

# UnitTest Framework - Signal Handling

More efficient handling of control-C during a test run is provided by The `-c/--catch` command-line option to `unittest`, along with the **`catchbreak`** parameter. With catch break behavior enabled, control-C will allow the currently running test to complete, and the test run will then end and report all the results so far. A second control-c will raise a `KeyboardInterrupt` in the usual way.

If the `unittest` handler is called but `signal.SIGINT` handler isn't installed, then it calls for the default handler. This will normally be the expected behavior by code that replaces an installed handler and delegates to it. For individual tests that need `unittest` control-c handling disabled, the `removeHandler()` decorator can be used.

The following utility functions enable control-c handling functionality within test frameworks –

## `unittest.installHandler()`

Install the control-c handler. When a **`signal.SIGINT`** is received all registered results have `TestResult.stop()` called.

## `unittest.registerResult(result)`

Register a **`TestResult`** object for control-c handling. Registering a result stores a weak reference to it, so it doesn't prevent the result from being garbage collected.

## `unittest.removeResult(result)`

Remove a registered result. Once a result has been removed then `TestResult.stop()` will no longer be called on that result object in response to a control-c.

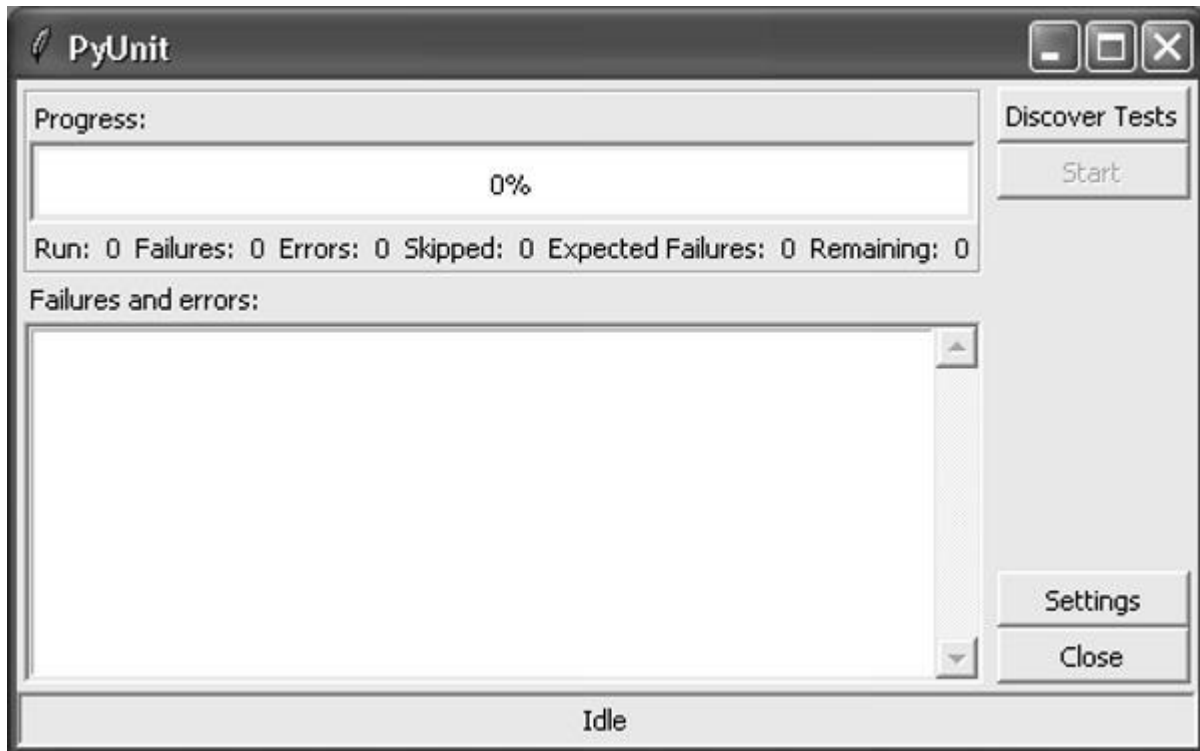
## `unittest.removeHandler(function = None)`

When called without arguments, this function removes the control-c handler if it has been installed. This function can also be used as a test decorator to temporarily remove the handler whilst the test is being executed.

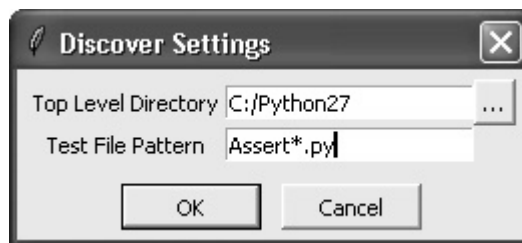
## GUI Test Runner

The `unittest` module is installed to discover and run tests interactively. This utility, a Python script 'inittestgui.py' uses Tkinter module which is a Python port for TK graphics tool kit. It gives an easy to use GUI for discovery and running tests.

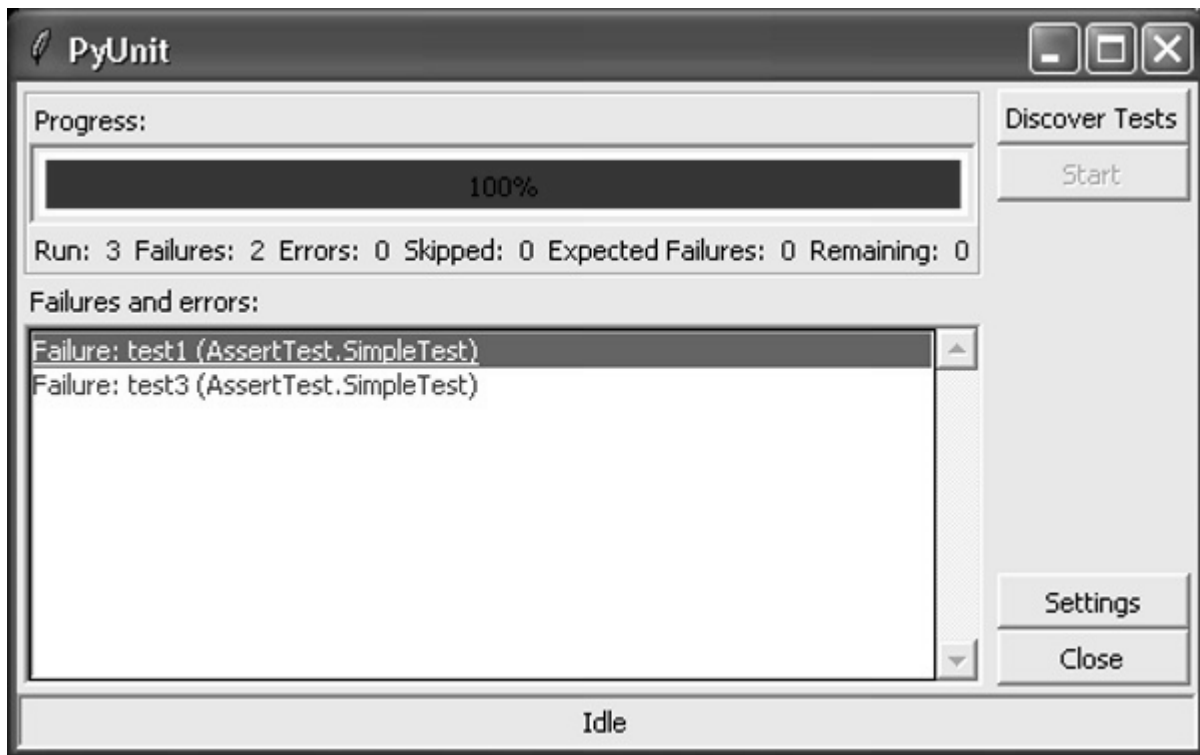
Python `inittestgui.py`



Click the 'Discover Tests' button. A small dialog box appears where you can select directory and modules from which test are to be run.



Finally, click the start button. Tests will be discovered from the selected path and module names, and the result pane will display the results.



In order to see the details of individual test, select and click on test in the result box –



If you do not find this utility in the Python installation, you can obtain it from the project page <http://pyunit.sourceforge.net/> .

Similar, utility based on wxpython toolkit is also available there.