Syntax: `task` `<override>` `<filter>` `<command>` `<modifications>`

TASKWARRIOR 2.3.0
Command Reference

Docs
http://taskwarrior.org

Q&A
http://answers.tasktools.org

Support
support@taskwarrior.org
#taskwarrior on freenode

Help
```
task help
man task
man taskrc
man task-color
man task-tutorial
man task-faq
man task-sync
```

Basic Usage
```
task add Pay the bills
task add Call the bank
task list
task 1 modify due:Friday
task 1 done
task 2 delete
```

Escape shell characters:
```
task add John\'s birthday
```

Quote shell characters:
```
task add "John's birthday"
```

Escape These:
```
$ ! ' " ( ) ; \ ` * ?
{ } [ ] < > | & % # ~
```

Debugging
```
task rc.debug=1
task diagnostics
task rc.debug.tls=<level> …
```

Copyright (C) 2006 - 2014
 P Beckingham
 F Hernandez
 D Deimeke

Revised 2014-05-11

Runtime Override
```
rc.<variable>=<value>
```

Config File Override
```
rc:<file>
```

Parser Override
```
task <parsed> -- <unparsed>
```

Configuration
```
task show [<var>]
task config <var> <value>
task config <var>
task config <var> ''
```

UDA Configuration
```
uda.NAME.type=TYPE
uda.NAME.label=LABEL
uda.NAME.value=VAL,VAL,…

TYPE:
string, date, duration,
numeric
```

Sync Configuration
```
taskd.server
taskd.ca or taskd.trust
taskd.certificate
taskd.key
taskd.credentials
taskd.ciphers
```

Date Formats
With Time
```
rc.dateformat=YMD_HNS
```
Just Date
```
rc.dateformat=YMD
```

Annotations
```
Add:    task <id> annotate foo
Remove: task <id> denotate foo
```

Annotation en-passant
```
task <id> done    Finished it
task <id> delete  Not needed
```

Synchronize
```
task sync init   (first time)
task sync
```

Import/Export
```
task [<filter>] export
task import <file>
```
Used for writing extensions, wrappers.

Filter
Expression of terms with values, operators.
```
<term> [<op> <term> …]
```

Terms
```
<attr>:<value>
<attr>.<modifier>:<value>
<DOM>
<ID range>
<UUID list>
```

ID Ranges - use ID or UUID
```
task 1              info
task 1-10           info
task 1,3,5,6-10,13  info
task 1 2 3          info
```

Searching for text
```
task desc.contains:foo
task foo
task /foo/
```

Operators
```
and       conjunction
or        disjunction
xor       exclusive or
+         add
-         subtract
*         multiply
/         divide
%         modulue
^         exponent
~         regex match
!~        regex non-match
<         less than
<=        less than or equal
==        equal
!=        inequal
>=        greater or equal
>         greater than
_hastag_  has tag
_notag_   hasn't tag
( )       precedence
```

Attribute Modifiers
```
task urgency.under:4    list
task proj.endswith:work list

before (under, below)
after (over, above)
none
any
is (equals)
isn't (not)
has (contains)
hasnt
startswith (left)
endswith (right)
word
noword
```

Report Commands
(Run 'task reports')
```
active
all
blocked
blocking
burndown.daily
burndown.monthly
burndown.weekly
completed
ghistory.annual
ghistory.monthly
history.annual
history.monthly
information
list
long
ls
minimal
newest
next
oldest
overdue
projects
ready
recurring
summary
tags
timesheet
unblocked
waiting
```

Write Commands
```
add
annotate, denotate
append
config
delete
done
duplicate
edit
import
log
modify
prepent
start, stop
synchronize
undo
```

Read Commands
```
<report>
info
count
export
```

Misc Read Commands
```
calendar
colors
columns
count
diagnostics
execute
export
help
ids, uuids
logo
report
show
stats
udas
version
```

Modifications

Set
```
<attribute>:<value>
```

Remove
```
<attribute>:
```

Substitute
```
/from/to/g
/from/to/
```

Add Dependency
```
depends:<id>[,…]
```

Remove Dependency
```
depends:-<id>[,…]
```

Tag
```
+foo
```

Untag
```
-foo
```

Date Symbols
```
yesterday
today
now
tomorrow
someday
easter
tuesday
23rd
```

Period Symbols
```
hourly     1hr
daily      5days
weekly     2wks
monthly    3months
quarterly  1q
annual     2years
```

Document Object Model (DOM)
```
<id|uuid>.<attribute>
rc.<name>
system.<version|os>
```

Examples
```
task _get 123.description
task add … due:123.due
task _get rc.dateformat
task _get 123.uuid
```

Abbreviations
```
task project:Home list
task projec:Home   lis
task pro:Home      li
```

Task Attributes
```
ID          unmodifiable
UUID        unmodifiable
urgency     unmodifiable

description
project
priority
tags
status

depends
recur

entry       created
wait        hide until
scheduled   ready
start       started
due         due
end         done/delete
until
modified    last mod
```
```
UDAs (Try: task uda)
```

Special Tags
```
+next
+nonag
+nocal
+nocolor
```

Virtual Tags
```
+DUE
+DUETODAY
+TODAY
+OVERDUE
+WEEK
+MONTH
+YEAR
+BLOCKED
+UNBLOCKED
+BLOCKING
+ACTIVE
+SCHEDULED
+UNTIL
+WAITING
+ANNOTATED
+PARENT
+CHILD
```

Examples
```
Stop parsing: task add — pro:a /from/to/ +foo
Rename proj:  task project:OLD modify project:NEW
Correction:   task <id> modify /incorrect/correct/g
Hide until:   task <id> modify wait:<date>
Expire task:  task <id> modify until:<date>
Just data:    task rc.verbose=nothing …

Filter:       task \( pro:Home or pro:Garden \) and +weekend
Filter:       task \( pro:Home or pro:Garden \) +weekend
```