

# Tunit.xml

From TinyOS Wiki

The tunit.xml file is the file responsible for telling TUnit which physical hardware platforms are attached to the computer. This includes information about which combinations of hardware to test, how to program them, how to talk to them, and any special compile flags necessary for that particular platform or test run.

TUnit comes with an example tunit.xml file with the name tunit\_example.xml ([http://tinysos.cvs.sourceforge.net/\\*checkout\\*/tinysos/tinysos-2.x-contrib/tunit/tunit\\_example.xml?content-type=text%2Fplain](http://tinysos.cvs.sourceforge.net/*checkout*/tinysos/tinysos-2.x-contrib/tunit/tunit_example.xml?content-type=text%2Fplain)) . Developers must create their own tunit.xml file to describe their hardware connected to the computer.

Each time the hardware connected to the computer is changed, the tunit.xml file must be updated to reflect the latest hardware.

## Contents

- 1 Structure
  - 1.1 Test Runs
  - 1.2 Motes
  - 1.3 Examples
    - 1.3.1 Single Nodes
    - 1.3.2 Multiple Nodes
    - 1.3.3 Differing Compile Time Options
- 2 See Also

## Structure

### Test Runs

The tunit.xml file is divided up into several Test Runs ([http://docs.tinysos.net/index.php/Test\\_Run](http://docs.tinysos.net/index.php/Test_Run)) containing one or more mote hardware platforms to test. Each Test Run is given a name which is used to identify tests that failed.

```
<tunit>
  <testrun name="tmote">
    ...
  </testrun>
</tunit>
```

## Motes

Inside of each Test Run, individual hardware is identified within the *mote* tags. The mote tag may take several arguments, some of which are optional:

```
<mote target="telosb" motecom="serial@COM6:tmote" installextras="bsl,5" builddextras="lpl" />
```

**target** is the name of the platform used at compile time, like *telosb* or *iris* or *micaz*

```

motecom describes how to connect and talk to the mote using serial forwarder
installextras are optionally anything else to include in the make command for installing the program to the node
buildextras are optionally anything else to always include in the make command for compiling the program

```

The above `<mote ... />` line would translate into the equivalent command-line commands:

```

make telosb lpl
make telosb reinstall.0 bsl,5
[serial forwarder] -comm serial@COM6:tmote -port 9000

```

## Examples

### Single Nodes

Setup a test run for a telosb or tmote platform currently connected to your computer:

```

<testrun name="single_telosb">
  <mote target="telosb" motecom="serial@COM6:tmote" installextras="bsl,5" />
</testrun>

```

Setup a test run for a micaz platform:

```

<testrun name="single_micaz">
  <mote target="micaz" motecom="serial@COM11:micaz" installextras="mib510,/dev/ttyS9" />
</testrun>

```

### Multiple Nodes

Run tests that apply to two telosb's. The application will compile a single time since the two platforms and compile time options do not differ.

```

<testrun name="single_micaz">
  <mote target="telosb" motecom="serial@COM6:tmote" installextras="bsl,5" />
  <mote target="telosb" motecom="serial@COM7:tmote" installextras="bsl,6" />
</testrun>

```

Equivalent command-line commands TUnit uses for compiling and installing test applications for this test run:

```

$ make telosb
$ make telosb reinstall.0 bsl,5
$ make telosb reinstall.1 bsl,6
$ [serial forwarder] -comm serial@COM6:tmote -port 9000
$ [serial forwarder] -comm serial@COM7:tmote -port 9001

```

Run tests that apply to both a telosb and a micaz. The application will be compiled once for each different platform, for a total of 2 compilations. The telosb, being the first mote entry of the test run, is the Driving Node of the test:

```

<testrun name="single_micaz">
  <mote target="telosb" motecom="serial@COM6:tmote" installextras="bsl,5" />
  <mote target="micaz" motecom="serial@COM11:micaz" installextras="mib510,/dev/ttyS9" />
</testrun>

```

Equivalent command-line commands TUnit uses for compiling and installing test applications for this test run:

```

$ make telosb

```

```
$ make telosb reinstall.0 bsl,5
$ make micaz
$ make micaz reinstall.1 mib510,/dev/ttyS9
$ [serial forwarder] -comm serial@COM6:tmote -port 9000
$ [serial forwarder] -comm serial@COM11:micaz -port 9001
```

## Differing Compile Time Options

Run a test with two telosb's, but include an extra compile flag for one of the platforms. Because the compile-time options differ, the application will be compiled once for each platform:

```
<testrun name="single_micaz">
  <mote target="telosb" motecom="serial@COM6:tmote" installextas="bsl,5" />
  <mote target="telosb" motecom="serial@COM7:tmote" installextas="bsl,6" builddextas="lpl" />
</testrun>
```

Equivalent command-line commands TUnit uses for compiling and installing test applications for this test run:

```
$ make telosb
$ make telosb reinstall.0 bsl,5
$ make telosb lpl
$ make telosb reinstall.1 bsl,6
$ [serial forwarder] -comm serial@COM6:tmote -port 9000
$ [serial forwarder] -comm serial@COM7:tmote -port 9001
```

## See Also

- TUnit
- Setting up TUnit
- TUnit Test Flow
- Flow of TUnit Java Execution

Retrieved from "<http://tinyos.stanford.edu/tinyos-wiki/index.php?title=Tunit.xml&oldid=551>"

- 
- This page was last modified on 15 January 2008, at 08:03.
  - This page has been accessed 9,909 times.