

Testing Nashorn

Testing of Nashorn takes place in the Nashorn repository directory. See [Building Nashorn](#).

Nashorn tests are TestNG based. Running tests requires downloading the TestNG library and placing its jar file into the test/lib subdirectory:

```
# download and install TestNG
wget http://testng.org/testng-x.y.z.zip
unzip testng-x.y.z.zip
cp testng-x.y.z/testng-x.y.z.jar ./test/lib/testng.jar
```

After that, you can run the tests using:

```
(cd make ; ant test)
```

There are several additional tests that can be added to the test repository, including the ECMA-262 compliance test suite. The tests can be downloaded using:

```
(cd make ; ant externals)
```

You can run the ECMA-262 test suite with Nashorn using:

```
(cd make ; ant test262)
```

These tests take time, so we have a parallelized runner for them that takes advantage of all processor cores on your computer:

```
(cd make ; ant test262parallel)
```

How to write your own test?

Nashorn uses its own simple test framework. Any .js file dropped under nashorn/test directory is considered as a test. A test file can optionally have .js.EXPECTED (foo.js.EXPECTED for foo.js) associated with it. The .EXPECTED file, if exists, should contain the output expected from compiling and/or running the test file.

The test runner crawls these directories for .js files and looks for JTReg-style @foo comments to identify tests.

- @test - A test is tagged with @test.
- @test/fail - Tests that are supposed to fail (compiling, see @run/fail for runtime) are tagged with @test/fail.
- @test/compile-error - Test expects compilation to fail, compares output.
- @test/warning - Test expects compiler warnings, compares output.
- @test/nocompare - Test expects to compile [and/or run?] successfully (may be warnings), does not compare output.
- @subtest - denotes necessary file for a main test file; itself is not a test.
- @run - A test that should be run is also tagged with @run (otherwise the test runner only compiles the test).
- @run/fail - A test that should compile but fail with a runtime error.
- @run/ignore-std-error - script may produce output on stderr, ignore this output.
- @argument - pass an argument to script.
- @option - pass option to engine.

Sample

```
/**
 * @option --dump-ir-graph
 * @test
 */
```