author.editor.full}
1323 { author empty$
1324   { editor empty$
1325     { "" }
1326     { editor format.full.names }
1327     if$
1328   }
1329   { author format.full.names }
1330 if$
1331 }
1332
1333 FUNCTION {make.full.names}
1334 { type$ "book" =
1335   type$ "inbook" =
1336   or
1337   'author.editor.full
1338   { type$ "proceedings" =
1339     'editor.full
1340     'author.full
1341     if$
1342   }
1343 if$
1344 }
1345
1346 % =====
1347 FUNCTION {output.bibitem}
1348 { newline$
1349   "\bibitem[" write$
1350   label write$
1351   "]" make.full.names duplicate$ short.list =
1352     { pop$ }
1353     { * }
1354   if$
1355   "]" * write$
1356   cite$ write$
1357   "]" write$
1358 %% language empty$
1359 %% { "english" 'curlanguage := }
1360 %% { language 'curlanguage := }
1361 %% if$
1362 langid empty$
1363   { language empty$
1364     { "english" 'curlanguage := }
1365     { language 'curlanguage := }
1366     if$
1367   }
1368   { langid 'curlanguage := }
1369 if$
1370 "\selectlanguageifdefined" curlanguage braceify * write$

```

```

1371 newline$
1372 ""
1373 before.all 'output.state :=
1374 }
1375 % =====
1376 %FUNCTION {output.bibitem}
1377 %{ newline$
1378 % "\bibitem" write$
1379 %% author.key.label
1380 %% year parenthesify *
1381 %% "; lbl:" label * *
1382 %% "; mfn:" make.full.names * *
1383 % label
1384 % make.full.names *
1385 % bracketise write$
1386 % cite$ braceify write$
1387 % newline$
1388 % language empty$
1389 % { "english" 'curlanguage := }
1390 % {language 'curlanguage := }
1391 % if$
1392 % "\selectlanguageifdefined" curlanguage braceify * write$
1393 % newline$
1394 % ""
1395 % before.all 'output.state :=
1396 %}
1397 % =====
1398 </natbib>
1399

```

n.dashify

```

1400 FUNCTION {n.dashify}
1401 { 't :=
1402 ""
1403 { t empty$ not }
1404 { t #1 #1 substring$ "-" =
1405   { t #1 #2 substring$ "--" = not
1406     { "--" *
1407       t #2 global.max$ substring$ 't :=
1408     }
1409     { { t #1 #1 substring$ "-" = }
1410       { "-" *
1411         t #2 global.max$ substring$ 't :=
1412       }
1413     while$
1414   }
1415   if$
1416 }
1417 { t #1 #1 substring$ *
1418   t #2 global.max$ substring$ 't :=
1419 }

```

```

1420     if$
1421   }
1422 while$
1423 }
1424

```

word.in

```

1425 FUNCTION {word.in}
1426 { bbl.iin
1427   " " * }
1428

```

format.btitle

```

1429 FUNCTION {format.btitle}
1430 { title
1431 }
1432

```

.or.space.connect

```

1433 FUNCTION {tie.or.space.connect}
1434 { duplicate$ text.length$ #3 <
1435   { "~" }
1436   { " " }
1437   if$
1438   swap$ * *
1439 }
1440

```

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

```

1441 FUNCTION {tie.connect}
1442 { "~"
1443   swap$ * *
1444 }
1445
1446

```

either.or.chec

```

1447 FUNCTION {either.or.check}
1448 { empty$
1449   'pop$
1450   { "can't use both " swap$ * " fields in " * cite$ * warning$ }
1451   if$
1452 }
1453

```

format.bvolume

```

1454 FUNCTION {format.bvolume}
1455 { volume empty$
1456   { "" }
1457   { bbl.vvol volume tie.connect
1458     series empty$
1459     'skip$
1460     { bbl.of space.word * series emphasize * }

```

```

1461     if$
1462     "volume and number" number either.or.check
1463   }
1464   if$
1465 }
1466

```

mat.number.series

```

1467 FUNCTION {format.number.series}
1468 { volume empty$
1469   { number empty$
1470     { series field.or.null }
1471     { series empty$
1472       { "there's a number but no series in " cite$ * warning$
1473         bbl.nnr }
1474       {
1475         %new.dblslash
1476         new.sentence
1477         series
1478         bbl.nr
1479         tie.or.space.connect}
1480       if$
1481       number tie.or.space.connect
1482     }
1483   } if$
1484 }
1485 { "" }
1486 if$
1487 }
1488

```

is.num

```

1489 FUNCTION {is.num}
1490 { chr.to.int$
1491   duplicate$ "0" chr.to.int$ < not
1492   swap$ "9" chr.to.int$ > not and
1493 }
1494

```

extract.num

```

1495 FUNCTION {extract.num}
1496 { duplicate$ 't :=
1497   "" 's :=
1498   { t empty$ not }
1499   { t #1 #1 substring$
1500     t #2 global.max$ substring$ 't :=
1501     duplicate$ is.num
1502     { s swap$ * 's := }
1503     { pop$ "" 't := }
1504   } if$
1505 }
1506 while$

```

```

1507 s empty$
1508 'skip$
1509 { pop$ s }
1510 if$
1511 }
1512
1513 (*debug)

```

eng.ord

```

1514 FUNCTION {eng.ord}
1515 { duplicate$ "1" swap$ *
1516 # -2 #1 substring$ "1" =
1517 { bbl.th * }
1518 { duplicate$ #-1 #1 substring$
1519 duplicate$ "1" =
1520 { pop$ bbl.st * }
1521 { duplicate$ "2" =
1522 { pop$ bbl.nd * }
1523 { "3" =
1524 { bbl.rd * }
1525 { bbl.th * }
1526 if$
1527 }
1528 if$
1529 }
1530 if$
1531 }
1532 if$
1533 }
1534 (/debug)
1535

```

convert.edition

```

1536 FUNCTION {convert.edition}
1537 { edition
1538 % edition extract.num "1" change.case$ 's :=
1539 % s "first" = s "1" = or
1540 % { bbl.first 't := }
1541 % { s "second" = s "2" = or
1542 % { bbl.second 't := }
1543 % { s "third" = s "3" = or
1544 % { bbl.third 't := }
1545 % { s "fourth" = s "4" = or
1546 % { bbl.fourth 't := }
1547 % { s "fifth" = s "5" = or
1548 % { bbl.fifth 't := }
1549 % { s #1 #1 substring$ is.num
1550 % { s eng.ord 't := }
1551 % { edition 't := }
1552 % if$
1553 % }

```



```

1554 %             if$
1555 %             }
1556 %             if$
1557 %             }
1558 %             if$
1559 %             }
1560 %             if$
1561 %             }
1562 % if$
1563 % t
1564 }
1565

```

format.edition

```

1566 FUNCTION {format.edition}
1567 { edition empty$
1568   { "" }
1569   { output.state mid.sentence =
1570     { convert.edition "l" change.case$ " " * bbl.edition * }
1571     { convert.edition "t" change.case$ " " * bbl.edition * }
1572     { convert.edition " " * bbl.edition * }
1573     { convert.edition " " * bbl.edition * }
1574     if$
1575   }
1576   if$
1577 }
1578
1579 INTEGERS { multiresult }
1580

```

multi.page.check

```

1581 FUNCTION {multi.page.check}
1582 { 't :=
1583   #0 'multiresult :=
1584   { multiresult not
1585     t empty$ not
1586     and
1587   }
1588   { t #1 #1 substring$
1589     duplicate$ "-" =
1590     swap$ duplicate$ "," =
1591     swap$ "+" =
1592     or or
1593     { #1 'multiresult := }
1594     { t #2 global.max$ substring$ 't := }
1595     if$
1596   }
1597   while$
1598   multiresult
1599 }
1600

```

format.pages

```
1601 %%FUNCTION {format.pages}
1602 %%{ pages empty$
1603 %%  { "" }
1604 %%  { pages multi.page.check
1605 %%    { bbl.ppages pages n.dashify tie.connect }
1606 %%    { bbl.ppage pages tie.connect }
1607 %%  if$
1608 %%  }
1609 %% if$
1610 %%}
1611 FUNCTION {format.pages}
1612 { eid empty$
1613   {
1614     pages empty$
1615     { "" }
1616     { pages multi.page.check
1617       { bbl.ppages pages n.dashify tie.connect }
1618       { bbl.ppage pages tie.connect }
1619     if$
1620   }
1621   if$
1622 }
1623 { eid multi.page.check
1624   { bbl.ppages eid n.dashify tie.connect }
1625   { bbl.ppage eid tie.connect }
1626   if$
1627 }
1628 if$
1629 }
1630
```

format.pages.page

```
1631 %%FUNCTION {format.pages.page}
1632 %%{ pages empty$
1633 %%  { pagetotal empty$
1634 %%  { "" }
1635 %%  { pagetotal bbl.pages tie.connect }
1636 %%  if$}
1637 %%  { format.pages}
1638 %% if$
1639 %%}
1640 FUNCTION {format.pages.page}
1641 { eid empty$
1642   { pages empty$
1643     { pagetotal empty$
1644       { "" }
1645       { pagetotal bbl.pages tie.connect }
1646     if$
1647   }
1648   { format.pages}

```

```

1649     if$
1650   }
1651   { format.pages }
1652   if$
1653 }
1654

```

mat.vol.num.pages

```

1655 FUNCTION {format.vol.num.pages}
1656 { volume field.or.null
1657   number empty$
1658   'skip$
1659   {
1660     ", no." number tie.or.space.connect *
1661     volume empty$
1662     { "there's a number but no volume in " cite$ * warning$ }
1663     'skip$
1664     if$
1665   }
1666   if$
1667   pages empty$
1668   'skip$
1669   { duplicate$ empty$
1670     { pop$ format.pages }
1671     { ": " * pages n.dashify * }
1672     if$
1673   }
1674   if$
1675 }
1676

```

format.volume

```

1677 FUNCTION {format.volume}
1678 { volume empty$
1679   { "" }
1680   { bbl.vvol volume tie.or.space.connect }
1681   if$
1682 }
1683

```

format.number

```

1684 FUNCTION {format.number}
1685 { number empty$
1686   { "" }
1687   { bbl.nr number tie.or.space.connect }
1688   if$
1689 }
1690
1691 < *debug >

```

mat.chapter.pages

```

1692 FUNCTION {format.chapter.pages}

```

```

1693 { chapter empty$
1694   'format.pages
1695   { type empty$
1696     { bbl.chapter }
1697     { type "l" change.case$ }
1698     if$
1699     chapter tie.or.space.connect
1700     pages empty$
1701     'skip$
1702     { ", " * format.pages * }
1703     if$
1704   }
1705 if$
1706 }
1707 </debug>
1708

```

empty.misc.check

```

1709 FUNCTION {empty.misc.check}
1710 { author empty$ title empty$ howpublished empty$
1711   month empty$ year empty$ note empty$
1712   and and and and and
1713   key empty$ not and
1714   { "all relevant fields are empty in " cite$ * warning$ }
1715   'skip$
1716   if$
1717 }
1718

```

ormat.thesis.type

```

1719 FUNCTION {format.thesis.type}
1720 { type empty$
1721   'skip$
1722   { pop$
1723     <!utf8> type "t" change.case$
1724     <utf8> type
1725   }
1726   if$
1727 }
1728

```

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

```

1729 %FUNCTION {format.techrep.type.number}
1730 %{ type empty$
1731 %   { bbl.techreport }
1732 %   'type
1733 %   if$
1734 %   number empty$
1735 %<!utf8> { "t" change.case$ }
1736 %<utf8> { "" }
1737 %   { number tie.or.space.connect }
1738 %   if$

```

```

1739 %}
1740
1741 FUNCTION {format.techreport.type}
1742 { type empty$
1743   { bbl.techreport }
1744   'type
1745   if$
1746 }
1747

```

**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`.

```

1748 <!*long>
1749 FUNCTION {author.before}
1750 {
1751   author empty$
1752   'skip$
1753   {author num.names$ #4 <
1754     {format.authors output
1755       new.sentence}
1756     'skip$
1757     if$}
1758   if$
1759 }
1760 </!long>

```

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

```

1761 <*long>
1762 FUNCTION {author.before}
1763 {
1764   author empty$
1765   'skip$
1766   { format.authors output
1767     new.sentence
1768   }
1769   if$
1770 }
1771 </long>
1772

```

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

```

1773 %<!*long>
1774 %%FUNCTION {bookauthor.before}

```

```

1775 %%{
1776 %% bookauthor empty$
1777 %% 'skip$
1778 %% {bookauthor num.names$ #4 <
1779 %% {format.bookauthors output
1780 %% new.sentence}
1781 %% 'skip$
1782 %% if$}
1783 %% if$
1784 %%}
1785 %%</*!long>
1786 %%<*long>
1787 %%FUNCTION {bookauthor.before}
1788 %%{
1789 %% bookauthor empty$
1790 %% 'skip$
1791 %% { format.bookauthors output
1792 %% new.sentence
1793 %% }
1794 %% if$
1795 %%}
1796 %%</long>
1797

```

`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).

```

1798 <!*modern>
1799 FUNCTION {author.after}
1800 {
1801 author empty$
1802 'skip$
1803 {format.authors.after output
1804 new.semicolumn }
1805 if$
1806 }
1807 </*!modern>
1808 <*modern>
1809 </*!long>
1810 FUNCTION {author.after}
1811 {
1812 author empty$
1813 'skip$
1814 {author num.names$ #3 >
1815 {format.authors.after output
1816 new.semicolumn }
1817 'skip$

```

```

1818     if$}
1819   if$
1820 }
1821 </!long>
1822 < *long>
1823 FUNCTION {author.after} { }
1824 </long>
1825 </modern>
1826

```

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

```

1827 FUNCTION {bookauthor.after}
1828 {
1829   bookauthor empty$
1830     'skip$
1831     {format.bookauthors.after output
1832       new.semicolumn }
1833   if$
1834 }
1835

```

`organization.after`

```

1836 FUNCTION {editor.organization.after}
1837 {
1838   compiler empty$
1839     {}
1840     { format.compiler.after output
1841       new.semicolumn
1842     }
1843   if$
1844   editor empty$
1845     {}
1846     { format.editors.after output
1847       new.semicolumn
1848     }
1849   if$
1850   organization empty$
1851     {}
1852     {organization output
1853       new.semicolumn
1854     }
1855   if$
1856 }
1857

```

`format.url`

```

1858 FUNCTION {format.url}
1859 { url empty$
1860   { "" }
1861   {

```

```

1862 (!(modern|strict))      "\BibUrl{ " url * "}" *
1863 (modern|strict)        "URL: \BibUrl{" url * "}" *
1864      urldate empty$
1865      { "" }
1866      { " (" bbl.urldate * ": " * urldate * ")" * }
1867      if$ *
1868    }
1869  if$
1870 }
1871

```

format.annotate

```

1872 FUNCTION {format.annotate}
1873 { annotate empty$
1874   { "" }
1875 { after.sentence 'output.state :=
1876   "\BibAnnote{" annotate add.period$ * "}" *
1877 }
1878 if$
1879 }
1880

```

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

```

1881 FUNCTION {format.isbn}
1882 {
1883   isbn empty$
1884   { "" }
1885   { "ISBN: ~\href{http://isbndb.com/search-all.html?kw=" isbn *
1886     "}{ " * isbn * "}" *
1887   }
1888   if$
1889 }
1890

```

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.

```

1891 (*eprint)
1892 FUNCTION {add.doi}
1893 { duplicate$ empty$
1894   'skip$
1895   { doi empty$
1896     'skip$
1897     { "\href{http://dx.doi.org/" doi * "}{ " * swap$ * "}" * }
1898     if$
1899   }
1900   if$

```



```

1901 }
1902 </eprint>
    If .bst style is compiled without eprint option, we just ignore doi field.
1903 <!*eprint>
1904 FUNCTION {add.doi} { }
1905 </!*eprint>
1906

```

`add.medium` New in version 2.

```

1907 <!*strict>
1908 FUNCTION {add.medium} { }
1909 </!*strict>
1910 <*strict>
1911 FUNCTION {add.medium}
1912 { duplicate$ empty$
1913   'skip$
1914   { medium empty$
1915     { " " * bbl.medium enclose.square.brackets * }
1916     { " " * medium enclose.square.brackets * }
1917 %%     { bbl.medium enclose.square.brackets * }
1918 %%     { medium enclose.square.brackets * }
1919     if$
1920   }
1921   if$
1922 }
1923 </strict>
1924

```

## 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.

```

1925 <!*eprint>
1926 %FUNCTION {format.eprint}
1927 %{ eprint empty$
1928 %   { "" }
1929 %   { eprintclass empty$

```

```

1930 % { " \href{http://arxiv.org/abs/" eprint * "}" * "{" * eprint * "}" * }
1931 % { eprinttype empty$
1932 % { " \href{http://arxiv.org/abs/" eprint * "}" *
1933 % {" * eprintclass * "/" * eprint * "}" *
1934 % }
1935 % { " \href{http://arxiv.org/abs/" eprint * "}" *
1936 % {" * eprinttype * ":" * eprintclass * "/" * eprint * "}" *
1937 % }
1938 % if$}
1939 % if$}
1940 %if$}
1941
1942 %FUNCTION {format.eprint}
1943 %{ eprint empty$
1944 % { "" }
1945 % { eprinttype empty$
1946 % { "" }
1947 % { eprinttype "~: " *}
1948 % if$
1949 % eprintclass empty$
1950 % { }
1951 % { eprintclass * "/" *}
1952 % if$
1953 % eprint *
1954 % }
1955 % if$
1956 % url empty$
1957 % { }
1958 % { "\href{" url * "}" * swap$ * "}" *}
1959 % if$
1960 %}
1961
1962 FUNCTION {format.eprint}
1963 { eprint empty$
1964 { "" }
1965 { eprinttype empty$
1966 { "" }
1967 { eprinttype "~: " *}
1968 if$
1969 eprintclass empty$
1970 { }
1971 { eprintclass * "/" *}
1972 if$
1973 url empty$
1974 { eprint * }
1975 { "\href{" * url * "}" * eprint * "}" *}
1976 if$
1977 }
1978 if$
1979 }

```

```

1980
1981 FUNCTION {output.eprint.url}
1982 {
1983   eprint.empty$
1984   { format.url output }
1985   { format.eprint output }
1986   if$
1987 }
1988
1989 </eprint>
1990
1991 <!eprint>
1992 FUNCTION {output.eprint.url}
1993 {
1994   format.url output
1995 }
1996 </!eprint>
1997

```

## 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`.

```

1998 FUNCTION {article}
1999 {
2000   output.bibitem
2001   author.before

```

```

2002 <natbib> author format.key output
2003 format.title add.medium "title" output.check
2004 new.slash
2005 author.after
2006 new.dblslash
2007 journal emphasize add.doi "journal" output.check % new in v.2
2008 new.block
2009 format.date "year" output.check
2010 new.block
2011 format.volume output
2012 format.number output
2013 new.block
2014 format.pages.page output
2015 new.block
2016 note output
2017 new.sentence
2018 % format.url output
2019 output.eprint.url
2020 format.annotate output
2021 fin.entry
2022 }
2023

```

**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.

```

2024 FUNCTION {book}
2025 {
2026   output.bibitem
2027   author.before
2028 <natbib> author format.key output
2029   format.btitle add.doi add.medium "title" output.check
2030   new.slash
2031   author.after
2032   editor.organization.after
2033   new.sentence
2034   format.number.series output
2035   new.block
2036   format.edition output
2037   new.block
2038   output.address.publisher
2039   format.date "year" output.check
2040   new.block
2041   format.bvolume output
2042   new.block

```

```

2043 format.pages.page output
2044 new.block
2045 (eprint) format.isbn output
2046 (eprint) new.block
2047 note output
2048 new.sentence
2049 % format.url output
2050 output.eprint.url
2051 format.annotate output
2052 fin.entry
2053 }
2054

```

**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`, `howpublished`, `type`, `note`, `location`, `chapter`, `pages`, `pagetotal`, `addendum`, `pubstate`, `doi`, `eprint`, `eprintclass`, `eprinttype`, `url`, `urldate`.

```

2055 FUNCTION {booklet}
2056 {
2057   output.bibitem
2058   author.before
2059 (natbib) author format.key output
2060 format.title add.doi add.medium "title" output.check
2061 new.slash
2062 author.after
2063 editor.organization.after
2064 new.block
2065 howpublished output
2066 address output
2067 format.date "year" output.check
2068 new.block
2069 note output
2070 new.sentence
2071 % format.url output
2072 output.eprint.url
2073 format.annotate output
2074 fin.entry
2075 }
2076

```

**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`.

```

2077 FUNCTION {inbook}

```

```

2078 {
2079   output.bibitem
2080   author.before
2081 (natbib)  author format.key output
2082   format.btitle add.doi add.medium "title" output.check
2083   new.slash
2084   author.after
2085   new.dblslash
2086 %   bookauthor.before
2087   booktitle "booktitle" output.check
2088   new.slash
2089   bookauthor.after
2090   editor.organization.after
2091   new.block
2092   format.edition output
2093   new.block
2094   format.number.series output
2095   new.sentence
2096   output.address.publisher
2097   format.date "year" output.check
2098   new.block
2099   format.bvolume output
2100   new.block
2101   format.pages.page output
2102   new.block
2103 (eprint)  format.isbn output
2104 (eprint)  new.block
2105   note output
2106   new.sentence
2107 %   format.url output
2108   output.eprint.url
2109   format.annotate output
2110   fin.entry
2111 }
2112

```

**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.

```

2113 FUNCTION {incollection}
2114 {
2115   output.bibitem
2116   author.before
2117 (natbib)  author format.key output
2118   new.sentence

```

```

2119 format.title add.doi add.medium "title" output.check
2120 new.slash
2121 author.after
2122 new.dblslash
2123 booktitle "booktitle" output.check
2124 new.slash
2125 editor.organization.after
2126 new.block
2127 output.address.publisher
2128 format.date "year" output.check
2129 new.block
2130 format.bvolume output
2131 format.number.series output
2132 new.block
2133 format.pages.page output
2134 new.block
2135 note output
2136 new.sentence
2137 % format.url output
2138 output.eprint.url
2139 format.annotate output
2140 fin.entry
2141 }
2142

```

**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.

```

2143 FUNCTION {proceedings}
2144 {
2145   output.bibitem
2146   (natbib) editor format.key output
2147   format.btitle add.doi add.medium "title" output.check
2148   new.slash
2149   editor.organization.after
2150   new.block
2151   format.bvolume output
2152   format.number.series output
2153   % address empty$
2154   %   { publisher output
2155   %     format.date "year" output.check
2156   %   }
2157   %   { address output.nonnull
2158   %     format.date "year" output.check
2159   %     new.sentence
2160   %     publisher output

```

```

2161 %   }
2162 %   if$
2163   output.address.publisher
2164   format.date "year" output.check
2165   new.block
2166   note output
2167   new.sentence
2168 %   format.url output
2169   output.eprint.url
2170   format.annotate output
2171   fin.entry
2172 }
2173

```

**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.

```

2174 FUNCTION {inproceedings}
2175 { output.bibitem
2176   author.before
2177   (natbib) author format.key output
2178   new.sentence
2179   format.title add.doi add.medium "title" output.check
2180   new.slash
2181   author.after
2182   new.dblslash
2183   booktitle "booktitle" output.check
2184   new.slash
2185   editor.organization.after
2186   new.block
2187   format.bvolume output
2188   format.number.series output
2189   new.block
2190 %   address empty$
2191 %   { publisher output
2192 %     format.date "year" output.check
2193 %   }
2194 %   { address output.nonnull
2195 %     new.column
2196 %     publisher output
2197 %     format.date "year" output.check
2198 %   }
2199 %   if$
2200   output.address.publisher
2201   format.date "year" output.check
2202   new.block

```



```

2203 format.pages.page output
2204 new.block
2205 note output
2206 new.sentence
2207 % format.url output
2208 output.eprint.url
2209 format.annotate output
2210 fin.entry
2211 }
2212

```

**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.

```

2213 FUNCTION {manual}
2214 { output.bibitem
2215   author empty$
2216     { organization empty$
2217       'skip$
2218         { organization output.nonnull
2219           address output
2220         }
2221       if$
2222     }
2223   { format.authors output.nonnull }
2224   if$
2225 (natbib) author format.key output
2226 new.block
2227 format.btitle add.doi add.medium "title" output.check
2228 author empty$
2229   { organization empty$
2230     {
2231       address new.block.checka
2232       address output
2233     }
2234     'skip$
2235     if$
2236   }
2237   {
2238     organization address new.block.checkb
2239     organization output
2240     address output
2241   }
2242   if$
2243   format.edition output
2244   format.date "year" output.check
2245   new.block
2246   note output

```

```

2247 new.sentence
2248 % format.url output
2249 output.eprint.url
2250 format.annotate output
2251 fin.entry
2252 }
2253

```

**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`.

```

2254 FUNCTION {misc}
2255 { output.bibitem
2256   format.authors output
2257   (natbib) author format.key output
2258   title howpublished new.sentence.checkb
2259   format.title add.medium output
2260   howpublished new.block.checka
2261   howpublished output
2262   new.block
2263   format.date "year" output.check
2264   new.block
2265   note output
2266   new.sentence
2267   % format.url output
2268   output.eprint.url
2269   format.annotate output
2270   fin.entry
2271 }
2272

```

**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`

```

2273 FUNCTION {unpublished}
2274 { output.bibitem
2275   author.before
2276   (natbib) author format.key output
2277   format.btitle "title" output.check
2278   new.slash
2279   author.after
2280   editor.organization.after
2281   new.block
2282   format.date "year" output.check
2283   new.block
2284   note "note" output.check

```

```

2285 new.sentence
2286 % format.url output
2287 output.eprint.url
2288 format.annotate output
2289 fin.entry
2290 }
2291

```

**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.

```

2292 FUNCTION {online}
2293 { output.bibitem
2294   format.authors output
2295 (natbib) author format.key output
2296   title howpublished new.sentence.checkb
2297   format.title add.doi add.medium "title" output.check
2298 %   howpublished new.block.checka
2299   howpublished new.dblslash.checka
2300   howpublished enclose.square.brackets output
2301   editor.organization.after
2302   new.sentence
2303   new.block
2304   output.address.publisher
2305   format.date output
2306   new.block
2307 %   format.url output
2308   output.eprint.url
2309   new.sentence
2310   note output
2311   format.annotate output
2312   fin.entry
2313 }
2314

```

**internet** New in version 2012.02.15.

```

www 2315 FUNCTION {internet} {online}
webpage 2316 FUNCTION {www} {online}
ielectronic 2317 FUNCTION {webpage} {online}
2318 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

```

2319 <!(modern | strict)>
2320 FUNCTION {thesis}
2321 { output.bibitem
2322   format.authors "author" output.check
2323 <(natbib)> author format.key output
2324   new.sentence
2325   format.btitle "title" output.check
2326   new.column
2327   bbl.phdthesis format.thesis.type output.nonnull
2328   new.slash
2329   school "school" output.check
2330   new.block
2331   output.address.publisher.date
2332   new.block
2333   format.pages.page output
2334   note output
2335   new.sentence
2336   format.url output
2337   format.annotate output
2338   fin.entry
2339 }
2340 </!(modern | strict)>
2341 <*modern | strict>
2342 FUNCTION {thesis}
2343 { output.bibitem
2344   format.authors "author" output.check
2345 <(natbib)> author format.key output
2346   new.sentence
2347   format.btitle add.doi add.medium "title" output.check
2348   new.column
2349 % bbl.phdthesis format.thesis.type output.nonnull
2350   type "type" output.check
2351   new.column
2352   number output
2353   new.slash
2354   format.authors.after output
2355   new.semicolumn
2356   school "school" output.check
2357   new.block
2358   output.address.publisher
2359   format.date "year" output.check
2360   new.block
2361   format.pages.page output
2362   new.block
2363   note output
2364   new.sentence
2365 % format.url output
2366   output.eprint.url
2367   format.annotate output
2368   fin.entry

```

```
2369 }
2370 </modern | strict>
2371
```

**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.

```
2372 %FUNCTION {report}
2373 %{
2374 %   output.bibitem
2375 %   author.before
2376 %   new.sentence
2377 %   format.title add.doi add.medium "title" output.check
2378 %   new.column
2379 %%   format.techrep.type.number output.nonnull
2380 %   type "type" output.check
2381 %   new.slash
2382 %   author.after
2383 %   editor.organization.after
2384 %   new.block
2385 %   address output
2386 %   new.column
2387 %   institution "institution" output.check
2388 %   format.date "year" output.check
2389 %   new.block
2390 %   note output
2391 %   new.block           % v.2
2392 %   format.pages.page output % v.2
2393 %   new.sentence
2394 %%   format.url output
2395 %   output.eprint.url
2396 %   format.annotate output
2397 %   fin.entry
2398 %}
2399 FUNCTION {report}
2400 {
2401   output.bibitem
2402 %   author.before
2403 %   new.sentence
2404   format.title add.doi add.medium "title" output.check
2405 <natbib>   title format.key output
2406   new.column
2407 %   format.techrep.type.number output.nonnull
2408 %   type "type" output.check
2409 %   format.report.type.number "type" output.check
2410   type "type" output.check
```

```

2411 new.column
2412 number output
2413 new.slash
2414 institution "institution" output.check
2415 new.semicolumn
2416 format.chief.after output % from editor field
2417 new.semicolumn
2418 format.executor.after output % from author field
2419 new.block
2420 address output
2421 new.column
2422 organization output
2423 format.date "year" output.check
2424 new.block % v.2
2425 format.pages.page output % v.2
2426 new.sentence % или new.block ?
2427 output.eprint.url
2428 new.block
2429 note output
2430 format.annotate output
2431 fin.entry
2432 }
2433

```

## 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.

```

2434 <*(modern | strict)>
2435 FUNCTION {phdthesis}
2436 { output.bibitem
2437   format.authors "author" output.check
2438   new.sentence
2439   format.btitle "title" output.check
2440   new.column
2441   bbl.phdthesis format.thesis.type output.nonnull
2442   new.slash
2443   school "school" output.check
2444   new.block
2445   output.address.publisher.date
2446   new.block
2447   format.pages.page output
2448   note output
2449   new.sentence
2450   format.url output
2451   format.annotate output

```

```

2452 fin.entry
2453 }
2454 <!(modern | strict)>
2455 <{*modern | strict}>
2456 FUNCTION {phdthesis}
2457 { output.bibitem
2458   format.authors "author" output.check
2459 <natbib> author format.key output
2460   new.sentence
2461   format.btitle add.doi add.medium "title" output.check
2462   new.column
2463   bbl.phdthesis format.thesis.type output.nonnull
2464   new.column
2465   number output
2466   new.slash
2467   format.authors.after output
2468   new.semicolumn
2469   school "school" output.check
2470   new.block
2471   output.address.publisher
2472   format.date "year" output.check
2473   new.block
2474   format.pages.page output
2475   new.block
2476   note output
2477   new.sentence
2478 % format.url output
2479   output.eprint.url
2480   format.annotate output
2481   fin.entry
2482 }
2483 </modern | strict>
2484

```

**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.

```

2485 <*(modern | strict)>
2486 FUNCTION {masterthesis}
2487 { output.bibitem
2488   format.authors "author" output.check
2489 <natbib> author format.key output
2490   new.sentence
2491   format.btitle "title" output.check
2492   new.column
2493   bbl.mthesis format.thesis.type output.nonnull
2494   new.slash
2495   school "school" output.check
2496   new.block
2497   output.address.publisher.date
2498   new.block
2499   format.pages.page output

```

```

2500 note output
2501 new.sentence
2502 format.url output
2503 format.annotate output
2504 fin.entry
2505 }
2506 <!(modern | strict)>
2507 <*modern | strict>
2508 FUNCTION {masterthesis}
2509 { output.bibitem
2510   format.authors "author" output.check
2511 <natbib> author format.key output
2512   new.sentence
2513   format.btitle add.doi add.medium "title" output.check
2514   new.column
2515   bbl.mthesis format.thesis.type output.nonnull
2516   new.column
2517   number output
2518   new.slash
2519   format.authors.after output
2520   new.semicolumn
2521   school "school" output.check
2522   new.block
2523   output.address.publisher
2524   format.date "year" output.check
2525   new.block
2526   format.pages.page output
2527   new.block
2528   note output
2529   new.sentence
2530 % format.url output
2531   output.eprint.url
2532   format.annotate output
2533   fin.entry
2534 }
2535 </modern | strict>
2536

```

**dscithesis** 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.

```

2537 <*(modern | strict)>
2538 FUNCTION {dscithesis}
2539 { output.bibitem
2540   format.authors "author" output.check
2541 <natbib> author format.key output
2542   new.sentence
2543   format.btitle "title" output.check
2544   new.column
2545   bbl.dscithesis format.thesis.type output.nonnull
2546   new.slash
2547   school "school" output.check

```



```

2548 new.block
2549 output.address.publisher.date
2550 new.block
2551 format.pages.page output
2552 note output
2553 new.sentence
2554 format.url output
2555 format.annotate output
2556 fin.entry
2557 }
2558 </!(modern | strict)>
2559 <*modern | strict>
2560 FUNCTION {dscithesis}
2561 { output.bibitem
2562   format.authors "author" output.check
2563   (natbib) author format.key output
2564   new.sentence
2565   format.btitle add.doi add.medium "title" output.check
2566   new.column
2567   bbl.dscithesis format.thesis.type output.nonnull
2568   new.column
2569   number output
2570   new.slash
2571   format.authors.after output
2572   new.semicolumn
2573   school "school" output.check
2574   new.block
2575   output.address.publisher
2576   format.date "year" output.check
2577   new.block
2578   format.pages.page output
2579   new.block
2580   note output
2581   new.sentence
2582 % format.url output
2583 output.eprint.url
2584 format.annotate output
2585 fin.entry
2586 }
2587 </modern | strict>
2588

```

conference

```

2589 FUNCTION {conference} { inproceedings }
2590

```

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.

```

2591 %FUNCTION {techreport}
2592 %{
2593 % output.bibitem

```

```

2594 % author.before
2595 % new.sentence
2596 % format.title add.doi add.medium "title" output.check
2597 % new.column
2598 % format.techrep.type.number output.nonnull
2599 % new.slash
2600 % author.after
2601 % editor.organization.after
2602 % new.block
2603 % address output
2604 % new.column
2605 % institution "institution" output.check
2606 % format.date "year" output.check
2607 % new.block
2608 % note output
2609 % new.block % v.2
2610 % format.pages.page output % v.2
2611 % new.sentence
2612 %% format.url output
2613 % output.eprint.url
2614 % format.annotate output
2615 % fin.entry
2616 %}
2617
2618 FUNCTION {techreport}
2619 {
2620 output.bibitem
2621 % author.before
2622 % new.sentence
2623 format.title add.doi add.medium "title" output.check
2624 (natbib) title format.key output
2625 new.column
2626 % format.techrep.type.number output.nonnull
2627 % type "type" output.check
2628 % format.report.type.number "type" output.check
2629 % type output
2630 format.techreport.type output
2631 new.column
2632 number output
2633 new.slash
2634 institution "institution" output.check
2635 new.semicolumn
2636 format.chief.after output % from editor field
2637 new.semicolumn
2638 format.executor.after output % from author field
2639 new.block
2640 address output
2641 new.column
2642 organization output
2643 format.date "year" output.check

```

```

2644 new.block          % v.2
2645 format.pages.page output % v.2
2646 new.sentence % или new.block ?
2647 output.eprint.url
2648 new.block
2649 note output
2650 format.annotate output
2651 fin.entry
2652 }
2653
2654

```

default.type

```

2655 FUNCTION {default.type} { misc }
2656

```

## 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.

```

2657 MACRO {jan} {"Jan."}
2658 MACRO {feb} {"Feb."}
2659 MACRO {mar} {"Mar."}
2660 MACRO {apr} {"Apr."}
2661 MACRO {may} {"May"}
2662 MACRO {jun} {"Jun."}
2663 MACRO {jul} {"Jul."}
2664 MACRO {aug} {"Aug."}
2665 MACRO {sep} {"Sep."}
2666 MACRO {oct} {"Oct."}
2667 MACRO {nov} {"Nov."}
2668 MACRO {dec} {"Dec."}

```

## 2.8 Journal Abbreviations

### 2.8.1 Physics and astronomy

Borrowed from physjour.mbs of package custom-bib.

```

2669 MACRO {aa}{"Astron. \& Astrophys."}
2670 MACRO {aasup}{"Astron. \& Astrophys. Suppl. Ser."}
2671 MACRO {aj} {"Astron. J."}
2672 MACRO {aph} {"Acta Phys."}
2673 MACRO {advp} {"Adv. Phys."}
2674 MACRO {ajp} {"Amer. J. Phys."}
2675 MACRO {ajm} {"Amer. J. Math."}
2676 MACRO {amsci} {"Amer. Sci."}
2677 MACRO {anofd} {"Ann. Fluid Dyn."}
2678 MACRO {am} {"Ann. Math."}
2679 MACRO {ap} {"Ann. Phys. (NY)"}

```

2680 MACRO {adp} {"Ann. Phys. (Leipzig)"}  
2681 MACRO {ao} {"Appl. Opt."}  
2682 MACRO {apl} {"Appl. Phys. Lett."}  
2683 MACRO {app} {"Astroparticle Phys."}  
2684 MACRO {apj} {"Astrophys. J."}  
2685 MACRO {apjsup} {"Astrophys. J. Suppl."}  
2686 MACRO {apss} {"Astrophys. Space Sci."}  
2687 MACRO {araa} {"Ann. Rev. Astron. Astrophys."}  
2688 MACRO {baas} {"Bull. Amer. Astron. Soc."}  
2689 MACRO {baps} {"Bull. Amer. Phys. Soc."}  
2690 MACRO {cmp} {"Comm. Math. Phys."}  
2691 MACRO {cpam} {"Commun. Pure Appl. Math."}  
2692 MACRO {cppcf} {"Comm. Plasma Phys. \& Controlled Fusion"}  
2693 MACRO {cpc} {"Comp. Phys. Comm."}  
2694 MACRO {cqg} {"Class. Quant. Grav."}  
2695 MACRO {cra} {"C. R. Acad. Sci. A"}  
2696 MACRO {fed} {"Fusion Eng. \& Design"}  
2697 MACRO {ft} {"Fusion Tech."}  
2698 MACRO {grg} {"Gen. Relativ. Gravit."}  
2699 MACRO {ieeens} {"IEEE Trans. Nucl. Sci."}  
2700 MACRO {ieeeps} {"IEEE Trans. Plasma Sci."}  
2701 MACRO {ijimw} {"Interntl. J. Infrared \& Millimeter Waves"}  
2702 MACRO {ip} {"Infrared Phys."}  
2703 MACRO {irp} {"Infrared Phys."}  
2704 MACRO {jap} {"J. Appl. Phys."}  
2705 MACRO {jasa} {"J. Acoust. Soc. America"}  
2706 MACRO {jcp} {"J. Comp. Phys."}  
2707 MACRO {jchp} {"J. Chem. Phys."}  
2708 MACRO {jetp} {"Sov. Phys.--JETP"}  
2709 MACRO {jfe} {"J. Fusion Energy"}  
2710 MACRO {jfm} {"J. Fluid Mech."}  
2711 MACRO {jmp} {"J. Math. Phys."}  
2712 MACRO {jne} {"J. Nucl. Energy"}  
2713 MACRO {jnec} {"J. Nucl. Energy, C: Plasma Phys., Accelerators, Thermonucl. Res."}  
2714 MACRO {jnm} {"J. Nucl. Mat."}  
2715 MACRO {jpc} {"J. Phys. Chem."}  
2716 MACRO {jpp} {"J. Plasma Phys."}  
2717 MACRO {jpsj} {"J. Phys. Soc. Japan"}  
2718 MACRO {jsi} {"J. Sci. Instrum."}  
2719 MACRO {jvst} {"J. Vac. Sci. \& Tech."}  
2720 MACRO {nat} {"Nature"}  
2721 MACRO {nature} {"Nature"}  
2722 MACRO {nedf} {"Nucl. Eng. \& Design/Fusion"}  
2723 MACRO {nf} {"Nucl. Fusion"}  
2724 MACRO {nim} {"Nucl. Inst. \& Meth."}  
2725 MACRO {nimpr} {"Nucl. Inst. \& Meth. in Phys. Res."}  
2726 MACRO {np} {"Nucl. Phys."}  
2727 MACRO {npb} {"Nucl. Phys. B"}  
2728 MACRO {nt/f} {"Nucl. Tech./Fusion"}  
2729 MACRO {npbpc} {"Nucl. Phys. B (Proc. Suppl.)"}

2730 MACRO {inc} {"Nuovo Cimento"}  
 2731 MACRO {nc} {"Nuovo Cimento"}  
 2732 MACRO {pf} {"Phys. Fluids"}  
 2733 MACRO {pfa} {"Phys. Fluids A: Fluid Dyn."}  
 2734 MACRO {pfb} {"Phys. Fluids B: Plasma Phys."}  
 2735 MACRO {pl} {"Phys. Lett."}  
 2736 MACRO {pla} {"Phys. Lett. A"}  
 2737 MACRO {plb} {"Phys. Lett. B"}  
 2738 MACRO {prep} {"Phys. Rep."}  
 2739 MACRO {pnas} {"Proc. Nat. Acad. Sci. USA"}  
 2740 MACRO {pp} {"Phys. Plasmas"}  
 2741 MACRO {pop} {"Phys. Plasmas"}  
 2742 MACRO {ppcf} {"Plasma Phys. \& Controlled Fusion"}  
 2743 MACRO {phitrsl} {"Philos. Trans. Roy. Soc. London"}  
 2744 MACRO {prl} {"Phys. Rev. Lett."}  
 2745 MACRO {pr} {"Phys. Rev."}  
 2746 MACRO {physrev} {"Phys. Rev."}  
 2747 MACRO {pra} {"Phys. Rev. A"}  
 2748 MACRO {prb} {"Phys. Rev. B"}  
 2749 MACRO {prc} {"Phys. Rev. C"}  
 2750 MACRO {prd} {"Phys. Rev. D"}  
 2751 MACRO {pre} {"Phys. Rev. E"}  
 2752 MACRO {ps} {"Phys. Scripta"}  
 2753 MACRO {procrsl} {"Proc. Roy. Soc. London"}  
 2754 MACRO {rmp} {"Rev. Mod. Phys."}  
 2755 MACRO {rsi} {"Rev. Sci. Inst."}  
 2756 MACRO {science} {"Science"}  
 2757 MACRO {sciam} {"Sci. Am."}  
 2758 MACRO {sam} {"Stud. Appl. Math."}  
 2759 MACRO {st} {"Sky and Telesc."}

## 2.8.2 Supplementary Journal Names

Borrowed from suppjour.mbs of package custom-bib.

2760 MACRO {cjp} {"Czech. J. Phys."}  
 2761 MACRO {el} {"Europhys. Lett."}  
 2762 MACRO {en} {"Europhys. News"}  
 2763 MACRO {fujitsustj} {"FUJITSU Sci. Tech. J."}  
 2764 MACRO {ieeed} {"IEEE Trans. Electron Devices"}  
 2765 MACRO {ieeeim} {"IEEE Trans. Instrum. Meas."}  
 2766 MACRO {ieeejqe} {"IEEE J. Quantum Electron."}  
 2767 MACRO {ieeem} {"IEEE Trans. Magn."}  
 2768 MACRO {ieeep1} {"IEEE Photonic Technol. Lett."}  
 2769 MACRO {ieeeuffc} {"IEEE Trans. Ultrason., Ferroelect., Freq. Cont."}  
 2770 MACRO {jem} {"J. Electron. Mater."}  
 2771 MACRO {jes} {"J. Electrochem. Soc."}  
 2772 MACRO {jetplett} {"JETP Lett."}  
 2773 MACRO {jjap} {"Japan. J. Appl. Phys."}  
 2774 MACRO {jpha} {"J. Phys. A: Math. Gen."}  
 2775 MACRO {jphb} {"J. Phys. B: At. Mol. Opt. Phys."}

2776 MACRO {jphbold} {"J. Phys. B: At. Mol. Phys."}  
 2777 MACRO {jphc} {"J. Phys.: Condens. Matter"}  
 2778 MACRO {jphcold} {"J. Phys. C: Solid State Phys."}  
 2779 MACRO {jphd} {"J. Phys. D: Appl. Phys."}  
 2780 MACRO {jvsta} {"J. Vac. Sci. Technol. A"}  
 2781 MACRO {jvstb} {"J. Vac. Sci. Technol. B"}  
 2782 MACRO {me} {"Microelectron. Eng."}  
 2783 MACRO {necrd} {"NEC Res.{\&} Develop."}  
 2784 MACRO {pa} {"Physica A"}  
 2785 MACRO {pb} {"Physica B"}  
 2786 MACRO {pc} {"Physica C"}  
 2787 MACRO {pd} {"Physica D"}  
 2788 MACRO {procieeee} {"Proc. IEEE"}  
 2789 MACRO {procspie} {"Proc. SPIE"}  
 2790 MACRO {pssa} {"Phys. Stat. Sol. A"}  
 2791 MACRO {pssb} {"Phys. Stat. Sol. B"}  
 2792 MACRO {rpp} {"Rep. Progr. Phys."}  
 2793 MACRO {sm} {"Synthet. Metal"}  
 2794 MACRO {sost} {"Solid State Technol."}  
 2795 MACRO {ss} {"Surf. Sci."}  
 2796 MACRO {ssc} {"Solid State Commun."}  
 2797 MACRO {sst} {"Semicond. Sci. Technol."}  
 2798 MACRO {suplatt} {"Superlatt. Microstr."}  
 2799 MACRO {sust} {"Supercond. Sci. Technol."}  
 2800 MACRO {znat} {"Z. Naturforsch."}

### 2.8.3 Optics

Borrowed from photjour.mbs.

2801 MACRO {appopt} {"Appl. Opt."}  
 2802 MACRO {bell} {"Bell Syst. Tech. J."}  
 2803 MACRO {ell} {"Electron. Lett."}  
 2804 MACRO {jasp} {"J. Appl. Spectr."}  
 2805 MACRO {jqe} {"IEEE J. Quantum Electron."}  
 2806 MACRO {jlwt} {"J. Lightwave Technol."}  
 2807 MACRO {jmo} {"J. Mod. Opt."}  
 2808 MACRO {josa} {"J. Opt. Soc. America"}  
 2809 MACRO {josaa} {"J. Opt. Soc. Amer.~A"}  
 2810 MACRO {josab} {"J. Opt. Soc. Amer.~B"}  
 2811 MACRO {jdp} {"J. Phys. (Paris)"}  
 2812 MACRO {oc} {"Opt. Commun."}  
 2813 MACRO {ol} {"Opt. Lett."}  
 2814 MACRO {os} {"Opt. Spectrosc."}  
 2815 MACRO {phtl} {"IEEE Photon. Technol. Lett."}  
 2816 MACRO {pspie} {"Proc. Soc. Photo-Opt. Instrum. Eng."}  
 2817 MACRO {vr} {"Vision Res."}  
 2818 MACRO {zph} {"Z. f. Physik"}  
 2819 MACRO {zphb} {"Z. f. Physik~B"}  
 2820 MACRO {zphd} {"Z. f. Physik~D"}

## 2.8.4 Physics of condensed Matter

```
2821 MACRO {sse} {"Solid-State Electron."}
2822 MACRO {pss} {"Phys. Sol. State"}
2823 MACRO {nl} {"Nano Lett."}
```

## 2.8.5 Soviet and Russian journals

To be extended.

```
2824 MACRO {sjpp} {"Sov. J. Plasma Phys."}
2825 MACRO {spd} {"Sov. Phys.--Doklady"}
2826 MACRO {sptp} {"Sov. Phys.--Tech. Phys."}
2827 MACRO      {spu} {"Sov. Phys.--Uspekhi"}
2828  $\langle$ !utf8 $\rangle$ MACRO {ufn} {"\CYRU\CYRF\CYRN"}
2829  $\langle$ utf8 $\rangle$ MACRO {ufn} {"\Y\phi H"}
2830 MACRO {pu} {"Phys.--Uspekhi"}
2831 MACRO {sjot} {"Sov. J. Opt. Technol."}
2832 MACRO {sjqe} {"Sov. J. Quantum Electron."}
2833 MACRO {sleb} {"Sov. Phys.--Leb. Inst. Rep."}
2834 MACRO {stph} {"Sov. Phys.--Techn. Phys."}
2835 MACRO {stphl} {"Sov. Techn. Phys. Lett."}
```

## 2.9 Main cycle

```
2836
2837 READ
2838
```

## 2.10 Sorting

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

```
2839  $\langle$ *sort | natbib $\rangle$ 
2840
```

sortify

```
2841 FUNCTION {sortify}
2842 { purify$
2843  $\langle$ !utf8 $\rangle$  "l" change.case$
2844 }
2845  $\langle$ /sort | natbib $\rangle$ 
2846
```

sort.format.names

```
2847  $\langle$ *sort $\rangle$ 
2848 %% =====
2849 %% This version from old Gost package
2850 %%<!*natbib>
2851 FUNCTION {sort.format.names}
2852 { 's :=
2853 #1 'nameptr :=
2854 ""
2855 s num.names$ 'numnames :=
```

```

2856 numnames 'namesleft :=
2857   { namesleft #0 > }
2858   { nameptr #1 >
2859     { " " * }
2860     'skip$
2861     if$
2862     s nameptr
2863     "{vv{ } }{ll{ }}{ f{ }}{ jj{ }}"
2864     format.name$ 't :=
2865     nameptr numnames = t "others" = and
2866     { "et al" * }
2867     %{ bbl.etal * }
2868     { t sortify * }
2869     if$
2870     nameptr #1 + 'nameptr :=
2871     namesleft #1 - 'namesleft :=
2872   }
2873   while$
2874 }
2875 %%</!natbib>
2876 %% This version from plainnat.bst
2877 %% It ignores second and subsequent authors but include year.
2878 %%<*natbib>
2879 %FUNCTION {sort.format.names}
2880 %{ 's :=
2881 % #1 'nameptr :=
2882 % ""
2883 % s num.names$ 'numnames :=
2884 % numnames 'namesleft :=
2885 %   { namesleft #0 > }
2886 %   {
2887 %     s nameptr "{vv{ } }{ll{ }}{ ff{ }}{ jj{ }}" format.name$ 't :=
2888 %     nameptr #1 >
2889 %     {
2890 %       " " *
2891 %       namesleft #1 = t "others" = and
2892 %       { "zzzz" * }
2893 %       { numnames #2 > nameptr #2 = and
2894 %         { "zz" * year field.or.null * " " * }
2895 %         'skip$
2896 %         if$
2897 %         t sortify *
2898 %       }
2899 %     }
2900 %   }
2901 %   { t sortify * }
2902 %   if$
2903 %   nameptr #1 + 'nameptr :=
2904 %   namesleft #1 - 'namesleft :=
2905 % }

```



```

2906 % while$
2907 %}
2908 %%</natbib>
2909 %% =====
2910

```

sort.format.title

```

2911 FUNCTION {sort.format.title}
2912 { 't :=
2913   "A " #2
2914   "An " #3
2915   "The " #4 t chop.word % Removes "The " if any
2916   chop.word           % Removes "An " if any
2917   chop.word           % Removes "A " if any
2918   sortify
2919   #1 global.max$ substring$
2920 }
2921

```

author.sort

```

2922 %% =====
2923 %% This version from old gost package.
2924 %%
2925 <!*natbib>
2926 FUNCTION {author.sort}
2927 { author empty$
2928   { key empty$
2929     { "to sort, need author or key in " cite$ * warning$
2930       ""
2931     }
2932     { key sortify }
2933   if$
2934 }
2935 {
2936   author num.names$ #4 <
2937     {author sort.format.names }
2938     {title sort.format.title}
2939   if$
2940 }
2941 if$
2942 }
2943 </!*natbib>
2944 %% This version from plainnat.bst
2945 <!*natbib>
2946 FUNCTION {author.sort}
2947 { author empty$
2948   { key empty$
2949     { "to sort, need author or key in " cite$ * warning$
2950       ""
2951     }
2952     { key sortify }

```

```

2953     if$
2954   }
2955   { author sort.format.names }
2956   if$
2957 }
2958 </natbib>
2959 %% =====
2960

```

author.title.sort

```

2961 <!*natbib|natbib>
2962 FUNCTION {author.title.sort}
2963 { author empty$
2964   { title empty$
2965     { key empty$
2966       { "to sort, need author, title, or key in " cite$ * warning$
2967         ""
2968       }
2969       { key sortify }
2970     }
2971   }
2972   { title sort.format.title }
2973 }
2974 }
2975 {
2976   author num.names$ #4 <
2977     {author sort.format.names }
2978     {title sort.format.title}
2979   }
2980 }
2981 if$
2982 }
2983 </!*natbib|natbib>
2984 <!*natbib|natbib>
2985 FUNCTION {author.editor.sort}
2986 { author empty$
2987   { editor empty$
2988     { key empty$
2989       { "to sort, need author, editor, or key in " cite$ * warning$
2990         ""
2991       }
2992       { key sortify }
2993     }
2994   }
2995   { editor sort.format.names }
2996 }
2997 }
2998 { author sort.format.names }
2999 if$
3000 }
3001

```

```

3002 FUNCTION {author.organization.sort}
3003 { author empty$
3004   { organization empty$
3005     { key empty$
3006       { "to sort, need author, organization, or key in " cite$ * warning$
3007         ""
3008       }
3009     { key sortify }
3010   } if$
3011 }
3012 { "The " #4 organization chop.word sortify }
3013 if$
3014 }
3015 { author sort.format.names }
3016 if$
3017 }
3018
3019 FUNCTION {editor.organization.sort}
3020 { editor empty$
3021   { organization empty$
3022     { key empty$
3023       { "to sort, need editor, organization, or key in " cite$ * warning$
3024         ""
3025       }
3026     { key sortify }
3027   } if$
3028 }
3029 { "The " #4 organization chop.word sortify }
3030 if$
3031 }
3032 { editor sort.format.names }
3033 if$
3034 }
3035 </natbib|natbib>
3036
3037

```

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

```

3038 <!*natbib>
3039 FUNCTION {presort}%#1
3040 {
3041   author.title.sort
3042   " "
3043   *
3044   year field.or.null sortify
3045   *
3046   " "
3047   *
3048   title field.or.null
3049   sort.format.title
3050   *

```

```

3051 #1 entry.max$ substring$
3052 'sort.key$ :=
3053 }
3054 </!natbib>
3055 <natbib>
3056 FUNCTION {presort}%#2
3057 { calc.label
3058   label sortify
3059   %author.title.sort
3060   " "
3061   *
3062   % ===== plainnat.bst =====
3063   % type$ "book" =
3064   % type$ "inbook" =
3065   % or
3066   % 'author.editor.sort
3067   % { type$ "proceedings" =
3068   %   'editor.organization.sort
3069   %   { type$ "manual" =
3070   %     'author.organization.sort
3071   %     'author.sort
3072   %     if$
3073   %   }
3074   %   if$
3075   % }
3076   % if$
3077   author.title.sort
3078   " "
3079   *
3080   year field.or.null sortify
3081   *
3082   " "
3083   *
3084   %cite$
3085   title field.or.null sort.format.title
3086   *
3087   #1 entry.max$ substring$
3088   'sort.label :=
3089   sort.label *
3090   % =====
3091   #1 entry.max$ substring$
3092   'sort.key$ :=
3093 }
3094 </natbib>
3095 </sort>
3096
3097 <!*sort>
3098 <natbib>
3099 INTEGERS { seq.num }
3100

```

```

3101 FUNCTION {init.seq}
3102 { #0 'seq.num :=}
3103
3104 EXECUTE {init.seq}
3105
3106 FUNCTION {int.to.fix}
3107 { "00000000" swap$ int.to.str$ *
3108  #-1 #10 substring$
3109 }
3110
3111 FUNCTION {presort}%#3
3112 {
3113   calc.label           % computes label
3114   label sortify       % initiates sort.label
3115   " "
3116   *
3117   seq.num #1 + 'seq.num := % advance seq.num
3118   seq.num int.to.fix   % prepend seq.num with 0s
3119   'sort.label :=      % set sort.label to seq.num
3120   sort.label *        % append seq.num to label
3121   #1 entry.max$ substring$ % cut if too long
3122   'sort.key$ :=      % set sort.key$
3123 }
3124 </natbib>
3125 </!sort>
3126
3127 < *sort | natbib >
3128 ITERATE {presort}
3129
3130 SORT
3131
3132 </sort | natbib >
3133

```

## 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.

```

3134 <!natbib>STRINGS { longest.label }
3135 <natbib>STRINGS { longest.label last.label next.extra }
3136
3137 <!natbib>INTEGERS { number.label longest.label.width }
3138 <natbib>INTEGERS { number.label longest.label.width last.extra.num }
3139

```

`ize.longest.label` Initialize those string.

```

3140 < *!natbib >
3141 FUNCTION {initialize.longest.label}

```

```

3142 { "" 'longest.label :=
3143 #1 'number.label :=
3144 #0 'longest.label.width :=
3145 }
3146 </!natbib>
3147 < *natbib>
3148 FUNCTION {initialize.longest.label}
3149 { "" 'longest.label :=
3150 #0 int.to.chr$ 'last.label :=
3151 "" 'next.extra :=
3152 #0 'longest.label.width :=
3153 #0 'last.extra.num :=
3154 #0 'number.label :=
3155 }
3156 </natbib>
3157
3158 EXECUTE {initialize.longest.label}
3159

```

ize.longest.label      Iterate though the list of entries to compute label.

```

3160 < *!natbib>
3161 FUNCTION {forward.pass}
3162 { number.label int.to.str$ 'label :=
3163 number.label #1 + 'number.label :=
3164 label width$ longest.label.width >
3165 { label 'longest.label :=
3166 label width$ 'longest.label.width :=
3167 }
3168 'skip$
3169 if$
3170 }
3171 </!natbib>
3172 < *natbib>
3173 FUNCTION {forward.pass}
3174 { last.label label =
3175 { last.extra.num #1 + 'last.extra.num :=
3176 last.extra.num int.to.chr$ 'extra.label :=
3177 }
3178 { "a" chr.to.int$ 'last.extra.num :=
3179 "" 'extra.label :=
3180 label 'last.label :=
3181 }
3182 if$
3183 number.label #1 + 'number.label :=
3184 }
3185 </natbib>
3186
3187 ITERATE {forward.pass}
3188

```

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

```

3189 (*natbib)
3190 FUNCTION {reverse.pass}
3191 { next.extra "b" =
3192   { "a" 'extra.label := }
3193   'skip$
3194   if$
3195   extra.label 'next.extra :=
3196   extra.label
3197   duplicate$ empty$
3198   'skip$
3199   { "{\natexlab{" swap$ * "}} * }
3200   if$
3201   'extra.label :=
3202   label extra.label * 'label :=
3203 }
3204
3205 REVERSE {reverse.pass}
3206
3207 FUNCTION {bib.sort.order}
3208 { sort.label 'sort.key$ :=
3209 }
3210
3211 ITERATE {bib.sort.order}
3212
3213 SORT
3214 </natbib>
3215

```

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

```

3216 FUNCTION {begin.bib}
3217 { "\begin{thebibliography}{ longest.label * }" * write$ newline$
3218   "\def\selectlanguageifdefined#1{" write$ newline$
3219     "\expandafter\ifx\csname date#1\endcsname\relax" write$ newline$
3220     % "\else\language\csname l@#1\endcsname\fi" write$ newline$
3221     "\else\selectlanguage{#1}\fi" write$ newline$
3222     "\providecommand*{\href}[2]{\small #2}" write$ newline$
3223     "\providecommand*{\url}[1]{\small #1}" write$ newline$
3224     "\providecommand*{\BibUrl}[1]{\url{#1}}" write$ newline$
3225     "\providecommand*{\BibAnnote}[1]{}" write$ newline$
3226     "\providecommand*{\BibEmph}[1]{#1}" write$ newline$
3227 (*modern|!modern)
3228 %"\ProvideTextCommandDefault{\cyrdash}{---}" write$ newline$
3229 %\DeclareUTFcharacter[UTFencname]{x2014}{\cyrdash}
3230 %"\let\cyrdash\textemdash" write$ newline$
3231 %"\ProvideTextCommandDefault{\cyrdash}{\hbox to.8em{--\hss--}}" write$ newline$
3232 %"\ProvideTextCommandDefault{\cyrdash}{\textemdash}" write$ newline$
3233 %"\ProvideTextCommandDefault{\cyrdash}{\iflanguage{russian}{\hbox to.8em{--\hss--}}{\textemdash}}" write$ n
3234 %%" \ProvideTextCommandDefault{\cyrdash}{%" write$ newline$
3235 %%" \iflanguage{russian}{\hbox to.8em{--\hss--}}{%" write$ newline$
3236 %%" \iflanguage{ukrainian}{\hbox to.8em{--\hss--}}{\textemdash}}" write$ newline$

```

```

3237 "\providecommand*\BibDash{\ifdim\lastskip>Opt\unskip\nobreak\hskip.2em plus 0.1em\fi" write$ newline$
3238 "\cyrdash\hskip.2em plus 0.1em\ignorespaces}" write$ newline$
3239 "\renewcommand\newblock{\ignorespaces}" write$ newline$
3240 </modern!modern>
3241 <natbib> "\providecommand\{natexlab}[1]{#1}" write$ newline$
3242 preamble$ empty$
3243 'skip$
3244 { preamble$ write$ newline$ }
3245 if$
3246 }
3247
3248
3249 EXECUTE {begin.bib}
3250
3251 EXECUTE {init.state.consts}
3252
3253 ITERATE {call.type$}
3254

```

end.bib

```

3255 FUNCTION {end.bib}
3256 { newline$
3257 % "\catcode'\/=11" write$ newline$
3258 "\end{thebibliography}" write$ newline$
3259 }
3260
3261 EXECUTE {end.bib}
3262
3263 </bst>

```

That's all, Folks!

## Change History

v0.8	Upload to CTAN	1
General: \BibAnnote added	v1.2	1
\BibEmph added	General: Entries eprint, eprintclass,	1
\BibUrl added	eprinttype	1
Entry ANNOTE added	Fix bbl.urldate for ukrainian (Andrey Shvajkoy)	1
v0.9	Medium field	1
General: Bug fix in INPROCEEDINGS	Options modern, long, eprint	1
Bug fix in names and date formatting	Refactoring, Documentation	1
v1.0	Strict option	1
General: Bug fix (long annote)	Support for natbib package	1
v1.1	Thesis entry, report entry	1
General: Added German, French, Italian languages	v1.2a	1
Entry ONLINE	General: Default for \cyrdash added	1
Gost705.dtx borrowed from Disser pkg		1



v1.2b		fix Gost2003:— replaced by \BibDash . . .	1
General: numpages renamed to pagetotal . . .	1	langid field added . . . . .	1
v1.2c		v1.2d	
General: eid field added . . . . .	1	General: \bbljan e.t.c. macros removed . . .	1

### 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.

Symbols			
\& . . . . .	2669, 2670, 2692, 2696, 2701, 2719, 2722, 2724, 2725, 2742, 2783	bbl.nnumber . . . . .	18
\' . . . . .	402	bbl.nopublisher . . . . .	23
\, . . . . .	698, 700, 707, 715, 722	bbl.nr . . . . .	18
\. . . . .	1121	bbl.number . . . . .	18
\/ . . . . .	3257	bbl.of . . . . .	17
add.blank . . . . .	11	bbl.pages . . . . .	20
add.doi . . . . .	48	bbl.page . . . . .	20
add.medium . . . . .	49	bbl.phdthesis . . . . .	22
and . . . . .	11	bbl.ppages . . . . .	20
article . . . . .	51	bbl.ppage . . . . .	20
author.after . . . . .	46	bbl.techreport . . . . .	21
author.before . . . . .	45	bbl.urldate . . . . .	21
author.editor.full . . . . .	34	bbl.vvolume . . . . .	15
author.full . . . . .	34	bbl.vvol . . . . .	16
author.key.label . . . . .	27	begin.bib . . . . .	79
author.sort . . . . .	73	bookauthor.after . . . . .	47
author.title.sort . . . . .	74	bookauthor.before . . . . .	45
bbl.and . . . . .	17	booklet . . . . .	53
bbl.chief . . . . .	24	book . . . . .	52
bbl.cmplr . . . . .	15	change.language . . . . .	8
bbl.dscithesis . . . . .	22	chop.word . . . . .	12
bbl.edby . . . . .	14	conference . . . . .	65
bbl.edition . . . . .	15	convert.edition . . . . .	40
bbl.etal . . . . .	17	default.type . . . . .	67
bbl.executor . . . . .	24	dscithesis . . . . .	64
bbl.iin . . . . .	19	editor.full . . . . .	34
bbl.iissue . . . . .	16	editor.organization.after . . . . .	47
bbl.iiss . . . . .	16	either.or.chec . . . . .	38
bbl.in . . . . .	19	emphasize . . . . .	13
bbl.medium.elres . . . . .	23	empty.misc.check . . . . .	44
bbl.medium.text . . . . .	23	enclose.square.brackets . . . . .	13
bbl.medium . . . . .	24	end.bib . . . . .	80
bbl.mthesis . . . . .	21	eng.ord . . . . .	40
bbl.nnoaddress . . . . .	22	extract.num . . . . .	39
bbl.nnopublisher . . . . .	23	field.or.null . . . . .	13
bbl.nnr . . . . .	19	fin.entry . . . . .	10
		format.address.publisher.date . . . . .	32
		format.address.publisher . . . . .	32
		format.annote . . . . .	48
		format.authors.after . . . . .	30
		format.authors . . . . .	27
		format.bookauthors.after . . . . .	30
		format.bookauthors . . . . .	29
		format.btitle . . . . .	38
		format.bvolume . . . . .	38
		format.chapter.pages . . . . .	43
		format.chief.after . . . . .	30
		format.compiler.after . . . . .	30
		format.date . . . . .	32
		format.edition . . . . .	41
		format.editors.after . . . . .	30
		format.eprint . . . . .	49
		format.executor.after . . . . .	30
		format.full.names . . . . .	34
		format.isbn . . . . .	48
		format.key . . . . .	27
		format.lab.names . . . . .	26
		format.month . . . . .	31
		format.names.rev . . . . .	26
		format.names . . . . .	25
		format.number.series . . . . .	39
		format.number . . . . .	43
		format.pages.page . . . . .	42
		format.pages . . . . .	42
		format.techrep.type.number . . . . .	44
		format.thesis.type . . . . .	44
		format.title . . . . .	31
		format.url . . . . .	47
		format.vol.num.pages . . . . .	43
		format.volume . . . . .	43
		ielectronic . . . . .	59
		inbook . . . . .	53
		incollection . . . . .	54
		init.state.consts . . . . .	8
		initialize.longest.label . . . . .	77, 78

inproceedings	56		
internet	59	\_ .	356, 358, 361, 365, 655,
is.num	39		658, 688, 702, 703, 704,
make.full.names	34		719, 720, 721, 729, 731,
manual	57		732, 733, 739, 741, 742,
mastersthesis	63		743, 749, 751, 752, 753,
misc	58		771, 774, 1062, 1069, 1083
multi.page.check	41		
n.dashify	37		
new.block.checka	12		
new.block.checkb	12		
new.block	10		
new.column	10		
new.dblslash.checka	13		
new.dblslash	10		
new.semicolumn	11		
new.sentence.checka	12		
new.sentence.checkb	13		
new.sentence	11		
new.slash	10		
non.stop	12		
not	11		
online	59		
or	11		
output.address.publisher.date	33		
output.bibitem	34		
output.check	9		
output.nonnull	8		
output	9		
phdthesis	62		
presort	75		
proceedings	55		
report	61		
reverse.pass	78		
sort.format.names	71		
sort.format.title	73		
sortify	71		
space.word	14		
techreport	65		
thesis	59		
tie.connect	38		
tie.or.space.connect	38		
unpublished	58		
webpage	59		
word.in	38		
www	59		
\‘	706		
\bbland	506		
\bbledby	354		
\bbledition	388		
\bblIn	589		
\bblin	579		
\bblIss	454		
\bblIssue	437		
\bblno	550, 566		
\bblNumber	524		
\bblnumber	537		
\bblof	471, 488		
\bblP	638		
\bblp	612		
\bblPp	625		
\bblpp	599		
\bblVol	424		
\bblVolume	411		
\begin	330, 3217		
\BibAnnote	1876, 3225		
\BibDash	116, 117, 1152, 1153, 3237		
\BibEmph	307, 3226		
\bibitem	11, 12, 13, 1233, 1247, 1349, 1378		
\BibUrl	1862, 1863, 3224		
\catcode	3257		
\cite	21, 24, 25, 26, 27, 28		
\citeauthor	29, 30		
\citet	22, 23		
\citeyear	31		
\csname	3219, 3220		
\cyn	1124		
\CYRA	1109, 1121		
\CYRB	731, 751		
\cyrch	674		
\cyrdash	3228, 3229, 3230, 3231, 3232, 3233, 3234, 3238		
\CYRE	774		
\CYREREV	771		
\CYRF	1103, 2828		
\cyrf	687		
\CYRI	1115, 1118		
\cyrishrt	771, 774, 1112		
\CYRO	674, 1127		
\cyrsoftsn	803, 1115, 1118		
\CYRYA	1100		
\cyrYu	1115, 1118		
\DeclareUTFcharacter	3229		
\def	3218		
\DescribeFunction	324, 325, 326		
\else	3220, 3221		
\end	3258		
\endcsname	3219, 3220		
\expandafter	3219		
\fi	3220, 3221, 3237		
\hbox	3231, 3233, 3235, 3236		
\href	1885, 1897, 1930, 1932, 1935, 1958, 1975, 3222		
\hskip	3237, 3238		
\hss	3231, 3233, 3235, 3236		
\ifdim	3237		
\iflanguage	3233, 3235, 3236		
\ifx	3219		
\ignorespaces	3238, 3239		
\language	3220		
\lastskip	3237		
\ldots	702, 704, 719, 721		
\let	3230		
\natexlab	3199, 3241		
\newblock	119, 3239		
\nobreak	3237		
\providecommand	3222, 3223, 3224, 3225, 3226, 3237, 3241		

<code>\ProvideTextCommandDefault</code>	<b>S</b>	.. 3230, 3232, 3233, 3236
..... 3228,	<code>\selectlanguage</code> .....	3221
3231, 3232, 3233, 3234	<code>\selectlanguageifdefined</code>	<code>\textnumero</code> .....
	..... 102, 1240,	556, 570
	1258, 1370, 1392, 3218	
<b>R</b>	<code>\small</code> .....	<b>U</b>
	3222, 3223	<code>\unskip</code> .....
<code>\relax</code> .....		3237
3219	<b>T</b>	<code>\url</code> .....
		3223, 3224
<code>\renewcommand</code> .....	<code>\textemdash</code> .....	<code>\UTFencname</code> .....
3239		3229