0.00.5em

0.0.00.5em

0.0.0.00.5em

0em

RESSL Documentation

Release 0.3

Bryan Morris, Jeff McNeal

March 25, 2015

CONTENTS

This document contains the user guide, installation guide and migration guide for the RESSL system.

CHAPTER

ONE

INTRODUCTION

RESSL is an acronym that stands for (Read Evaluate Start Sequence Loop) and is a system inspired by the Lisp REPL (Read Evaluate Print Loop) providing an iterative environment in SystemVerilog to launch UVM-based sequences.

It includes a command line interpreter (Read-Evalue) and an API to create an start UVM sequences.

More details are provided in the paper “RESSL UVM Sequences to the Mat” presented at the SNUG Conference in March 2015 in San Jose.

CHAPTER

TWO

INSTALLATION GUIDE

This document outlines the requirements and installation for the RESSL system.

The last section of this document provides details on how you can run a "RESSLized" version of the UVM `ubus` example provided in UVM 1.2 library.

2.1 Requirements

RESSL has the following system requirements:

- GNU `readline`: The RESSL front-end uses the GNU `readline` library that provides command line editing, completion, and history. Most Linux distributions include the libreadline library. If the library isn't present, you'll need to install it. Any recent version will be sufficient.
- `svlib` (version 0.4): Verilab's svlib library is needed for some of the processing available in the String.
- `ressl` (version 0.3): Verilab's RESSL library.
- UVM 1.2: Universal Verification Methodology. This can be downloaded from Accellera. We provide a Unix `patch` file for a *pristine* version of the library.

Optionally, the User Guide is written using ReStructured Text and generated with the Python ‘‘Sphinx‘‘_ documentation system. You'll need to download and install the ‘‘Sphinx‘‘_ system, and LaTeX distribution for your system if you want to generate a PDF version of the documentation.

2.2 Step 1. Extract ressl-v0.3.tar.gz

First ensure that the `ressl-v0.3.tar.gz` for is correct and complete by comparing the MD5 checksum provided at the Verilab web page where you downloaded this package.

```
# md5sum ressl-v0.3.tar.gz
```

The `ressl-v0.3.tar.gz` contains the required `svlib`, UVM 1.2 (updated to work with `RESSL`),‘‘ressl‘‘ libraries and a copy of the `README.md` file.

Unzip and untar the `ressl-v0.3.tar.gz` file into a temporary location:

```
# gunzip ressl-v0.3.tar.gz
# tar xvf ressl-v0.3.tar
```

This extracts the following files:

- `svlib.tar`
- `ressl.tar`

- uvm-1.2.tar

2.3 Step 2. Installing svlib

Untar the svlib.tar into a directory of your choice.

2.4 Step 3. Installing ressl

Untar the ressl.tar into a directory of your choice.

2.5 Step 4. Installing and Patching UVM

We have provided a pristine UVM 1.2 tarball that you can patch with our updates created for RESSL. The main update adds object introspection for any field defined with a uvm_field_* macro.

Untar the uvm-1.2.tar into a directory of your choice (for the remainder of let this document be called <UVM_ROOT_DIR>).

The ressl directory contains a sub-directory patch which contains the following files:

uvm-1.2-object.patch a unix patch file that provides the object introspection and UVM field macro updates required for RESSL to get/set fields in a sequence or object.

uvm-1.2-ubus.patch a unix patch file that updates the UVM 1.2 ubus example to launch the RESSL system as an example and demo of RESSL.

uvm-1.2.patch a unix patch file that aggregates the previous two patch files.

All these patches must be run from the <UVM_ROOT_DIR>. As above, let <UVM_ROOT_DIR> represent the root directory. However, the patches are found in the RESSL directory you installed in step 3 (<RESSL_ROOT_DIR>). For the examples below, let's assume that you've installed RESSL in the directory ~/lib/ressl, and the UVM 1.2 directory in ~/lib/uvm-1.2.

The patch commands to add the object introspection and update the ubus example is:

```
# cd ~/lib/uvm1.2
# patch -p1 < ~/lib/ressl/uvm-1.2.patch
```

***Please* remember that the *patch* command uses an input redirection < to** pull in the patch file.

To add the object introspection only:

```
# cd ~/lib/uvm1.2
# patch -p1 < ~/lib/ressl/uvm-1.2-object.patch
```

Or to update the ubus examples only:

```
# cd ~/lib/uvm1.2
# patch -p1 < ~/lib/ressl/uvm-1.2-ubus.patch
```

2.6 Step 5. Running the demo

To run the ubus demo, change into the <UVM_ROOT_DIR>/examples/integrated/ubus/examples directory and re-build the example. This demo has only been tested with the Syynopsys VCS.

There are two environment variables that must be defined prior compiling the demo:

RESSL_ROOT The root directory where you installed the RESSL library (i.e., the contents of the ressl.tar extracted in a previous step).

SVLIB_ROOT The root directory where you installed the SVLIB library (i.e., the contents of the svlib.tar extracted in a previous step).

To run the demo you can either:

- Set the RESSL_ROOT and SVLIB_ROOT environment variables on the command line, and run the make command to compile and run the demo:

```
# make -f Makefile.vcs comp run \
  RESSL_ROOT=/home/bmorris/proj/ressl \
  SVLIB_ROOT=/home/bmorris/proj/svlib
```

- Or set the same variables as environment variables:

```
# export RESSL_ROOT=/home/bmorris/proj/ressl
# export SVLIB_ROOT=/home/bmorris/proj/svlib
# make -f Makefile.vcs comp run
```

CHAPTER

THREE

MIGRATION GUIDE

This document outlines how you can integrate RESSL into your environment.

See Also:

The InstallationGuide.rst to install the RESSL system and its dependent libraries.

CHAPTER

FOUR

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*