Simple Use Case: TestCase Object

Create a TestCase object with desired dests, add your TestCase to the TestRunner object, and run the test:

```
//set up a test case:
var oTestCase = new YAHOO.tool.TestCase({
  name: "Simple Math",
  testEquality: function () {
    YAHOO.util.Assert.areEqual(4, (2+2), "2+2=4");
  };
});

//add the test case to the TestRunner:
YAHOO.tool.TestRunner.add(oTestCase);

YAHOO.tool.TestRunner.run(); //run the test
```

Key Members of the TestCase Object

name	The name of the TestCase. This will be visible in the logging of TestCase events.
test*name*	A function that tests functional code via one or more assertions; name must begin with the string "test". A TestCase can have one or more test members.
setUp	Method that prepares your test case to run by, for example, creating needed data constructs or objects.
tearDown	Method that nulls out variables in use by the TestCase, detaches event handlers, etc.
_should	Special object that provides granular configuration of the test case. It can have the following members:

	ignore	A name:value pair consisting of a testname and Boolean indicating whether to ignore the test (e.g.: `ignore: {testOne : true /*ignore testOne*/}.`
	error	A name:value pair consisting of a testname and Boolean indicating whether the test is should fail: `error: {testTwo : true /*testTwo should fail*/}` Or, a specific error string can substitute for the Boolean: `error: {testTwo : "Expected string." /*testTwo should fail with this specific error message*/}.`

All TestCase configurations should be passed into the TestCase constructor as an object literal; see Simple Use Case.

Key Events in YUITest

See online docs for full list of custom events, including TestSuite- and TestRunner-level events.

All YUITest events are subscribed to at the TestRunner level; e.g.:
`YAHOO.tool.TestRunner.subscribe(YAHOO.tool.TestRunner.TEST_FAIL_EVENT, myFn);`

Test-level Events

Event:	Fires:
TEST_PASS_EVENT	When an individual test passes.
TEST_FAIL_EVENT	When an individual test fails.

Argument Data Object for Test-level Events:

type	Type of event.
testCase	The testCase object to which this test belongs.
testName	The string name of this test.

TestCase-level Events

Event:	Fires:
TEST_CASE_BEGIN_EVENT	Before the TestCase is run.
TEST_CASE_COMPLETE_EVENT	After the TestCase is run.

Argument Data Object for TestCase-level Events:

type	Type of event.		
testCase	Current TestCase instance.		
results (TEST_CASE_END event only)	**passed**	Number of tests that passed.	
	failed	Number of tests that failed.	
	testname	**result**	*pass* or *fail*.
		message	String returned by the test.

Note: TestSuite- and TestRunner-level events are also available, containing summary data in addition to specific TestCase results objects. See online docs for full details.

Assertions

Assertions are accessed via `YAHOO.util.Assert`

Equality assertions:
areEqual(expected, actual)
areNotEqual(expected, actual)
equivalent to == test

Sameness assertions:
areSame(expected, actual)
areNotSame(expected, actual)
equivalent to === test

Data-type assertions:
isArray(arg)
isBoolean(arg)
isFunction(arg)
isNumber(arg)
isObject(arg) *object and function return true*
isString(arg)

isTypeOf assertion:
isTypeOf(sType, sTest, sFailureMessage)
```
YAHOO.util.Assert.isTypeOf("string
    ", 5, "Expected string.");
    //fails
```

isInstanceOf assertion:
isInstanceOf(oConstructor, oTestObject, sFailureMessage)
```
YAHOO.util.Assert.isInstanceOf(Str
    ing, "Madrone", "Expected
    string."); //passes
```
can be used to test non-native objects, too

Special Value Assertions:
isFalse, isTrue, isNaN, isNotNaN, isNull, isNotNull, isUndefined, isNotUndefined
```
YAHOO.util.Assert.isNull(7,
    "Expected null."); //fails
```

YAHOO.tool.TestSuite Methods:

add(obj *testCase* or *testSuite*) adds a testCase or testSuite object to a testSuite

YAHOO.tool.TestRunner Methods:

add(obj *testCase* or *testSuite*) adds a testCase or testSuite object to the list of items to run
clear() removes all test objects from the runner
run() runs all testCase and testSuites currently queued in the TestRunner

YAHOO.tool.TestCase Methods:

wait([fn *segment*, int *delay*]) causes the TestCase to delay the specified number of milliseconds before *segment* is executed
resume([fn *segment*]) resmes a paused test; if *segment* is omitted, the test automatically passes

YAHOO.util.UserAction Methods:

click(obj *target*, *obj options*) simulates a click on the target
mousedown(obj *target*, *obj options*) simulates a mousedown on the target
mouseup(obj *target*, *obj options*) simulates a mouseup on the target
mouseover(obj *target*, *obj options*) simulates a mouseover on the target
mouseout(obj *target*, *obj options*) simulates a mouseout on the target
mousemove(obj *target*, *obj options*) simulates a mousemove on the target
keydown(obj *target*, *obj options*) simulates a keydown on the target
keyup(obj *target*, *obj options*) simulates a keyup on the target
keypress(obj *target*, *obj options*) simulates a keypress on the target

Dependencies

The YUI Test Utility requires the Yahoo Global Object, Dom Collection, Event Utility, and Logger Control. The Logger's CSS file and the YUI Test CSS file are also required.