HDL Obfuscator and Name Change Tool

Reference Guide

Technical Support:

Website:	https://sourceforge.net/projects/hdlobf/
Email:	vispicassod@hotmail.com

Trademark:

Verilog® is a registered trademark of Cadence Design Systems, Inc.
Windows® is a registered trademark of Microsoft Corporation.

Index:

1 Document history

Version	ID	Date	Change
1.0	--	09/28/2004	Initial version
1.1		11/07/2004	VHDL support added

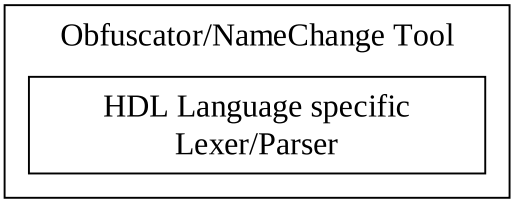
2 Introduction

HDLObf is intended to be a HDL Obfuscator and identifier name change utility. Primarily designed for Verilog/SystemVerilog and initial support for VHDL. Feature to obfuscate based on the scope of the identifier will be added in feature release. This will allow modules/entity not supplied to the obfuscator or name change to be left alone.

System requirements JDK 1.4, **ANTLR 2.7.4** (antlr.jar in classpath). Tested with window/cygwin environment.

3 Architecture

The tool uses ANTLR tool to generate lexer/parser. **ANTLR**, ANother Tool for Language Recognition, (formerly PCCTS) is a language tool that provides a framework for constructing recognizers, compilers, and translators from grammatical descriptions containing Java, C#, or C++ actions. ANTLR provides excellent support for tree construction, tree walking, and translation.
The language specific lexer and parser are used in a general purpose obfuscator / Name Change Tool


Obfuscator/NameChange Tool
HDL Language specific Lexer/Parser

4 Usage

The usage options are

```
java HDLObf  (GUI mode)
java NameChange <Language Name> <Map file> <input file name> <output file name>
java Obfuscator     <Language Name> <Map file> <input file name> <output file name>
```

Operation: supported values **NC** (name change) or **OB** (obfuscate)
Map file: file name containing name mapping.
Language Name: support values **ver**, **sv** (system verilog), **vhd** and **sc** (systemc)

The map file is updated every time Obfuscator is used this will help running Obfuscator in batch mode or for back tracking. By keeping the map file same and repeating the command for different files will yield identifier map reuse in batch mode.

4.1 Common problems:

4.1.1 Preserving module/entity names

If there is a need to preserve a particular entity/module name during obfuscation please make sure that you have that module name and its port entry in the map file, e.g.:
MyModule=MyModule
MyModuleport1= MyModuleport1
MyModuleport2= MyModuleport2
A preserve module and preserve non-declared module feature is planned for the next release.

4.1.2 No output file generated

If there is a message “Could not create output file: <filename>” please check the following:

1. Make sure the **path exists** and is not read-only**.**
2. Make sure the **file dose not exist** in the path. By default the program will not overwrite and existing file.

For any other problems please send me a example if you can or give me as much detail as you can.

4.1.3 Class not found

If you receive something like this:
Exception in thread "main" java.lang.NoClassDefFoundError: <classname>
Make sure you have the <HDLObf_HOME>/bin directory and antlar.jar in your classpath. For confirmation use echo %CLASSPATH% on windows, if you are using linux/unix you would know what to do.

4.2 VHDL Feature Special Note:

When using VHDL feature the IEEE or the STD library are not built in to the lexer/parser. To avoid ieee, std functions and declarations to be obfuscated please append the map file vhdl_map_ieee_std.map.dat located in <HDLObf_HOME>/src and/or <HDLObf_HOME>/test.

5 Planned Features

(1) Scope based identifier name modification support.
   a. To better obfuscate the code.
   b. Ability to have modules that can be excluded form obfuscation (automatically if not included in the list).

(2) Multiple files loading instead of one by one, like a –f option to have a file with list of file names and output directory.

  a. Improves performance, as the map needs to be loaded only once.
  b. Required to implement feature (1).