OSVVM Settings User Guide

User Guide for Release 2026.01

By

Jim Lewis

SynthWorks VHDL Training

Jim@SynthWorks.com

http://www.SynthWorks.com

Table of Contents

1 OSVVM Settings Overview

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

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.

2 OSVVM Settings Directory for VHDL and Script Settings

The OSVVM Settings Directory for VHDL and Script settings is found in the directory specified by joining the Tcl variables $SettingsRootDirectory, $SettingsDirectory, and $::osvvm::OsvvmSettingsSubDirectory.

The SettingsRootDirectory is $OsvvmLibraries. $OsvvmLibraries is the location of your OsvvmLibraries install. $OsvvmLibraries is a Tcl variable set by the scripts when started with $OsvvmLibraries/Scripts/StartUp.tcl.

The SettingsDirectory is determined by the first of the following found:

- Environment variable OSVVM_SETTINGS_DIR - (highest precedence)
- Directory $OsvvmLibraries/../OsvvmSettings
- Directory $OsvvmLibraries/osvvm for vhdl settings and $OsvvmLibraries/Scripts for script settings - (lowest precedence)

The OSVVM_SETTINGS_DIR variable can contain either an absolute path or a relative path. If OSVVM_SETTINGS_DIR contains an absolute path, then the path of $OsvvmLibraries in the join is ignored.

$::osvvm::OsvvmSettingsSubDirectory is set by OsvvmSettingsLocal.tcl. As a result, this value is "" for scripts (since OsvvmSettingsLocal.tcl has not been called yet). If multiple simulators are being used, set OsvvmSettingsSubDirectory as follows:

```
set OsvvmSettingsSubDirectory ${ToolName}
```

3 OsvvmScriptSettingsPkg

OsvvmScriptSettingsPkg provides Tcl project settings to the VHDL code in the OSVVM Utility Library. It uses deferred constants that are declared in the package declaration OsvvmScriptSettingsPkg.vhd. Typically deferred constants get their value in the package body OsvvmScriptSettingsPkg_generated.vhd.

3.1 OsvvmScriptSettingsPkg_generated.vhd

The values in the generated package body are derived from the Tcl settings. Most ensure that the Tcl scripts and OSVVM Utility library are using the same file names and paths.

OSVVM_SETTINGS_REVISION is important. It is used in OsvvmSettingsPkg_default.vhd to pick between new and old (backward compatible) settings in the OSVVM Utility library. If you need to update the OSVVM libraries during a project, you can set the Tcl variable ::osvvm::OsvvmVersionCompatibility to the version of the OSVVM version when the project started to maintain compatible settings.

```
-- 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 ;
```

3.2 Package body options

If for some reason, these values are not appropriate, you may override them by creating a package named OsvvmScriptSettingsPkg_local.vhd in the Settings Directory. If the package OsvvmScriptSettingsPkg_generated.vhd could not be created (due to directory or file permissions), the package OsvvmScriptSettingsPkg_default.vhd (in OsvvmLibraries/osvvm) will be used.

4 OsvvmSettingsPkg

OsvvmSettingsPkg defines constants that used by the OSVVM Utility Library. Mostly these constants are used to allow changes in behavior to be made to the OSVVM utilities while supporting alternate settings to maintain backward compatibility.

These settings are important as improving the quality of the random seed hash functions will change the results produced by all tests cases that use randomization. Having these change late in a project when test cases are stable and producing required coverage would be bad. Having these change at the beginning of a new project has no impact at all.

Note many of these settings are conditional based on the constant OSVVM_SETTINGS_REVISION from the package OsvvmScriptSettingsPkg – which in turn generates this constant from the Tcl variable ::osvvm::OsvvmVersionCompatibility.

4.1 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 ;
```

4.2 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.

4.3 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, base name to be used with WriteScoreboardYaml, otherwise, the name is the exact file name – which is required to be in a particular format (with a proscribed prefix and suffix).

4.4 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 ;
```

4.5 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 ;
```

4.6 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 ;
```

5 Variables Set by Simulator Scripts

The following tcl variables are set by VendorScripts_<ScriptBaseName>.tcl. Where ScriptBaseName is a tcl variable that uniquely identifies the tool being run.

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

6 OsvvmSettingsLocal.tcl

OsvvmSettingsLocal.tcl is used to configure script settings. A template is provided in the OsvvmSettingsLocal_Example.tcl in directory OsvvmLibraries/Scripts. See the next section Settings Directory for where you should put OsvvmSettingsLocal.tcl.

6.1 Structure of OsvvmSettingsLocal.tcl

```
# Global Settings
namespace eval ::osvvm {
# OSVVM Name Space Settings
  #  Version Compatibility Settings
  #  VHDL File Viewer 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
}
```

6.2 Global Settings

Set variables to your project variables here.

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

This will allow you to build your project as:

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

6.3 Version Compatibility Settings (OSVVM_SETTINGS_REVISION)

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
```

6.4 VHDL File Viewer Settings

In the Test Case report, a link to the VHDL test case file is provided. Set variable VhdlFileViewerPrefix to "vscode://file/" to view the file in vscode. The default value, "", views the file in the html viewer.

```
    #  variable VhdlFileViewerPrefix      ""                 ; # Default
    #  variable VhdlFileViewerPrefix      "vscode://file/"   ; # view with vscode
```

6.5 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
```

6.6 Supporting Different VHDL Settings for Different Simulators

Setting OsvvmSettingsSubDirectory adds a subdirectory to the search path or generated file path for VHDL settings. For details see Settings Directory. A good setting for supporting multiple simulators is:

```
    variable OsvvmSettingsSubDirectory      $ToolName
```

If multiple simulators are not being used, the default value of "" is just fine.

6.7 Supporting Different Script Settings for Different Simulators

OsvvmSettingsSubDirectory cannot be used to select a simulator specific version of OsvvmSettingsLocal.tcl since OsvvmSettingsLocal.tcl sets OsvvmSettingsSubDirectory – a chicken and egg problem.

Instead, in addition to running OsvvmSettingsLocal.tcl (if it exists), StartUpShared.tcl will also run OsvvmSettingsLocal_<ScriptBaseName>.tcl (if it exists).

If different naming than <ScriptBaseName> is needed, then OsvvmSettingsLocal.tcl can include something like the following to run simulator specific settings:

```
    include OsvvmSettingsLocal_${ToolName}.tcl
```

Small customizations can be added to OsvvmSettingsLocal.tcl using conditionals such as the following:

```
        if {($ToolVendor eq "Siemens") || ($ToolVendor eq "Aldec")} {
          variable ExtendedAnalyzeOptions   "-quiet"
          variable ExtendedSimulateOptions  "-quiet"
        }
```

6.8 What Should Generate TCL Errors

```
    variable FailOnBuildErrors        "true"    ;# simulator command had errors
    variable FailOnReportErrors       "false"   ;# yaml reports caused html failure
    variable FailOnTestCaseErrors     "false"   ;# one or more test case(s) had errors
```

6.9 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
```

6.10 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
```

6.11 VHDL Simulation Settings

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

6.12 Default Coverage Options

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

6.13 Default Simulation Controls

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

6.14 Extended Analyze and Simulate Options

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

6.15 GHDL and NVC Analyze and Simulate Options

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

6.16 OSVVM Developer Controls

```
      variable TclDebug                    "false"  ;# print $::errorInfo on failures
      variable ReportDebug                 "false"  ;# also print on report failures
      variable OsvvmDevDeriveArchitectures  "false"  ;# Derive 2008 arch from 2019
```

6.17 Name of Second Top Level

```
    variable SecondSimulationTopLevel ""
```

6.18 Deprecated Settings

The setting SettingsAreRelativeToSimulationDirectory is deprecated and the only valid value is "false". The value will be "false" in StartUpShared, since OsvvmSettingsLocal.tcl has not run yet. As a result, if set to "true", VHDL settings will be in a different directory from the script settings. In a future revision of OSVVM, it is likely that setting this to "true" will result in a tcl error.

7 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.