BibTool Quick Reference

Version 2.67 — Gerd Neugebauer — ©2017

http://www.gerd.neugebauer.de/software/TeX/BibTool/
gene@gerd-neugebauer.de

Command

```
rsc                  – _command
                       P
                       file.
yp       -A t
         e           Determine
         a, A,
         -d
                     Chec
rmat     -f k  _fo_
                     Generate        ey        k     _format
         -F
                     Enable
         -h        ld
                     Prin
         -i input_file
                     Mark
         -k
                     Mak
         -K
                     Mak
         -o output_file
                     Send          output_file.
         -q
                     Suppress
         -r resource_file
                     Read          resource_file.
         -R
                     Load
         -s
                     Sort
         -S
                     Sort
         -v
                     T
                     tions
         -x aux_file  ress.initial.newline
                     Extract       aux_file.
         -X regex.t
                     Extract       er,     regex.
```

General

```
                              pattern
resource.sea          {dir1:dir2. }
resource {file}
bibtex.sea            {dir1:dir2.
bibtex.env.name       {ENV_NAME}
env.sepa              {c}
dir.file.sepa         {c}

rint             p      {message}
                 quiet        OnOff
                 verb         OnOff
```

Reading

```
                              t          e
                              field
input {bib_file}
output.file    {file}
pa                    OnOff
pass.comments         OnOff
new.entry             {t    }
```

Manipulation

```
rint.align       p           n
rint.align.k p               n
rint.align.p p               n
rint.align.comment           n
rint.b           p      OnOff
rint.comma at.end       OnOff
rint.deleted entries    OnOff
rint.deleted p          refix    {p    }
rint.indent p                n
rint.line.length  all        n
rint.newline    #            n
rint.pa         p            OnOff
rint.terminal.comma          f      OnOff
rint.use.tab p               OnOff
rint.wide.equal              OnOff
                supp                 OnOff
new.field.t          {new=old}
symb     yp          t           =
                upper
```

Sorting

```
so      OnOff            #
so           OnOff
so           OnOff
rmat         {fo  }
so      .    {.  }
so           OnOff
```

Searching

```
tex.define {macro[a        ]}
extract.file {file}
select {field1.      n "regex"}
yplect {t     1.    n }
select.b         {field1.    n "regex"}
select.b         {cha }
select.case.sensitive    OnOff
select.fields    {field1,field2,.. }
                 r.crossref.map
```

Field

```
add.field {field="value"}
delete.field {field}
eep.field  k      {field}          b          i
eep.field  k      {field    2="pattern"}
rename.field {old=new}
rename.field {old=new              }
            alias
rewrite.rule { pattern }
            include
            mo
rewrite.rule { pattern }            }
            mak
rewpattrmle {f1.   n #}
```

rewriting

```
rewrite.case.sensitive    OnOff
rewrite.limit    {n}
```

Checking

```
check.double      OnOff
check.do.delete   OnOff
check.rule {field                    }
check.case.sensitive    OnOff
```

Strings

```
macro.file {file}
rint.all.strings            OnOff
expand.macros     OnOff
```

Inheritance

```
crossref.map      OnOff
clea              {}
crossref.limit    {n}
expand.crossref   OnOff
expand.xdata      OnOff
```

```
apply        OnOff
apply        OnOff
apply        OnOff
k            OnOff
```

Counting

```
count.all    OnOff
count.used   OnOff
```

Key

"junior"   w

reserve.k   p   w **OnOff**
reserve.k   p   ignored) **OnOff**
ey   rmkat   **{fo   }**
cial   v   **ey**   as
                    sp   **ey**
ey   k   8859/1   **OnOff tk**
          defau lt.k   **{k   }**
ey e   k   **base**   the **yp**
      alues: v   **yp**
ey   k   **y   {s}** the
ey   k   **OnOff**
r   **fmt.name.title   {s}** strings
          fmt.title.title   {s}   **ea**
          fmt.name.name   {s}
          fmt.inter.name   {s}
          fmt.name.p   {s}   try
          fmt.et.al   {s}
          fmt.w   **{s}**
          new.fo   **{n="sp   }**   **ormatting**
                          een

Name

      Use n letters. m name pr
efo the   b et mid according p after
slate   T   is
'*').
.   at **%sn.mf[snid][p][p   ]** of   b
              format
          %sremvd[snid][p][p   ]   b
              format
          %sn.mf[snid][p][p   ]   is
              format   c

Pseudo

$k
$default.k   ISO
$so
$source   ords
$t
@t
$da   whether
$month   een
$mon
$y   whether
$hour   een   sort
$minute
$second   whether
$user   een
$hostname
                  whether
%±x.y n(field) w
names. character ysc x last
%±x.y N(field)   whether
      format ysc   een x names.
%±x.y p(field)   whether
      format x names
      mat y.   ords
%±x.y d(field)   whether
      format x digitsum y^{th} n
                      een
%±x.y D(field)
      format x digitsum y^{th} n
      truncation.   whether
%±x s(field)   ords
      format x string

%sn.mf[snid][p][p   ]
      format   tence
                  the
                         tence   field
                  ords

%±x.y t(field)
      format x sen y.
%±x.y T(field)   k
      format x sen
      (W
%±x.y w(field)
      ords format x impro y.
%±x W(field)
      ords format ISO   (W y.
      nored)
%±x.y #n(field)   only
      test   yles.
           t x and y.   only
%±x.y #N(field)   yles.
      duce
           t x and y.
%±x.y #p(field)
      test   T
           t x and y.
%±x.y #s(field)
      test
           b x and y.
%±x.y #t(field)
      test
           t x and y.
%±x.y #T(field)
      test
      w x and y.
%±x.y #w(field)
      test
           t x and y.
%±x.y #W(field)
      test
      w x and y.

Libraries

                      `check_y`   Chec
          default   `default`   All
                      `field`   Redefine
          braces   `brace`   Use
                      `improve`   Apply
                      `iso2test`   T
                      `iso_def`   Define
                                       formatting.
                      `keep_bibtex`   Keep
                  st   B   Bɪ Tɛ X
                      `keep_biblatex` Keep
                  st   bibLATEX
                      `month`   In
                      `opt`   Remo
          ecify   `sort_fld`   Sp
                      `tex_def` macro Define ɛX
                      `biblatex`   Define
                      bibL   to   ᴀTEX.