Interfacing Other Alert and Verbosity Filtering Utilities to OSVVM AlertLogPkg

For Release 2015.03

By

Jim Lewis

SynthWorks VHDL Training

Jim@SynthWorks.com

http://www.SynthWorks.com

Table of Contents

1 Overview

For details on using AlertLogPkg, see AlertLogPkg_user_guide.pdf.

This document is for those already using another package for alerts and verbosity control (such as BitVis Utility Library). In this case, what should you do with OSVVM alerts? Your choices are:

- Handle OSVVM Alerts from just like asserts from other packages
- Use AlertLogPkg to get a summary of OSVVM errors
- Interface AlertLogPkg to another package.

2 Handle OSVVM Alerts just like Asserts from other packages

The previous version of OSVVM used asserts rather than alerts. Hence, you can continue handling OSVVM errors using the same mechanism you used with previous versions of the packages and be no worse off.

One way to handle this is to use scripts to inspect transcript files for FAILURE, ERROR, or WARNING.

3 Use AlertLogPkg to get a summary of OSVVM Errors

If you use other mechanisms for error counting, you can use a minimal amount of AlertLogPkg, to retrieve a count of OSVVM alerts for a test using GetAlertCount. An abbreviated command reference for GetAlertCount is below.

3.1 Package References

Using AlertLogPkg requires the following package references:

```
library osvvm ;
use osvvm.OsvvmGlobalPkg.all ;
use osvvm.AlertLogPkg.all ;
```

3.2 AlertType

Alert levels can be FAILURE, ERROR, or WARNING.

```
type AlertType is (FAILURE, ERROR, WARNING) ;
```

3.3 AlertCountType

Alerts are stored as a value of AlertCountType.

```
subtype AlertIndexType is AlertType range FAILURE to WARNING ;
type    AlertCountType is array (AlertIndexType) of integer ;
```

CAUTION: When working with values of AlertCountType, be sure to use named association as the type ordering may change in the future.

3.4 GetAlertCount

GetAlertCount returns the AlertCount value at AlertLogID. GetAlertCount is overloaded to return either AlertCountType or integer.

```
impure function GetAlertCount(AlertLogID : AlertLogIDType := ALERTLOG_BASE_ID)
    return AlertCountType ;
impure function GetAlertCount(AlertLogID : AlertLogIDType := ALERTLOG_BASE_ID)
    return integer ;
. . .
TopTotalErrors  := GetAlertCount ;          -- AlertCount for OSVVM
```

4 Interface AlertLogPkg to another package

This step is a little more complicated than you probably want to use. This step involves replacing the package body of AlertLogPkg with one that connects to your package. Since AlertLogPkg will change in the future, your package body will need to also.

Currently, the only subprograms of AlertLogPkg that are used by OSVVM are Alert, Log, and IsLoggingEnabled. The following section describes how to edit the package body.

4.1 Alert

In the procedure body for AlertLogPkg, the implementation of Alert shown below needs to be replaced with code that calls the desired package.

```
procedure alert(
  AlertLogID   : AlertLogIDType ;
  Message      : string  ;
  Level        : AlertType := ERROR
) is
begin
  AlertLogStruct.Alert(AlertLogID, Message, Level) ;
end procedure alert ;
```

Change all other calls to AlertLogStruct.Alert to a call to Alert above.

4.2 Log

In the procedure body for AlertLogPkg, the implementation of Log shown below needs to be replaced with code that calls the desired package.

```
procedure log(
  AlertLogID   : AlertLogIDType ;
  Message      : string ;
  Level        : LogType := ALWAYS
) is
begin
  AlertLogStruct.Log(AlertLogID, Message, Level) ;
end procedure log ;
```

Change all other calls to AlertLogStruct.Log to a call to Log above.

4.3 IsLoggingEnabled

In the procedure body for AlertLogPkg, the implementation of IsLoggingEnabled shown below needs to be replaced with code that calls the desired package.

```
impure function IsLoggingEnabled(
    AlertLogID : AlertLogIDType ; Level : LogType
) return boolean is
begin
  return AlertLogStruct.IsLoggingEnabled(AlertLogID, Level) ;
end function IsLoggingEnabled ;
```

Change all other calls to AlertLogStruct.IsLoggingEnabled to a call to IsLoggingEnabled above.

4.4 Remove Protected Type and Shared Variable

Remove the protected type declaration and body for AlertLogStructPType. Remove the shared variable declaration AlertLogStruct.

4.5 Compile your replacement package body for AlertLogPkg

Compile your newly created package body for AlertLogPkg into your OSVVM library. Now all OSVVM alerts will be reported through your verbosity control system.

5 About AlertLogPkg

AlertLogPkg is a living package and will be updated from time to time. It is part of the OSVVM library. It is available through http://osvvm.org and http://www.synthworks.com/downloads.