Transcript Package User Guide

User Guide for Release 2015.01

By

Jim Lewis

SynthWorks VHDL Training

Jim@SynthWorks.com

http://www.SynthWorks.com

Table of Contents

1 TranscriptPkg Overview

TranscriptPkg allows different parts of a testbench to use a common transcript file. It does this by providing an internal file identifier (TranscriptFile), and subprograms for opening (TranscriptOpen) files, closing (TranscriptClose) files, printing (print and writeline), and checking if the file is open (IsTranscriptOpen).

2 TranscriptPkg Use Models

TranscriptPkg offers three printing methods, Print, WriteLine, and direct usage of TranscriptFile. If the TranscriptFile is open, Print and WriteLine write to the transcript file, otherwise, they write to std.textio.OUTPUT.

```
use osvvm.TranscriptPkg.all ;
architecture Test1 of tb is

  process
  begin
    TranscriptOpen("./results/Transcript_t1_nominal.txt") ;
    Print("Print 1") ;
    swrite(buf, "Write 1") ;
    writeline(buf) ;
    . . .
    TranscriptClose ;

    if IsTranscriptOpen then
      swrite(buf, "Write 1 - transcript open") ;
      writeline(TranscriptFile, buf) ;
    end if;
```

TranscriptPkg also supports a mirror mode in which Print and WriteLine write to both TranscriptFile and std.textio.OUTPUT. Entering mirror mode is via SetTranscriptMirror.

```
SetTranscriptMirror ;
SetAlertStopCount(ERROR, 20) ;
```

3 Usage by other OSVVM packages

Print and WriteLine are used by other OSVVM packages.

In CoveragePkg, WriteBin without a file specification uses IsTranscriptOpen to determine whether to use std.textio.OUTPUT or TranscriptFile.

In AlertLogPkg, alert, log, and ReportAlerts all use osvvm.TranscriptPkg.WriteLine.

4 Method Reference

4.1 Package Reference

Using TranscriptPkg requires the following package reference:

```
library osvvm ;
use osvvm.TranscriptPkg.all ;
```

4.2 Transcript File

TranscriptFile is an internal file to the declarative region of TranscriptPkg.

```
file TranscriptFile : text ;
```

4.3 TranscriptOpen

TranscriptOpen opens the TranscriptFile. When the transcript is open, alert, log, ReportAlerts, WriteLine, and print use the TranscriptFile, otherwise, they use OUTPUT.

```
procedure TranscriptOpen (Status: out FILE_OPEN_STATUS; ExternalName: STRING;
    OpenKind: FILE_OPEN_KIND := WRITE_MODE) ;
procedure TranscriptOpen (ExternalName: STRING;
    OpenKind: FILE_OPEN_KIND := WRITE_MODE) ;
impure function  TranscriptOpen (ExternalName: STRING;
    OpenKind: FILE_OPEN_KIND := WRITE_MODE) return FILE_OPEN_STATUS ;
. . .
--   file open status,   File name,       file open kind
TranscriptOpen(Status,   "./Results.txt", WRITE_MODE) ;
TranscriptOpen(          "./Results.txt", WRITE_MODE) ;
```

WRITE_MODE is the default. If READ_MODE is used, an error is generated.

Usage of the function form allows a file to be opened in a declarative region (such as in a package).

```
Constant OPENED_RESULTS : FILE_OPEN_STATUS :=
      TranscriptOpen("./Results.txt", WRITE_MODE) ;
```

4.4 TranscriptClose

TranscriptClose opens the TranscriptFile.

```
  procedure TranscriptClose ;
```

4.5 IsTranscriptOpen

IsTranscriptOpen returns true when the TranscriptFile is open.

```
impure function IsTranscriptOpen return boolean ;
. . .
if IsTranscriptOpen then
  swrite(buf, "Write 1 - transcript open") ;
  writeline(TranscriptFile, buf) ;
end if;
```

4.6 WriteLine

Writeline writes the buffer to the TranscriptFile when it is open, otherwise, to OUTPUT.

```
procedure WriteLine(buf : inout line)  ;
. . .
swrite(buf, "A Message") ;
writeline(buf) ;
```

4.7 Print

Print writes the string to the TranscriptFile if it is open, otherwise, to OUTPUT.

```
procedure Print(s : string) ;
. . .
Print("Another Message") ;
```

4.8 SetTranscriptMirror

SetTranscriptMirror sets mirror mode in which Print and WriteLine write to both TranscriptFile and std.textio.OUTPUT.

```
procedure SetTranscriptMirror (A : boolean := TRUE) ;
. . .
SetTranscriptMirror ;
```

4.9 IsTranscriptMirrored

IsTranscriptMirrored returns true when the transcript is mirrored.

```
impure function  IsTranscriptMirrored return boolean ;
. . .
if IsTranscriptMirrored then
  . . .
```

5 Compiling TranscriptPkg and Friends

Use of AlertLogPkg requires use NamePkg, OsvvmGlobalPkg, and AlertLogBasePkg. The compile order is: NamePkg.vhd, OsvvmGlobalPkg.vhd, AlertLogBasePkg.vhd, and AlertLogPkg.vhd. Compiling the packages requires VHDL-2008.

6 About TranscriptPkg

TranscriptPkg was developed and is maintained by Jim Lewis of SynthWorks VHDL Training. It originated in conjunction with AlertLogPkg.

Please support our effort in supporting TranscriptPkg and OSVVM by purchasing your VHDL training from SynthWorks.

TranscriptPkg is released under the Perl Artistic open source license. It is free (both to download and use - there are no license fees). You can download it from http://www.synthworks.com/downloads. It will be updated from time to time. Currently there are numerous planned revisions.

If you add features to the package, please donate them back under the same license as candidates to be added to the standard version of the package. If you need features, be sure to contact us. I blog about the packages at http://www.synthworks.com/blog. We also support the OSVVM user community and blogs through http://www.osvvm.org.

Find any innovative usage for the package? Let us know, you can blog about it at osvvm.org.

7 Future Work

TranscriptPkg.vhd is a work in progress and will be updated from time to time.

Caution, undocumented items are experimental and may be removed in a future version.

8 About the Author - Jim Lewis

Jim Lewis, the founder of SynthWorks, has twenty-eight 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.