Harness - Feature #1554

enhance the command-line to provide better user feedback on the state of testing

09/18/2012 06:15 PM - Greg Shah

Status:	WIP	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Tomasz Domin	**% Done:**	90%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**version:**	
vendor_id:	GCD		

Description

History

#1 - 09/18/2012 06:15 PM - Greg Shah

Enhance the Harness command line driver to:

1. Report % completion on the fly.
2. Summarize the pass/fail/run statistics.

#2 - 11/28/2012 11:47 AM - Greg Shah

Make sure the driver also returns an overall success/failure return code so that scripts can be used to automate running the harness.

Likewise, the console output should be done in a way that can easily be compatible with the above mentioned scripts.

#3 - 04/10/2013 08:54 AM - Constantin Asofiei

> Make sure the driver also returns an overall success/failure return code so that scripts can be used to automate running the harness.

The ca_upd20130410a.zip update from #1552 fixes this.

#4 - 07/21/2022 05:38 AM - Greg Shah

- Assignee set to Tomasz Domin

- Start date deleted (09/18/2012)

#5 - 08/04/2022 12:39 PM - Tomasz Domin

- Status changed from New to WIP

#6 - 08/04/2022 12:48 PM - Tomasz Domin

- % Done changed from 0 to 50

I've implemented two features:
- dumping reports on CTRL-C - so partial results are available for review in case test is interrupted
- logging - to give better overview what is going on with test scenarios

For now both options are always enabled
Should I add these features as command line enabled options that are disabled by default ?

I've also upgraded code to compile with no warnings and to run with Java 11

Logging gives following sample output:

```
Running test plan...
[2022-08-04 18:42:46] [FINE] TestPlan - Start: APP_tc_po_item_003
[2022-08-04 18:42:46] [FINE] TestSet  - Start: tc_tests
[2022-08-04 18:42:46] [FINE] TestSet  - [Driver 0] Start
[2022-08-04 18:42:46] [FINE] TestSet  - [Driver 1] Start
[2022-08-04 18:42:46] [FINE] Test     - [Driver 0][login_to_main_menu] Start
[2022-08-04 18:42:46] [FINE] Test     - [Driver 0][login_to_main_menu] Step 1: RUN-CODE:Login to APP and get t
o a main menu.
[2022-08-04 18:43:41] [FINE] Test     - [Driver 0][login_to_main_menu] Step 2: WAIT-FOR-SCREEN-BUFFER:Login to
 APP and get to a main menu.
[2022-08-04 18:43:41] [FINE] Test     - [Driver 0][login_to_main_menu] Step 3: SEND-TEXT:Login to APP and get 
to a main menu.
[2022-08-04 18:43:44] [FINE] Test     - [Driver 0][login_to_main_menu] Step 4: CHECK-SCREEN-BUFFER:Login to AP
P and get to a main menu.
[2022-08-04 18:43:45] [FINE] Test     - [Driver 0][login_to_main_menu] Step 5: SEND-TEXT:Login to APP and get 
to a main menu.
[2022-08-04 18:43:45] [FINE] Test     - [Driver 0][login_to_main_menu] Step 6: SEND-TEXT:Login to APP and get 
to a main menu.
[2022-08-04 18:43:46] [FINE] Test     - [Driver 0][login_to_main_menu] Step 7: CHECK-SCREEN-BUFFER:Login to AP
P and get to a main menu.
[2022-08-04 18:43:47] [FINE] Test     - [Driver 0][login_to_main_menu] End
[2022-08-04 18:43:47] [FINE] Driver   - [Driver 0] PASSED
[2022-08-04 18:43:47] [FINE] Test     - [Driver 0][logout_from_main_menu] Start
[2022-08-04 18:43:47] [FINE] Test     - [Driver 0][logout_from_main_menu] Step 1: CHECK-SCREEN-BUFFER:Logout f
rom the APP main menu.
[2022-08-04 18:43:47] [FINE] Test     - [Driver 0][logout_from_main_menu] Step 2: SEND-TEXT:Logout from the AP
P main menu.
[2022-08-04 18:43:47] [FINE] Test     - [Driver 0][logout_from_main_menu] Step 3: CHECK-SCREEN-BUFFER:Logout f
rom the APP main menu.
[2022-08-04 18:43:48] [FINE] Test     - [Driver 0][logout_from_main_menu] Step 4: SEND-TEXT:Logout from the AP
P main menu.
[2022-08-04 18:43:49] [FINE] Test     - [Driver 0][logout_from_main_menu] Step 5: LOG-SCREEN-BUFFER:Logout fro
m the APP main menu.
[2022-08-04 18:43:49] [FINE] Test     - [Driver 0][logout_from_main_menu] Step 6: WAIT-FOR-SCREEN-BUFFER:Logou
t from the APP main menu.
[2022-08-04 18:43:49] [FINE] Test     - [Driver 0][logout_from_main_menu] End
[2022-08-04 18:43:49] [FINE] Driver   - [Driver 0] PASSED
[2022-08-04 18:43:49] [FINE] TestSet  - End: tc_tests
[2022-08-04 18:43:49] [FINE] TestPlan - End: APP_tc_po_item_003
Completed in 1 minutes and 3.631 seconds. Status = PASSED.
```

#7 - 08/04/2022 01:15 PM - Tomasz Domin

Committed revision 19.

#8 - 08/08/2022 11:06 AM - Tomasz Domin

- % Done changed from 50 to 60

Implemented TestSet abort when early initialization of Test fails - in that case the entire TestSet queue is aborted
Changed log formatting to be a bit more readable.
CTRL-C report now dumps main index.html, so report is browsable starting from the root.

Committed revision 20.

#9 - 08/08/2022 01:07 PM - Greg Shah

Code Review Revisions 19, 20

Overall, I like the changes.

1. In build.xml, are both of these conditions needed?

```
     <condition property="manifest.name" value="${manifest.dir}/harness_1.8.mf">
        <matches string="${java.version}" pattern="^1\.8"/>
     </condition>
     <condition property="manifest.name" value="${manifest.dir}/harness_1.8.mf">
        <matches string="${java.version}" pattern="^1.\."/>
     </condition>
```

It seems like the 2nd condition handles both cases.

2. I think the dumping of output on CTRL-C seems like a feature we can always enable. Is there value in making it conditional?

3. I would like to split the logging output a little differently.

- WARN - most critical output, all errors are this level
  - Driver.run() pre-failures and catch block
- INFO - summary output
  - Harness.main() start and end
  - TestPlan.execute() start and end
  - TestSet.run() start and end
- FINE - detailed output
  - Driver.run() main while loop contents
  - Test.execute() start and end
- FINEST - verbose output
  - Test.execute() step details
  - Driver.run() finally block

4. I'd like to eliminate the dependency on the logging.properties. In addition, it would be good to control the logging level from the command line. Default the logging level to FINE.

5. Should we move all System.out usage to the logger?

6. JobQueue is missing a history entry.

7. In LogHelper, public static void warn(String message, Throwable e) needs javadoc.

8. In LogHelper, please move the private static String getCallingClassName(int level) to after the public methods.

9. All history entries in a branch that is not a "shared task branch" should have their own history number. This "harness" branch is not a shared task branch, so each of your entries should have history numbers.

#10 - 08/09/2022 10:28 AM - Tomasz Domin

- % Done changed from 60 to 90

Greg Shah wrote:

> Code Review Revisions 19, 20
>
> Overall, I like the changes.
>
> 1. In build.xml, are both of these conditions needed?

It is needed as java 1.8 configuration needs to be used for Java 11+, and since Java 9 version string format has been changed.
The first line matched Java 1.8, the second line matches java 10+ (e.g. "17.0.4"), I've added a separate entry for Java 9 (I cant test it anyway)
I kept the other line for pre-Java1.8 compatibility, which I cannot test and I presume is still working.

> 2. I think the dumping of output on CTRL-C seems like a feature we can always enable. Is there value in making it conditional?

I made a separate option "-c" to control CTRL-C dumps feature only (disabled by default)

> 3. I would like to split the logging output a little differently.
>
> - WARN - most critical output, all errors are this level
>   - Driver.run() pre-failures and catch block
> - INFO - summary output
>   - Harness.main() start and end
>   - TestPlan.execute() start and end
>   - TestSet.run() start and end
> - FINE - detailed output
>   - Driver.run() main while loop contents
>   - Test.execute() start and end
> - FINEST - verbose output
>   - Test.execute() step details
>   - Driver.run() finally block

Done

> 4. I'd like to eliminate the dependency on the logging.properties. In addition, it would be good to control the logging level from the command line. Default the logging level to FINE.

logging.properties is removed
Default log level is FINE now
It is increased to FINER when -d option is added
Added new option -l to specify desired log level (WARNING,INFO,FINE,FINER)

> 5. Should we move all System.out usage to the logger?

It was part of my plan, I've done it except for part that IMO require console access (via System.err)

> 6. JobQueue is missing a history entry.

Added history entry

> 7. In LogHelper, public static void warn(String message, Throwable e) needs javadoc.

Added javadoc

8. In LogHelper, please move the private static String getCallingClassName(int level) to after the public methods.

Moved

9. All history entries in a branch that is not a "shared task branch" should have their own history number. This "harness" branch is not a shared task branch, so each of your entries should have history numbers.

Added missing history entries.

Committed revision 21.

#11 - 08/10/2022 03:22 PM - Greg Shah

Code Review Revision 21

No objections.

#12 - 08/15/2022 10:58 AM - Tomasz Domin

Added logging of progress of test plan every 60 seconds:

```
2022-08-15 16:44:09 FINER   TestPlan regression_testing/PRE_CONDITION Total: 3 left: 3 failed: 0 passed: 0 successful: 0.00% progress: 0.00%
```

New option -x - if it is set the tests will abort anytime in case there is any not-test related failure (e.g. pre- fails). This is needed to make sure that environmental issues like terminal connection, out of memory etc are not affecting tests results, as test are only considered complete if there are no environmental issues

Committed revision 22.

#13 - 08/15/2022 11:43 AM - Greg Shah

Code Review Rev 22

1. In ExitCode, please set PRE_FAILED to -13 instead of -12 (which is a duplicate error number).

2. In TestPlan there should be a quick exit if FINER logging is not enabled. Otherwise we calculate stats but never output them.

#14 - 08/16/2022 05:17 AM - Tomasz Domin

Some ideas about how to make a kind of benchmark with harness

1. use exiting scenarios to get more real-world like results, like existing regression test
2. remove all failing tests, all tests that make up the benchmark must be successful
3. remove/override/reduce all pauses/wait elements, so there is no unnecessary waiting
4. increase poll frequency (e.g. for screen check) to make test results are more granular. Optionally there could be a feature that checks screen contents every time terminal sends data, so screen checks would be done ASAP
5. repeat tests few times to make sure results are repeatable thus reliable in given environment - with external script

Two more options are needed for that:

1. option to turn benchmark mode globally (to enable options 3 and 4 above, and any other)
2. option to skip tests with provided in a file, tests are excluded by test name

#15 - 08/16/2022 05:42 AM - Greg Shah

I have no objection to this idea. However, it probably should be implemented in a new task.

#16 - 01/19/2023 11:29 AM - Tomasz Domin

I've implemented two new features:
- trace mode - basically it allows to log even more details only when needed - enabled with -d -d command line options.
- timeout multiplier option - this allows to increase test step timeout for slower/overloaded machines ;) - its enabled with -g <int> command line option, e.g. -g 2 will multiply all timeouts in all scenario by 2.
Both features are under testing, I will commit them once they are tested.

#17 - 01/20/2023 05:24 AM - Tomasz Domin

There is an issue with threads when backout tasks fails - in most cases all tests executed on the same thread (Driver) following the failed one were also failing as backout task is usually used to restore initial test state to allow thread reusage.
To fix this I've implemented a new feature - when in debug mode if a backout task fails it is terminated. In case there are enough threads the test will continue to run but number of false negatives should be reduced. In case there are not enough thread test scenario will stop indefinitely, that's why it is only available in debug mode.
The feature is currently in testing.

#18 - 01/20/2023 02:27 PM - Tomasz Domin

I've implementing a feature that allows for controlling number of TestSet executed in parallel.