

# HOW TO: Use the tester package.

Patrick Chabrier

patrick.chabrier@toulouse.inra.fr

May 13, 2011

## Abstract

The "tester" package provide a simple solution to build simulation test for Difference Equation models. It is well integrated inside the VLE framework, and it does enable to provide test by inserting a dedicated model inside a simulator. No programing is needed.

## 1 General explanation

The package provide two things : a model and an application. The model called DETester can be inserted inside a vpz simulator. It need also to be connected to a Difference Equation model. As soon, as this is done, and when running the simulator, each time the tester model receive an event, it does store it by time, date an value. This is true when the DETester model is used in the "store" mode. But when the model is used in the "check" mode, what was previously stored is now compared to what is received. When the comparison fails, the simulation fails as well.

In order to make the boost::test usable inside a simulation, a tester command is provided.

As it is explained below, realizing a test, is in fact modelling a simulator with a test inside. And creating a list of test is completing a list of test inside a CMakeList.txt.

## 2 using Tester

**Step by step, a test simulator:** The first step consist in designing a test simulator.

- The first thing to do is of course to install the tester package like all the other one.
- Then you need to add a model to the simulator you want to test. The name of the package where it is defined is tester and the model you want to use is DETester.
- Now you can connect a DE from your simulator to your new DETester Model
- Configuring the DETester model is the next step, three experimental condition ports are available:
  - file (string)(mandatory): the file name you want to use to store the datas of the test (file will be stored in the /data folder of the package).
  - action (string)(mandatory): the action you want to do is either "store" or "check".
  - tolerance (double)(optional): the tolerance we accept in the value comparison, the default value is set to 0.0000000001.
- Launch the simulator, this will register the datas in the /data folder of the package.
- Change the value of the port action to "check". This what you want to do next time a test is launched. (Warning : if action is set to "store" when you run the test it won't test the values but will tell you it did so take care to keep your testing vpz with action set to "check" after registering the reference data)

You can, just by changing the parameters of the model build as many test as you want.

**step by step, configuring the package you want to test:** Now, we have many test simulators stored where ever you want, but we need to mention to the package that they are test. The next step consists in adding the test inside the dedicated list.

- Inside the CMakeLists.txt (the one in package base folder), you need to add the program\_options component inside the list of BOOST component.
- In order to initialize the directory of test of your package you can copy the contents of the test directory of the tester package inside the test directory of the package you want to test.
- Now you want to complete the list of test by editing the CMakeList.txt (the one in the /test folder you just copied) and adding line like this at the end of the file :

```
ADD_TEST(test_0 tester --package theNameOfYourPackage --vpz theNameOfYourVPZ.vpz)
ADD_TEST(test_1 tester --package theNameOfYourPackage --vpz theNameOfYourVPZ.vpz)
...
```

where test\_n is the name of the test, tester the name of the command that launch the test.

- Finally you can run the test like this :

```
> vle -P theNameOfYourPackage test
```

If you want to analyze an encountered problem just go to the /test directory of your package and run the test like that :

```
> ./tester --package theNameOfYourPackage --vpz theNameOfYourVPZ.vpz
```

**Note:** that the tester package has as example a test using the wwdm package. And you can launch it like this:

```
> vle -P tester configure build test
```

**Note:** also that you can study the directory /exp and /test of the tester package to have a clearer idea on how the tester package can be used.