SQUASH-TA FRAMEWORK 1.10

Framework TA-1.10.1 Changelog (25/05/2018)

[Saop UI] Changed version of embeded 'commons-codec' library, after witnessing its incompatibility.

The list build hangs with a NullPointerException in version 1.10.0-RELEASE. Now fixed.

[Integration]
The Squash TA framework wasn't building in the new forge, due to Groovy compiler / jdk8 compatibility issues. Fixed.

Framework TA-1.10.1 improvements (20/04/2018)

Evolutions of Squash-TA 1.10.0 Framework are plentiful. They affect both the driver layer and the plugins part of Squash-TA

Driver layer

This section describes all changes in the framework's interface components. These are :

- The Squash TA maven plugin
- The scripting language
- The test project model definition
- The reporting system

Squash TA maven plugin

Tweaking the effect of Squash TA status on maven build failure

Up to version 1.9.0-RELEASE, the effect of Squash TA test failures on the maven build as a whole was controlled by a single boolean parameter : by default, Squash TA test failures failed the whole maven build. If the alwaysSuccess boolean parameter was set to true, Squash TA test failures never caused the maven build to fail.
As of version 1.10.0-RELEASE, a new additional parameter allows to fine tune that behavior by setting up a failure threshold. This allows to fail the build if an error (technical failure of Squash TA tests) is detected, but not if the failure is functional (failed assertion).

Test suite definition

Error mesasges emitted when the test suite definition is not valid now include the test defintion content.

Scripting language

Macro calls in macros

Beginning with the 1.10.0-RELEASE version of the Squash TA framework, macro definitions (also known as shortcuts) may include the use of other macros. To avoid infinite recursion, a given macro can only be called 16 times recursively from a top macro definition before trigger the following DSL parsing error:

```
Macro call '<# macro call instruction>' exceeded the recursion limit 16
```

Specific syntax for VoidResource

As of version 1.10.0-RELEASE, the DSL parser recognizes the syntax `{void}` as the builtin VoidResource instance used for components that take no INPUT resource, or return no OUTPUT resource.
NB : The `{void}` name cannot conflict with user defined resources, as single curly braces `{...}` are not allowed in user defined resource names (whereas `{{…}}` are allowed in order to specify a temporary resource name).

Test project model définition

Loading test resources from classpath

Beginning with the 1.10.0-RELEASE version, test resources can be loaded from jars in the engine classpath. This allows the use of resources packaged in Squash TA plugins, as well as any third party classpath element, in Squash TA test scripts.
The resource name given to the load instruction is the java style resource name of the resource file, where dots are replaced by slashes. Here is an example :

```
LOAD com/example/plugins/sampleplugin/dataset.xml AS dataset.xml
```

Execution logs

Execution message logs were added or amended for the following processes :

- resource search : better log when a resource is not found in a given repository
- macro definition enumeration :
  - warning if a macro file is rejected as invalid
  - more information in debug messages about macro enumeration when macro enumerator debug messages are enabled

Reporting

Adding an 'HTML summary' report

A new, simlified HTML execution report has been added to the 1.10.0-RELEASE version of the framework. This report, called 'HTML summary' only gives a global OK/Functional KO/technical KO result for each test script (including attachmed files, if any), without the results of individual test steps. While the complete report holds valuable information for test automation engineers, we believe the HTML Summary to be easier to read for non-tech savy people, such as pure business analysts.

Enhancement of the TA to TM callback connector

Two features were added to the Squash TA to Squiash TM callback connector through which test results are posted into automated Squash TM executions :

- to avoid garbled display of the error message in the Squash TM when some third party assertion components report error summaries including HTML formatting, a link parameter was added to escape HTML tags so that they are properly integrated and displayed in Squash TM. Such failure summaries are discouraged, but if you need to use a third party plugin that does not apply these guidelines, this option may come in handy.

For the record, Squash TA assertions should give detailed (and possibly formatted) failure information as Failure context elements that are output as attachments to the detailed Squash TA HTML report.

- The debug messages from the TM/TA link component have been enhanced to help debugging callback failures, including some link configuration hosting environment related problems.

Squash TA engine

Error reporting

Better debugging inforamtion was added to execution log messages in two cases :

- when a Target component (adaptor for a test target system type, like the WebTarget or DatabaseTarget) is not recognized (from the previous message diagnosing the problem as an unrecognized target type was difficult)
- some cases of mismatched plugin combination are more clearly described

Squash TA plugins

This section describes all changes to Squash TA plugins.

Common Components Plugin

FileResource parametrization optimization

The File to File(param) FileResource converter used for property substitution in FileResource test resources has been modified to load and treat the FileResource in chunks. This allows to parametrize bigger files without exhausting the JVM memory.

FileResource date formulae enhancement

New functions have been added to the date formulae used by the File to File(param.date) converter :

- the `date(${property.key},<input date format>)` allows to get a date from a property value er from the converter USING clause instead of always computing based on the current date.
- The `ensureWorkingDay()` function adjusts the computed date to the nearest working day before or after the input date, depending on its parameters. Working days are read from component configuration through the USING clause.
- The addWorkingDay() function allows to add a given (positive or negative) number of working days to its input date. Working days are read from component configuration through the USING clause.

[USER_MESSAGE] custom test execution logs

The log command has been modified to allow logging of FileResource contents as custom test execution log entries. A matching macro has been added.
In addition, a macro to load several messages from one test line has been added. One use of this might be logging an resource description message just before logging a FileRessource content.

Use of CSVResource

A converter was added to produce a CSVConfiguration resource from a FileResource. This allows to describe specific CSV format parameters to the File to CSV(structured) by adding a CSVConfigurationResource to the converter USING clause. This allows to check and parse CSV files with a different separator or quote character,

Translation dictionnary resource

Components were added in order to set up translation dictionnaries as test resources to be used in component USING clauses. The primary goal at this stage is not internationalization, but changing messages in generic assertion reports such as database comparison reports into more explicit messages in the specific comparison context of the test.

XmlResource conversion and validation

A `xml.well.formed` unary assertion has been added to ensure that the contents of a FileResource test resource are valid XML. This may be used when getting XML output from a SUT. As an aside, error messages emitted when trying to convert a FileResource that is not valid XML into an XMLResource have been enhanced.

PropertiesResource

Properties file content parsing and validation :

- Defining empty properties was not possible. These property definitions are now accepted, and a warning is issued in the execution log.
- When a property file entry is indeed invalid, the error message now includes the culprit line's number and contents.
- The default properties file encoding may now be overriden is need be through use of the USING clause of the converter.

Database Plugin

dataset to XmlResource converter

A Converter has been added to convert SQL result set resources to XMLResources to allow further use of datbase extracts in tests.

Execution logs

Database Plugin components now log a warning when their USING clause contains one or more useless resource.

Filechecker Plugin

Error reporting

Assertions from that plugin now add their report to failure context as an attachment when they fail.
A warning is issued in the execution loàg when useless resources are detected in USING clauses.

Selenium Plugin

Error reporting

The converter used to create the selenium java bundle from the test code resources now detects that no java compiler is available and throws an explicit exception. Diagnosis is thus made easier when Squash TA is executed whith a JRE instead of a JDK.
Some case when the java compiler error messages were lost have been fixed.

Java compiler configuration for Selenium2 tests

The selenium2 components and macro have been enhanced to allow passing compiler flags when compiling a Selenium2 test before execution. This may be used to debug or fix some compilation errors.

Failure report attachments generated from the Selenium test code

To allow adding custom failure report attachments generated from the java code of a Selenium test, components scan the Selenium code's System.out stream for specific lines formatted in the following way :

```
[[ATTACHMENT--|<report absolute path>]]
```

for example :

```
[[ATTACHMENT--|/tmp/selenium56254.txt]]
```

The path may be a directory, in which cas the whole subtree (for example, an HTML page with its resources) will be attached.

SoapUI plugin

SoapUI test case properties

Test case properties declared in the executed SoapUI workspace may be set by passing configuration resourfces in TA script USING clauses.

SSH Plugin

Key-based authentication

The SSH Plugin now supports key-based SSH authentication.