Version 5.5

Release Notes

Disclaimer

This document is part of JSystem open source project. JSystem is released under Apache 2.0 license. The license can be found here.

© 2005-2010 Ignis Software Tools Ltd. All rights reserved.

Table of Contents

er*

1.Purpose of this Document

These release notes describe the new features and bug fixes for version 5.5 of ***JSystem Automation Platform***.

1.1   New Features Summary

JSystem 5.5 includes features that span over all of the automation development lifecycle and users functionality. The new features include the following:

- HTML Reporter changes
- Model changes
- Flow Control Elements Changes

1.2   Licensing Note

JSystem 5.5 is released under Apache version 2.0 license.

JSystem 5.5 uses many open source projects. A complete list of all projects and their licenses can be found in JSystem Trac.

Note that JSystem core uses only projects with BSD like licenses (BSD, Apache 2.0, LGPL etc.) with the exception of java2html (GPL). Java2html project is removed in JSystem 5.5. If you don't wish to use it you can download the nightly build (Windows, Linux).

2.HTML Reporter changes

The HTML reporter main purpose is to be a fancy log of scenario execution. Nevertheless the HTML report is being viewed by people without technical orientation. In Jsystem 5.5 we have added some capabilities which will help the developers of the automation project make the HTML report more "managers friendly".

2.1 HTML style support with css

Starting at Jsystem 5.5 the user have partial control on the look and formatting of the HTML report by supplying his css.

- css classes were embedded in the html report which is created by the runner.
  List of css classes + a very short explanation of the class can be found in default.css.

- User can define a list of css files which will be incorporated in the html report by going to the jsystem properties dialog --> "Reporter" tab --> css.path.
  css paths should be separated with a semicolon, css files are searched in the classpath and are added to the html in their order
  in the css.path property.

2.2 Controlling package list area

Starting at JSystem 5.5 the user can control the list of links which appear in the package list area.

Altering the package list area involves creating a properties file, and defining the html.package.list property to point to the properties file.

The package list area is redrawn on the end of each test; when drawn, the properties are read from the properties file. The key of the property is link text and the value is link reference.

2.2.1 Example

Here is a code example which creates a properties file and adds a link to the test itself:

```java
/**
 * This test demonstrates how to use the package list feature
 * which was added in jsystem 5.5. The feature enables the developer
 *  to control the left area
 * of the html report.
 * This example shows how to put a link to a test and a link to
 *  jsystemtest.org in the upper area.
 *
 * To make the example work, in jsystem.properties add the following
 *  property:
 * html.package.list=package.properties
 * Now run this test.
 * When opening the html report you will see the in the left upper area
 * of the report you can see the link to this test.
 *
 */
@Test
public void addLinkToTest() throws Exception {
    //create reference to test.
    String currentFullFileName =
            HtmlReporterUtils.getCurrentTestFileName();
    File reportPath = new
        File(report.getCurrentTestFolder()).getParentFile();
    String ref = FileUtils.getRelativePath(new
                File(currentFullFileName),reportPath);
    ref = "./"+ref;

    //create properties file and add links
    Properties p = new Properties();
    p.setProperty("link to test",ref);
    p.setProperty("jsystem site","http://jsystemtest.org");
    FileUtils.savePropertiesToFile(p, "package.properties");
}
```

This is how the html report will look after executing the test:

2.3   Hide test in HTML

When running long scenarios some of the tests are actually administrative steps which have no meaning (if they pass) and when shown in the html report, they make it cumbersome and overloaded with information.

To hide a test in HTML report go to the scenario, right click on the test and select the "Hide in Html" menu item.

After marking a test as "hidden in HTML", it is marked with "(h)"

When executing scenario and viewing log, you will see that the test doesn't appear in the HTML log:

2.4   Flow control meaningful name

In order to improve html scenario tree readability, the ability to set flow elements meaningful name was added.

To set flow element meaningful name, right click on the element and select the "Update meaningful name"
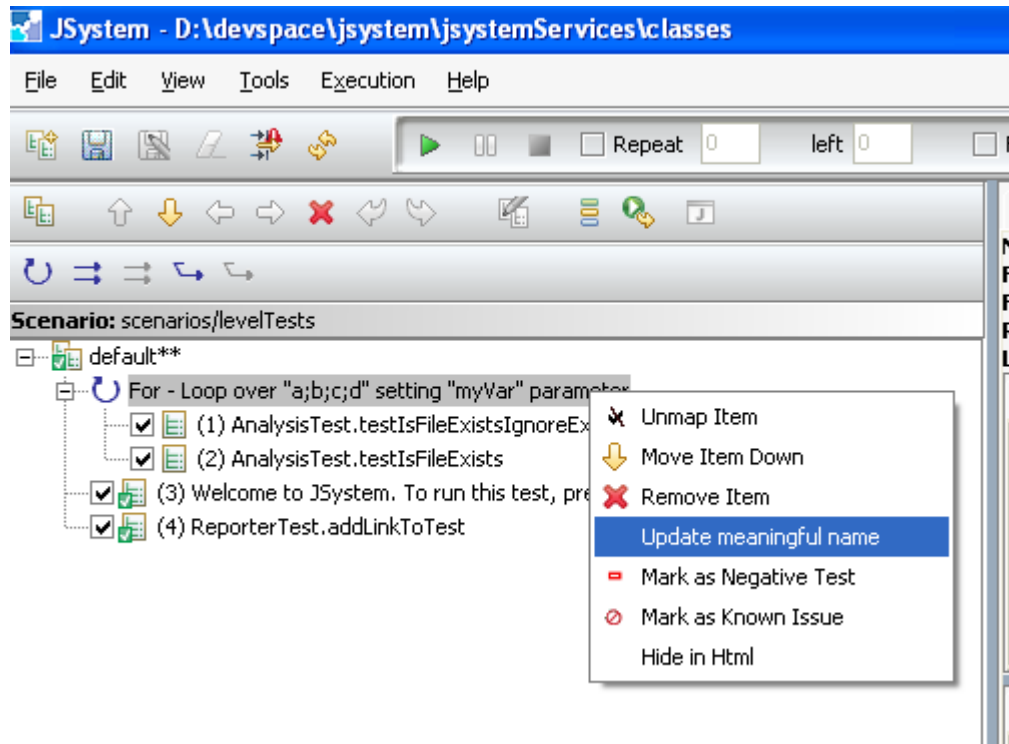
When selecting the menu item, an input dialog is opened:
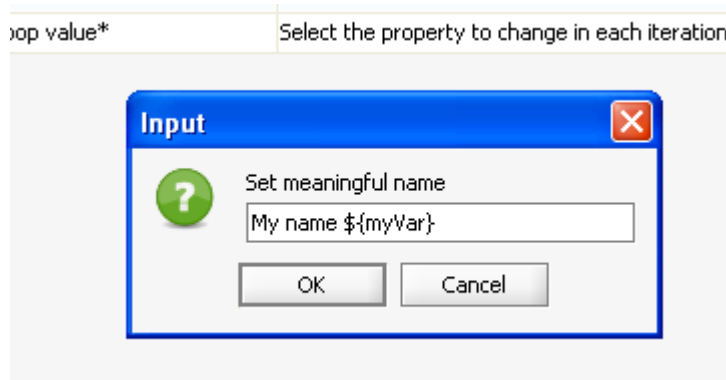
Note that you can incorporate parameters in the meaningful name.

Once meaningful name is updated it is also updated in the scenario tree and in the HTML report.

3.Model changes

JSystem 5.5 introduces some model enhancements which improve the Usability of the runner scenario studio and HTML reporter, and

Increase the power of the JRunner as a development tool.

3.1   Mark a Negative Test

For each test that exists in the system, a Negative test can also be created.

In order to allow using the same test for Negative testing without the need to create a new test, a flag was added to signal that a test is a negative test.

To mark a test (or a Scenario marked as test) as a Negative test,

Right click and select "Mark as Negative Test".
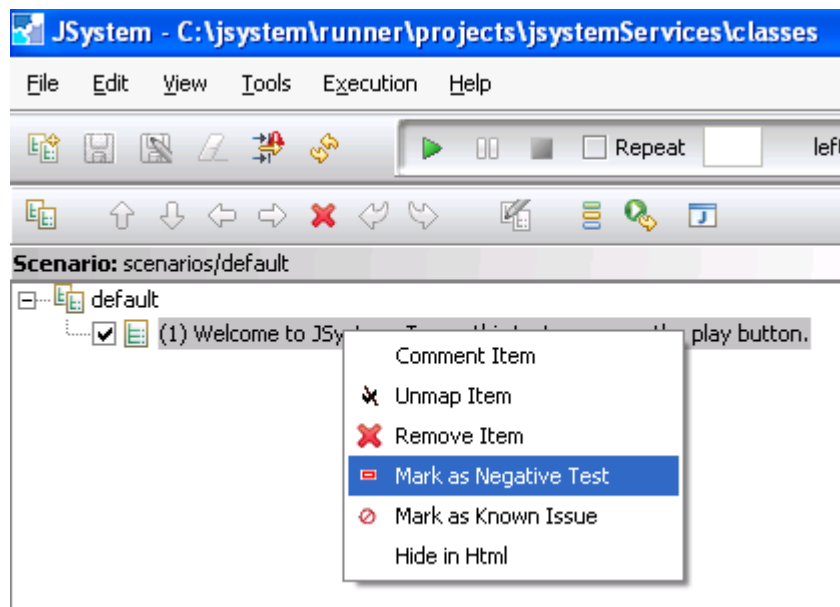
After marking a test as negative, the test will be marked as follows:

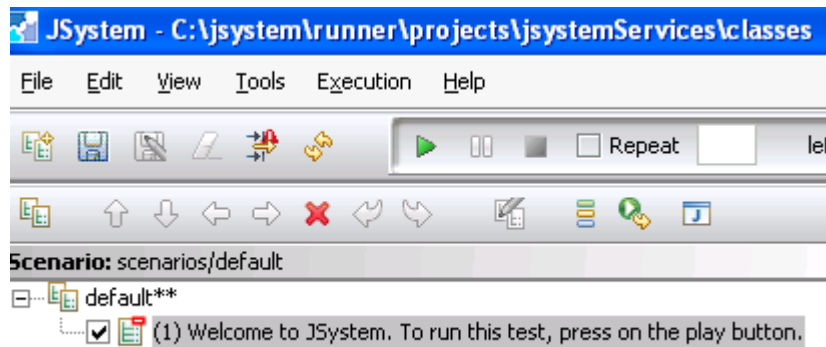A Negative test will act as follows:

a) if test passes – it will fail, reporting

"**Negative test - Test was supposed to fail but passed**".

b) if a test fails on assertion it will pass, reporting

"Negative test - Test failed as expected".

Note: tests with errors (Exceptions) will still be considered failed.

3.2   Test Status Parameter Expose

Following the Return parameters feature added in prior versions,

Starting at JSystem 5.5 each test exposes its status using a Boolean parameter called "Pass".

The parameter is changed after each test execution and holds the status of the last run test.
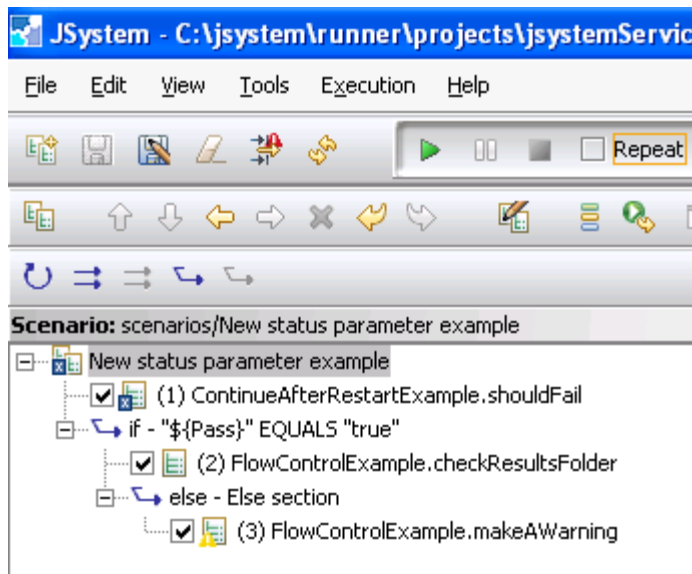
This can be used for flow control usage\statistics, etc.

3.2.1   Example

In the following example, a test is executed and fails.

The flow element after it checks the value of the "Pass" parameter and acts accordingly.

The parameter is referenced like any other return parameter, using the ${parameter name} format.

Note: The parameter default is "true".

4.Flow Control Elements Changes

Starting JSystem 5.5, Flow elements abilities were extended to allow wider flow options.

4.1 Flow Parameters in Sub Scenarios

Following prior support for test parameters in sub-scenarios, Starting at JSystem 5.5, the user can assign different values for Flow elements Parameters in different instances of sub-scenario.

In the following example, sub scenario "Base if scenario" has different values for different instances.
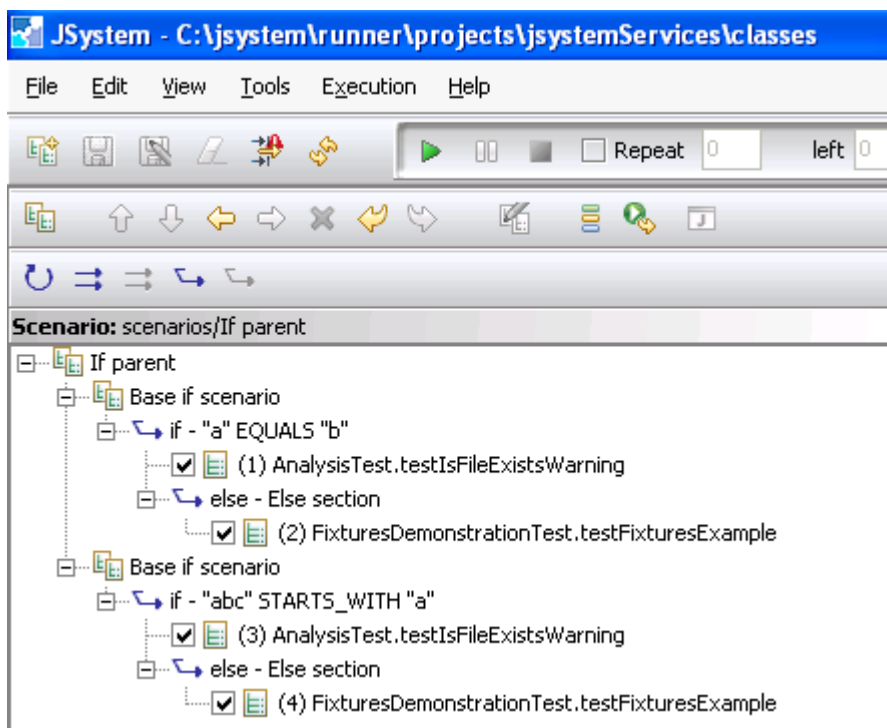
4.2 Reference Parameters Support

Reference Parameters were already introduced to Jsystem test parameters in prior versions including reference to:

1) Values from run.properties file

2) Values from summary.properties file

3) Sut XPath elements

4) Random values

5) Scenario parameters

6) Environment variables

In Jsystem 5.5, Flow elements (if, switch and for), can also use reference parameters as input.

Addressing reference parameters is used the same way it was used for test parameters and is addressed in older documentation.

4.2.1   Example – using reference to random value
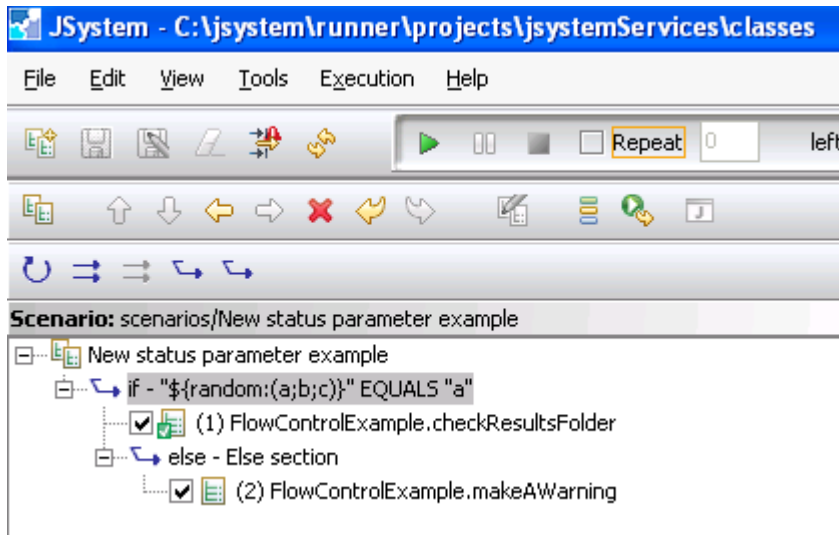
4.3   If Condition Expanding

Till Jsystem 5.5, the flow element "if" only supported String equality.

The "if" was extended to support both Mathematical and String comparisons.

For Mathematical comparisons:

   a) Equals (=)
   b) Not Equals (!=)
   c) Less (<)
   d) Greater (>)
   e) Less or Equal (<=)
   f)   Greater or Equal (>=)

For String comparisons:

   a) Equals
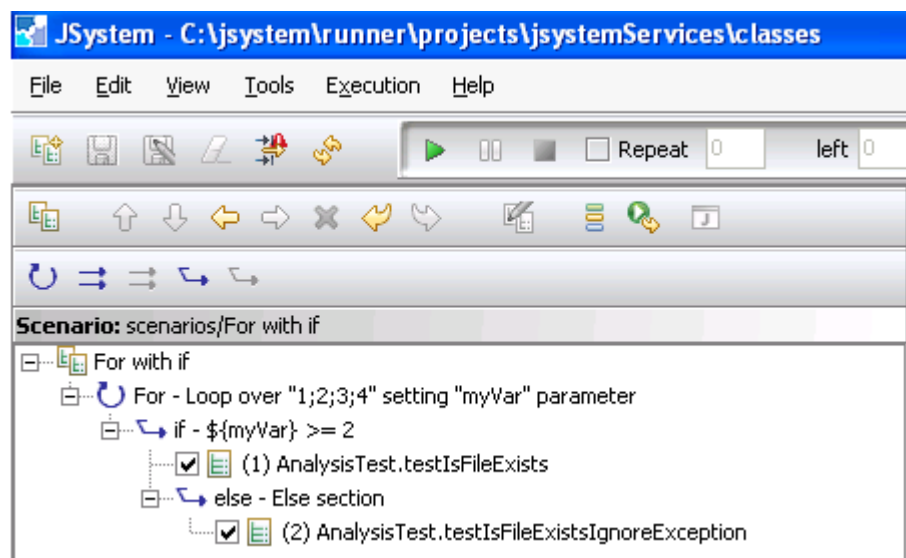   b) Not Equals
   c) Contains
   d) Starts with
   e) Ends with

When selecting the "if" element, you can choose which comparison type to use (String\Mathematical).

Then, you can select which comparison to use.

The following shows the "if" element representation in the Scenario editor for a mathematical greater or equal comparison.
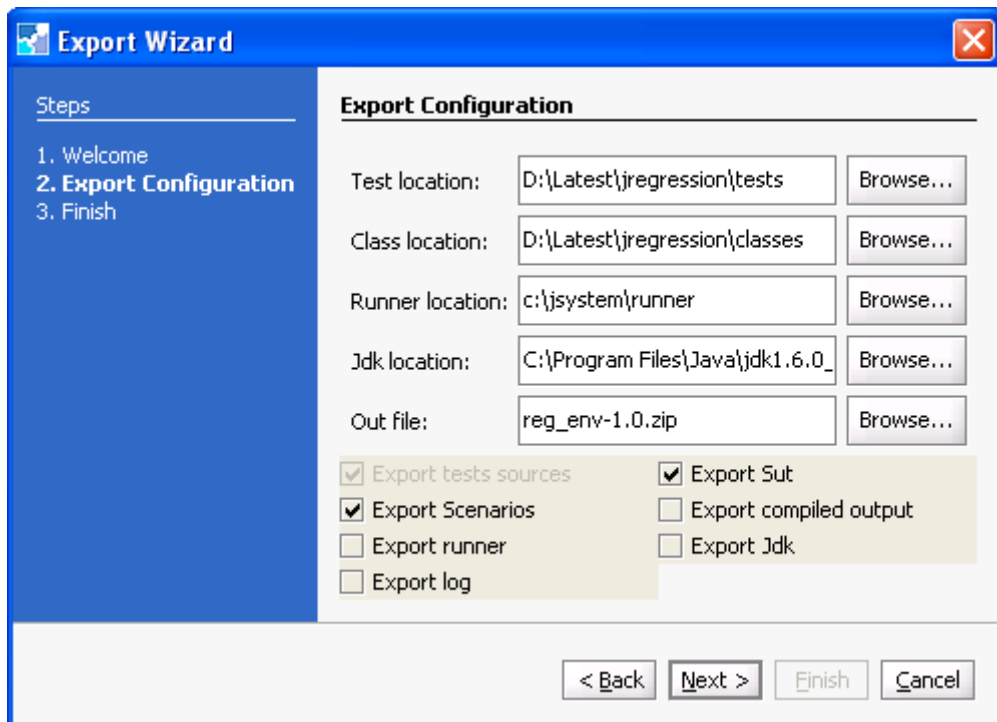

In this example, the "if" inner test will run twice (for values 1 & 2), then the "else" inner test will run twice (for values 3 & 4).

5.Export\Import project

5.1   Export wizard extension

The export wizard was extended to allow exporting of the project files without exporting the Sut or Scenarios files.

By default, the Sut and Scenarios will be checked for export.

These options will affect both source and classes export (if compiled output is also exported).
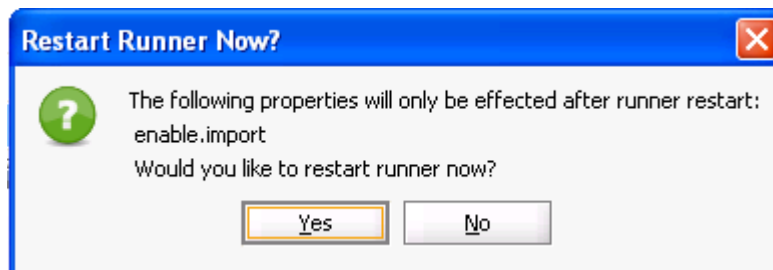
5.2   Import project

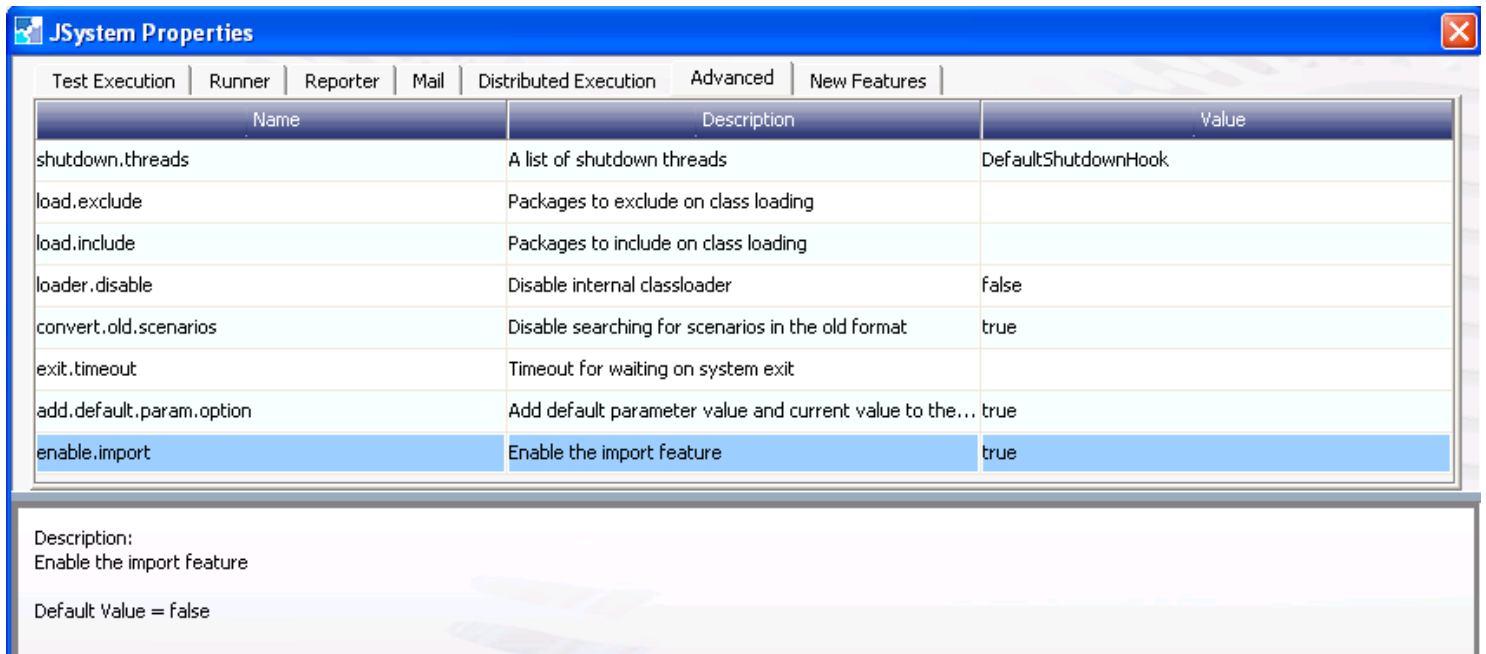In order to support Merging of different projects, the import option was added.

To enable the import button:

Go to Tools-> JSystem Properties-> Advanced tab.

Select the "enable.import" property and change its value to "true". After applying the changes, you will be asked to restart the runner.
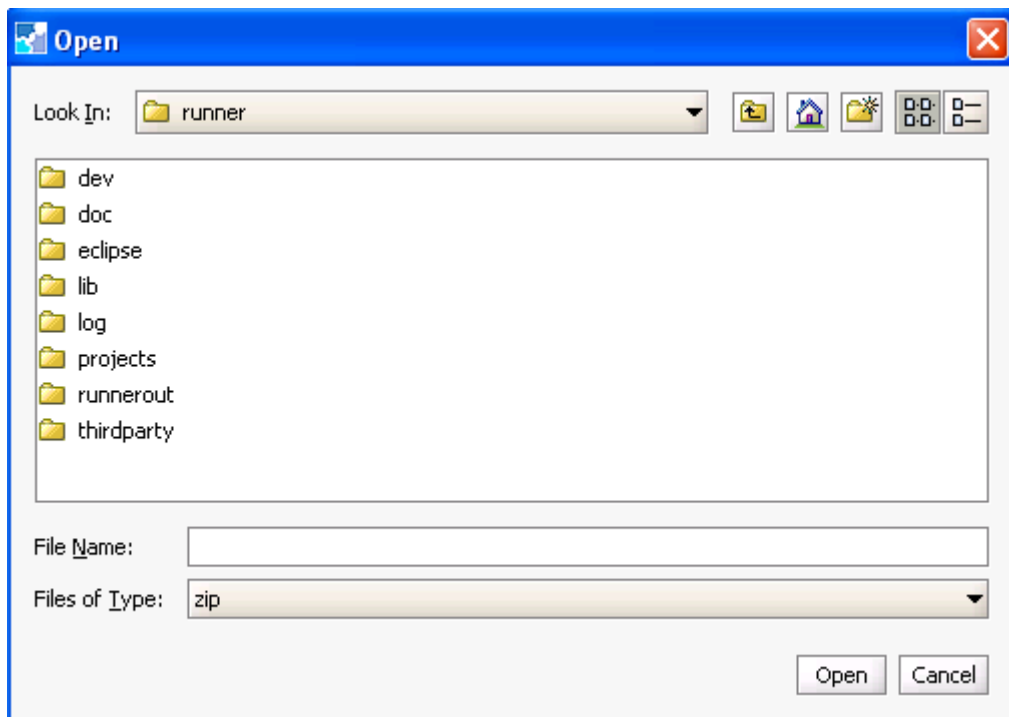
After restart, the upper right bar will have an additional button added, for the import project action.

Pressing this button, will open a file chooser:

Now, select the exported zip file from another project.

After the import is done, additional test\suts and scenarios will appear in the runner.

Note: duplicate tests\suts\scenarios will be overwritten.

6.Auto-Save

When enabling auto-save, by changing the interval value to a   positive value in JSystem properties, the JSystem will activate the save scenario action to save the scenario every '*interval*' seconds.

To set the JSystem property: go to JSystem Properties dialog (Tools → JSystem Properties), into the runner tab, and change the auto.save.interval value.
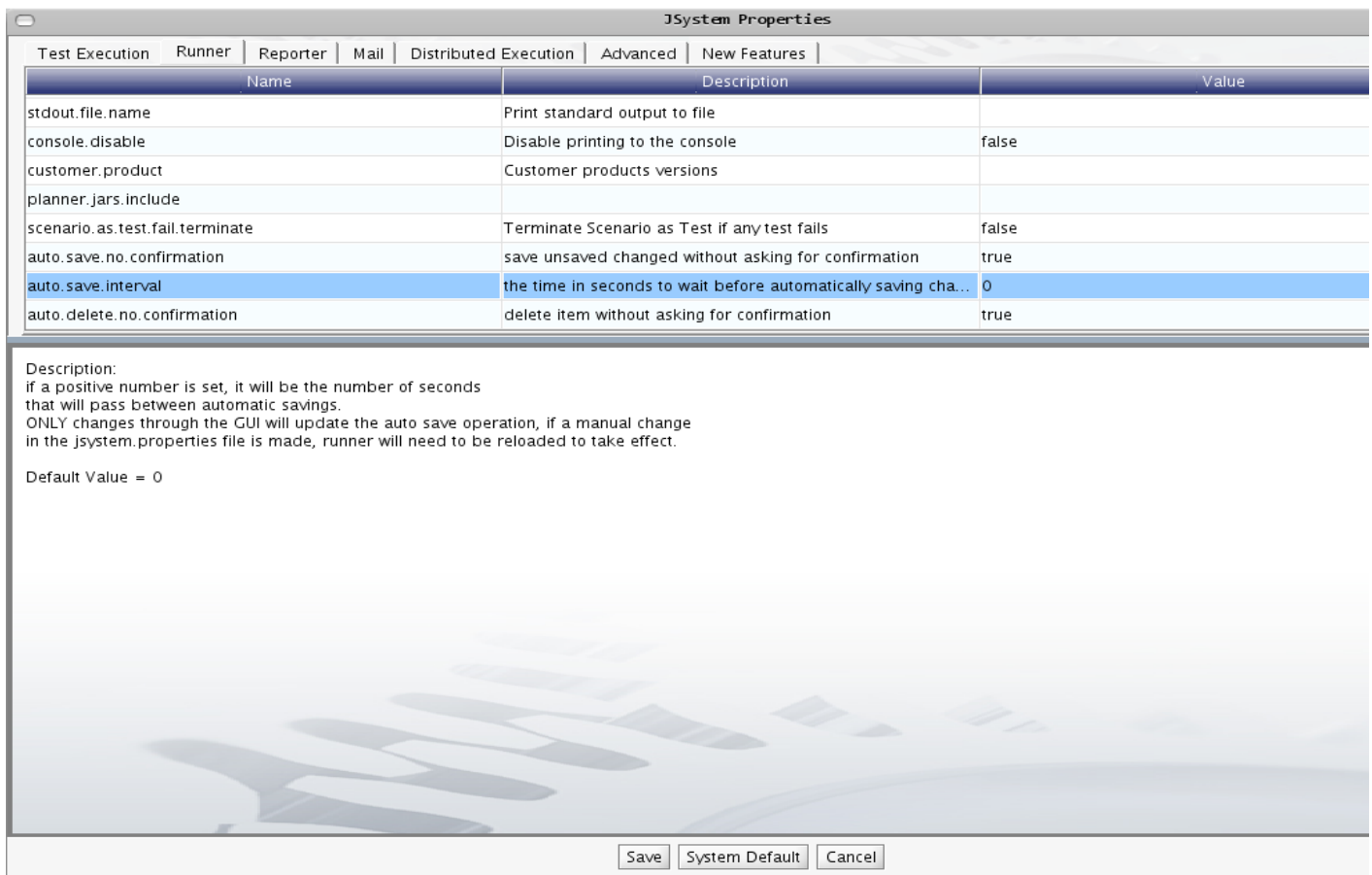
Note: To disable auto save behavior set the value to 0. This is also the default value.

If the value of auto.save.interval parameter will be change directly in the jsystem.properties file, new settings will not be available until next JRunner restart.

7.Known Limitations

7.1   Hierarchical Scenario with For Loops

When executing a "for" loop with a first child which is a flow element, the Html Scenario Hierarchical structure might look wrong.

If a test child is added to the "for" loop, the problem is solved.

This will be fixed in future releases.

7.2   Scenario Marked as Test and Inner Features

When running a Scenario marked as test, with inner tests marked as:

Negative tests\Known issue\Hide in html,

Inner tests functionality is unexpected.

This will be fixed in future releases.

8. Important note regarding System object jars

Untill JSystem 5.5, additional System Objects jars were part of the installation and were available in the runner\lib folder, such as

Ixia, WireShark, etc….

These jars were removed and are no longer part of the open source JSystem project.

For obtaining these jars and additional info, please contact jsystemtest@gmail.com.