The GOST package

Igor Kotelnikov, Leonid Sinev
https://github.com/kia999/GOST/

Version ? released ?

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.

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Kotelnikov to the present status and some code was borrowed from disser package developed by Stanislav Kruchinin and unpublished work by Artem Petrenkov.

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

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

Etc.

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) versions of each BibTeX style, in each case offering a choice of sorted and unsorted.

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.
<table>
<thead>
<tr>
<th>Style</th>
<th>utf8</th>
<th>strict</th>
<th>eprint</th>
<th>long</th>
<th>sort</th>
<th>natbib</th>
</tr>
</thead>
<tbody>
<tr>
<td>gost2003</td>
<td></td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gost2003s</td>
<td></td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>gost2008</td>
<td></td>
<td></td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gost2008n</td>
<td></td>
<td></td>
<td>+</td>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>gost2008l</td>
<td></td>
<td></td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>gost2008s</td>
<td></td>
<td></td>
<td>+</td>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>gost2008ns</td>
<td></td>
<td></td>
<td>+</td>
<td></td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>gost2008ls</td>
<td></td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Style</th>
<th>utf8</th>
<th>strict</th>
<th>eprint</th>
<th>long</th>
<th>sort</th>
<th>natbib</th>
</tr>
</thead>
<tbody>
<tr>
<td>ugost2003</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ugost2003s</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>ugost2008</td>
<td>+</td>
<td></td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ugost2008n</td>
<td>+</td>
<td></td>
<td>+</td>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>ugost2008l</td>
<td>+</td>
<td></td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ugost2008s</td>
<td>+</td>
<td></td>
<td>+</td>
<td></td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>ugost2008ns</td>
<td>+</td>
<td></td>
<td>+</td>
<td></td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>ugost2008ls</td>
<td>+</td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

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.

The strict option provides conformance to the Standard 7.1-2003. The bibstyles 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 bibstyles compiled without strict option meets the Standard 7.0.5-2008 which can be thought off as a relaxed version of the Standards 7.1-2003. These bibstyles bear the name gost2008 with possible suffixes s, l, n.

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).

<table>
<thead>
<tr>
<th>Encoding</th>
<th>CSF</th>
<th>Sorting order</th>
</tr>
</thead>
<tbody>
<tr>
<td>cp866</td>
<td>rusci.csf</td>
<td>Cyrillic first, Latin</td>
</tr>
<tr>
<td>cp1251</td>
<td>cp1251.csf</td>
<td>Cyrillic first, Latin</td>
</tr>
<tr>
<td>koi8-u</td>
<td>koi8u.csf</td>
<td>Cyrillic first, Latin</td>
</tr>
<tr>
<td>utf8</td>
<td>utf8cyrillic.csf</td>
<td>Cyrillic first, Latin</td>
</tr>
</tbody>
</table>

In addition, `BibTeX8` distribution comes with few more CSFs.

<table>
<thead>
<tr>
<th>Encoding</th>
<th>CSF</th>
<th>Sorting order</th>
</tr>
</thead>
<tbody>
<tr>
<td>cp866</td>
<td>cp866rus.csf</td>
<td>Latin first, Cyrillic</td>
</tr>
</tbody>
</table>

2 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 `langid="ukrainian"` or `langid="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 and ugost*.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 bibtex.exe or bibtex8.exe 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.

3 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}{\small #1}
\providecommand*{\BibUrl}{\url{#1}}
\providecommand{\BibAnnote}{}
\providecommand*{\BibEmph}{#1}

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

\providecommand*{\BibDash}{ }

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

4 Where to get

5 Version history

Version 1.2l (2021.02.05)
Bug fixed: removed \endofline.

Version 1.2k (2020.12.29)
1. All stuff is now generated from gost.dtx.
2. Limited support of the date field added.
3. Formatting of doi field updated: http://dx.doi.org changed to https://doi.org.
4. Restricted support of date field added.
5. Documentation and examples update (thanks to Leonid Sinev).

Version 1.2i (2017.01.12)
1. Documentation and examples update (thanks to Leonid Sinev).
2. Restored @MastersThesis instead of @MasterThesis (thanks to Leonid Sinev).
3. media="eresource" is introduced in addition to media="online" and media="text"; if present, the media field is not ignored any more in modern bst-styles compiled without the strict option.
4. location field is introduced as an alias of address field.
5. @DSCISTHESIS entry renamed to @DOCTHESIS.
6. school field in @THESIS and similar entries is replaced by institution to comply with biblatex-gost style.

Version 1.2h (2016.08.21)
1. Minor changes in documentation.

Version 1.2g (2016.07.25)
1. Minor changes in documentation.

Version 1.2f (2016.07.12)
1. Support for patent entry added (thanks to Stanislav Kruchinin).
2. medium field renamed to media field for compatibility with biblatex.

Version 1.2e (2016.07.07)
1. Hard coded "URL" string replaced with a language sensitive string (thanks to Roman Budnyi).
Version 1.2d (2015.02.18)
1. jan, feb, etc. macros fixed.
2. New macro \texttt{format.month}.

Version 1.2c (2015.01.10)
1. \texttt{langid} field is added. It has same meaning as \texttt{language} which is now obsolete but is still supported for backward compatibility; \texttt{langid} has priority over \texttt{language}.
2. \texttt{eid} field is added. It has priority over \texttt{pages}.
3. The ligature "--- has been substituted with \texttt{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 \texttt{BibDash} has been improved.
5. \texttt{BibDash} now typesets short em-dash (\texttt{cyrdash}) only for \texttt{russian} and \texttt{ukrainian} languages. In earlier versions, it produces short em-dash for all languages.
   (This feature was removed since it did not work with all engines.)

Version 1.2a (2012.08.31)
1. \texttt{cyrdash} is now defined via \texttt{\ProvideTextCommand} rather than \texttt{\providecommand}.

Version 1.2 (2012.02.22)
1. Code refactoring. All styles are now generated from single source file.
2. Support for GOST 7.1-2003. The field \texttt{medium} is added to reflect type of material. For most entry types \texttt{medium} defaults to \texttt{text}.
3. Support for \texttt{natbib} package.
4. All modern styles are now compiled with the \texttt{eprint} option.

Version 1.1 (2012.01.21)
1. Support for GOST 7.0.5-2008 and GOST 7.1-2003 is provided.
2. \texttt{@Online} entry is added to format a reference to electronic resource on Internet.
3. \texttt{@MastersThesis} entry is added to format a reference to master’s thesis.
4. \texttt{@DSciThesis} entry is added to format a reference to doctor of sciences thesis.
5. \texttt{Urldate}, \texttt{eprint}, \texttt{eprintclass}, \texttt{eprinttype} fields are added.
5.1 Older versions

2012.02.22 Support for natbib package.

2012.02.02 Adaptation to GOST 7.0.5, electronic publishing.

2005.08.12 First version uploaded to CTAN.

2003.06.06 First public version.

6 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}.

\begin{verbatim}
\selectlanguage{russian}
\end{verbatim}

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

Enlist all entry fields allowed in a bibliographic database. Most fields are common for many standard bst styles. Nonlisted fields are just ignored by Bib\TeX.

ENTRY

{ address
  annotate
  author
  booktitle
  bookauthor
  chapter
  edition
  editor
  compiler
  howpublished
  institution
  journal
  key
  %major % new in v.1.2i, alias for speciality, not implemented
  majorcode % new in v.1.2i, alias for specialitycode
  month
  note
  number
  organization
  pages
  eid % new in v1.2c
  publisher
  % school % alias for institution
  series
  %speciality % new in v.1.2i, eqv. to major in biblatex-gost, not implemented yet...
  specialitycode % new in v.1.2i, alias of number, eqv. to majorcode in biblatex-gost
  title
  %medium % new in v1.2; renamed to media.
  media % new in v1.2f
  type
  volume
  year
  language
  langid % new in v1.2c
  booklanguage

Entries borrowed from biblatex.

date % new in v1.2i
pagetotal
url
urldate
isbn
doi
eprint
eprinttype % = archivePrefix
eprintclass % = primaryClass
% new in v1.2f:
% appear in biblatex:
%addendum % not implemented yet...
holder % see patent
location % new in v.1.2i, alias of address
%subtitle % not implemented yet...
titleaddon % new in v.1.2i, see @thesis
%version % not implemented yet...
% Appear in biblatex-gost for @patent entry:
authorcountry % country of the patent authors
credits % statement of responsibility, other than provided in Biblatex
ipc % Code of the International Patent Classification
%media % General material designation NOTE: medium in the above
requestnumber % Registration number of the application to the patent document
publicationdate % Date of publication
publication % and information on the official gazette, which published patent
prioritydate % Information about the convention priority: the date of filing of the application,
prioritynumber % number and
prioritycountry % country name of convention priority.
requestdate % ??
}

6.2 Output functions

Declare internal variables and constants used for formatting bibliographic records. Other variables are defined below when needed.

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

STRINGS { curlanguage }

STRINGS { s t }

STRINGS { y m d } % new in v.1.2j

init.state.consts Set constants that designate various output states which are kept by `output.state` integer. The
latter is checked and updated by a set of functions such as ‘output’ to be defined below.

FUNCTION {init.state.consts}
{ #0 'before.all :=
 #1 'mid.sentence :=
 #2 'after.sentence :=
 #3 'after.block :=
 #4 'after.dblslash :=
 #5 'after.slash :=
 #6 'after.colon :=
 #7 'after.semicolon :=
}

set.language Sets current language curlanguage. Called by bibitem.output before any other function.

FUNCTION {set.language}
{ langid empty$ 
   { language empty$ 
     { "english" 'curlanguage := } 
     { language 'curlanguage := } 
     if$ 
   } 
   { langid 'curlanguage := } 
   if$ 
}

reset.language Reset current language to booklanguage if provided. Called by outputnonnull after double slash so that the rest of the record is formatted with booklanguage.

FUNCTION {reset.language}
{ booklanguage empty$ 
   { "" } 
   { booklanguage 'curlanguage := 
     "\selectlanguageifdefined{" 
     curlanguage * 
     "}" * 
   } 
   if$ 
}

outputnonnull Writes the last literal in the stack to output buffer assuming that it is not empty and adds an appropriate punctuation symbol.

FUNCTION {outputnonnull}
{ swap$ 
   output.state mid.sentence = 
   { ", " * write$ } 
   output.state after.block = 
   { add.period$ write$ 
     " \BibDash " write$ 
   }
}
newline$ "\newblock " write$
{ output.state before.all =
'write$
{ output.state after.dblslash =
  ""/" * reset.language * " " * write$ }
{ output.state after.slash =
  "/ " * write$ }
{ output.state after.colon =
  ": " * write$ }
{ output.state after.semicolon =
  "; " * write$ }
{ add.period$ " " * write$ }
if$
}
if$
}
if$
}
if$
}
if$
}
if$ mid.sentence 'output.state :=
}
if$
}
if$

output  Calls outputnonnull if the last literal string in the stack is not empty; otherwise it discards
the literal.
FUNCTION {output}
{ duplicate$ empty$
'pop$
'outputnonnull
if$
}

output.check  Does the same but also warns if the indicated field is empty. Needs two literals in the stack:
the field and the name of the field, e.g., author "author" output.check.
FUNCTION {output.check}
{ 't :=
duplicate$ empty$
{ pop$
"empty " t * " in " * cite$ * warning$
}
'outputnonnull
if$
fin.entry  fin.entry finalizes current entry. It writes dot, if no dot is found in stack, and starts new line.

FUNCTION {fin.entry}
{ add.period$
write$
newline$
}

Declare a family of functions to put punctuation marks depending on the current status of the output stack.

new.block
This just checks output state and revert it to another state if required. Checking output state prevents occasional doubling of punctuation marks.

FUNCTION {new.block}
{ output.state before.all = 'skip$
{ after.block 'output.state := }
if$
}

new.dblslash
FUNCTION {new.dblslash}
{ output.state before.all = 'skip$
{ after.dblslash 'output.state := }
if$
}

new.slash
FUNCTION {new.slash}
{ output.state before.all = 'skip$
{ after.slash 'output.state := }
if$
}

new.colon
FUNCTION {new.colon}
{ output.state before.all = 'skip$
{ after.colon 'output.state := }
if$
}

new.semicolon
FUNCTION {new.semicolon}
6.3 Logical functions and various checks

Declare few logical functions.

\texttt{not}

\begin{verbatim}
FUNCTION {not}
{ { #0 } { #1 } if$ }
\end{verbatim}

\texttt{and}

\begin{verbatim}
FUNCTION {and}
{ 'skip$ { pop$ #0 } { pop$ #0 } if$ }
\end{verbatim}

\texttt{or}

\begin{verbatim}
FUNCTION {or}
{ { pop$ #1 } { pop$ #1 } 'skip$ if$ }
\end{verbatim}
non.stop

(NB: What’s the hell? Never used.)

FUNCTION {non.stop}
{ duplicate$ }
"]" * add.period$
"." =
\}

 new.block.checka  Adds new.block if the last literal in stack is not empty.

FUNCTION {new.block.checka}
{ empty$ }
 'skip$
 'new.block
 if$
 }

 new.block.checkb  Adds new.block if either of the two last literals in the stack is not empty.

FUNCTION {new.block.checkb}
{ empty$ }
 swap$ empty$
 and
 'skip$
 'new.block
 if$
 }

 new.sentence.checka  Adds new.sentence if the last literal in stack is not empty.

FUNCTION {new.sentence.checka}
{ empty$ }
 'skip$
 'new.sentence
 if$
 }

 new.sentence.checkb  Adds new.sentence if either of the two last literals in the stack is not empty.

FUNCTION {new.sentence.checkb}
{ empty$ }
 swap$ empty$
 and
 'skip$
 'new.sentence
 if$
 }

 new.dbldlash.checka  For online entry.

FUNCTION {new.dbldlash.checka}
{ empty$ }
 'skip$
field.or.null  Replaces an empty field with null string "".

FUNCTION {field.or.null}
{ duplicate$ empty$
{ pop$ "" }
'skip$
if$
}

either.or.check  (NB: Move upwards)

FUNCTION {either.or.check}
{ empty$
'pop$
{ "can't use both swap$ " fields in " * cite$ * warning$ }
if$ }

6.4 String Functions

spaces.around  Inserts a space before and after last literal in the stack.

FUNCTION {spaces.around}
{ " " swap$ * " " * }

emphasize  Emphasizes the last literal in the stack if it is not empty. v1.2k: emphasize now returns empty field if last literal in the stack was empty.

FUNCTION {emphasize}
{ duplicate$ empty$
{ pop$ "" }
'skip$  % v.1.2k
{ "\BibEmph{swap$ "}" * }
if$
}

bracify  New in v.1.2. An idea borrowed from apsrev4-1.bst. Encloses last literal in the stack by braces even if it is empty. Note that braces are normally not printed by \LaTeX.

FUNCTION {bracify}
{ duplicate$ empty$
{ pop$ "{}" }
{ "{} swap$ * "" * }
if$
}
This and the next functions are used to enclose last word by square and round brackets. In contrast to \texttt{bracify} function they push null string if the last literal is empty.

\begin{verbatim}
FUNCTION \{bracketise\}
{ duplicate$ empty$ { pop$ "" } { "[" swap$ * "]" * } if$ }
\end{verbatim}

\begin{verbatim}
FUNCTION \{paranthesify\}
{ duplicate$ empty$ { pop$ "" } { "(" swap$ * ")" * } if$ }
\end{verbatim}

The function \texttt{chop.word} in the context ‘\texttt{sstr len str chop.word}’ tries to remove given substring \texttt{sstr} of the length \texttt{len} from the beginning of the string \texttt{str}. It trims \texttt{str} only if first \texttt{len} symbols in \texttt{str} coincides with \texttt{sstr}. See examples in Section 6.8.

\begin{verbatim}
FUNCTION \{chop.word\}
{ 's := 'len := s #1 len substring$ = { s len #1 + global.max$ substring$ } 's if$ }
\end{verbatim}

Currently not used.

\begin{verbatim}
FUNCTION \{is.num\}
{ chr.to.int$ duplicate$ "0" chr.to.int$ < not swap$ "9" chr.to.int$ > not and }
\end{verbatim}

Currently not used.
FUNCTION {extract.num}

{ duplicate$ 't :=
  "" 's :=
  { t empty$ not }   
  { t #1 #1 substring$ 
    t #2 global.max$ substring$ 't :=
    duplicate$ is.num
    { s swap$ * ?s := }    
    { pop$ "" 't := }    
    if$    
  }   
  while$    
  s empty$    
  'skip$    
  { pop$ s }    
  if$    
}

FUNCTION {tie.connect}

{"" swap$ * * }

FUNCTION {tie.or.space.connect}

{ duplicate$ text.length$ #3 <
  { "--" }    
  { " " }    
  if$    
  swap$ * * }

FUNCTION {n.dashify}

{ 't :=
  ""    
  { t empty$ not }    
  { t #1 #1 substring$ "-" =
    { t #1 #2 substring$ "--" = not
      { "--" *
        t #2 global.max$ substring$ 't :=
      }    
      { { t #1 #1 substring$ "-" = }
        { "-" *
          t #2 global.max$ substring$ 't :=
      }    
  }    
  }
while$
} if$
} { t #1 #1 substring$ *
  t #2 global.max$ substring$ 't :=
} if$
} while$
) Returns 1 if the last literal (usually page) contains either ‘-’, ‘,’ or ‘+’; otherwise returns 0.
Used in connection with n.dashify.
⟨NB: Заметим несогласованность функций multi.page.check и n.dashify. Последняя функция проверяет дефис, а первая ещё и минус и запятую.⟩

INTEGERS { multiresult }
FUNCTION {multi.page.check}
{ 't :=
  #0 'multiresult :=
  { multiresult not
    t empty$ not
    and
    }
  { t #1 #1 substring$
    duplicate$ "-" =
    swap$ duplicate$ "," =
    swap$ "+" =
    or or
    { #1 'multiresult := }
    { t #2 global.max$ substring$ 't := }
    if$
  } } while$
multiresult

6.5 Language-sensitive abbreviations

Declare language-sensitive abbreviations. We provide two versions of any abbreviation for unicode and non-Unicode styles. The language-sensitive functions push to the stack a string that depends of the the current value of the string curlanguage. It is set for every entry by output.bibitem function. Abbreviations for russian, ukrainian and english values of the string curlanguage are always provided, and sometimes for french and german.
FUNCTION {bbl.edby}
{ curlanguage "english" =
  {"ed. \ by"}
  { curlanguage "ukrainian" =
    {utf8} \"{\cyry\cyri\cyrii\cyrd \cyry\cyre\cyrd.}\"}
    {utf8} \"пiд ред.\"}
    { curlanguage "russian" =
      {utf8} \"под ред.\"}
      "language is not defined: " curlanguage " in bbl.edby" * * warning$ "Ed. \ by"}
if$}
if$
if$
if$
if$
if$
if$

FUNCTION {bbl.compiler}
{ curlanguage "english" =
  "Compiler"
  { curlanguage "german" =
    "Hrsg."
    { curlanguage "ukrainian" =
      {utf8} \"{\cyr\CU\cyrk\cyr1.}\"
      {utf8} \"{Укл.}\"
      { curlanguage "russian" =
        {utf8} \"{\cyr\cyrz\cyrd.}\"
        {utf8} \"{Сост.}\"
        "language is not defined: " curlanguage " in bbl.compiler" * * warning$ "Compiler"}
      if$
      if$
    if$
  if$}

FUNCTION {bbl.edition}
{ curlanguage "english" =
  {"ed."
  { curlanguage "ukrainian" =
    {utf8} \"{\cyry\cyri\cyrd.}\"
    {utf8} \"{вид.}\"
    { curlanguage "russian" =
      {utf8} \"{\cyr\cyrz\cyrd.}\"
      {utf8} \"{изд.}\"
      { curlanguage "german" =
        {utf8} \"{\cyr\cyro\cyrs\cyrt.}\"
        {utf8} \"{Aus.}\"
        { utf8 } \"{Aufl.}\%
        { "aufl.\" } %%%% { "Aufl. " } ??
      if$
      if$}
  if$}

\bbl.compiler}

\bbl.edition
{ curlanguage "french" =
    {"\"{e}dition\"
     {"language is not defined: " curlanguage " in bbl.edition" * warning"ed."}
    if$
    if$
    if$
    if$
    if$
    if$

bbl.vvolume

FUNCTION {bbl.vvolume}
{ curlanguage "english" = curlanguage "french" = or curlanguage "italian" = or
{ "Volume"
{ curlanguage "ukrainian" = curlanguage "russian" = or
{\utf8} { "\CYRT\cyro\cyrm" }
{\utf8} { "Том" }
{ curlanguage "german" =
{ "\{Band\}" } { "Volumen" }
{ "language is not defined: " curlanguage " in bbl.vvolume" * warning"Volume"
if$
if$
if$

bbl.vvol

FUNCTION {bbl.vvol}
{ curlanguage "english" = curlanguage "french" = or curlanguage "italian" = or
{ "Vol."}
{ curlanguage "ukrainian" = curlanguage "russian" = or
{\utf8} { "\CYRT.\" }
{\utf8} { "Т." }
{ curlanguage "german" =
{ "\{Bd.\}" } { "Vol." }
{ "language is not defined: " curlanguage " in bbl.vvol" * warning"Vol."}
if$
if$
if$

bbl.issue

FUNCTION {bbl.issue}
{ curlanguage "english" =
{ "Issue"
{ curlanguage "ukrainian" =
{\utf8} { "\CYRV\cyri\cyro\cyru\cyrs\cyrk"}
{\utf8} { "\BYmcyxk\" }
{ curlanguage "russian" =
{\utf8} { "\CYRV\cyreyr\cyro\cyru\cyrs\cyrk"}
{\utf8} { "\BYmcyxk\" }
{ curlanguage "german" =

20
"{Heft}" "Ausgabe"

"language is not defined: " curlanguage " in bbl.iissue" * * warning$ "Iss." if$

if$

FUNCTION {bbl.iiss}

{ curlanguage "english" =
  {"Iss."}

{ curlanguage "ukrainian" =
  \utf8"\CYRV\cyri\cyrp."}

{ curlanguage "russian" =
  \utf8"\CYRV\cyrery\cyrp."}

{ curlanguage "german" =
  {"{H.}"}

"language is not defined: " curlanguage " in bbl.iiss" * * warning$ "Iss." if$

if$

FUNCTION {bbl.of}

{ curlanguage "english" =
  {"of"}

{ curlanguage "german" =
  { "von" }

{ curlanguage "ukrainian" =
  \utf8"\cyri\cyrir\cyrz" }

{ curlanguage "russian" =
  \utf8"\cyri\cyrir\cyrz" }

"language is not defined: " curlanguage " in bbl.of" * * warning$ "of" if$

if$

FUNCTION {bbl.etal}

{ curlanguage "english" =
  {"et~al."}

{ curlanguage "german" =
  { "u.~a." }
FUNCTION {bbl.nr}
{ curlanguage "english" =
{ "no." }
{ curlanguage "italian" =
{ "no." }
{ curlanguage "ukrainian" = curlanguage "russian" = or
{utf8}  { "{\textnumero}" }
  {utf8}  { "{\textnumero}" }
{ curlanguage "german" =
  { "{\textnumero}" }% { "an." }
{ curlanguage "french" =
  { "no." }
  "language is not defined: " curlanguage " in bbl.nr" * * warning$ "no."}
if$}
if$}
else$}
else$}
bbl.nnr
FUNCTION {bbl.nnr}
{ curlanguage "english" =
"No."
{ curlanguage "ukrainian" = curlanguage "russian" = or
{utf8}  { "{\textnumero}" }
  {utf8}  { "{\textnumero}" }
{ curlanguage "german" =
  { "{\textnumero}" }% { "an." }
{ curlanguage "french" =
  { "no." }
  "language is not defined: " curlanguage " in bbl.nnr" * * warning$ "No."}
if$}
if$}
if$}
if$}
bbl.in
FUNCTION {bbl.in}
{ curlanguage "english" = curlanguage "german" = or
Currently not used.

FUNCTION {bbl.iin}
{ curlanguage "english" = curlanguage "german" = or
 { "In" }
 { curlanguage "ukrainian" = curlanguage "russian" = or
 { \utf8{ "\cyrv" } }
 { utf8{ "{В}" } }
 {"language is not defined: " curlanguage " in bbl.iin" * * warning$ "In"}
}
if$
if$

FUNCTION {bbl.pages}
{ curlanguage "english" = curlanguage "french" = or curlanguage "italian" = or
 { "p." } %%% { "pp." }
 { curlanguage "ukrainian" = curlanguage "russian" = or
 { \utf8{ "{\cyrs.}" } }
 { utf8{ "{с.}" } }
 { curlanguage "german" =
 { "S." } %%% { "s." }
 {"language is not defined: " curlanguage " in bbl.pages" * * warning$ "p."}
}
if$
if$
if$

FUNCTION {bbl.page}
{ curlanguage "english" = curlanguage "french" = or curlanguage "italian" = or
 { "p." }
 { curlanguage "ukrainian" = curlanguage "russian" = or
 { \utf8{ "{\cyrs.}" } }
 { utf8{ "(c.)" } }
 { curlanguage "german" =
 { "S." } %%% { "s." }
 {"language is not defined: " curlanguage " in bbl.page" * * warning$ "p."}
}
if$
if$
if$

FUNCTION {bbl.ppages}
FUNCTION {bbl.ppages}
752 { curlanguage "english" = curlanguage "french" = or curlanguage "italian" = or
753 { "P." } \%\%\% \{ "Pp." \}
754 { curlanguage "ukrainian" = curlanguage "russian" = or
755 { curlanguage "german" =
756 { "language is not defined: " curlanguage " in bbl.ppages" * * warning$ "P." } if$
757 if$
758 if$
759 if$
760 bbl.ppage
761 FUNCTION {bbl.ppage}
762 { curlanguage "english" = curlanguage "french" = or curlanguage "italian" = or
763 { "P." }
764 { curlanguage "ukrainian" = curlanguage "russian" = or
765 { curlanguage "german" =
766 { "language is not defined: " curlanguage " in bbl.ppage" * * warning$ "P." } if$
767 if$
768 if$
769 if$
770 bbl.url
771 Added in version 2016.07.07.
772 FUNCTION {bbl.url}
773 { curlanguage "english" =
774 { "Access mode" }
775 { curlanguage "ukrainian" =
776 { "language is not defined: " curlanguage " in bbl.url" * * warning$ "online; accessed" } if$
777 if$
778 if$
779 if$
780 bbl.urldate
781 Added in version 2012.01.15.
782 FUNCTION {bbl.urldate}
798 { curlanguage "english" =
799  {"online; accessed"}
800  { curlanguage "ukrainian" =
801  {"(дата звернення)"}
802  { curlanguage "russian" =
803  {"(дата обращения)"}
804  { curlanguage "german" =
805  {"(online; abgerufen)"}
806  { curlanguage "french" =
807  {"en ligne; accédé"}
808  { "language is not defined: " curlanguage " in bbl.urldate" * * warning$ "online; accessed" "}
809  if$}
810  if$}
811  if$}
812  if$}
813  if$}
814  if$}
815 if$}
816
bbl.techreport
817 FUNCTION {bbl.techreport}
818  { curlanguage "english" =
819    { "Rep." }
820  { curlanguage "german" =
821    { "Bericht" }
822  { curlanguage "russian" =
823    { "Отчет" }
824    { "(en ligne; acc\'{e}d\'{e})" }
825    { "language is not defined: " curlanguage " in bbl.techrep" * * warning$ "Rep." }
826    if$
827    if$
828    if$
829
bbl.mathesis
830 FUNCTION {bbl.mathesis}
831  { curlanguage "english" =
832    { "Master's thesis" }
833  { curlanguage "german" =
834    { "diss.-mag." }
835  { curlanguage "russian" =
836    {"квалификационная работа магистра" }
837    { "(en ligne; acc\'{e}d\'{e})" }
838    { "(en ligne; acc\'{e}d\'{e})" }
839    { "(language is not defined: " curlanguage " in bbl.mthesis" * * warning$ "Master's thesis" }
840    if$
841    if$
842    if$
843
26
FUNCTION {bbl.nnopublisher}

FUNCTION {bbl.media.text}

FUNCTION {bbl.media.eresource}

FUNCTION {bbl.media.online}
FUNCTION {bbl.chief}

{ "chief" }

FUNCTION {bbl.executor}

{ "executor" }

FUNCTION {bbl.media}

{ "online" }
{ media "eresource" =
    { bbl.media.eresource }
    { bbl.media.text }
    if$
    if$
}

FUNCTION {bbl.req}
{
    curlanguage "english" =
        { "req." }
    curlanguage "german" =
        { "ang." }
    curlanguage "russian" =
        { "language is not defined: " curlanguage " in bbl.req" * * warning$ "req" }
        if$
    if$
    if$
}

FUNCTION {bbl.publ}
{
    curlanguage "english" =
        { "publ." }
    curlanguage "german" =
        { "ausg." }
    curlanguage "russian" =
        { "language is not defined: " curlanguage " in bbl.publication" * * warning$ "publication" }
        if$
    if$
    if$
}

FUNCTION {bbl.priority}
{
    curlanguage "english" =
        { "priority" }
    curlanguage "german" =
        { "Prioritat" }
}
New in version 1.2k.
FUNCTION \{bbl.apr\}
{ curlanguage "english" =
  {"Apr."} }
{ curlanguage "ukrainian" =
  {"Квіт."} }
{ curlanguage "russian" =
  { "Апр." } }
{ curlanguage "german" =
  {"März"} }
"language is not defined: bbl.apr for " curlanguage * warning$ "Apr." }
if$}
FUNCTION \{bbl.may\}
{ curlanguage "english" =
  {"May"} }
{ curlanguage "ukrainian" =
  {"Трав."} }
{ curlanguage "russian" =
  { "Май" } }
{ curlanguage "german" =
  {"Mai"} }
"language is not defined: bbl.may for " curlanguage * warning$ "May" }
if$}
FUNCTION \{bbl.jun\}
{ curlanguage "english" =
  {"June"} }
FUNCTION {bbl.jun}
{ curlanguage "ukrainian" =
  "Чер." } % Червень
{ curlanguage "russian" =
  "Июнь" } % Июнь
{ curlanguage "german" =
  "Juni" }
"language is not defined: bbl.jun for " curlanguage * warning$ "June" }
if$
if$
if$
if$
FUNCTION {bbl.jul}
{ curlanguage "ukrainian" =
  "Липень" } % Липень
{ curlanguage "russian" =
  "Июль" } % Июль
{ curlanguage "german" =
  "Juli" }
"language is not defined: bbl.jul for " curlanguage * warning$ "July" }
if$
if$
if$
FUNCTION {bbl.aug}
{ curlanguage "ukrainian" =
  "Серпень" } % Серпень
{ curlanguage "russian" =
  "Авг." } % Август
{ curlanguage "german" =
  "Aug." }
"language is not defined: bbl.aug for " curlanguage * warning$ "Aug." }
if$
if$
FUNCTION {bbl.sep}
{ curlanguage "english" =
  "Jul" 
  "Aug." 
  "Sep." }
FUNCTION {bbl.oct}

FUNCTION {bbl.nov}

FUNCTION {bbl.dec}
bbl.arxiv  New in version 1.2k.

FUNCTION {bbl.arxiv}

bbl.jstor FUNCTION {bbl.arxiv}

bbl.pubmed FUNCTION {bbl.arxiv}

FUNCTION {bbl.jstor}

FUNCTION {bbl.googlebooks}
FUNCTION {bbl.pubmed}
1274 { curlanguage "english" =
1275   { "PubMed" }
1276   { curlanguage "ukrainian" =
1277     { "PubMed" }
1278     { utf8 "PubMed" }
1279   } curlanguage "russian" =
1280     { utf8 "PubMed" }
1281     { "PubMed" }
1282     { "language is not defined: bbl.pubmed for " curlanguage * warning$ "PubMed" }
1283   if$
1284   if$
1285   if$
1286   if$
1287 FUNCTION {bbl.googlebooks}
1288 { curlanguage "english" =
1289   { "Google Books" }
1290   { curlanguage "ukrainian" =
1291     { utf8 "Google Книги" }
1292     { utf8 "Google Книги" }
1293   } curlanguage "russian" =
1294     { utf8 "Google Книги" }
1295     { utf8 "Google Книги" }
1296   } curlanguage "german" =
1297     { utf8 "Google Books" }
1298     { "Google Books" }
1299     { "language is not defined: bbl.googlebooks for " curlanguage * warning$ "Google Books" }
1300   if$
1301   if$
1302   if$
1303   if$
1304 FUNCTION {bbl.hdl}
1305 { curlanguage "english" =
1306   { "Handle.Net" }
1307   { curlanguage "ukrainian" =
1308     { utf8 "Handle.Net" }
1309     { utf8 "Handle.Net" }
1310   } curlanguage "russian" =
1311     { utf8 "Handle.Net" }
1312     { utf8 "Handle.Net" }
1313   } curlanguage "german" =
1314     { utf8 "Handle.Net" }
1315     { utf8 "Handle.Net" }
1316     { "language is not defined: bbl.hdl for " curlanguage * warning$ "Handle.Net" }
1317   if$
1318   if$
1319   if$
1320   if$
1321   if$
1322 if$
6.6 Aliases

Aliases to some fields are introduced with the help of `field1.or.field2` functions.

`address.or.location` Pushes `address` field if not empty; otherwise pushes `location` value even if it is empty.

```
FUNCTION {address.or.location}
{
  address empty$
  { location empty$
    'skip$
    { location }
    if$
  }
  { address }
  if$
}
```

`specialitycode.or.number` Pushes `specialitycode` value if not empty; otherwise returns `number` value even if the latter is empty.

```
FUNCTION {specialitycode.or.number}
{
  specialitycode empty$
  { number }
  { specialitycode }
  if$
}
```

`institution.or.school` Pushes `institution` value if not empty; otherwise returns `school` value even if the latter is empty.

```
FUNCTION {institution.or.school}
{
  institution empty$
  { school}
  { institution }
  if$
}
```

6.7 Formatting dates

(NEEDS to be located before natbib labels. This is experimental section. Needs to be upgraded.)

`date.to.year` New in version 1.2k. Extracts year from date. Currently, only the date of the form either `YYYY-MM-DD`, or `YYYY-MM`, or `YYYY` can be processed as expected. Returns `date` if it is empty or has wrong format.

```
FUNCTION {date.to.year}
```
FUNCTION {date.to.year}
{ date empty$ 
  { date }
  { 
    ⟨*debug⟩
    "date.to.year::: date in " cite$ * " =" * date * warning$
    "date.to.year::: length of date is " date text.length$ int.to.str$ * warning$
    ⟨/debug⟩
    date text.length$ #3 > 
    
    { 
      ⟨debug⟩
      "::: y=" y * warning$
      ⟨/debug⟩
      y 
    } 
    {
      "wrong format of date in " cite$ * ": date=" * date * warning$
      date 
    } if$ 
    if$
  }
}

FUNCTION {date.to.month}
{ date empty$ 
  { date }
  { 
    ⟨*debug⟩
    "date.to.month::: date in " cite$ * " =" * date * warning$
    "date.to.month::: length of date is " date text.length$ int.to.str$ * warning$
    ⟨/debug⟩
    date text.length$ #6 > 
    
    { 
      ⟨debug⟩
      "::: m=" m * warning$
      ⟨/debug⟩
      m 
    } 
    {
      "wrong format of date in " cite$ * ": date=" * date * warning$
      date 
    } if$
  }
}
FUNCTION {date.to.day}
{
  date empty$
  {
    date
  }
  {
    ⟨*debug⟩
    "date.to.day::: date in " cite$ * " =" * date * warning$
    "date.to.day::: length of date is " date text.length$ int.to.str$ * warning$
    ⟨/debug⟩
    date text.length$ #10 =
    {
      date #9 #2 substring$ 'd :=
      ⟨*debug⟩
      ":::: d=" d * warning$
      ⟨/debug⟩
      d
    }
    {
      "wrong format of date in " cite$ * ": date=" * date * warning$
      date
    } if$
  } if$
}

FUNCTION {year.or.date.to.year}
{
  year empty$
  {
    ⟨*debug⟩
    "year.or.date.to.year::: empty year in " cite$ * warning$
    ⟨/debug⟩
    date.to.year
  } if$
  { year } if$
}

FUNCTION {format.month}

Reads month field and translate standard English abbreviation of months (as defined by

New in version 1.2k. Extracts year from date. Returns date if it is empty or has wrong format.

New in version 1.2k. Returns year if not empty; otherwise call date.to.year.

⟨NB: ВОЗМОЖНО, ПОДОБНЫЕ ФУНКЦИИ ЛУЧШЕ назвать this.year.)
FUNCTION {format.month}
{ month empty$ }
{ "" }
{ month "Jan." = bbl.jan }
{ month "Feb." = bbl.feb }
{ month "Mar." = bbl.mar }
{ month "Apr." = bbl.apr }
{ month "May" = bbl.may }
{ month "Jun." = bbl.jun }
{ month "Jul." = bbl.jul }
{ month "Aug." = bbl.aug }
{ month "Sep." = bbl.sep }
{ month "Oct." = "Окт." }
{ month "Nov." = bbl.nov }
{ month "Dec." = bbl.dec }
{ "unknown month in " cite$ * warning$ month if$ } if$

FUNCTION {format.date}
% year empty$
{ year.or.date.to.year empty$ }
{ month empty$ }
{ "" }
{ "there's a month but no year in " cite$ * warning$ format.month }
{ "year

40
6.8 Formatting names

Declare functions to format separate elements of a bibliographic reference.

Important note
Neither \texttt{bibtex} nor \texttt{bibtex8} can handle uncoded text without troubles. In particular, they fail to reduce a Cyrillic name to initials. Therefore we avoid using \texttt{f.} primitive (which trims first name to first letter) when option \texttt{utf8} is in effect; in the latter case \texttt{ff} primitive is called instead.

New version of functions formatting names. Function name indicates number of persons printed. These functions look for last literal in the stack which should not be empty; hence, they should be called by other functions that checks if the last literal is empty.

\begin{verbatim}
FUNCTION {fmt.names.first}
{ #1
  ⟨strict&utf8⟩ "vv"{11}{jj}{,f.}" \\
  ⟨strict & utf8⟩ "vv"{11}{jj}{,ff}" \\
  ⟨!strict&utf8⟩ "vv"{11}{jj}{f.}" \\
  ⟨!strict & utf8⟩ "vv"{11}{jj}{ff}" \\
  format.name$ 't :=
} nameptr #1 >

FUNCTION {fmt.names.three}
{s :=
  #1 'nameptr :=
  s num.names$ 'numnames :=
  numnames 'namesleft :=
  { namesleft #0 > 
    { s nameptr
      ⟨strict&utf8⟩ "vv"{11}{jj}{,f.}" \\
      ⟨strict & utf8⟩ "vv"{11}{jj}{,ff}" \\
      ⟨!strict&utf8⟩ "vv"{11}{jj}{f.}" \\
      ⟨!strict & utf8⟩ "vv"{11}{jj}{ff}" \\
      format.name$ 't :=
      nameptr #1 >
    } nameptr #4 = numnames #4 > and
    { "others" 't :=
      #1 'namesleft :=
    }
}

FUNCTION {fmt.names.all}
{ year ". " \texttt{\textbackslash BibDash} " format.month ** }
{ year.or.date.to.year " \texttt{\textbackslash BibDash} " format.month ** }
  if$
  }
  if$
}
\end{verbatim}
byt
  if$
  namesleft #1 >
  { "," * t * }
  { t "others" = t "others" = or
  { " " * bbl.etal * }
  { " " * bbl.etal * }
  { "," * t * }
  if$
  }
  if$
  }
  if$
  nameptr #1 + 'nameptr :=
  namesleft #1 - 'namesleft :=
  }
  while$
  }
  }
FUNCTION {fmt.names.all}
  { 's :=
  #1 'nameptr :=
  s num.names$ 'numnames :=
  numnames 'namesleft :=
  { namesleft #0 > }
  { s nameptr
  \%"{vv}{{ll}}" format.name$ 't :=
  { strict&utf8} "{vv}{{ll}}{"jj}{"f.}" 
  { strict & utf8} "{vv}{{ll}}{"jj}{"ff}" 
  { strict & utf8} "{vv}{{ll}}{"jj}{"ff}" 
  { strict & utf8} "{vv}{{ll}}{"jj}{"ff}" 
  format.name$ 't :=
  nameptr #1 + 'nameptr :=
  { namesleft #1 >
  { "," * t * }
  { numnames #2 > curlanguage "english" = and
  { "," * }
  'skip$
  if$
  t "others" = t "others" = or
  { " " * bbl.etal * }
  { " " * bbl.etal * }
  { " " * bbl.etal * }
  if$
  }
  if$
  }
  t
  if$
6.9 Formatting names (cont.)

format.author Formats a list of authors for the heading part of a bibliographic record by applying either fmt.names.first, fmt.names.three or fmt.names.all to the field author, if it is not empty; otherwise it pushes empty author. This allows making a check as in the case of format.author "author" output.check.

⟨NB: Однако заметим, что все другие функции format.... пустое поле замещают пустой строкой.⟩

%FUNCTION {format.author}
{ author empty$
% %%%"" % < v.1.2k
% { author } % v.1.2k
% %{ author format.names emphasize} % 1st if strict, <=3 otherwise
% %<long> { author fmt.names.all emphasize}
%<!long&strict> { author fmt.names.first emphasize}
%<!long&!strict> { author fmt.names.three emphasize}
% if$
%}
FUNCTION {format.author}
{
author empty$
{ author } %%%'skip$
⟨*
long⟩
{ author fmt.names.all }
⟨/long⟩
⟨*!long⟩
{author num.names$ #4 <$
{strict} author fmt.names.first
⟨!strict⟩ author fmt.names.three
}
{ "" } %%%'skip$
if$
⟨!/long⟩
if$
⟩
}
FUNCTION {format.bookauthors}
{bookauthor empty$
{ "" }
{ bookauthor format.names}% cuts to 4 persons if !strict option

format.bookauthors Is used only once by bookauthor.head called in inbook entry. ⟨NB: Not used anymore!⟩
Formats author to be placed after a slash in the zone of responsibility of a bibliographic record. In contrast to output.author.rest does not check number of items in the author field; it is recommended to use output.author.rest instead whenever possible. (NB: Not used any more.)

FUNCTION {format.author.rest}
{
  % Does this work?
  %%skip$ % this seemed to work
  %%"" % this seemed to work
  }%author empty$
  %< v.1.2k
  % v.1.2k
  if$
  }%!long

FUNCTION {format.bookauthors.rest}
{
bookauthor empty$

FUNCTION {format.editors.rest}
{
editor empty$

FUNCTION {format.chief.rest}
{ bbl.chief "\ " * editor fmt.names.all * }
6.10 Formatting natbib keys

```
FUNCTION {fmt.names.brief}
{s := #1 'nameptr := s num.names$ 'numnames := numnames 'namesleft := namesleft #0 > }
{s nameptr "{{v}}\{ll}" format.name$ 't := nameptr #1 > }
{n namesleft #1 > }
{ "", " * t * }
{n numnames #2 > curlanguage "english" = and ""," * }
'skip$ if$ t "others" = t "-others" = or
```
Declares function to go to left part of optional argument of `\bibitem` in the bibstyles generated with the option natbib. It cuts the list of person to 2 at most and drops first name of every person.

```latex
FUNCTION {format.names.key}
{ 's :=
  set.language %%% уже вызвана в output.bibitem, но вроде бы нужна и здесь
  s #1 "{vv~}{ll}" format.name$
  s num.names$ duplicate$ #2 >
  { pop$ " bbl.etal * * } #2 <
  { 'skip$
    { s #2 "{ff }{vv }{ll}{ jj}" format.name$ "others" =
      { " " bbl.etal * * }
      { " " bbl.and " * * * s #2 "{vv~}{ll}" format.name$ * }
    if$
    } if$
  if$
 } if$
if$
)

Substitute an empty last literal in the stack (usually, author) with the key field if provided. Used as a heading of bibliographic record of author is empty.

```latex
FUNCTION {format.key}
{ empty$
  { key field.or.null }
  { " " }
  if$
  if$
)

Composes a key to be used as a reference label with natbib styles. If author field is empty an attempt is made to retrieve key field. If it is also empty 3 first letters are retrieved from the citation key cite$. If author field is not empty format.names.key defined above is called.

```latex
FUNCTION {author.key.label}
{ author empty$
  46
}
These functions operate similarly but engage a different set of fields. They are called below
by calc.short.list.

FUNCTION {author.editor.key.label}
{ author empty$ }
{ editor empty$ }
{ key empty$ }
{ cite$ #1 #3 substring$ }
"key % causes lost of year
{ "{}" key * } % Bug in bibtex8 ??
if$
}
{ editor format.names.key }
if$
}
{ author format.names.key }
if$
}

FUNCTION {author.key.organization.label}
{ author empty$ }
{ key empty$ }
{ cite$ #1 #3 substring$ }
"The " #4 organization chop.word #3 text.prefix$ }
if$
}
{ author format.names.key }
if$
}

FUNCTION {editor.key.organization.label}
{ editor empty$ }
{ key empty$ }
{ cite$ #1 #3 substring$ }
"The " #4 organization chop.word #3 text.prefix$ }
if$
}
"key
if$
}
{ author format.names.key }
if$
}
calc.short.list Calculates short.list for \bibitem label in natbib styles depending on the type of entry.

\textit{(NB: Нужна внимательная ревизия логики вычислений. В общем-то, она имеет значения в особом случае, когда поле author пустое.)}

\begin{verbatim}
FUNCTION {calc.short.list}
{ type$ "book" = type$ "inbook" = or
  'author.editor.key.label
  { type$ "proceedings" =
    'editor.key.organization.label
    { type$ "manual" =
      'author.key.organization.label
      'author.key.label
     if$
  }
  }
  if$
}
if$
'short.list :=
}
\end{verbatim}

\textit{NB: Вопреки названию, вычисляет только часть метки в [] для \bibitem. Скобка после года не закрыта, так как еще может быть добавлено a, b, .... Лучше перенести весь код output.bibitem.}

\begin{verbatim}
FUNCTION {calc.label}
{ calc.short.list
  short.list
  "("
  *
  % year duplicate$ empty$
  year.or.date.to.year duplicate$ empty$
  short.list.key field.or.null = or
  { pop$ "" }
  'skip$
  if$
  *
  'label :=
  25 }
}
\end{verbatim}

calc.long.list In case of natbib option, we need calc.long.list to compose output.bibitem, and the latter, in its turn, requires some more functions.

\begin{verbatim}
FUNCTION {calc.long.list} % called 1 time only
{ type$ "book" = type$ "inbook" = or
  %'format.author.editor.brief
\end{verbatim}
6.11 Output functions (continued)

address.publisher Outputs address (or location) and publisher fields separated by colon if both fields are available; otherwise outputs that field which is not empty.
\begin{verbatim}
FUNCTION {output.bibitem}
{
  set.language
  newline$
  "\bibitem" write$
  ⟨*natbib⟩
  label extra.label * ) *
  calc.long.list *
  bracketise write$
  ⟨/natbib⟩
  cite$ bracify write$
  newline$
  "\selectlanguageifdefined" curlanguage bracify * write$
  newline$
  ""
  before.all 'output.state :=
}
⟨/strict⟩
\end{verbatim}

\textbf{output.bibitem} \quad Is called at the beginning of any entry. It sets \texttt{curlanguage} string variable based on \texttt{langid} or \texttt{language} field.

\begin{verbatim}
FUNCTION {output.bibitem}
{
  set.language
  newline$
  "\bibitem" write$
  ⟨*natbib⟩
  label extra.label * ) *
  calc.long.list *
  bracketise write$
  ⟨/natbib⟩
  cite$ bracify write$
  newline$
  "\selectlanguageifdefined" curlanguage bracify * write$
  newline$
  ""
  before.all 'output.state :=
}
⟨/strict⟩
\end{verbatim}

\textbf{output.bibitem} \quad Is called at the beginning of any entry. It sets \texttt{curlanguage} string variable based on \texttt{langid} or \texttt{language} field.

\begin{verbatim}
FUNCTION {output.bibitem}
{
  set.language
  newline$
  "\bibitem" write$
  ⟨*natbib⟩
  label extra.label * ) *
  calc.long.list *
  bracketise write$
  ⟨/natbib⟩
  cite$ bracify write$
  newline$
  "\selectlanguageifdefined" curlanguage bracify * write$
  newline$
  ""
  before.all 'output.state :=
}
⟨/strict⟩
\end{verbatim}

\textbf{output.bibitem} \quad Is called at the beginning of any entry. It sets \texttt{curlanguage} string variable based on \texttt{langid} or \texttt{language} field.
6.12 Formatting title, booktitle, etc.

Important note
Neither \texttt{bibtex.exe} nor \texttt{bibtex8.exe} can handle unicoded text without troubles. In particular, \texttt{bibtex8} fails to change case of a string if it contains Cyrillic letter. Therefore we avoid using \texttt{change.case} when option \texttt{utf8} is applied.

\begin{verbatim}
FUNCTION {format.bvolume}
  { volume empty$ }
  { "" }
  { bbl.vvol volume tie.connect
    series empty$
    "skip"
    { bbl.of spaces.around * series emphasize * }
    if$ }
  { "volume and number" number either.or.check }
  if$ }

FUNCTION {format.number.series}
  { volume empty$ }
  { number empty$ }
  { series field.or.null }
  { series empty$
    { "there's a number but no series in " cite$ * warning$
      bbl.nnr }
    }
  { %new.dblslash
    new.sentence

51
eng.ord Is not currently used. (NB: Note that bbl.st, bbl.nd, bbl.rd, bbl.th are not defined.)

convert.edition

1962 series
1963 bbl.nr
1964 tie.or.space.connect}
1965 if$
1966 number tie.or.space.connect
1967$
1968 if$
1969$
1970 { "=" }
1971 if$
1972$
1973

FUNCTION {eng.ord}
1974 { duplicate$ "1" swap$ *
1975 #-2 #1 substring$ "1" =
1976 { bbl.th * }
1977 { duplicate$ #-1 #1 substring$
1978  duplicate$ "1" =
1979  { pop$ bbl.st * }
1980  { duplicate$ "2" =
1981  { pop$ bbl.nd * }
1982  { "3" =
1983  { bbl.rd * }
1984  { bbl.th * }
1985  if$}
1986 $
1987 if$
1988$
1989 if$
1990 $
1991 if$
1992$
1993 if$
1994$
1995 (/debug)
1996

convert.edition

1997 FUNCTION {convert.edition}
1998 { edition
1999 % edition extract.num "1" change.case$ 's :=
2000 % s "first" = s "1" = or
2001 % { bbl.first 't := }
2002 % { s "second" = s "2" = or
2003 % { bbl.second 't := }
2004 % { s "third" = s "3" = or
2005 % { bbl.third 't := }
2006 % { s "fourth" = s "4" = or
2007 % { bbl.fourth 't := }
2008 % { s "fifth" = s "5" = or

52
function: format.edition

function: format.pages
format.pages.page

FUNCTION {format.pages.page}
{ eid empty$ 
{ pages empty$ 
{ pagetotal empty$ 
{ "" } 
{ pagetotal bbl.pages tie.connect } 
if$ 
} 
} 
{ format.pages} 
if$ 
} 
{ format.pages } 
if$ 
} 
if$ 
}

format.vol.num.pages

FUNCTION {format.vol.num.pages}
{ volume field.or.null 
 number empty$ 
 'skip$ 
 { "", no." number tie.or.space.connect * 
 volume empty$ 
 { "there's a number but no volume in " cite$ * warning$ } 
 'skip$ 
 if$ 
} 
if$ 
} 
if$ 
}

format.volume

FUNCTION {format.volume}
{ volume empty$ 
 { "" } 
{ bbl.vvol volume tie.or.space.connect } 
if$ 
}
format.number

FUNCTION {format.number}
{ number empty$ 
  "" }
{ bbl.nr number tie.or.space.connect } if$
}

mat.chapter.pages

(*debug)
FUNCTION {format.chapter.pages}
{ chapter empty$ 
  "format.pages 
  { type empty$ 
    { bbl.chapter }
    { type "I" change.case$ } if$
    chapter tie.or.space.connect 
    pages empty$ 
    'skip$
    { ", " * format.pages * } if$
    } if$
  } if$
}

empty.misc.check

FUNCTION {empty.misc.check}
{ author empty$ title empty$ howpublished empty$ 
  month empty$ year empty$ note empty$ 
  and and and and and 
  key empty$ not and 
  { "all relevant fields are empty in " cite$ * warning$ } 
  'skip$
  if$
}

bbl.thesis.type

FUNCTION {bbl.thesis.type}
{ type "mathesis" = 
  { bbl.mathesis } 
  { type "phdthesis" = 
    { bbl.phdthesis } 
  }
}

55
Functions format.thesis.type and format.techreport.type number.

Function to format report type and number.

Formats and writes list of authors in the heading of a bibliographic record. The GOST bibstyles skip the list of authors in the beginning of the bibliographic record if the number of authors is 4.
or larger except for the styles with suffix 1 which prints all authors in the heading. The bibstyles compiled with the option \texttt{struct print} print at most 1 person name in the heading.

\begin{verbatim}
FUNCTION {output.author.head} {
  author empty$
  'skip$
  (*long)
  { author fmt.names.all outputnonnull
    new.sentence
  }
  (/long)
  (*!long)
  {author num.names$ #4 <
    { (*strict) author fmt.names.first output
      (*!strict) author fmt.names.three output
      new.sentence
    }
    'skip$
    if$
  }
  if$
}

FUNCTION {output.author.rest} {
  (*!long)
  author empty$
  'skip$
  (*long)
  {author num.names$ #1 >
    (*strict) author num.names$ #3 >
    { author fmt.names.all output
      newsemicolon
    }
    'skip$
    if$
  }
  if$
  (/!long)
  if$
}

{bookauthor.head} There are also 2 version of the function \texttt{bookauthor.head}. Not used anymore!

{output.author.rest} Writes the rest of authors list after the entry title followed by a slash. In modern bibstyles, the list of authors in the heading part of a bibliographic record is cut to either 3 persons (if no \texttt{!strict} option is applied) or 1 person (if \texttt{strict} option). By idea, \texttt{output.author.rest} should print the rest of the authors in the \texttt{author} field but currently it also repeats the persons printed in the heading part. \texttt{output.author.rest} returns a null string ("") if all persons are printed in the heading part of the record. Note that all persons are always printed in the heading part if \texttt{long} option is applied.

\end{verbatim}
bookauthor.rest  This function is used only in inbook entry. It always cuts list to 4 persons since format.bookauthors.rest does that.

```
FUNCTION {bookauthor.rest}
{ bookauthor empty$ 'skip$ }
(strict) bookauthor fmt.names.all output
(!strict) bookauthor fmt.names.three output
new.semicolon if$
}
```

editor.organization.rest

```
FUNCTION {editor.organization.rest}
{ compiler empty$ { }
(format.compiler.rest output new.semicolon if$
editor empty$ { }
(format.editors.rest outputnonnull new.semicolon if$
organization empty$ { }
(organization outputnonnull new.semicolon if$
if$
)
}

format.url

```
FUNCTION {format.url}
{ url empty$ ""
{ bbl.url ": \BibUrl\(* *\)\) * urldate empty$ ""
{ "( " bbl.urldate ": * * urldate *\)\) * if$ *
}

if$
```
output.url

FUNCTION {output.url}
{ url empty$ 'skip$ { format.url output } if$ }

format.annotate

FUNCTION {format.annotate}
{ annotate empty$ "" if$ }
{ after.sentence 'output.state := "\BibAnnote{" annotate add.period$ * "}" * if$ }

format.isbn

FUNCTION {format.isbn}
{ isbn empty$ "" if$ }
{ "ISBN:\href{http://isbndb.com/search-all.html?kw=" isbn * "}" * if$ }

add.doi

FUNCTION {add.doi}
{ duplicate$ empty$ 'skip$ { doi empty$ 'skip$ { "\href{https://doi.org/" doi * "}(" * swap$ * ")" * } if$ }

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 identificator at https://doi.org/ web-site.
If `.bst` style is compiled without `eprint` option, we just ignore `doi` field.

```latex
FUNCTION {add.doi} { }
```

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

```latex
FUNCTION {add.media} { duplicate$ empty$ 'skip$ { media empty$ 'skip$ { " " * bbl.media bracketise * } if$ } if$ } if$ } if$ }
```

6.13 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”:

\begin{verbatim}
eprint = {identifier},
eprinttype = {Resource},
\end{verbatim}

\texttt{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.

\begin{verbatim}
function should use url. TO BE DONE YET.}
\end{verbatim}
Functions added in v1.2f to format patent entry (thanks to Stanislav Kruchinin).

add.number

format.type.number
6.14 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.
6.14.1 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: editor, editora, editorb, editorc, journalsubtitle, issuetitle, language, origlanguage, series, volume, number, eid, issue, month, pages, version, note, issn, addendum, pubstate, doi, eprint, eprintclass, eprinttype, url, urldate.

FUNCTION {article}

```
{output.bibitem
  output.author.head
  new.sentence
  ⟨natbib⟩author format.key output
  title add.media "title" output.check
  new.slash
  output.author.rest
  new.dblslash
  journal emphasize add.doi "journal" output.check % new in v1.2
  new.block
  format.date "year/date" output.check
  new.block
  format.volume output
  format.number output
  new.block
  format.pages.page output
  new.block
  note output
  new.sentence
  format.url output
  format.annote output
  fin.entry}
```

**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.

FUNCTION {book}

```
{2487
  output.bibitem
  output.author.head
  new.sentence
  ⟨natbib⟩author format.key output
  title add.media "title" output.check
  new.slash
  output.author.rest
  new.dblslash
  journal emphasize add.doi "journal" output.check % new in v1.2
  new.block
  format.date "year/date" output.check
  new.block
  format.volume output
  format.number output
  new.block
  format.pages.page output
  new.block
  note output
  new.sentence
  format.url output
  format.annote output
  fin.entry
  }
2488
```
A book-like work without a formal publisher or sponsoring institution. Use the field howpublished to supply publishing information in free format, if applicable. The field type may be useful as well.

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

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

FUNCTION {booklet}
{
  output.bibitem
  output.author.head
  new.sentence
  {natbib} author format.key output
  title add.doi add.media "title" output.check
  new.colon % added in v.1.2k
  titleaddon output % added in v.1.2k
  new.slash
  output.author.rest
  editor.organization.rest
  new.sentence
  format.number.series output
  new.block
  format.edition output
  new.block
  output.address.publisher
  format.date "year/date" output.check
  new.block
  format.bvolume output
  new.block
  format.pages.page output
  new.block
  (eprint) format.isbn output
  (eprint) new.block
  note output
  new.sentence
  % format.url output
  output.eprint.url
  format.annote output
  fin.entry
  }

booklet
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.

FUNCTION {inbook}
{
output.bibitem
output.author.head
new.sentence
(natbib) author format.key output
new.colon  \% added in v.1.2k
new.slash
output.author.rest
new.dblslash
\% bookauthor.head
booktitle "booktitle" output.check
new.slash
bookauthor.rest
editor.organization.rest
new.block
format.edition output
new.block
format.number.series output
new.sentence
output.address.publisher
format.date "year/date" output.check
new.block
format.bvolume output
new.block
}
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.

FUNCTION {incollection}
{
  output.bibitem
  output.author.head
  new.sentence
  {natbib} author format.key output
  new.sentence
  title add.doi add.media "title" output.check
  new.colon  % added in v.1.2k
  titleaddon output  % added in v.1.2k
  new.slash
  output.author.rest
  new.dbslash
  booktitle "booktitle" output.check
  new.slash
  editor.organization.rest
  new.block
  output.address.publisher
  format.date "year/date" output.check
  new.block
  format.bvolume output
  format.number.series output
  new.block
  format.pages.page output
  new.block
  note output
  new.sentence
  % format.url output
  output.eprint.url
} fin.entry
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.

FUNCTION {proceedings}
{
  output.bibitem
  editor format.key output
  title add.doi add.media "title" output.check
  new.colon % added in v.1.2k
  titleaddon output % added in v.1.2k
  new.slash
  editor.organization.rest
  new.block
  output.address.publisher % 1.2k moved from below
  new.block % added in v.1.2k
  format.date "year/date" output.check
  new.block
  format.bvolume output
  format.number.series output
  new.block
  format.pages.series output
  %%%output.address.publisher % 1.2k moved upper
  new.block
  note output
  new.sentence
  % format.url output
  output.eprint.url
  output.eprint.url
  format.annote output
  fin.entry
}

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.

FUNCTION {inproceedings}
{
  output.bibitem
}
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.

FUNCTION {manual}

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

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

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

FUNCTION {patent}
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.

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

FUNCTION {unpublished}

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.

internet  New in version 2012.02.15.

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
report

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.

%FUNCTION {report}
{ report

% output.bibitem
% output.author.head
% new.sentence
% title.add.doi.add.media "title" output.check
% new.colon
% format.techrep.type.number outputnonnull
% type "type" output.check
% new.slash
% output.author.rest
% editor.organization.rest
% new.block
% address output
% new.colon
% institution "institution" output.check
% format.date "year/date" output.check
% new.block
% note output
FUNCTION {report}
{
output.bibitem
% output.author.head
% new.sentence
{title add.doi add.media "title" output.check
⟨natbib⟩ title format.key output
new.colon % added in v.1.2k
titleaddon output % added in v.1.2k
new.colon
% format.techrep.type.number output.nonnull
% type "type" output.check
% format.report.type.number "type" output.check
type "type" output.check
new.colon
number output
new.slash
%institution "institution" output.check
institution.or.school "institution/school" output.check
newsemicolon
format.chief.rest output % from editor field
ewsemicolon
format.executor.rest output % from author field
new.block
address.or.location output
new.colon
organization output
%format.date "year/date" output.check
new.block % v.2
% new.sentence % или new.block ?
output.eprint.url
new.block
note output
format.annote output
fin.entry
}
}
6.14.2 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.

```latex
FUNCTION {phdthesis}
{ output.bibitem
  format.author "author" output.check
  ⟨natbib⟩ author format.key output
  new.sentence
  title add.doi add.media "title" output.check
  new.colon
  bbl.phdthesis format.thesis.type outputnonnull
  new.colon
  %%number output % code of the speciality
  specialitycode.or.number output % code of the speciality, new in v.1.2i
  new.colon
  titleaddon output % date of defence and approval; new in v.1.2i
  new.slash
  %%%format.author.rest output %% duplicates athours
  output.author.rest % prints if num.names$ > 3 or > 1
  new.semicolon
  %institution "institution" output.check
  institution.or.school "institution/school" output.check
  new.block
  output.address.publisher
  format.date "year/date" output.check
  new.block
  format.pages.page output
  new.block
  new.sentence
  % format.url output
  output.eprint.url
  new.sentence
  format.annote output
  fin.entry
}
```

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.

```latex
FUNCTION {mastersthesis}
{ output.bibitem
  format.author "author" output.check
  ⟨natbib⟩ author format.key output
  new.sentence
  title add.doi add.media "title" output.check
  new.colon
```
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.

FUNCTION {docthesis}
{ output.bibitem
  format.author "author" output.check
  {natbib} author format.key output
  new.sentence
  title add.doi add.media "title" output.check
  new.block
  bbl.docthesis format.thesis.type outputnonnull
  new.colon
  specialitycode.or.number output % code of the speciality, new in v.1.2i
  new.colon
  titleaddon output % date of defence and approvement; new in v.1.2i
  new.slash
  %format.author.rest output % duplicates authors
  output.author.rest % prints if num.names$ > 3 or > 1
  new.semicolon
  institution "institution" output.check
  institution.or.school "institution/school" output.check
  new.block
  output.address.publisher
  format.date "year/date" output.check
  new.block
  format.pages.page output
  new.block
  note output
  new.sentence
  format.url output
  output.eprint.url
  format.annote output
  fin.entry
}
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.
6.15 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.

MACRO {jan} {"Jan."}
MACRO {feb} {"Feb."}
MACRO {mar} {"Mar."}
MACRO {apr} {"Apr."}
MACRO {may} {"May"}
MACRO {jun} {"Jun."}
MACRO {jul} {"Jul."}
MACRO {aug} {"Aug."}
MACRO {sep} {"Sep."}
MACRO {oct} {"Oct."}
MACRO {nov} {"Nov."}
6.16 Journal Abbreviations

6.16.1 Physics and astronomy

Borrowed from physjour.mbs of package custom-bib.
6.16.2 Supplementary Journal Names

Borrowed from suppjour.mbs of package custom-bib.

3211 MACRO {cjp} {"Czech. J. Phys."}
3212 MACRO {el} {"Europhys. Lett."}
3213 MACRO {en} {"Europhys. News"}
3214 MACRO {fujitsustj} {"FUJITSU Sci. Tech. J."}
3215 MACRO {ieeceed} {"IEEE Trans. Electron Devices"}
3216 MACRO {ieeeim} {"IEEE Trans. Instrum. Meas."}
3217 MACRO {ieeejqe} {"IEEE J. Quantum Electron."}
3218 MACRO {ieeeem} {"IEEE Trans. Magn."}
3219 MACRO {ieeeptl} {"IEEE Photonic Technol. Lett."}
3220 MACRO {ieeeuffc} {"IEEE Trans. Ultrason., Ferroelect., Freq. Cont."}
3221 MACRO {jem} {"J. Electron. Mater."}
3222 MACRO {jes} {"J. Electrochem. Soc."}
3223 MACRO {jetplet} {"JETP Lett."}
3224 MACRO {jjap} {"Japan. J. Appl. Phys."}
3228 MACRO {jphc} {"J. Phys.: Condens. Matter"}
3229 MACRO {jphcold} {"J. Phys. C: Solid State Phys."}
3230 MACRO {jphd} {"J. Phys. D: Appl. Phys."}
3231 MACRO {jvsta} {"J. Vac. Sci. Technol. A"}
3232 MACRO {jvstb} {"J. Vac. Sci. Technol. B"}
3233 MACRO {me} {"Microelectron. Eng."}
3234 MACRO {necrd} {"NEC Res. Develop."}
3235 MACRO {pa} {"Physica A"}
3236 MACRO {pb} {"Physica B"}
3237 MACRO {pc} {"Physica C"}
3238 MACRO {pd} {"Physica D"}
3239 MACRO {procieee} {"Proc. IEEE"}
3240 MACRO {procspie} {"Proc. SPIE"}
3243 MACRO {rpp} {"Rep. Progr. Phys."}
3244 MACRO {sm} {"Synthet. Metal"}
3245 MACRO {sost} {"Solid State Technol."}
3246 MACRO {ss} {"Surf. Sci."}
3247 MACRO {ssc} {"Solid State Commun."}
3248 MACRO {sst} {"Semicond. Sci. Technol."}
3249 MACRO {suplatt} {"Superlatt. Microstr."}
3250 MACRO {sst} {"Supercond. Sci. Technol."}
3251 MACRO {znat} {"Z. Naturforsch."}

6.16.3 Optics

Borrowed from photjour.mbs.

3252 MACRO {appopt} {"Appl. Opt."}
3253 MACRO {bell} {"Bell Syst. Tech. J."}

82
6.16.4 Physics of condensed Matter

6.16.5 Soviet and Russian journals

6.17 Main cycle

6.18 Sorting

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

FUNCTION {sortify}
{ purify$ ⟨utf8⟩ "l" change.case$ }
⟨/sort | natbib⟩

FUNCTION {sort.format.names}
{ 's := num.names$ 'numnames :=
numnames 'namesleft :=
{ namesleft #0 > }:
{nameptr #1 >
" " " * }
'skip$
if$
nombre nameptr
"{vv{ }}{ll{ }}{ ff{ } }{ jj{ } }"
format.name$ 't :=
nombre numnames = t "others" = and
{ "et al" * }
{ bbl.etal * }
{ t sortify * }
if$

nameptr #1 + 'nameptr :=

while$
}

FUNCTION {sort.format.names}
{ 's :=
#1 'nameptr :=
""
's num.names$ 'numnames :=
numnames 'namesleft :=
{ namesleft #0 > }
{nameptr #1 >
" " " * }
'skip$
if$
nombre nameptr
"{vv{ }}{ff{ } }{ jj{ } }"
format.name$ 't :=
nombre nameptr #1 + 'nameptr :=

while$
}
sort.format.title

FUNCTION {sort.format.title}
{ 't :=
  "A " #2
  "An " #3
  "The " #4 t chop.word % Removes "The " if any
  chop.word % Removes "An " if any
  chop.word % Removes "A " if any
  sortify
  #1 global.max$ substring$
}

author.sort

%% This version from old gost package.
%%
{!*natbib}
FUNCTION {author.sort}
{ author empty$
  { key empty$
    { "to sort, need author or key in " cite$ * warning$ ""
    }
    { key sortify }
  } if$
  }
  }
  { author num.names$ #4 <
The function \texttt{author.title.sort} is used in the \texttt{presort} function only.

\begin{verbatim}
FUNCTION \{author.title.sort\}
{ author empty$ 
  { title empty$
    { key empty$
      { "to sort, need author, title, or key in " cite$ * warning$ ""
        { key sortify }
        if$
      } 
      if$
    } 
    { title sort.format.title }
    if$
  } 
  if$
  }

{ author.num.names$ #4 <
  {author.sort.format.names }
  {title.sort.format.name}
  if$
}

if$
\end{verbatim}

The function \texttt{author.editor.sort}, \texttt{author.organization sort}, \texttt{editor.organization sort} are not currently used. See commented text in function \texttt{presort#2}. \textbf{(NB: WE NEED TO CORRECT THAT!)}
%FUNCTION {author.editor.sort}
%{ author empty$
%   { editor empty$
%      { key empty$
%         { "to sort, need author, editor, or key in " cite$ * warning$
%         ""
%         }
%      }% key sortify
%      if$
%   }% editor sort.format.names}
% if$
% }% if$
% %
%FUNCTION {author.organization.sort}
%{ author empty$
%{ organization empty$
% { key empty$
%   { "to sort, need author, organization, or key in " cite$ * warning$
%   ""
%   }
% }% key sortify
% if$
% }% if$
% { "The " #4 organization chop.word sortify }%
% if$
% }% if$
% %
%FUNCTION {editor.organization.sort}
%{ editor empty$
%{ organization empty$
% { key empty$
%   { "to sort, need editor, organization, or key in " cite$ * warning$
%   ""
%   }
% }% key sortify
% if$
% }% if$
% { "The " #4 organization chop.word sortify }
% if$
% }% if$
Function to compute \texttt{sort.key}. What is the space string "\hspace{3em}" for? Version \#1 is for 'sort' and '!natbib' options. Version \#2 is for 'sort' and 'natbib' options. Version \#3 is for '!sort' and 'natbib' options.

```latex
\begin{verbatim}
FUNCTION {presort}%#1
{
  author.title.sort " " *
  year field.or.null sortify *
  " "
  title field.or.null
  sort.format.title *
  #1 entry.max$ substring$
  'sort.key$ :=
}
\end{verbatim}
```

---

```latex
\begin{verbatim}
FUNCTION {presort}%#2
{
  calc.label
  label sortify
  %author.title.sort
  " "
  *
  % ========= plainnat.bst =========
  % type$ "book" =
  % type$ "inbook" =
  % or
  % author.editor.sort
  % { type$ "proceedings" =
  %   'editor.organization.sort
  %   { type$ "manual" =
  %     'author.organization.sort
  %     'author.sort
  %   if$
  % } if$
  % } if$
  % if$
  author.title.sort " " *
  " "
  year field.or.null sortify *
  " "
\end{verbatim}
```
\%cite$
title field.or.null sort.format.title
*
#1 entry.max$ substring$
'sort.label :=
sort.label *
\% =================================================
#1 entry.max$ substring$
'sort.key$ :=

\langle /natbib \rangle
\langle /sort \rangle
\langle* !sort \rangle
\langle* natbib \rangle
INTEGERS { seq.num }

FUNCTION {init.seq}
{ #0 'seq.num :=}
EXECUTE {init.seq}

FUNCTION {int.to.fix}
{ "000000000" swap$ int.to.str$ *
#-1 #10 substring$
}

FUNCTION {presort}%#3
{
calc.label % computes label
label sortify % initiates sort.label
" "
*
seq.num #1 + 'seq.num := % advance seq.num
seq.num int.to.fix % prepend seq.num with 0s
'sort.label := % set sort.label to seq.num
sort.label * % append seq.num to label
#1 entry.max$ substring% cut if too long
'sort.key$ := % set sort.key$
}

\langle /natbib \rangle
\langle /sort \rangle
\langle* sort | natbib \rangle
ITERATE {presort}

SORT

\langle /sort | natbib \rangle
6.19 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.

```latex
\begin{verbatim}
\begin{verbatim}
\newcommand{\longestlabel}{longest.label}
\newcommand{\lastlabel}{last.label}
\newcommand{\nextextra}{next.extra}
\newcommand{\numberlabel}{number.label}
\newcommand{\longestlabelwidth}{longest.label.width}
\newcommand{\lastextranum}{last.extra.num}
\end{verbatim}
\end{verbatim}
\end{verbatim}
```

```latex
\begin{verbatim}
\begin{verbatim}
\newcommand{\numberlabel}{number.label}
\newcommand{\longestlabelwidth}{longest.label.width}
\newcommand{\lastextranum}{last.extra.num}
\end{verbatim}
\end{verbatim}
```

Iterate though the list of entries to compute label.

```latex
\begin{verbatim}
\begin{verbatim}
\newcommand{\forwardpass}{forward.pass}
\newcommand{\numberlabel}{number.label}
\newcommand{\longestlabelwidth}{longest.label.width}
\newcommand{\lastextranum}{last.extra.num}
\end{verbatim}
\end{verbatim}
```

```latex
\begin{verbatim}
\begin{verbatim}
\newcommand{\forwardpass}{forward.pass}
\newcommand{\numberlabel}{number.label}
\newcommand{\longestlabelwidth}{longest.label.width}
\newcommand{\lastextranum}{last.extra.num}
\end{verbatim}
\end{verbatim}
```
within the \texttt{thebibliography} environment we define few formatting macros for user to customize how the reference list is formatted.

\begin{verbatim}
FUNCTION {begin.bib}
{ \"begin\{thebibliography\}\" longest.label * \"\} * write$ newline$
\end{verbatim}
7 Change History

v0.8
General: entry field annote added ........... 8
macro \BibAnnote added ........... 59, 91
macro \BibEmph added ........... 15, 91
macro \BibUrl added ........... 58, 91
v0.9
General: bug fix in @inproceedings entry
type ......................... 68

v1.1
General: Entry type @online added ........... 73
entry type @online added ........... 14
v1.2
General: @report entry ..................... 74
@thesis entry ..................... 73
entry field medium added ........... 8
entry fields eprint, eprintclass,
8 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.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>add.blank</td>
<td>13</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
</tr>
<tr>
<td>add.doi</td>
<td>59</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
</tr>
<tr>
<td>add.media</td>
<td>60</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
</tr>
<tr>
<td>add.number</td>
<td>62</td>
<td>19</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
</tr>
<tr>
<td>address.or.location</td>
<td>37</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
</tr>
<tr>
<td>and</td>
<td>13</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>article</td>
<td>64</td>
<td>18</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
</tr>
<tr>
<td>author.editor.key.label</td>
<td>47</td>
<td>19</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>author.editor.sort</td>
<td>86</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>author.key.label</td>
<td>46</td>
<td>29</td>
<td>29</td>
<td>29</td>
<td>29</td>
<td>29</td>
<td>29</td>
<td>29</td>
<td>29</td>
<td>29</td>
<td>29</td>
<td>29</td>
<td>29</td>
<td>29</td>
<td>29</td>
</tr>
<tr>
<td>author.key.organization.label</td>
<td>47</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
</tr>
<tr>
<td>author.organization.sort</td>
<td>86</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>35</td>
</tr>
<tr>
<td>author.sort</td>
<td>85</td>
<td>24</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
</tr>
<tr>
<td>author.title.sort</td>
<td>86</td>
<td>20</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>bbl.and</td>
<td>22</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>21</td>
</tr>
</tbody>
</table>

93
<table>
<thead>
<tr>
<th>Field or Null</th>
<th>Transform</th>
<th>Constant</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>extract.num</td>
<td>16</td>
</tr>
<tr>
<td>15</td>
<td>field.or.null</td>
<td>15</td>
</tr>
<tr>
<td>18</td>
<td>multi.page.check</td>
<td>18</td>
</tr>
<tr>
<td>14</td>
<td>new.dblslash.checka</td>
<td>14</td>
</tr>
<tr>
<td>14</td>
<td>new.dblslash.checkb</td>
<td>14</td>
</tr>
<tr>
<td>14</td>
<td>new.dblslash</td>
<td>14</td>
</tr>
<tr>
<td>14</td>
<td>new.sentence.checkb</td>
<td>14</td>
</tr>
<tr>
<td>14</td>
<td>new.sentence.checka</td>
<td>14</td>
</tr>
<tr>
<td>14</td>
<td>new.sentence</td>
<td>14</td>
</tr>
<tr>
<td>13</td>
<td>new.stop</td>
<td>13</td>
</tr>
<tr>
<td>13</td>
<td>not</td>
<td>13</td>
</tr>
<tr>
<td>12</td>
<td>n.dashify</td>
<td>12</td>
</tr>
<tr>
<td>12</td>
<td>reset.language</td>
<td>12</td>
</tr>
<tr>
<td>12</td>
<td>reverse.pass</td>
<td>12</td>
</tr>
<tr>
<td>11</td>
<td>report</td>
<td>11</td>
</tr>
<tr>
<td>11</td>
<td>output.author</td>
<td>11</td>
</tr>
<tr>
<td>11</td>
<td>output.nonnull</td>
<td>11</td>
</tr>
<tr>
<td>11</td>
<td>output</td>
<td>11</td>
</tr>
<tr>
<td>10</td>
<td>set.language</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>sort.format.names</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>sort.format.title</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>sort</td>
<td>10</td>
</tr>
<tr>
<td>9</td>
<td>tie.connect</td>
<td>9</td>
</tr>
<tr>
<td>9</td>
<td>tie.or.space.connect</td>
<td>9</td>
</tr>
<tr>
<td>9</td>
<td>www</td>
<td>9</td>
</tr>
<tr>
<td>9</td>
<td>year.or.date.to.year</td>
<td>9</td>
</tr>
<tr>
<td>8</td>
<td>Initialize.longest.label</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>ln.dashify</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>ln.dashify.checka</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>ln.dashify.checkb</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>ln.dashify</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>ln.dashify.checka</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>ln.dashify.checkb</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>ln.dashify</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>ln.dashify.checka</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>ln.dashify.checkb</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>ln.dashify</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>ln.dashify.checka</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>ln.dashify.checkb</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>ln.dashify</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>ln.dashify.checka</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>ln.dashify.checkb</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>ln.dashify</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>ln.dashify.checka</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>ln.dashify.checkb</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>ln.dashify</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>ln.dashify.checka</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>ln.dashify.checkb</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>ln.dashify</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>ln.dashify.checka</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>ln.dashify.checkb</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>ln.dashify</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>ln.dashify.checka</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>ln.dashify.checkb</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>ln.dashify</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>ln.dashify.checka</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>ln.dashify.checkb</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>ln.dashify</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>ln.dashify.checka</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>ln.dashify.checkb</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>ln.dashify</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>ln.dashify.checka</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>ln.dashify.checkb</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>ln.dashify</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>ln.dashify.checka</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>ln.dashify.checkb</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>ln.dashify</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>ln.dashify.checka</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>ln.dashify.checkb</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>ln.dashify</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>ln.dashify.checka</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>ln.dashify.checkb</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>ln.dashify</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>ln.dashify.checka</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>ln.dashify.checkb</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>ln.dashify</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>ln.dashify.checka</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>ln.dashify.checkb</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>ln.dashify</td>
<td>0</td>
</tr>
</tbody>
</table>