

Command	General	Sorting	
- _command P file.	resource.sea {dir ₁ :dir ₂ . }	so OnOff	check.double OnOff
-A t Determine t in 0, a, A,	resource {file} bibtex.sea {dir ₁ :dir ₂ . } bibtex.env.name {ENV_NAME} env.sepa {c} dir.file.sepa {c}	so OnOff so OnOff so {fo } so {. so OnOff	check.do.delete OnOff check.rule {field } check.case.sensitive OnOff
-d Chec	p {message}		Strings
-f k _fo Generate k _format	quiet OnOff verb OnOff	Searc	macro.file {file} p OnOff expand.macros OnOff
-F Enable		tex.define {macro[a]} extract.file {file} select {field ₁ . _n "regex"} select {t ₁ . _n } select.b {field ₁ . _n "regex"} select.b {cha } select.case.sensitive OnOff select.fields {field ₁ ,field ₂ ,. }	Inheritance
-h Prin	Reading		
-i input_file Mark	input {bib_file} output.file {file}	Field	crossref.map OnOff clea {}
-k Mak	pa OnOff	add.field {field="value"} delete.field {field}	crossref.limit {n}
-K Mak	pass.comments OnOff	k {field } k {field ₂ ="pattern"} rename.field {old=new} rename.field {old=new } rewrite.rule { pattern } delete rewrite.rule { pattern }	expand.crossref OnOff expand.xdata OnOff
-o output_file Send	new.entry {t }		B E _X 1.0
-q Suppress	p n p n p n p OnOff p OnOff p OnOff p {p }		apply OnOff apply OnOff apply OnOff k OnOff
-r resource_file Read	p n p n p OnOff p OnOff p {p }		Coun
-R Load	p n p n p OnOff p OnOff p OnOff p OnOff p OnOff p OnOff p OnOff		
-s Sort	p n p n p OnOff p OnOff p OnOff p OnOff p OnOff p OnOff		
-S Sort	p n p n p OnOff p OnOff p OnOff p OnOff p OnOff p OnOff		
-v T tions	p n p n p OnOff p OnOff p OnOff p OnOff p OnOff p OnOff		
-x aux_file Extract	supp OnOff new.field.t {new=old}	rewrite rewrite.case.sensitive OnOff rewrite.limit {n}	
-X regex Extract	symb t upp	Chec	count.all OnOff count.used OnOff

Key	%sn.mj[mid][p][p] format	%±x.y t(field) format x sen	Libraries
OnOff	F	%±x.y T(field) format x sen	check_y Chec
OnOff	Pseudo	%±x.y W(field) format x sen (W	default All
{fo }	\$k	%±x.y w(field) format x w	field Redefine
sp	\$default.k	%±x W(field) format x w	brace Use
long.need,	\$so	%±x y W(field) format x w	improve Apply
OnOff	\$source	%±x.y #n(field) test	iso2tex T
default.k {k }	\$t	%±x.y #N(field) test	iso.def Define
k base	@t	%±x.y #p(field) test	keep_bibtex Keep
v	\$da	%±x.y #s(field) test	keep_biblatex Keep
{s}	\$month	%±x.y #t(field) test	month In
OnOff	\$mon	%±x.y #w(field) test	opt Remo
fmt.name.title {s}	\$y	%±x.y #W(field) test	sort fld Sp
fmt.title.title {s}	\$hour	%±x.y #s(field) test	tex.def Define
fmt.name.name {s}	\$minute	%±x.y #t(field) test	biblatex to
fmt.inter.name {s}	\$second	%±x.y #w(field) test	TeX
fmt.name.p {s}	\$user	%±x.y #T(field) test	TeX
fmt.et.al {s}	\$hostname	%±x.y #s(field) test	TeX
fmt.w {s}	F	%±x.y #t(field) test	TeX
new.fo {n="sp }	%±x.y n(field) format y c	%±x.y #w(field) test	TeX
	%±x.y N(field) format y c	%±x.y #T(field) test	TeX
	%±x.y p(field) format x names	%±x.y #W(field) test	TeX
	%±x.y d(field) format	%±x.y #s(field) test	TeX
	%±x.y D(field) format x digits	%±x.y #t(field) test	TeX
	%±x s(field) format x string	%±x.y #w(field) test	TeX