

# Struts test

[Back to Test Tracking](#)

## Summary

[Mission and Status](#)

[How to run Struts scenario](#)

1. [Configuration](#)

3. [Setup](#)

3. [Run](#)

4. [Results](#)

[How to customize Struts scenario](#)

## Mission and Status

Struts scenario is about running standard Struts examples on top of Apache Tomcat and Apache Harmony. The test suites consists of two parts: driver and tests. Driver is a web client implemented using [HTMLUnit](#) toolkit. Normally it should be run on top of some stable VM. The tests part is a web server part. It consists of Struts with its examples embedded into Tomcat installation.

[Back to Summary](#)

## How to run Struts tests

### 1. Configuration

Make sure that PATH environment variable contains JDK 1.5 (use RI) and Apache Ant (v1.6.5 or later), and that JAVA\_HOME and ANT\_HOME are properly set up. If you use proxy server, specify proxy settings by setting ANT\_OPTS environment variable:

Windows:

```
set ANT_OPTS="-Dhttp.proxyHost=proxy.com -Dhttp.proxyPort=1111"
```

Linux:

```
export ANT_OPTS="-Dhttp.proxyHost=proxy.com -Dhttp.proxyPort=1111"
```

Edit your *framework.local.properties* file:

- Specify driver JRE home by setting *struts\_test.parameters.required.req.driver.java.home*. Normally it should point to some stable JRE.
- Specify JDK to test *struts\_test.parameters.required.req.test.java.home*. It may require a java compiler to compile JSP pages so a path to JDK should be supplied here rather than a path to JRE.
- Make sure the locations for HTMLUnit, JUnit, Tomcat, Struts and Xerces are correctly defined in *adaptors/struts\_test/parameters.xml*
- You may also pass some system properties to test suite by defining a value for *struts\_test.parameters.optional.tested.vm.options* property. The list of valid properties is below.

The following system properties are used to control the scenario execution:

- *webapptest.app.name* - Name of concrete test. We will refer it's value as '<app.name>' below;
- *webapptest.debug* - optional, default is false. Whether to print info about visited anchors to stdout in both modes or not;
- *webapptest.spider.mode* - optional, false by default. Whether to run in 'write' mode or not;
- *webapptest.spider.nestinglevel* - optional for both modes, default is '10'. Maximum nesting level for visiting anchors starting from web application home. MUST be the same for 'spider' and 'test' modes for tested web application;
- *<app.name>.resource.path* - optional, default is relative to the test working dir path 'resources/<app.name>';
- *<app.name>.app.host* - optional, default is 'localhost' (<app.host> below)
- *<app.name>.app.port* - optional, default is '8080' (<app.port> below)
- *webapptest.spider.startfile* - mandatory for 'spider' mode. App home URL's tail (<spider.startfile> below) for 'spider' mode. The URL is constructed as:

```
http://<app.host>:<app.port><spider.startfile>
```

- *http.proxy.host* - optional, specify if connection to the app. host is through proxy;
- *http.proxy.port* - optional, default is '-1'.

[Back to Summary](#)

## 2. Setup

```
$ cd buildtest/infra
$ ant -Dtest.suites=struts_test install
$ ant -Dtest.suites=struts_test setup
```

The last command copies required parameters from *framework.local.properties* file to *required-parameters.properties* file. You should run it after each modification to *framework.local.properties*.

[Back to Summary](#)

## 3. Run

```
$ cd buildtest/infra
$ ant -Dtest.suites=struts_test run
```

[Back to Summary](#)

## 4. Results

Results summary can be found at `buildtest/infra/build/results/struts_test/test-results/`

[Back to Summary](#)

## How to customize Struts scenario

There are two main working modes for this test suite:

- 'spider' (write) mode for 'golden' data/test scenario generation;
- 'test' (read) mode for test scenario playback. Text data returned from server during scenario execution compared against pregenerated 'golden' data.

The following files are produced/used by the scenario:

- 'scenario.properties' - Main test scenario generated during 'spider' mode run and played during 'test' run. May be updated manually after generation. Sample (beginning fragment, note - form submission added manually here):

```
10000.URL./struts2-showcase-2.0.6/showcase.jsp :file520468537.html
10010.ANCHOR.3 :file1722219095.html
10020.ANCHOR.0 :file-1080589601.html
10030.ANCHOR.20 :file857505531.html
10040.ANCHOR.19 :file1376914575.html
10050.BREAK_LEVEL :4
10060.BREAK_LEVEL :3
10070.BREAK_LEVEL :2
10080.ANCHOR.1 :file-2140644304.html
10085.FORM.0.data.test.Update\ Content :update_content.html
10086.BREAK_LEVEL :3
10090.ANCHOR.20 :file-263602398.html
...
```

- 'gen.patterns.properties' - regex patterns to be applied to the 'golden' content (text server responses) before writing it to the 'golden' file. Must be presented if running in 'spider' mode. May contain no patterns. Use pattern description line syntax as follows:

```
nnnn.<regex>
```

in this case replacement string is "". OR

```
nnnn.<regex> :<replacement-string>
```

The patterns will be applied in their 'nnnn' ascending order. Sample:

```
0010.;jsessionid\\w{32}
...
```

- 'compare.patterns.properties' - regex patterns to be applied to the 'golden' content and server responses before actual comparison. Must be presented if running in 'test' mode. May contain no patterns if there is no test execution context related info in golden files and server responses (such as timestamps, references to files with concrete paths etc). Use the same pattern description line syntax as specified for 'gen.patterns.properties' above. Sample:

```
0010.;jsessionid\\w{32}
0020.[0-9]{2,4}[\\/.-][0-9]{2}[\\/.-][0-9]{2,4}
0030.[0-9]{1,2}\\:[0-9]{2}[\\:[0-9]{2}]*
0040.[0-9]*\\ @\\ [0-9]*
0050.<td>.*@\\w{5,}.*</td>
0060."struts\\.token"\\ value\\"\\w{30,}" : "struts\\.token"\\ value\\"
0080.file%3A.*%2Fjakarta-tomcat-5\\.0\\.30 :file%3A.*%2Fjakarta-tomcat-5\\.0\\.30
0090.nifty\\.js\\(?:config\\=.*" : "
0100.<td>.*\\ ago</td> :<td></td>
0110.[A-Z][a-z]{2},*\\ [0-9]{1,2},*\\ [0-9]{4}
0120.[AP]M</td> :</td>
0130.[A-Z][a-z]{2}\\ [A-Z][a-z]{2}\\ [0-9]{2}
0140.[A-Z]{4,5}\\ [0-9]{4}
...
```

- 'fileXXXXXXXXX.yyyy' - 'golden' file containing test data as it was returned from server and processed by all replacement patterns from 'gen.patterns.properties' file during 'spider' mode run.

NOTE: ALL THE ABOVE FILES MUST BE IN THE DIR SPECIFIED BY '<app.name>.resource.path' SYSTEM PROPERTY DESCRIBED ABOVE.

[Back to Summary](#)

[Back to Test Tracking](#)