OSVVM Settings User Guide

User Guide for Release 2024.11

By

Jim Lewis

SynthWorks VHDL Training

Jim@SynthWorks.com

http://www.SynthWorks.com

Table of Contents

1 OSVVM Settings Overview

The default OSVVM settings are intended to be the best settings for new users of OSVVM. So if you are just getting started with OSVVM, you can skip reading this document for now.

OSVVM supports VHDL settings in OsvvmScriptSettingsPkg_generated.vhd and OsvvmSettingsPkg_default.vhd, and script settings in OsvvmSettingsLocal.tcl.

2 OsvvmScriptSettingsPkg

OsvvmScriptSettingsPkg is used to help the VHDL code locate items set by the scripts. It uses deferred constants, with the constant declarations in the file OsvvmScriptSettingsPkg.vhd and the package body generated as OsvvmScriptSettingsPkg_generated.vhd.

2.1 OsvvmScriptSettingsPkg_generated.vhd

```
-- This file is autogenerated by CreateOsvvmScriptSettingsPkg
package body OsvvmScriptSettingsPkg is
  constant OSVVM_HOME_DIRECTORY         : string := "C:/OsvvmLibraries" ;
  constant OSVVM_RAW_OUTPUT_DIRECTORY   : string := "" ;
  constant OSVVM_BASE_OUTPUT_DIRECTORY  : string := "" ;
  constant OSVVM_BUILD_YAML_FILE        : string := "OsvvmRun.yml" ;
  constant OSVVM_TRANSCRIPT_YAML_FILE   : string := "OSVVM_transcript.yml" ;
  constant OSVVM_REVISION               : string := "2024.09" ;
  constant OSVVM_SETTINGS_REVISION      : string := "2024.09" ;
end package body OsvvmScriptSettingsPkg ;
```

2.2 Location of OsvvmScriptSettingsPkg_generated.vhd

The package OsvvmScriptSettingsPkg_generated.vhd is generated by the OSVVM scripts. It is created in either OsvvmLibraries/Scripts/$OsvvmSettingsSubDirectory or $CurrentSimulationDirectory/$OsvvmSettingsSubDirectory depending on the setting of TCL variable SettingsAreRelativeToSimulationDirectory.

If OsvvmScriptSettingsPkg_local.vhd exists, it will be analyzed, otherwise, if OsvvmScriptSettingsPkg_generated.vhd exists, it will be analyzed, otherwise, OsvvmScriptSettingsPkg_default.vhd will be analyzed.

3 OsvvmSettingsPkg

OsvvmSettingsPkg is used to maintain backward compatibility.

3.1 Package Body and Interface Groups

OsvvmSettingsPkg_default.vhd is a package body that sets deferred constants. It is organized as follows.

```
use work.IfElsePkg.all ;
use work.OsvvmScriptSettingsPkg.all ;

package body OsvvmSettingsPkg is
  -- Settings for RandomPkg
  -- Settings for ScoreboardGenericPkg
  -- Settings shared by AlertLogPkg and CoveragePkg
  -- Settings for CoveragePkg
  -- Settings for AlertLogPkg
end package body OsvvmSettingsPkg ;
```

3.2 Settings for RandomPkg

```
constant RANDOM_USE_NEW_SEED_METHODS : boolean :=
    IfElse(OSVVM_SETTINGS_REVISION >= "2024", TRUE, FALSE) ;  -- Historic FALSE
```

RANDOM_USE_NEW_SEED_METHODS selects higher quality seed generation methods in InitSeed. Make it TRUE for new designs. Keep it FALSE for older test cases that use OSVVM randomization.

3.3 Settings for ScoreboardGenericPkg

```
constant SCOREBOARD_YAML_IS_BASE_FILE_NAME : boolean :=
    IfElse(OSVVM_SETTINGS_REVISION >= "2024", TRUE, FALSE) ;  -- Historic FALSE
```

WriteScoreboardYaml is used when creating a custom scoreboard instance. Setting SCOREBOARD_YAML_IS_BASE_FILE_NAME to TRUE allows a short name to be used with WriteScoreboardYaml, otherwise, a name of an exact format must be used.

3.4 Shared Settings for AlertLogPkg and CoveragePkg

These determine the names printed by output generated by AlertLogPkg and CoveragePkg.

```
constant  OSVVM_PRINT_PREFIX         : string := "%% " ;
constant  OSVVM_DONE_NAME            : string := "DONE" ;
constant  OSVVM_PASS_NAME            : string := "PASSED" ;
constant  OSVVM_FAIL_NAME            : string := "FAILED" ;
constant  OSVVM_DEFAULT_TIME_UNITS   : time := 1 ns ;
```

3.5 Settings for CoveragePkg

COVERAGE_DEFAULT_WEIGHT_MODE sets the default weight mode for coverage randomization.

```
constant  COVERAGE_DEFAULT_WEIGHT_MODE  : string :=
    IfElse(OSVVM_SETTINGS_REVISION >= "2024", string'("REMAIN"), "AT_LEAST") ;
```

COVERAGE_USE_NEW_SEED_METHODS is like RANDOM_USE_NEW_SEED_METHODS for the coverage package.

```
constant  COVERAGE_USE_NEW_SEED_METHODS : boolean := TRUE ;
```

The following are used for names printed in coverage reports.

```
constant  COVERAGE_PRINT_PREFIX         : string := OSVVM_PRINT_PREFIX ;
constant  COVERAGE_PASS_NAME            : string := OSVVM_PASS_NAME ;
constant  COVERAGE_FAIL_NAME            : string := OSVVM_FAIL_NAME ;
```

The following are default settings used in the WriteBin reports. These are not relevang for the HTML reports.

```
constant COVERAGE_WRITE_PASS_FAIL   : boolean := FALSE ;
constant COVERAGE_WRITE_BIN_INFO    : boolean := TRUE ;
constant COVERAGE_WRITE_COUNT       : boolean := TRUE ;
constant COVERAGE_WRITE_ANY_ILLEGAL : boolean := FALSE ;
```

3.6 Settings for AlertLogPkg

The following controls specify that Alerts and Logs are to be printed justified with time first and time is justified by 9.

```
constant  ALERT_LOG_JUSTIFY_ENABLE  : boolean :=
    IfElse(OSVVM_SETTINGS_REVISION >= "2024", TRUE, FALSE) ; -- Historic FALSE
constant  ALERT_LOG_WRITE_TIME_FIRST : boolean :=
    IfElse(OSVVM_SETTINGS_REVISION >= "2024", TRUE, FALSE) ; -- Historic FALSE
constant  ALERT_LOG_WRITE_TIME_LAST : boolean := not ALERT_LOG_WRITE_TIME_FIRST ;
constant  ALERT_LOG_TIME_JUSTIFY_AMOUNT : integer :=
  IfElse(OSVVM_SETTINGS_REVISION >= "2024", 9, 0) ;  -- Justify time
```

The following controls defaults for AffirmIfFilesMatch and AlertIfFilesNotMatch.

```
constant  ALERT_LOG_IGNORE_SPACES  : boolean := FALSE ; -- Historic FALSE
constant  ALERT_LOG_IGNORE_EMPTY_LINES  : boolean := FALSE ; -- Historic FALSE
```

The following controls defaults for printing Alerts and Logs.

```
-- prefix message with # of errors - requested by Marco for Mike P.
constant  ALERT_LOG_WRITE_ERRORCOUNT : boolean := FALSE ;
constant  ALERT_LOG_WRITE_NAME    : boolean := TRUE ;   -- Print name Alert/Log
constant  ALERT_LOG_WRITE_LEVEL   : boolean := TRUE ;   -- Print Level Name
constant  ALERT_LOG_WRITE_TIME    : boolean := TRUE ;   -- Print Time
constant  ALERT_LOG_ALERT_NAME    : string := "Alert" ;
constant  ALERT_LOG_LOG_NAME      : string := "Log  " ;
constant  ALERT_LOG_ID_SEPARATOR  : string := ": " ;
constant  ALERT_LOG_PRINT_PREFIX  : string := OSVVM_PRINT_PREFIX ;
constant  ALERT_LOG_DONE_NAME     : string := OSVVM_DONE_NAME ;
constant  ALERT_LOG_PASS_NAME     : string := OSVVM_PASS_NAME ;
constant  ALERT_LOG_FAIL_NAME     : string := OSVVM_FAIL_NAME ;
constant ALERT_LOG_NOCHECKS_NAME  : string := "NOCHECKS" ;
constant ALERT_LOG_TIMEOUT_NAME   : string := "TIMEOUT" ;
```

Default stop counts for Alert FAILURE, ERROR, and WARNING.

```
constant  ALERT_LOG_STOP_COUNT_FAILURE         : integer := 0 ;
constant  ALERT_LOG_STOP_COUNT_ERROR           : integer := integer'high ;
constant  ALERT_LOG_STOP_COUNT_WARNING         : integer := integer'high ;
```

Default stop counts for Alert FAILURE, ERROR, and WARNING.

```
constant  ALERT_LOG_GLOBAL_ALERT_ENABLE : boolean := TRUE ;
```

Default for requirements goals.

```
constant  ALERT_LOG_DEFAULT_PASSED_GOAL        : integer := 1 ;
```

Control what makes a test failure.   Recommendation set all to TRUE.

```
constant  ALERT_LOG_FAIL_ON_WARNING            : boolean := TRUE ;
constant  ALERT_LOG_FAIL_ON_DISABLED_ERRORS    : boolean := TRUE ;
constant  ALERT_LOG_FAIL_ON_REQUIREMENT_ERRORS : boolean := TRUE ;
```

The following controls defaults for printing with ReportAlerts.

```
constant ALERT_LOG_REPORT_HIERARCHY            : boolean := TRUE ;
constant ALERT_LOG_PRINT_PASSED                : boolean := TRUE ;
constant ALERT_LOG_PRINT_AFFIRMATIONS          : boolean := FALSE ;
constant ALERT_LOG_PRINT_DISABLED_ALERTS       : boolean := FALSE ;
constant ALERT_LOG_PRINT_REQUIREMENTS          : boolean := FALSE ;
constant ALERT_LOG_PRINT_IF_HAVE_REQUIREMENTS  : boolean := TRUE ;
```

4 OsvvmSettingsLocal

OsvvmSettingsLocal.tcl is used configure script settings. It is placed in the directory OsvvmLibraries/Scripts. If this file exists, it is used to change settings. The file OsvvmSettingsLocal_Example.tcl is provided in OsvvmLibraries/Scripts as a template.

4.1 Structure of OsvvmSettingsLocal.tcl

```
# Global Settings
namespace eval ::osvvm {
# OSVVM Name Space Settings
  #  Version Compatibility Settings
  #  Directory structure and results file management
  #  TCL Error signaling during a build
  #  Stop Counts for Failures seen by Analyze and Simulate
  #  Generate transcripts
  #  VHDL Simulation Settings
  #  Default Coverage Options
  #  Simulation Controls
  #  Extended Analyze and Simulate Options
  #  GHDL Analyze and Simulate Options
  #  Second Top
}
```

4.2 Global Settings

Set variables to your project variables here.

```
variable Proj1 C:/Dev/Proj1
```

This will allow you to do builds for your project as:

```
build $Proj1/build.pro
```

4.3 Version Compatibility Settings

Version compatibility allows OSVVM's settings to work like previous versions.

```
variable OsvvmVersionCompatibility $OsvvmVersion  ;# 2023.00
```

Generate failure on status NOCHECKS (when 1), otherwise, passed.

```
variable FailOnNoChecks 1                         ;# 0 ;# false
```

Use the old version of CreateClock and CreateReset procedures.

```
variable ClockResetVersion $OsvvmVersion          ;# 2023.00
```

4.4 Directory Structure and Results File Management

Base directory to contain other outputs.

```
    variable OutputBaseDirectory  ""        ;# "osvvm"
```

Subdirectories of the base directory.

```
    variable LogSubdirectory      "logs"    ;# logs in $OutputBaseDirectory/logs
    variable ReportsSubdirectory  "reports" ;# reports in $OutputBaseDirectory/reports
    variable ResultsSubdirectory  "results" ;# results in $OutputBaseDirectory/results
    variable CoverageSubdirectory "CodeCoverage" ;# Code Coverage in . . .
```

Formulation of Library Directory

```
    # Library Directory = [file join $VhdlLibraryParentDirectory \
    #           $OutputBaseDirectory $VhdlLibraryDirectory $VhdlLibrarySubdirectory]
    variable VhdlLibraryParentDirectory   ""          ;# "C:/tools"
    variable VhdlLibraryDirectory         "VHDL_LIBS"
    variable VhdlLibrarySubdirectory    "${ToolNameVersion}"      ;# default value
```

Directory to hold temporary files.   Typically same as OutputBaseDirectory.

```
    variable OsvvmTemporaryOutputDirectory   ""  ;# temporary files
```

4.5 Where to Put OsvvmScriptSettingsPkg_generated.vhd

Location to place OsvvmScriptSettingsPkg_generated.vhd

```
    variable SettingsAreRelativeToSimulationDirectory "false"
    variable OsvvmSettingsSubDirectory      ""
```

4.6 What Should Generate TCL Errors

```
    variable FailOnBuildErrors        "true"
    variable FailOnReportErrors       "false"
    variable FailOnTestCaseErrors     "false"
```

4.7 Stop Build on Number of Analyze and Simulate Errors

Stop build when AnalyzeErrorStopCount and SimulteErrorStopCount errors have occurred.   If the value is 0, then do not stop.   Do not stop is appropriate for regression runs.

```
    variable AnalyzeErrorStopCount       0
    variable SimulateErrorStopCount      0
```

4.8 Type of Transcripts to Generate

Generate HTML Transcript

```
    variable TranscriptExtension      "html"    ;# Generate log and html transcripts
```

Generate Simulation Script for entire run

```
    variable CreateSimScripts         "false"    ;# Create a simulator script
```

Generate Log file with just OSVVM Output

```
    variable CreateOsvvmOutput        "false"    ;# log file with just OSVVM output
```

4.9 VHDL Simulation Settings

```
    # Valid VHDL Version values 1993, 2002, 2008, 2019
    variable DefaultVHDLVersion   "2008" ; # OSVVM requires > 2008.
    variable SimulateTimeUnits      "ps"
    variable DefaultLibraryName     "DefaultLib"
```

4.10 Default Coverage Options

```
    variable CoverageEnable           "true"
    variable CoverageAnalyzeEnable    "false"
    variable CoverageSimulateEnable   "false"
    variable CoverageAnalyzeOptions   [vendor_SetCoverageAnalyzeDefaults]
    variable CoverageSimulateOptions  [vendor_SetCoverageSimulateDefaults]
```

4.11 Default Simulation Controls

```
    variable SimulateInteractive "false"
    variable DebugIsSet          "false"
    variable Debug               "false"
    variable LogSignalsIsSet     "false"
    variable LogSignals          "false"
    variable ScriptDebug         "false"
```

4.12 Extended Analyze and Simulate Options

```
    variable VhdlAnalyzeOptions        ""
    variable VerilogAnalyzeOptions     ""
    variable ExtendedAnalyzeOptions    ""  ;# "-quiet"
    variable ExtendedSimulateOptions   ""  ;# "-quiet"
```

4.13 GHDL and NVC Analyze and Simulate Options

```
    variable ExtendedElaborateOptions  ""
    variable ExtendedRunOptions        ""
    variable SaveWaves                 "false"
    variable SimulateInteractive       "false"
```

4.14 Name of Second Top Level

```
    variable SecondSimulationTopLevel ""
```

4.15 Variables Set by Simulator Scripts

```
        #  variable ToolType
        #  variable ToolVendor
        #  variable ToolName
        #  variable ToolNameVersion
        #  variable ToolArgs
        #  variable NoGui
        #  variable ToolSupportsGenericPackages
```

5 About the Author - Jim Lewis

Jim Lewis, the founder of SynthWorks, has thirty plus years of design, teaching, and problem solving experience. In addition to working as a Principal Trainer for SynthWorks, Mr Lewis has done ASIC and FPGA design, custom model development, and consulting.

Mr. Lewis is chair of the IEEE 1076 VHDL Working Group (VASG) and is the primary developer of the Open Source VHDL Verification Methodology (OSVVM.org) packages. Neither of these activities generate revenue.

Please support our volunteer efforts by buying your VHDL training from SynthWorks.

If you find bugs these packages or would like to request enhancements, you can reach me at jim@synthworks.com.