# VRM APP LAYER USER MANUAL

TABLE OF CONTENTS

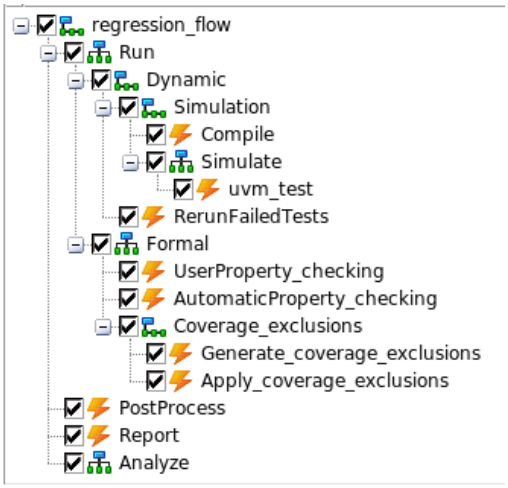
## INTRODUCTION

This document intends to document the regression flow Run Manager DataBase (RMDB) template provided with Questa Verification Run Manager (VRM) to speed up Questa VRM integration in your regression environment.

The RMDB template encapsulate all generic tasks of a regression flow, such as simulation, formal, compilation, etc. … It is limited to verification but can be easily expanded on demand to either enhance the existing verification flow or add custom tasks part of your regression. Eventually it aims at serving projects with minimum customization/parameterization and maximum flexibility.

## TEMPLATE FLOW DESCRIPTION

The regression flow described in the template RMDB is as depicted below:




4 top tasks are defined:

1. **Run**
   - Run regression tests with different tools
2. **PostProcess**
   - Placeholder for any tasks other than simulation such as adding trending metrics
3. **Report**
   - Generate regression run reports
4. **Analyze**
   - Optional step analyzing regression run performance

The top Run task is the main task and execute the following sub-tasks:

- **Dynamic**

- Run dynamic simulation

➢ **Formal**

- Run Static formal tools

Again each task or sub-task can be enhanced to meet your specific flow needs.

## KNOWN LIMITATIONS

Current version has the following limitations:

- **Doesn't take care of compile scripts**
  - 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**
  - Can be customized to support 3rd party simulator if needed
- **Only support UVM tests**
  - Support for non UVM tests may be added later
  - Shall not need huge work/redesign of RMDB

## PREREQUISITES

There are no dependencies or prerequisites on the existing project structure and environment. To read the remaining of the document it is recommended that you get familiar with the basics of Questa VRM as we will use terms coming from the Questa VRM documentation and architecture.

You can find the appropriate documentation under the Questa install tree documentation directory.

## QUICKSTART GUIDE

That quickstart will guide you through the mandatory steps to quickly integrate the template regression flow and get your regression up and running with minimum customization and effort.

Refer to the additional sections of that user guide to customize further the regression environment and add additional features if required.

## SETUP YOUR REGRESSION FLOW

During that step you will need to set parameters of the template RMDB:

- To set 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.
  - Syntax is –G<parameter>=<value>

### STEP 1 SETUP YOUR COMPILE FLOW

You can choose to compile your design and test bench with Questa VRM or leave it to your existing flow:

- Compilation outside of Questa VRM
  - Make sure that you untick/exclude the Compile task when invoking vrun
  - i.e. *vrun –exclude regression_flow/Dynamic/Simulation/Compile <other vrun options>*

- Compilation with Questa VRM
  - Override "CompileCommand" parameter with your compile command or script
  - i.e. *vrun -GCompileCommand="make compile" <other vrun options>*

### STEP 2 SETUP YOUR REGRESSION TESTS LIST

The tests to be ran must be capture in either of these 2 format:

- csv space separated
- soffice or Excel spreadsheet

By default the entry format is a spreadsheet, it can be controlled via "testfileformat" parameter:

- i.e. *vrun –Gtestfileformat="csv" <other vrun options>*

#### CSV FORMAT

The csv format follow the syntax as defined below:

```
# File Syntax is
# <testname> {<simulation options>} <repeat_count> <1st seed>...<nth seed>
# If not enough seeds then random is used to pad seeds.
```

| Field | testname | Options | repeat count | Seed |
|---|---|---|---|---|
| Level | REQUIRED | OPTIONAL | REQUIRED | OPTIONAL |
| Definition | **UVM test name** | Options passed at simulator invocation | **# of iterations** | Seeds (random by default) |

**Commented [GA1]:** This is not fully supported yet, will be added soon
For now spreadsheet is the working format
We need to add the simulation options as well from the text file

As an example we have 2 test files shown below.

The 1st example define a set of directed and random tests used to reach coverage. The seeds are left empty except for 1 test where it is set to 20. When not defined, seed is set as random and will be generated by the simulator. For the directed tests it is a don't care as there are no random objects.

```
# File Syntax is
# <testname> {<simulation options>} <repeat_count> <1st seed>...<nth 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
#random tests
ace_rw_txn_system_random_test {+SCRAMBLING=off +UVM_VERBOSITY=UVM_DEBUG} 1 20
ace_rw_txn_nonshareable_random_test {} 1
ace_rw_txn_innershareable_random_test {} 1
```

you can see as well that for the fixed seed test we are passing additional options to the simulation so we can debug and turn off a part of the design for instance.

### SPREADSHEET FORMAT

**Commented [GA2]:** This is not fully supported yet, will be added soon
For now spreadsheet is the working format

The spreadsheet format is pretty similar except it is captured in a spreadsheet. As for the csv file, you will capture testname, simulation options, count of repetition and optionally the seed.

An example is shown below:

| | 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 PARAMETERS

The last step is to set the minimal set of mandatory parameters. The following table show the list of required parameters.

For additional customization, Optional parameters are discussed in more details into section "Regression configuration parameters". With that minimum set of parameters you should be good to go with the regression run.

| Description | Name | Example |
|---|---|---|
| *if Compile task is ran, mandatory compile command to compile design and TB* | **CompileCommand** | Make compile |
| *If you want to link a test plan, set it to the value of your test plan file with full path (i.e. if not in excel format, set as well tplanoptions accordingly)* | **tplanfile** | /home/project/testplan.xls |

| | | |
|---|---|---|
| *snapshot loaded in simulator* | **SNAPSHOT** | top_dut_tb_opt |
| *spreadsheet/csv file regression tests list* | **testfile** | TestsList.xls |
| *spreadsheet sheet/tab to extract the tests list* | **testfile_tab** | SmokeTests |

## RUN YOUR REGRESSION

Now that you have setup the regression flow you are good to go and run your regression. You can refer to “Questa VRM user manual” to have all the possible options of vrun command, below is an example of invoking vrun:

*vrun -Gtestfile=testslist.ods -Gtestfile_tab=SmokeTests -GSNAPSHOT=TOP.top_hdl_hvl_opt -GCompileCommand="cd (%VRUNDIR%);make all" -include regression_flow -exclude regression_flow/Run/Dynamic/RerunFailedTests -exclude regression_flow/Run/Formal -exclude regression_flow/Analyze -Gmergeoptions="-testassociated -du cpu68hc11\(rtl\) -recursive" -Gtplanfile=$PRJ_ROOT/run/reqtracer/CPUCORE_68HC11_TP.xml*

That online command will run a regression with Questa VRM in batch mode with the following characteristics:

- grab regression tests to be ran in the spreadsheet testslist.ods at tab/sheet SmokeTests
- load the snapshot defined by parameter SNAPSHOT
- launch a compile command as defined per parameter CompileCommand
- only run the Dynamic simulation, PostProcess and Report tasks
- merge only the coverage for a specific design unit
- link the test plan define by parameter tplanfile to simulation coverage results

This is given as an example command and parameters overriding, much more parameterizations can be done and is documented in the section “Regression configuration parameters”.

## LOOK AT REGRESSION RESULTS

Commented [GA3]: TODO document here the outputs generated by the regression and where they are located

During and after the regression run, a number of outputs are generated and can be displayed and analyzed to steer the verification tasks to be done next (i.e. debugging failed test, cover coverage holes, eliminating noncontributing tests, etc. ….). The different outputs are defined in that section.

### TESTS SIMULATION OUTPUTS

All simulation outputs are available under <regression dir>/regression_flow/Run/Dynamic/Simulation/Simulate directory. These directories can be reused as is outside of Questa VRM to replay a specific tests or ship a debug environment to another engineer, indeed the test directories are self-contained and can be reused outside of the VRMDATA directory structure.

This is a powerful and productive way of passing failing tests to debug for instance to a specific engineer.

### FAILED TEST REPORT

At the end of regression, a list of all failed tests is automatically generated. The list can be generated either with only the current regression results or with the cumulative regression results, thus listing all failed tests from all regressions run.

The list is cleared if “v-clean” or “-realclean” switch is passed to vrun or can be removed manually from Questa VRM data directory. Upcoming release of the template RMDB shall support an automatic deletion via a RMDB parameter. (i.e. FAILED_TEST=cumulative or current)

### 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 data’s under <regression dir> except for the logs directory contents.

### MERGED COVERAGE OF ALL REGRESSIONS

At the end of regression, coverage result of the regression is merged with the previous regressions result in <regression dir>/logs. It enables to keep the merged coverage of all regressions run from the beginning of the project till present even after a “-clean”.

### TREND COVERAGE OF ALL REGRESSIONS

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

### RANKING REPORT

At the end of regression, a ranking process is launched automatically and provide the following outputs which are all generated under the Questa VRM data directory:

- List of contributing tests (.contrib file extension)
- List of noncontributing tests (.noncontrib file extension)
- Log of ranking process containing the contributing,, non-contributing and missing tests (cf. tests linked to the test plan but not run) (.rank.log file extension)
- Optimized tests list (_optimized_tesfile suffix file)

The optimized test list is generated by processing the contributing tests list and the ucdb to recreate a test file in csv format so one can rerun automatically or on-demand an optimized regression for instance after a fix to a bug is done.

## COVERAGE REPORT

At the end of regression, 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

At the end of regression, 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.

## QUESTA VRM REPORT

At the end of regression, a regression run report in HTML format is generated and stored under <regression dir>/report/vrun. Refer to Questa user manual on “trend report” for further details.

## COVERAGE EXCLUSIONS

**Commented [GA4]:** TODO add formal exclusion definition

# ADVANCED CUSTOMIZATION

## REGRESSION CONFIGURATION PARAMETERS

Commented [GA5]: Would be productive if we could extract/update the table from the RMDB using tags/comment in the XML as this is a moving target

### GLOBAL PARAMETERS

| Description | Name | Default Value |
|---|---|---|
| *Prefix used for filenames such as ucdb merge file, report, etc. …* | **regPrefix** | reg |
| *Root directory of TB source, may be used for trending on TB code* | **PRJ_TB_SRC_ROOT** | none |
| *Root directory of DUT source, may be used for trending on DUT code* | **PRJ_DUT_SRC_ROOT** | none |
| *DUT Release #, may be used for trending on DUT code version* | **PRJ_DUT_VERSION** | none |
| *Modelsim ini variable to define Questa libraries mapping* | **MODELSIMINI** | (%VRUNDIR%)/modelsim.ini |

### COMPILATION PARAMETERS

| Description | Name | Default Value |
|---|---|---|
| *if Compile task is ran, mandatory compile command to compile design and TB* | **CompileCommand** | TBD |

### COVERAGE PARAMETERS

| Description | Name | Default Value |
|---|---|---|
| *Questa VRM predefined parameter for regression merged ucdb* | **mergefile** | (%regPrefix%)_merge.ucdb |
| *Questa VRM predefined parameter for regression merge options* | **mergeoptions** | -testassociated |
| *Questa VRM predefined parameter for trending ucdb* | **trendfile** | (%DATADIR%)/logs/(%regPrefix%)_trend.ucdb |
| *Questa VRM predefined parameter for trending options* | **trendoptions** | none |
| *Questa VRM predefined parameter for test plan linking to ucdb* | **tplanfile** | TBD |
| *Questa VRM predefined parameter for test plan linking options* | **tplanoptions** | -format Excel -verbose |
| *Merge file preserved even after a –clean,*<br>*deleted if -realclean* | **mergefileAll** | (%DATADIR%)/logs/(%regPrefix%)_merge_all.ucdb |

| | | |
|---|---|---|
| *Placeholder to prevent ucdb test status to be failed/warning by excluding error message patterns using questasim –ucdbteststatusmsgfilter options (refer to Questa VM documentation for usage)* | **UCDBFILTER** | none |
| *Optional code coverage exclude file applied on ucdb merge file* | **CoverageManualExcludeFile** | TBD |
| *Optional code coverage exclude file generated by Questa Covercheck and applied on ucdb merge file* | **CoverageAutoExcludeFile** | (%DATADIR%)/covercheck_exclude.do |

## SIMULATION PARAMETERS

| Description | Name | Default Value |
|---|---|---|
| *snapshot loaded in simulator* | **SNAPSHOT** | REQUIRED |
| *Define test list format which can be either "csv" or "sheet"* | **testfileformat** | sheet |
| *spreadsheet/csv file with regression tests list* | **testfile** | REQUIRED |
| *spreadsheet sheet/tab to extract the tests list* | **testfile_tab** | REQUIRED if testfileformat="sheet" |
| *Options passed to simulator* | **vsimoptions** | -modelsimini (%MODELSIMINI%) -do "run.do" -wlf (%INSTANCE%).wlf -l (%INSTANCE%).log -title (%INSTANCE%) -sv_seed (%seed%) (%UCDBFILTER%) -cvgprecollect (%mergefile%) |
| *switch debug mode if set to 1 -> run.do file will source debug.do file* | **DEBUGMODE** | 0 |
| *switch to batch or interactive simulation mode* | **RUNMODE** | -c |
| *contents of run.do file source at simulation time 0* | **vsimRundo** | coverage save -cvg -codeAll -assert -onexit (%ucdbfile%);run -a;q -f |
| *contents of debug.do file source in run.do if debug mode is on* | **vsimDebugdo** | add log -r /* -depth 4 |
| *placeholder to launch Unix commands prior starting all simulations* | **SimulatePrecommand** | none |
| *placeholder to launch Unix commands prior each test simulation* | **testPrecommand** | none |
| *placeholder to launch Unix commands after each test simulation* | **testPostcommand** | none |

## REPORT PARAMETERS

| Description | Name | Default Value |
|---|---|---|

| | | |
|---|---|---|
| *Questa VRM predefined parameter for regression ranking output* | **rankfile** | (%DATADIR%)/(%regPrefix%).rank |
| *Questa VRM predefined parameter for regression ranking options* | **rankoptions** | -fewest -log (%rankfile%).log |
| *Questa VRM predefined parameter for result analysis database* | **triagefile** | (%regPrefix%)_triage.tdb |
| *Questa VRM predefined parameter for result analysis options* | **triageoptions** | -severity IFE -teststatus FEW -verbose |
| *Options for regression report output, by default HTML report is generated with full details* | **reportoptions** | -html -details -source -code bcestxf -assert -cvg -htmldir (%DATADIR%)/report/coverage |
| *Questa VRM predefined parameter for regression failed tests log file* | **faillog** | (%regPrefix%)_failed_tests.log |

## FORMAL PARAMETERS

| Description | Name | Default Value |
|---|---|---|
| | **COMPILE_PARAMS** | |
| | **DUTMODULE** | |
| | **DUTLIB** | |
| | **DUTPREFIX** | |
| | **CLKNAME** | clk |
| | **CLKDUTY** | 0 50 |
| | **CLKPERIOD** | 100 |
| | **RSTNAME** | rst_n |
| | **RSTACTIVE** | low |
| | **DIRECTIVES** | |
| | **VERIFY_PARAMS** | -effort low |