

BibTool Quick Reference Card

for BibTool version 2.68 — see also <http://www.gerd.neugebauer.de/software/TeX/BibTool/>
©2019 Gerd Neugebauer (gene@gerd-neugebauer.de)

Command line options	General	Sorting	Checks
<code>-rsc_command</code> Perform resource command as if given in a file.	<code>resource.search.path = {dir₁:dir₂...}</code> <code>resource {file}</code> <code>bibtex.search.path = {dir₁:dir₂...}</code> <code>bibtex.env.name = {ENV_NAME}</code> <code>env.separator = {c}</code> <code>dir.file.separator = {c}</code> <code>print {message}</code> <code>quiet = OnOff</code> <code>verbose = OnOff</code>	<code>sort = OnOff</code> <code>sort.cased = OnOff</code> <code>sort.reverse = OnOff</code> <code>sort.format = {format}</code> <code>sort.order {...}</code> <code>sort.macros = OnOff</code>	<code>check.double = OnOff</code> <code>check.do.delete = OnOff</code> <code>check.rule {field # pattern # message}</code> <code>check.warning.rule {field # pattern # message}</code> <code>check.error.rule {field # pattern # message}</code> <code>check.case.sensitive = OnOff</code> <code>unique.field {field}</code>
<code>-A type</code> Determine key disambiguation. type in 0, a, A,			
<code>-d</code> Check double entries.			
<code>-f key_format</code> Generate keys according to key_format			
<code>-F</code> Enable key generation with free key format.			
<code>-h</code> Print short help and exit.			
<code>-i input_file</code> Mark a file to be processed later.			
<code>-k</code> Make keys with the short format.			
<code>-K</code> Make keys with the long format.			
<code>-o output_file</code> Send the output to output_file.			
<code>-q</code> Suppress warning messages.			
<code>-r resource_file</code> Read the resource file resource_file.			
<code>-R</code> Load the default resource file now.			
<code>-s</code> Sort the result.			
<code>-S</code> Sort the result in reverse order.			
<code>-v</code> Turn on verbose messages about the actions performed.			
<code>-x aux_file</code> Extract those entries mentioned in aux_file.			
<code>-X regex</code> Extract entries matching regex.			
	<h3>Reading and Printing</h3> <code>input {bib_file}</code> <code>output.file = {file}</code> <code>parse.exit.on.error = OnOff</code> <code>pass.comments = OnOff</code> <code>new.entry.type {type}</code> <code>print.align = n</code> <code>print.align.key = n</code> <code>print.align.preamble = n</code> <code>print.align.comment = n</code> <code>print.braces = OnOff</code> <code>print.comma.at.end = OnOff</code> <code>print.deleted.entries = OnOff</code> <code>print.deleted.prefix = {prefix}</code> <code>print.indent = n</code> <code>print.line.length = n</code> <code>print.newline = n</code> <code>print.parentheses = OnOff</code> <code>print.terminal.comma = OnOff</code> <code>print.use.tab = OnOff</code> <code>print.wide.equal = OnOff</code> <code>suppress.initial.newline = OnOff</code> <code>new.field.type {new=old}</code> <code>symbol.type = type</code> <code>upper, lower, cased</code>	<h3>Searching (Extraction)</h3> <code>tex.define {macro[arg]=text}</code> <code>extract.file {file}</code> <code>select {field₁...field_n "regex"}</code> <code>select {type₁...type_n}</code> <code>select.by.string {field₁...field_n "regex"}</code> <code>select.by.string.ignore {chars}</code> <code>select.case.sensitive = OnOff</code> <code>select.fields = {field₁,field₂,...}</code> <h3>Field Manipulation</h3> <code>add.field {field="value"}</code> <code>delete.field {field}</code> <code>keep.field {field}</code> <code>keep.field {field if field₂="pattern"}</code> <code>rename.field {old=new}</code> <code>rename.field {old=new if field="pattern"}</code> <code>rewrite.rule { pattern }</code> <code>delete all matching fields</code> <code>rewrite.rule { pattern # replacement }</code> <code>rewrite.all fields</code> <code>rewrite.rule {f₁...f_n # pattern # replacement }</code> <code>rewrite some fields</code> <code>rewrite.case.sensitive = OnOff</code> <code>rewrite.limit = {n}</code>	<h3>Strings</h3> <code>macro.file {file}</code> <code>print.all.strings = OnOff</code> <code>expand.macros = OnOff</code> <h3>Inheritance</h3> <code>crossref.map = OnOff</code> <code>clear.crossref.map {}</code> <code>crossref.limit = {n}</code> <code>expand.crossref = OnOff</code> <code>expand.xdata = OnOff</code> <h3>Bib_TE_X1.0</h3> <code>apply.alias = OnOff</code> <code>apply.include = OnOff</code> <code>apply.modify = OnOff</code> <code>key.make.alias = OnOff</code> <h3>Counting</h3> <code>count.all = OnOff</code> <code>count.used = OnOff</code>

Key Generation

preserve.keys = OnOff
preserve.key.case = OnOff
key.format = {format}
 special values: short, long, short.need,
 long.need, empty
key.generation = OnOff
default.key = {key}
key.base = base
 values: upper, lower, digit
key.number.separator = {s}
key.expand.macros = OnOff
fmt.name.title = {s}
fmt.title.title = {s}
fmt.name.name = {s}
fmt.inter.name = {s}
fmt.name.pre = {s}
fmt.et.al = {s}
fmt.word.separator = {s}
new.format.type = {n="spec"}

Name Formatting Specification

Use n letters. Use m name parts. Insert pre before, mid between, and post after the words. Translate according to the s parameter ('+', '-', '*').

%sn.mf[mid][pre][post]
 format first names.
%sn.mv[mid][pre][post]
 format "von" part.
%sn.ml[mid][pre][post]
 format last name.

%sn.mj[mid][pre][post]
 format "junior" part.

Format Specifications

Pseudo fields:

\$key
\$default.key
\$sortkey
\$source
\$type
@type
\$day
\$month
\$mon
\$year
\$hour
\$minute
\$second
\$user
\$hostname

Formatting Fields:

%±x.y n(field)
 format y characters of x last names.
%±x.y N(field)
 format y characters of x names.
%±x.y p(field)
 format x names according to the name format y.
%±x.y d(field)
 format at most x digits of the yth number.
%±x.y D(field)
 format x digits of the yth number without truncation.
%±x s(field)
 format x string characters.

%±x.y t(field)
 format x sentence words of length y.
%±x.y T(field)
 format x sentence words of length y.
 (Words ignored)
%±x.y w(field)
 format x words of length y.
%±x W(field)
 format x words of length y. (Words ignored)
%±x.y #n(field)
 test whether the number of names is between x and y.
%±x.y #N(field)
 test whether the number of names is between x and y.
%±x.y #p(field)
 test whether the number of names is between x and y.
%±x.y #s(field)
 test whether the number of characters is between x and y.
%±x.y #t(field)
 test whether the number of words is between x and y.
%±x.y #T(field)
 test whether the number of not ignored words is between x and y.
%±x.y #w(field)
 test whether the number of words is between x and y.
%±x.y #W(field)
 test whether the number of not ignored words is between x and y.

Libraries

check_y Check the value of the year.
default All default settings.
field Redefine field names.
brace Use braces as delimiters.
improve Apply improvements.
iso2tex Translate ISO 8859/1 characters.
iso_def Define ISO 8859/1 characters for formatting.
keep_bibtex Keep only the fields of standard Bib_{TEX} styles.
keep_biblatex Keep only the fields of standard bib_{La}TEX styles.
month Introduce strings for month names.
opt Remove OPT in field names.
sort_fld Specify sort order for fields.
tex_def Define _{TEX} macros for formatting.
biblatex Define entry types and fields known to bib_{La}TEX.

Environment Variables

BIBTOOL Search path for rsc files
BIBINPUTS Search path for bib files
HOME Home directory for .bibtoolrsc