VendorCovApiPkg Package

User Guide

User Guide for Release 2016.11

By

Jim Lewis

SynthWorks VHDL Training

Jim@SynthWorks.com

http://www.SynthWorks.com

Table of Contents

1 VendorCovApiPkg Overview

VendorCovApiPkg provides an interface between the creation of coverage and recording of coverage in OSVVM CoveragePkg and a simulator.

2 Type Definitions

2.1 VendorCovHandleType

VendorCovHandleType is the type for the handle that OSVVM uses to identify the coverage object to the tool. The tool will set the value for the handle with a call to either function VendorCovPointCreate or VendorCovCrossCreate.

```
subtype VendorCovHandleType is integer;
```

The type may be changed to any valid VHDL type.

2.2 Coverage Bins

A coverage bin value is represented by an integer range stored in a record. The type used for the interface is VendorCovRangeArrayType which is an array of the VendorCovRangeType.

```
type VendorCovRangeType is record
  min: integer;
  max: integer;
end record;

type VendorCovRangeArrayType is array ( integer range <> ) of VendorCovRangeType;
```

3 Creating a Point/Item Coverage bin

The function VendorCovPointCreate creates the initial data structure for a point coverage bin and returns the handle to the bin.

```
impure function VendorCovPointCreate( name: string ) return VendorCovHandleType;
```

4 Creating a Cross Coverage bin

The function VendorCovCrossCreate creates the initial data structure for a cross coverage bin and returns the handle to the bin.

```
impure function VendorCovCrossCreate( name: string ) return VendorCovHandleType;
```

5 Updating a Coverage Bin name

OSVVM Functional Coverage modeling allows the bin to be named after the initial model is created. The procedure VendorCovSetName allows the bin name to be revised after the coverage model has already been created.

```
impure function VendorCovCrossCreate( name: string ) return VendorCovHandleType;
```

6 Adding Coverage Items to a Bin

For either point or cross coverage, bins are added to the model using VendorCovBinAdd.

```
procedure VendorCovBinAdd(
  obj      : VendorCovHandleType;
  bins     : VendorCovRangeArrayType;
  Action   : integer;
  atleast  : integer;
  name     : string
);
```

The types VendorCovHandleType and VendorCovRangeArrayType are explained in Type Definitions of this document. Action is 1 for a count bin, 0 for an ignore bin, and -1 for an illegal bin. The atleast parameter is the coverage goal. Note that the coverage goal can be different for different bins - this makes an OSVVM coverage bin more similar to a SystemVerilog/UCIS coverage group. Name is the string name of this bin.

7 Incrementing the Coverage

VendorCovBinInc increments the coverage of a bin. The value of index ranges from 1 to N where N is the number of coverage bins and is based on the order of capture of the coverage bins.

```
procedure VendorCovBinInc( obj: VendorCovHandleType; index: integer );
```

8 Binding the Procedures to the Tool.

Each of the functions and procedures need to resolve to a call to an item in the vendor tools. This can be either by calling the appropriate functions or procedures in the subprogram bodies (as demonstrated in VendorCovApiPkg.vhd) or by binding in foreign subprograms (as demonstrated in VendorCovApiPkg_Aldec.vhd).

9 Compiling VendorCovApiPkg and Friends

See OSVVM_release_notes.pdf for the current compilation directions. Rather than referencing individual packages, we recommend using the context declaration:

```
library OSVVM ;
  context osvvm.OsvvmContext ;
```

10 About VendorCovApiPkg

VendorCovApiPkg was developed and is maintained by Jim Lewis of SynthWorks VHDL Training. Prior to its release to OSVVM it was used in SynthWorks' VHDL classes.

Please support our effort in supporting the OSVVM library of packages by purchasing your VHDL training from SynthWorks.

VendorCovApiPkg 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 osvvm.org or from our development area on GitHub.

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.

11 Future Work

VendorCovApiPkg.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.

12 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.