

Author: Alan Richardson

Date: 22nd September 2014

- www.compendiumdev.co.uk
- www.eviltester.com
- www.seleniumsimplified.com
- www.javafortesters.com

My notes on exploratory testing, investigation, thoughts, ideas and logs for the Black Ops Testing Webinar focused on testing the terminal app TaskWarrior

- <http://taskwarrior.org>
- <http://blackopstesting.com>

Install And setup environments:

Investigation to make sure I can observe and control system

11:15 monitoring tooling

11:35 Read documentation and see what we can trigger for our testing

11:55 experiment with minimal data

Some of the values are calculated id, urgency, - would not expect to be able to modify these

Try and modify the entry value

Check that future entries have lower urgency than current tasks

Recurring task seems interesting, how does that work?

Are there better ways of tracking the testing?

Install And setup environments:

- Initially had it installed on vmware fusion on mac under ubuntu
- reinstall into vmware under windows
 - <http://taskwarrior.org/download/>
 - <http://www.ubuntu.com/download/desktop/thank-you?country=GB&version=14.04.1&architecture=amd64>
 - setup vm
 - install task warrior
 - ctrl + alt T
 - `sudo apt-get install task`
 - installed

```
alan@ubuntu:~$ task --version
2.2.0
```

The 'early' testing that I did on the mac VM is still there, as is the history of that session. (now copied that to dropbox for later analysis as I remember that I had an issue where I had deleted tasks but they still showed - I hadn't learned anything about task at the time so I may have been wrong). Copied details to an archive in the testing data see *earlyMacTesting* folder.

Investigation to make sure I can observe and control system

Would like to be able to:

- backup and restore taskwarrior files to keep and restore state
 - backup <http://taskwarrior.org/docs/taskserver/backup.html>
 - or **"zip -r ~/Dropbox/shared/taskData.zip ~/.task"**
 - possibly add `$(date +%Y%m%d)` into the above
 - **"zip -r ~/Dropbox/shared/task\$(date +%Y%m%d)Data.zip ~/.task"**
- clean taskwarrior down to base state
 - `rm ~/.task/*`
- monitor taskwarrior files
 - multitail `"multitail -Q 1 ~/.task/*"`
 - files open in a text editor - refresh when switch between tabs
- extract history as a 'test log'
 - add a `' '` before each command so it does not enter history
 - use `'echo '` as a 'comment entry in the history log
 - `"history -d #number"` to delete an entry in the history file
 - add time stamps to the history log `"export HISTTIMEFORMAT='%F %T ' "`
 - *since this is an export would need to do for each new terminal*
 - `"history > ~/Dropbox/shared/.../logFileName.txt"`
 - clear history `"history -c"`

22/09/2014 env setup 10:00

Install dropbox on to VMs (Mac & Windows)

- <https://www.dropbox.com/install?os=lnx>
- Sync the 'shared' folder
- `~/Dropbox/shared`

10:20

copy old unthinking install testing from Mac to shared as a log for analysis

```
export HISTTIMEFORMAT='%F %T ' "
```

```
history > ~/Dropbox/shared/earlyMacTesting/historyLogTimes.txt
```

REMEMBER: need to have “~/dropbox-dist/dropboxd” dropbox daemon running for sync to work -

TODO: investigate how to make that automatic for all distros - currently just running from comand line

Try and Delete files from a new install and start task

- partly to see how task works
- but also to see if I can clear down task data, and can use history as a test log tool
- task handles this
- see logfile task20140922_1100_startTaskNoFiles.txt
- history > ~/Dropbox/shared/task20140922_1100_startTaskNoFiles.txt

11:00

clear the history and start again

export HISTTIMEFORMAT='%F %T ' ' '

OK - my linux clock is not telling the correct time (it is set to 3:03 AM, current time is 11:03)

<http://www.cyberciti.biz/faq/howto-set-date-time-from-linux-command-prompt/>

sudo date +%T -s "11:06:00"

Set data so that my history logs reflect the actual time and testing

11:10 environments ready

11:15 monitoring tooling

multitail? - 11:25 seems useful for small files

<http://en.wikipedia.org/wiki/MultiTail>

<http://www.vanheusden.com/multitail/examples.php>

Log for experimenting with multitail

summary use “multitail -Q 1 ~/.task/*”

history > ~/Dropbox/shared/task20140922_1125_installAndStartMultiTail.txt

To see if multitail was useful, I created a task and deleted it to observe the change in the pending.data file

to see the history >

~/Dropbox/shared/task20140922_1125_multiTailCreateDeleteDupeNameTask.txt

Notes: need to do an HISTTIMEFORMAT export prior to each history - Todo: figure out how to fix this. Next time try this workaround

export HISTTIMEFORMAT='%F %T '; history >

~/Dropbox/shared/task20140922_1125_installAndStartMultiTail.txt

Took screenshot of current environment choosing open terminals - one for adhoc work, one for task, one for multitail, and one running dropbox daemon

2014-09-22_11_35_19_current_working_environment.png

Would have liked to find better logging, but thought I better start testing, otherwise I risk spending all the time on tools and environment investigation and no 'testing'.

11:35 Read documentation and see what we can trigger for our testing

Documents: <http://taskwarrior.org/docs/>

<http://taskwarrior.org/docs/tutorials/30second.html>

- task add <taskName>
- task add priority:<priority> <taskName>
- task next
 - shows tasks in priority order
 - how to amend priority of existing task?
 - *when you type 'task' it automatically does a 'task next' so this is a default action*
- task <id> done
- task <id> delete
- 'all you need to know' add, done, delete, next

export HISTTIMEFORMAT='%F %T '; history >

~/Dropbox/shared/task20140922_1145_30secondTutorial.txt

At this point I have more files in the .task

- pending.data
- history
- undo.data
- completed.data

“zip -r ~/Dropbox/shared/data/task\$(date + '%Y%m%d')Data.zip ~/.task”

taskAfter30SecondTutorialData.zip

11:55 experiment with minimal data

What if I do the minimal commands incorrectly?

- if I give a wrong command?
 - e.g. app, defeat
 - Answer: command might exist as a truncated command, otherwise ignored
- if I miss out an id?
 - is is a filter so the command then works on all tasks
- if I repeat a command e.g. task 1 done done?
 - 2nd done is treated as an annotation
- if task id does not exist?
 - error message ‘No tasks specified’
- if I use a priority that does not exist e.g. priority:Z?
 - Priority values may be 'H', 'M' or 'L', not 'Z'.
- if I use an attribute that does not exist e.g. priority?
 - it does not detect it as an attribute so the full text becomes
 - priority:H this is priority z
 - i.e. the attribute is added to the text

Q: how does it know what ID it is, since that doesn't appear in the entry anywhere? possible source of error injection?

Q: undo information tracked in file - possible rich source of interaction in the undo file - does it always keep pace with the state amendments? can you undo enough?

12:05

Note: need to find a different tool than multital, since this amends the files, rather than just writing a log of them, it doesn't help as much as a static view of each file

- opened all files in gedit and ...
- 'h' shows help for multital
- 'O' clears all windows so can see file amendments and then view in gedit

Now... experiment with minimal data

“task app bob” asked about a filter - check if app is a command with ‘task help’

is app treated as a filter?

re-issue command

this will modify all commands - seems to want to append bob on the task in there already

'no' to all commands

'task not appended'

- suspect 'app' is an append command I didn't see listed - will try with different filter

As it gave me a list of commands it gave me three commands to decide if I wanted to amend

```
alan@ubuntu:~$ task app bob
```

This command has no filter, and will modify all tasks. Are you sure?

(yes/no) yes

- Description will be changed from 'read the taskwarrior documentation' to 'read the taskwarrior documentation bob'.

Append to task 1 'read the taskwarrior documentation'?

(yes/no/all/quit) no

Task not appended.

- Description will be changed from 'experiment with 30 second tutorial' to 'experiment with 30 second tutorial bob'.

Append to task 0 'experiment with 30 second tutorial'?

(yes/no/all/quit) no

Task not appended.

- Description will be changed from 'read the taskwarrior documentation' to 'read the taskwarrior documentation bob'.

Append to task 0 'read the taskwarrior documentation'?

(yes/no/all/quit) no

Task not appended.

Appended 0 tasks.

But I only have one valid task

```
alan@ubuntu:~$ task
```

[task next]

ID	Age	Urg	Description
----	-----	-----	-------------

1	1h	0	read the taskwarrior documentation
---	----	---	------------------------------------

1 task

I did not expect to be able to amend a deleted or 'done' task.

See if 'app' was append, 'task zzz bob'

'append' is a command 'app' is not

Idea: investigate command name truncation as possible future test

missing description?

'task add' not added

can truncate done to don

task undo

and do == done

d does not == done, command might need to be 2 chars

cannot modify without any text - tried ", ' ' and spaces

```
task modify all same name
```

would amend all tasks, so <id> is a filter

Tried to repeat a command 'done done'

This didn't do anything untoward, but I noticed that the extra 'done' is treated as an annotation

12:50 - end of minor command and attribute test session - have learned a little more about the app: archived the session to:

```
export HISTTIMEFORMAT='%F %T  '; history >
```

```
~/Dropbox/shared/task20140922_1250_minimaldata.txt
```

```
zip -r ~/Dropbox/shared/data/task20140922_1250_minimaldata.zip ~/.task
```

<break>

Debrief:

- Documentation isn't always the easiest to find what you need. But seems like an app where you gradually learn more functionality as you need it.

- The basic test setup I have where I use the history as a log seems to work well, and zipping out the task files to leave a data trail is useful.
- Multitail works well in combination of the 'O' to refresh prior to a command, so I can see the difference, and having the files open in an editor means I can view the information in context.
- Need to find an area to zoom in on

14:23

Mechanism for id is explained here <http://taskwarrior.org/docs/ids.html>

basically the order in the pending.data file

Currently pending has

```
[description:"read the taskwarrior documentation" entry:"1411410185" status:"pending"
uuid:"0892003f-5f13-4095-8c88-eabbe126acc5"]
[description:"priority:H this is priority z" entry:"1411415318" status:"pending"
uuid:"0348665b-aec8-4a9b-bb60-63de734b594f"]
```

Modify description of 2 and change priorities so 'read ... documentation' is lower priority

```
alan@ubuntu:~$ task 2 modify This is now highest priority
- Description will be changed from 'priority:H this is priority z' to
'This is now highest priority'.
Modify task 2 'This is now highest priority'? (yes/no) yes
Modifying task 2 'This is now highest priority'.
Modified 1 task.
alan@ubuntu:~$ task
[task next]
```

```
ID Age Urg Description
1 3h 0 read the taskwarrior documentation
2 1h 0 This is now highest priority
```

2 tasks

```
alan@ubuntu:~$ task 1 modify priority:1
Modifying task 1 'read the taskwarrior documentation'.
Modified 1 task.
alan@ubuntu:~$ task
[task next]
```

```
ID Pri Age Urg Description
```



```
1 L    3h    1.8 read the taskwarrior documentation
2      1h      0 This is now highest priority
```

2 tasks

```
alan@ubuntu:~$ task 2 modify priority:h
```

```
Modifying task 2 'This is now highest priority'.
```

```
Modified 1 task.
```

```
alan@ubuntu:~$ task
```

```
[task next]
```

```
ID Pri Age Urg  Description
 2 H   1h    6 This is now highest priority
 1 L   3h   1.8 read the taskwarrior documentation
```

2 tasks

I half expected the order of the items to change, but they don't the id does not reflect priority - just order in the file. So id only changes when items move to completed.data

DONE: script, or terminal that supports this (initially found terminator and stopped looking) *to do: for future work, find out how to have the terminal automatically write out to a file all sysin and sysout to capture full log of testing.*

reading through docs

'task all' shows all tasks (but not deleted tasks)

task completed - shows completed

can't see a way of showing deleted items

How to create due dates? A: using attributes

Built-in attributes are:

```
description:    Task description text
status:         Status of task - pending, completed, deleted,
waiting
project:        Project name
priority:       Priority
due:            Due date
recur:          Recurrence frequency
```

until:	Expiration date of a task
limit:	Desired number of rows in report, or 'page'
wait:	Date until task becomes pending
entry:	Date task was created
end:	Date task was completed/deleted
start:	Date task was started
scheduled:	Date task is scheduled to start
depends:	Other tasks that this task depends upon

Also noticed can use the uuid as a filter - can I see deleted this way?

A: yes.

task 0f386335-4046-4d61-94eb-43160707c852 info

... shows deleted task info

entry:"..." works as filter

description does not

status:... works

info can show multiple items

also shows the 'undo' information associated with the 'task'

info shows a lot of information about the task - can I modify at that level?

priority

hmmm, amended task 2 to status:deleted

it marked it as deleted

I was able to do a 'task 2 info' on the task even though it was deleted

when I checked if the task was still there it was not in the task list again.

When does it move from one list to the other?

when I amend the status using the modify command, the task is not moved to completed.data

I can still "task 2 info" on the task - although it is in the undo data

So I have a pending move to completed.data waiting

Can I Queue up the completed move as well? Yes, after modifying 3 to status:completed I now have 2 in a Q to move

I suspect it is the task list that 'tidies up' - yes the 'task all' tidied up the hanging amends.

this could be classed as a minor 'bug', but I don't imagine it would really be fixed as it is easier to 'delete' a task and 'done' a task than modify the status.

What else can I modify?

try to modify uuid to a different uuid and it failed

what about blank?

```
alan@ubuntu:~$ task 1 modify uuid:
Modifying task 1 'read the taskwarrior documentation'.
Modified 1 task.
```

Hmmm, that's odd. A new line has been added to the undo buffer

```
time 1411423172
old []
new [description:"read the taskwarrior documentation"
entry:"1411410185" modified:"1411423172" priority:"L"
status:"pending" uuid:"eb9f30a8-4d51-4eb4-8fb1-51c48cdf203e"]
```

Old is blank - which I suspect should not happen, and it has pulled in the current pending task and says it has a new uuid , but the pending data is still the same

```
[description:"read the taskwarrior documentation" entry:"1411410185"
modified:"1411421199" priority:"L" status:"pending"
uuid:"0892003f-5f13-4095-8c88-eabbe126acc5"]
```

no seeming impact on the task 1 info, or task all report - the new uuid is not shown from the undo in the info display

what happens if I 'undo' this non change?

```
alan@ubuntu:~$ task undo
```

The last modification was made 9/22/2014
Taskwarrior no longer supports file format 1, originally used between
27 November 2006 and 31 December 2007.

Bug: taskwarrior has created a blank undo record that it can't undo when modify uuid to empty

```
export HISTTIMEFORMAT='%F %T '; history >  
~/Dropbox/shared/task20140922_1507_bugInvalidUndo.txt
```

```
zip -r ~/Dropbox/shared/data/task20140922_1507_bugInvalidUndo.zip ~/.task
```

Since the 'undo.data' is in error, I'm going to remove the last line from the undo list to 'fix' the app and carry on testing

removed

```
---  
time 1411423172  
old []  
new [description:"read the taskwarrior documentation"  
entry:"1411410185" modified:"1411423172" priority:"L"  
status:"pending" uuid:"eb9f30a8-4d51-4eb4-8fb1-51c48cdf203e"]
```

15:10 Carry on with 'modify' testing

```
alan@ubuntu:~$ task 1 modify status:boogaloo  
Modifying task 1 'read the taskwarrior documentation'.  
Modified 1 task.
```

What?

```
time 1411423856  
old [description:"read the taskwarrior documentation"  
entry:"1411410185" modified:"1411421199" priority:"L"  
status:"pending" uuid:"0892003f-5f13-4095-8c88-eabbe126acc5"]  
new [description:"read the taskwarrior documentation"  
entry:"1411410185" modified:"1411423856" priority:"L"  
status:"pending" uuid:"0892003f-5f13-4095-8c88-eabbe126acc5"]
```

Bug: Entry reported as modified, when no change made, after 'modify status:' to invalid status. Invalid status correctly ignored, but record marked as modified - only the modified date was modified.

Some of the values are calculated id, urgency, - would not expect to be able to modify these

```
alan@ubuntu:~$ task 1 modify id:56
Modifying task 1 'read the taskwarrior documentation'.
Modified 1 task.
```

it added an 'id' field to the record. task info doesn't show any difference, so I can add arbitrary attributes using the modify

but I can't modify the ID

```
alan@ubuntu:~$ task 1 modify ID:46
- Description will be changed from 'read the taskwarrior
documentation' to 'ID:46'.
Modify task 1 'ID:46'? (yes/no)
```

```
[description:"read the taskwarrior documentation" entry:"1411410185"
id:"56" modified:"1411424094" priority:"L" status:"pending"
uuid:"0892003f-5f13-4095-8c88-eabbel26acc5"]
```

The task 1 info, for the above does not show the id in it, was 'id' just a luck hit in modify?

Try other 'new attributes'

```
failed:
newAttrib:new
bob:bob
bob:3
```

all tried to amend description - only id slipped through (**bug?**)

try urgency - urgency accepted

```
alan@ubuntu:~$ task 1 modify urgency:67
Modifying task 1 'read the taskwarrior documentation'.
Modified 1 task.
```

Again - no impact on the task 1 info, but changes are shown in the modification list

```
alan@ubuntu:~$ task 1 info
```

Name	Value
ID	1
Description	read the taskwarrior documentation
Status	Pending
Priority	L
UUID	0892003f-5f13-4095-8c88-eabbel26acc5
Entered	9/22/2014 11:23:05 (3 hrs)
Urgency	1.8
Last modified	9/22/2014 15:20:56 (52 secs)

Date	Modification
9/22/2014 14:26:39	Modified set to '9/22/2014 14:26:39'. Priority set to 'L'.
9/22/2014 15:10:56	Modified changed from '9/22/2014 14:26:39' to '9/22/2014 15:10:56'.
9/22/2014 15:14:54	Id set to '56'. Modified changed from '9/22/2014 15:10:56' to '9/22/2014 15:14:54'.
9/22/2014 15:20:56	Urgency set to '67'. Modified changed from '9/22/2014 15:14:54' to '9/22/2014 15:20:56'.

I can task undo this and the field is removed.

I don't know if this is a bug or not, as I'm not sure if I can create extra attributes. I suspect it is a bug, because I've only been able to create extra attributes for fields which are 'calculated'

Bug: can 'modify' calculated fields id and urgency, and this adds a new attribute to the record, which can be removed with undo, but is not reflected in the task info

Try and modify the entry value

```
alan@ubuntu:~$ task 1 modify entry:1
The duration '1' was not recognized as valid, with correct units like
'3days'.
```

If I modify entry to current value

```
alan@ubuntu:~$ task 1 modify entry:1411410185
Modified 0 tasks.
```

to slightly before

```
alan@ubuntu:~$ task 1 modify entry:"1411410184"
```

Modifying task 1 'read the taskwarrior documentation'.

Modified 1 task.

Bug: Can modify entry date so that the task will be entered in the future

```
task 1 modify entry:"0010000001"
```

```
alan@ubuntu:~$ task
```

```
[task next]
```

ID	Pri	Age	Urg	Description
1	L	-3mo	1.17	read the taskwarrior documentation

Can use "year" value to modify entry date 68 years into the future

```
alan@ubuntu:~$ task 1 modify entry:"68year"
```

Modifying task 1 'read the taskwarrior documentation'.

Modified 1 task.

```
alan@ubuntu:~$ task
```

```
[task next]
```

ID	Pri	Age	Urg	Description
1	L	-23.2y	-44.7	read the taskwarrior documentation

1 task

Note: I guess this could be argued that this is a feature, and with more tasks, because of the urgency, I assume it will move lower down - I could check that.

...

Check that future entries have lower urgency than current tasks

Add a task with a due date and it shows before the -ve one, but add a normal task and it has a lower urgency than the one entered into the future, so assume this is a bug.

```
task add this is due today due:2mins
```

```
task add this has no due date
```

```
alan@ubuntu:~$ task
[task next]
```

ID	Pri	Due	Age	Urg	Description
2		9/22/2014	21s	8.64	this is due today
1	L		-3mo	1.17	read the taskwarrior documentation
3			1s	0	this has no due date

```
3 tasks
```

Can I modify the modified date?

System says I can, and the undo record has this, but the task info shows correct modified date

The modified date is always set to the current modified date, even though a 'modify' to the modify date is the request. Really the system should just ignore this modify, same as id, etc.

Minor Bug. Unless a way to exploit this is found.

```
export HISTTIMEFORMAT='%F %T  '; history >
~/Dropbox/shared/task20140922_1551_modifyTesting.txt
```

```
zip -r ~/Dropbox/shared/data/task20140922_1551_modifyTesting.zip ~/.task
```

Recurring task seems interesting, how does that work?

Does it have to be a big gap? docs suggest weekly
<http://taskwarrior.org/docs/commands/columns.html>

Create one for 1 second - what happens?

```
task add do me again recur:1secs due:1secs
```

everytime you do a 'task' or 'task all' i.e. the list functionality (which seems to be what triggers updates, it creates new items. So not really a useful function for humans, but the system creates a new pendind.data entry for every second that passes between the call to 'task'

I don't think this is a 'bug' since it uses the defined functionality, but it certainly makes an easy way to create lots of tasks as test data. Shame I can't set it up to be milliseconds.

Have a look at the recurring task docs:

<http://taskwarrior.org/docs/recurrence.html>

The 'mask' field on the parent task is a string with an "-" char for each entry it has created - I guess in theory this could blow up as too long if there were a lot of tasks.

How do I stop recurring? interact with the 'recurring' item?

task 4?

```
alan@ubuntu:~$ task 4 done
Task 4 'do me again' is neither pending nor waiting.
Completed 0 tasks.
```

```
alan@ubuntu:~$ task 4 delete
Task 4 'do me again' is not deletable.
Deleted 0 tasks.
```

Hmm, up to 968 tasks pending - I've created a Sorcerer's Apprentice moment

```
alan@ubuntu:~$ task 4 modify status:deleted
Modifying task 4 'do me again'.
Modified 1 task.
```

And now they have stopped recurring.

use built in functionality to complete all tasks

```
task 5-967 done
```

Save this session off - there are few bugs in there to discuss, and could spend some time on - better use of terminal for catching full session log

```
export HISTTIMEFORMAT='%F %T  '; history >
~/Dropbox/shared/task20140922_1625_recurringTaskTesting.txt
```

```
zip -r ~/Dropbox/shared/data/task20140922_1625_recurringTaskTesting.zip ~/.task
```

Are there better ways of tracking the testing?

Showing the input and output?

<http://unix.stackexchange.com/questions/111572/pipe-all-input-and-output-of-terminal-to-file>

script command writes everything to a file, but it is all esc chars, because it is the terminal output.

But - handy thing is if I do

```
script ~/Dropbox/shared/20140922_1640_sessionlog.txt
```

Then the script is output automatically to drop box and synced as the session progresses

Really want a simpler - text view

Using the terminator application, `"sudo apt-get install terminator"`

There is a 'logger' plugin.

Right click, preferences, plugins, activate logger

Back on terminal, right click, start logger - choose a file location e.g. the dropbox sync location with a filename suggestive of the session

```
20140922_1650_terminator_logger_experiment.txt
```

And the output is plain text to the text file, making it a very good sysin, and sysout logging tool.

DONE 16:55

Tidy up this log file - headings, a few minor comments, TOC - done 17:10

---- end ---