

# VectorBlox MXP Quickstart for MXP Simulator

March 13, 2018 VectorBlox

# **Contents**

1	Installation		
	1.1	Prerequisites	3
	1.2	Installing	3
2	Soft	ftware	
	2.1	Running the software	2

### 1 Installation

This guide describes installing VectorBlox MXP Simulation Library to run on an x86\_64 host compiling with GCC

#### 1.1 Prerequisites

Before you begin, make sure you have:

- GCC installed. Your native compiler should work on any linux workstation. On Windows you want to install mingw-w64 (Note that this is not to be confused with mingw from mingw.org, though you may want to download some utilities from that project such as GNU Make)
- (Windows Only) A unix-like environment we recommend msys from mingw. But make sure that the gcc command still points to the compiler described in the point above.

#### 1.2 Installing

- There is no need to install the simulator, just remember where it is when you link to the library. This is done automatically in the example makefiles.
- To use the License File set the environment variable export VECTORBLOX\_SIM\_LICENSE=/path/to/license

## 2 Software

This section describes how to compile and run software when linked to the simulator.

The string **EXAMPLES** refers to the examples subdirectory of the extracted release.

#### 2.1 Running the software

#### 1. Make the executable

Navigate to a test application such as vbw\_vec\_add\_t, located in EXAMPLES/software/bmark/vbw\_vec\_add\_t.

```
make clean_all all SIMULATOR=true
```

This command builds the vbxapi and vbxsim libraries, the test.c, and then links them.

Note that we used the <code>clean\_all</code> target first to ensure that any libraries that might have been compiled for a different target in the past were cleaned and re-compiled. If you know that the libraries were already compiled for the correct target, you can omit the <code>clean\_all</code> target.

#### 2. Execute the generated program

In order to execute the program that you just created make sure the environment variable  $\mbox{\sc blox\_SIM\_LICENSE}$  points to the correct path, and it is exported (child processes will inherit the value). Then run the program with the command ./test.elf