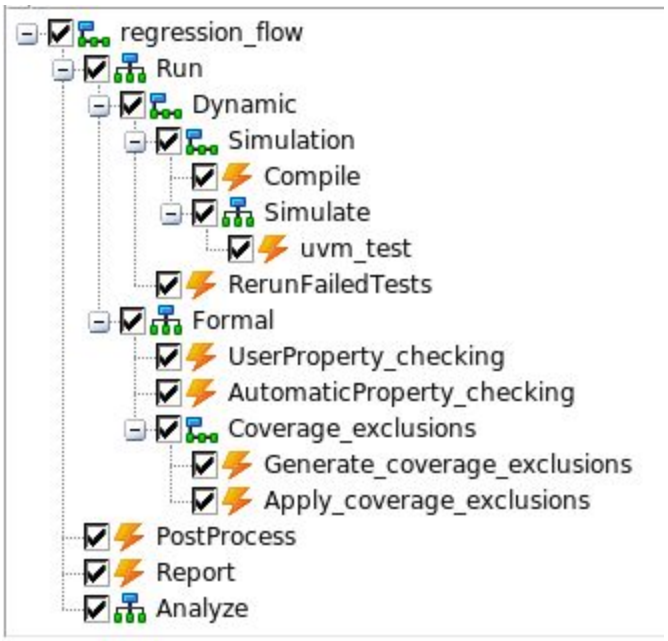
VRM App Layer User Manual

Table of contents

- Introduction
  - Known limitations
  - Prerequisites
- Quickstart guide
  - Setup your regression
    - step 1 Setup your compile flow
    - step 2 Setup your list of tests
    - step 3 Setup your regression run
  - Run your regression
  - Look at regression results
    - Ranking report
    - Coverage report
    - Trend report
    - merged coverage of current regression
    - merged coverage of all regressions
    - trend coverage of all regressions
    - Coverage exclusions
    - Summary of regression results
- Advanced Customization
  - VRM configuration parameters
  - Coverage exclusions
  - Profiling
  - Optional Steps
    - apply coverage exclusions

Introduction

This document intends to document the use model of the “generic” RMDB provided as a template to our end users as part of the Questa VRM tool and encapsulating all the generic tasks of a regression flow. it is limited to verification but can be expanded by each user to add custom features and tasks. Eventually it aims at serving projects with minimum customization/parameterization with maximum flexibility to add specific tasks.
As of today the flow is as depicted below:



There are 3 top tasks inside the regression_flow:

1. Run
    a. Run regression tests with different tools
2. PostProcess
    a. Placeholder for any tasks other than simulation such as adding trending metrics
3. Report
    a. Generate regression run reports
4. Analyze
    a. optional step to analyze a regression run performance

Run task is the main task and execute:

1. Dynamic
    a. Run regression tests with dynamic tools such as simulation
2. Formal
    a. Run regression tests with formal tools such as property checking

Known limitations

the current version has the following limitations:

- doesn't take care of compile scripts
    - a template Makefile is provided which is generic enough to be reused but is not part of the delivery
    - to be implemented by the end user using makefile or any other ways of his preference, he will just need to override the parameter "CompileCommand" so it is called accordingly by Compile task
- only support Questa tool suite
- only support UVM tests
    - support for non UVM tests may be added later
    - shall not need huge work/redesign of RMDB

Prerequisites

As of today the template is generic enough so that it shouldn't have any dependencies or prerequisites on the project structure and environment

Quickstart guide

this is a quickstart guide to get a regression flow up and running quickly with minimum customization but as well minimum effort. You can refer to the other sections of the user guide to customize further the regression environment and add additional features if required.

Setup your regression

step 1 Setup your compile flow

you have the choice of using Questa VRM to compile or not:

if you want to have the compile done by VRM then you need to override the parameter "CompileCommand" with your compile command or script, for instance:

- vrun -GCompileCommand="make -f MyMakefile compile"

if you want to do a separate compile, just make sure that you untick/exclude the Compile task when invoking vrun.

step 2 Setup your list of tests

Tests to be run by the regression must be captured into a file. You have the choice of 2 formats:

- csv format
- spreadsheet (soffice or excel)

the csv format follow the following structure:

	testname	**repeat count**	**seed**
level	***compulsory***	***compulsory***	***optional*** (set to 'random' by default)
description	name of your test	# of iterations	list of seeds space separated

The test file is basically a list of tests with their associated options (seed, # of repeat). 2 examples of test file are shown below.
The 1st one define a set of directed and random tests used for running tests to reach coverage. All tests leave the seeds empty and thus set it as random.

```
# File Syntax is
# <testname> <repeat_count> <1st seed>...<the seed>
# If not enough seeds then random is used to pad seeds.
```

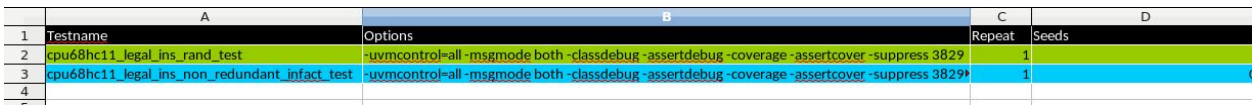
```
#directed tests
ace_rw_generic_test 1
ace_rw_generic_reordering_test 1
ace_rw_phase_test 1
ace_rw_txn_test 1
ace_rw_txn_len_size_incr_test 1
ace_generic_test 1
ace_txn_test 1

#random tests
ace_rw_txn_system_random_test 1
ace_rw_txn_nonshareable_random_test 1
ace_rw_txn_innershareable_random_test 1
ace_rw_txn_outershareable_random_test 1
ace_rw_txn_random_test 16
```

the 2nd one shows a test file of a regression running only the contributing tests, it has been generated automatically after analyzing all tests contribution and rebuilding the test file from there. not that each test now has a defined seed number and is only run once.

```
ace_rw_txn_random_test 1 1857278929
ace_rw_txn_random_test 1 1356686004
ace_rw_txn_random_test 1 1987789029
ace_rw_txn_random_test 1 950649920
ace_rw_txn_random_test 1 44670287
ace_rw_txn_random_test 1 206765227
ace_rw_txn_random_test 1 1183696954
ace_rw_txn_random_test 1 1077398618
ace_rw_txn_random_test 1 1973792859
ace_rw_txn_random_test 1 54993495
ace_rw_txn_random_test 1 59206403
ace_rw_txn_random_test 1 830722182
ace_rw_txn_random_test 1 500472126
ace_rw_txn_random_test 1 1069218584
ace_rw_txn_random_test 1 1632976657
ace_rw_txn_random_test 1 632504634
ace_rw_txn_random_test 1 1860941171
ace_rw_txn_random_test 1 1972545130
ace_rw_txn_random_test 1 2074607036
ace_rw_txn_random_test 1 344078625
ace_rw_txn_random_test 1 581068967
ace_rw_txn_random_test 1 651866545
ace_rw_txn_random_test 1 873574746
ace_rw_txn_random_test 1 886426089
```

The spreadsheet format is pretty similar except it is presented as a spreadsheet, as for csv you will capture the testname, simulation options, count of repetition and optionally the seed. Snapshot example below illustrate what it will look like



	A	B	C	D
1	Testname	Options	Repeat	Seeds
2	cpu68hc11_legal_ins_rand_test	-uvmcontrol=all -msgmode both -classdebug -assertdebug -coverage -assertcover -suppress 3829	1	
3	cpu68hc11_legal_ins_non_redundant_infact_test	-uvmcontrol=all -msgmode both -classdebug -assertdebug -coverage -assertcover -suppress 3829	1	
4				

step 3 Setup your regression run

Last step is to setup the minimal set of parameters required by Questa VRM to run properly.
The list of required parameters are below:

description	name	default
filename of tests spreadsheet	testfile	none
Sheet name of test spreadsheet to be picked up	testfile_tab	none
path to Questa library ini file	MODELSIMINI	modelsim.ini in run directory
path to snapshot to simulate	SNAPSHOT	none

That section only show required parameters, optional parameters are discussed in more details into section “Regression configuration parameters”.
To set the parameters via the GUI refer to chapter “Adding New Configurations to the Project File” and “Edit VRM Configurations” of Questa VRM documentation.
To set the parameters via the command line refer to chapter “Override Parameter Values from Command Line”of Questa VRM documentation.

Run your regression

TO DO
<refer here to Questa VRM command line and GUI to launch the regression with the parameters setup in the previous section.
will be done later as that doesn’t add value>

Look at regression results

TODO document here the outputs generated by the regression and where they are located
the regression run generate a number of outputs that can be analyzed at the end or in the course of the regression
TODO put the exact reference and filename of the files depicted below

Ranking report

after all tests are ran, a ranking process is launched and provide the following outputs:

- list of contributing tests
- list of non contributing test
- optimized regression list which allow to rerun a regression with only contributing tests

Coverage report

after all tests are ran a coverage report in HTML format is generated and stored under <regression dir>/report/coverage. refer to Questa user manual on “coverage report” for further details.

Trend report

After each regression run a trending report in HTML format is generated and stored under <regression dir>/report/trend. Refer to Questa user manual on “trend report” for further details.

merged coverage of current regression

at the end of the regression run, the merged coverage of the regression is available under <regression dir> and is saved as well under <regression dir>/logs with a timestamp suffix. the 1st merged coverage can be used to check the coverage of the specific regression, do analysis query (which test contributed to what, etc …) while the latter is saved to make sure that one can go back and do these queries even after a regression clean that delete all datas under <regression dir> except for the logs directory contents.

merged coverage of all regressions

at the end of regression run, the coverage result of the regression is merged with the previous regressions result in <regression dir>/logs. that enables one to have the merged coverage of all regressions run from the beginning of the project till present.

trend coverage of all regressions

at the end of regression run, the trendable coverage result of the regression is merged with the trend coverage file to track the regression trend.

Coverage exclusions

Summary of regression results

TODO put here as a table all the files above with names and location

description	name	location
merged coverage	<reg name>_merge.ucdb	<regression dir>

Advanced Customization

VRM configuration parameters

TODO describe all configurable parameters of RMDB, for now put the output of "grep "<parameter name=" below and shall make a table out of it with parameter description.

```
                <parameter name="PRJ_TB_SRC_ROOT"></parameter>
        <parameter name="PRJ_DUT_SRC_ROOT"></parameter>
        <parameter name="PRJ_DUT_VERSION" export="yes"></parameter>
        <parameter name="TIMESTAMP" type="tcl">[exec date
+%y%m%d_%H%M%S]</parameter>
        <parameter name="MODELSIMINI" type =
"tcl">(%VRUNDIR%)/modelsim.ini</parameter>
        <parameter name="regPrefix">reg</parameter>
        <parameter name="mergefile" export="yes">(%regPrefix%)_merge.ucdb</parameter>
        <parameter name="tplanfile">(%VRUNDIR%)/yourproject_testplan.xml</parameter>
        <parameter name="tplanoptions">-format Excel -verbose</parameter>
        <parameter name="mergeoptions" >-testassociated</parameter>
        <!--parameter name="trendoptions" ></parameter-->
        <parameter name="triagefile">triage.tdb</parameter>
        <parameter name="triageoptions">-severity IFE -teststatus FEW -rulesfile
(%RMDBDIR%)/transform.txt -verbose</parameter>
        <parameter
name="mergefileAll">(%DATADIR%)/logs/(%regPrefix%)_merge_all.ucdb</parameter>
        <parameter name="rankfile">(%DATADIR%)/(%regPrefix%).rank</parameter>
        <parameter name="rankoptions">-fewest -log (%rankfile%).log</parameter>
        <parameter name="trendfile"
>(%DATADIR%)/logs/(%regPrefix%)_trend.ucdb</parameter>
        <parameter name="reportoptions">-html -details -source -code bcestxf -assert -cvg
-htmldir</parameter>
        <parameter
name="CoverageAutoExcludeFile">(%DATADIR%)/covercheck_exclude.do</parameter>
        <parameter name="CoverageManualExcludeFile">manual_exclude.do</parameter>
        <parameter name="DEBUGMODE">0</parameter>
        <parameter name="SNAPSHOT" >TBD</parameter>
        <parameter name="runmode">-c</parameter>
        <parameter name="testfile">TBD</parameter>
        <parameter name="testfile_tab">TBD</parameter>
        <parameter name="TESTCASES" type = "tcl">[ReadCalc (%testfile%)
(%testfile_tab%)]</parameter>
```

```
        <parameter name="TBLinesOfCode" type = "tcl">[expr [GetLinesOfCodeFromPathlist
(%PRJ_TB_SRC_ROOT%) *.sv] + [GetLinesOfCodeFromPathlist (%PRJ_TB_SRC_ROOT%)
*.svh]]</parameter>
        <parameter name="DUTLinesOfCode" type = "tcl">[GetLinesOfCodeFromFilelist
(%PRJ_DUT_SRC_ROOT%)]</parameter>
        <parameter name="SimulatePrecommand"></parameter>
        <parameter name="testname">(%testcase%)</parameter>
        <parameter name="uvm_testname" type="tcl">[lindex [split (%testcase%) "."]
0]</parameter>
        <parameter name="seed" type="tcl">[lindex [split (%testcase%) "."] 1]</parameter>
        <parameter name="testoptions" type="tcl">[lindex [GetTestSettings (%testcase%)] 0]
+UVM_TESTNAME=(%uvm_testname%)</parameter>
        <parameter name="seed">random</parameter>
        <parameter name="ucdbfile">(%INSTANCE%).ucdb</parameter>
        <parameter name="UCDBFILTER"></parameter>
        <parameter name="vsimoptions"> -modelsimini (%MODELSIMINI%) -do "run.do" -wlf
(%INSTANCE%).wlf -l (%INSTANCE%).log -title (%INSTANCE%) -sv_seed (%seed%)
(%UCDBFILTER%) -cvgprecollect (%mergefile%)</parameter>
        <parameter name="vsimPrecommand"></parameter>
        <parameter name="vsimRundo">coverage save -cvg -codeAll -assert -onexit
(%ucdbfile%);run -a;q -f</parameter>
        <parameter name="vsimPostcommand"></parameter>
        <parameter name="COMPILE_PARAMS"></parameter>
        <parameter name="DUTMODULE" ></parameter>
        <parameter name="DUTLIB" ></parameter>
        <parameter name="DUTPREFIX"></parameter>
        <parameter name="CLKNAME">clk</parameter>
        <parameter name="CLKDUTY">0 50</parameter>
        <parameter name="CLKPERIOD">100</parameter>
        <parameter name="RSTNAME">rst_n</parameter>
        <parameter name="RSTACTIVE">low</parameter>
        <parameter name="DIRECTIVES"></parameter>
        <parameter name="VERIFY_PARAMS">-effort low</parameter>
        <parameter name="ucdbfile">(%INSTANCE%).ucdb</parameter>
        <parameter name="VERIFY_PARAMS">-effort low</parameter>
        <parameter name="VERIFY_PARAMS">-effort low</parameter>
```