TESTMAX

TestMax user guide

FAQ

Srimanta Jana

1/15/2014

Contents

Introduction:

The main advantage of this TestMax architecture is separating test data configuration from test case run time environment. It reduces the maintenance of test cases only limited to different test set and test data to run the same test into different QA env without writing or maintaining any test code. This tool provides automation framework for end-to-end testing for various database like Oracle, MySql ,PostgreSql,Cassandra etc. including any JDBC compliance database with extensive SQL and PL/SQL database scripting capabilities along with Selenium Webdriver for frontend integration, Web Service testing, JUNIT & TestNg tests also part of this framework, Performance testing of database API, Java API and Webservice are very easy using this framework, This framework introduced a new concept for reusable libraries using XML configuration without writing any code. SQLLib for back end and TagLib, Injectors for front end can be configured using XML and different tests can be injected into injection points in runtime. It also supports more complex test like data driven tests with static and dynamic dataset with configuration only.

How to setup TestMax Env?

1) Download the TestMaxProject.zip from source forge.com using below URL https://sourceforge.net/projects/testmax/

2) Unzip the project TestMaxProject.zip

3) In Eclipse create a new Java project pointing to the path you unzipped

4) Eclipse→File→New→Project→Java Project

5) In “Create Project Screen” select “Create Project From Existing Source” and select the path where you unzipped and then select Next , then select Finish.

6) Select the project and go to the project properties in Eclipse TestMaxProject-->Properties-->Java Build Path and make sure there all jars are added from the lib directories. If you are using Cassandra database add all jars from \lib\jdbc

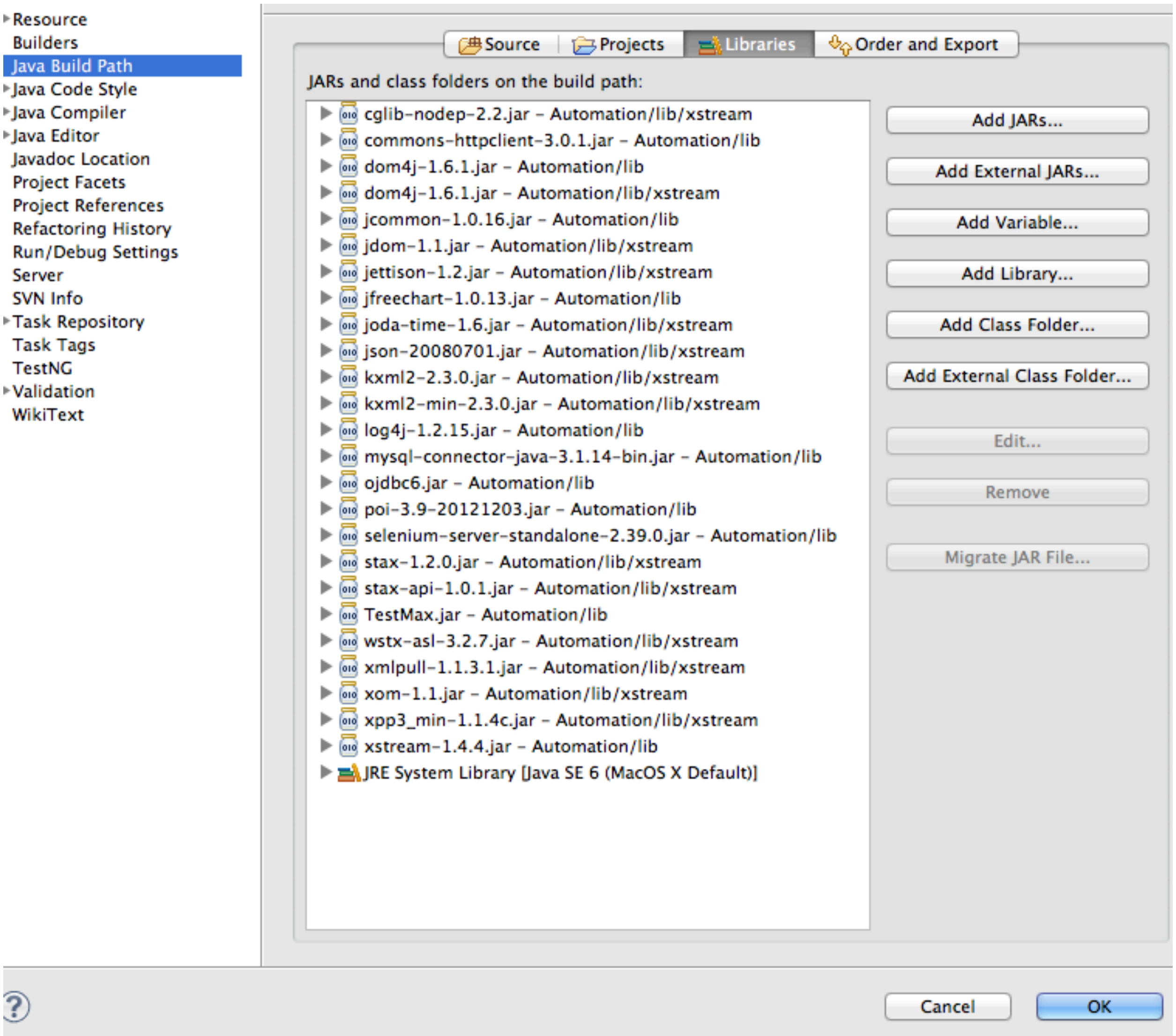
7) Select TestMaxProject -->Run-->Run Configurator-->Java Application -->Click New --> Select Main Class Name=scm.wm.runner.TestRunner. Give Configuration Name=TestMaxProject which will appear with this name under Java Application in the Run Configurator

8) Open Folder TestSuite\SampleTest.xml. Uncomment the test suite you want to run

9) Select TestMax dropdown menu under the Eclipse Run Green triangular button.

What are those classes I need in eclipse class path?

Based on your database you can change jdbc jar.



JARs and class folders on the build path:

```
cglib-nodep-2.2.jar - Automation/lib/xstream
commons-httpclient-3.0.1.jar - Automation/lib
dom4j-1.6.1.jar - Automation/lib
dom4j-1.6.1.jar - Automation/lib/xstream
jcommon-1.0.16.jar - Automation/lib
jdom-1.1.jar - Automation/lib/xstream
jettison-1.2.jar - Automation/lib/xstream
jfreechart-1.0.13.jar - Automation/lib
joda-time-1.6.jar - Automation/lib/xstream
json-20080701.jar - Automation/lib/xstream
kxml2-2.3.0.jar - Automation/lib/xstream
kxml2-min-2.3.0.jar - Automation/lib/xstream
log4j-1.2.15.jar - Automation/lib
mysql-connector-java-3.1.14-bin.jar - Automation/lib
ojdbc6.jar - Automation/lib
poi-3.9-20121203.jar - Automation/lib
selenium-server-standalone-2.39.0.jar - Automation/lib
stax-1.2.0.jar - Automation/lib/xstream
stax-api-1.0.1.jar - Automation/lib/xstream
TestMax.jar - Automation/lib
wstx-asl-3.2.7.jar - Automation/lib/xstream
xmlpull-1.1.3.1.jar - Automation/lib/xstream
xom-1.1.jar - Automation/lib/xstream
xpp3_min-1.1.4c.jar - Automation/lib/xstream
xstream-1.4.4.jar - Automation/lib
JRE System Library [Java SE 6 (MacOS X Default)]
```

How to create database for demo?

1) Create demo db instance using Cassandra, mysql or oracle. Your database instance name can be anything

2) Create 2 tables which is available in TestMaxProject/data/config/create_database.sql

3) Create table using this script

4) Make changes in db_demo.xml with you db info

5) You can create multiple tags for different database connections and use these tag as dbname in your test configuration

6) <sql name="q1" dbname="demodb" >

7) Look at SQLTemplate.xml file located under TestMaxProject\data\module\Template path

8) Use these below attribute for oracle database only if you only returning dbms_output as csv (xls with column name and data row) file format

9) dbmsoutput="yes" dbmstable="yes" dbmsoutputkey="Assert"

How to run your TestSuite?

1) Go to \data\TestSuite folder

2) Add your Test Suite under uncommented portion in the XML file

3) For Non browser Test like Database, JUNIT or TestNg, remove browsers, baseurl attributes

4) For browser Test use these 2 attributes. Use FireFox – release 16 for firefox test

5) If you want to run your test in multiple browsers then set browsers=firefox;chrome for example.

6) If you are running dataset driven frontend tests then, you can use more thread maximum up to 4 for all browsers together

How to setup configuration properties?

1) Go to the TestMaxProject\data\config folder

2) Open Config.properties file

3) Read the comment section of the property and if needed modify

How to add new database in the configuration?

1) Go to the TestMaxProject\data\config folder

2) If you have different test env in QA, Dev or release you can copy the db_demo.xml and rename the file with db_<target env>.xml

3) Read the instruction and example in the file and add new tag to identify your db

4) While configuring you test case using <sql> tag you should pass dbname=<db tag>

How to run Cassandra Demo TestNg Test?

Go to the TestMaxProject\data\config folder

2) Copy the create Test table script and create a table in Cassandra or any database which is pointed to database as demodb tag in your db_demo.xml

3) Open the file SampleText.xml under data\TestSuite

4) Add below configuration in uncommented section

```
<Test name="CassandraTest"

                  page="CassandraTest"

                  env="demo"

                  action="unit"

                  threads="1"

                  timeout="60" />
```

5) Comment all other test suite in this file

6) Open up Run Configurator and select TestMaxProject you configured in eclipse based on “How to setup Env” item 7)

7) Click Run button

How to run any test suite you develop?

1) Open the file SampleText.xml under data\TestSuite

2) Add below configuration in uncommented section

```
<Test name="Any Name of your Test suite"

        page="Unique Page Name given in your XML Test configuration under data/module
        folder"

        env="Choose your <target env> file name under config excluding db_ prefix "

        overrideattributes="Any variable need to override its value in runtime"

        groupbythread="Any column name in dataset you like to group by "

        action="unit"

        threads="1"

        timeout="60" />
```

3) Comment all other test suite in this file

4) Open up Run Configurator and select TestMaxProject you configured in eclipse based on “How to setup Env” item 7)

5) Click Run button

How to debug single or multiple tests (actions) in a test suite?

1) Every test suite has a unique page attribute which is an XML configuration file in data/module/<your project folder>

2) If you want to debug any individual action then select the action name attribute value which you want to run

3) In the SampleTest.xml file under /data/TestSuite use the page name= target page name and pass action=”unit@<target action to run>

4) If you want to run multiple actions then use action=”unit@ action1;action2..”

5) Comment all other test suite in this file

6) Open up Run Configurator and select TestMaxProject you configured in eclipse based on “How to setup Env” item 7)

7) Click Run button

How to build new backend tests?

1) Open data/module/template folder and choose any template file suites your types of test

2) Copy and rename the file to /data/module/<your project> folder

3) Open the XML file in source view in eclipse

4) Give a unique page name using name attribute under <page name=" unique page name" ../>

5) Make sure no files under /data/module have duplicate page name

6) Copy or rename any action and then change validators under that action based on your test

How to build frontend Tests?

1) First go to /data/taglib folder and create a new folder for your target application or project

2) Copy the Apigee.xml from /data/taglib/apigee folder to your target folder and rename the file

3) Rename name attribute for <taglib > with a new unique name

4) Modify loginToApigee Tag and add new tags as per your application

5) Modify invokeDashboard tag to appropriate name and you can add group of abstract tags based on your application flow. The sequence of tags should be how it will be called during runtime

6) Go to the injector folder and create a new folder as per your project or application name

7) Implement each of these abstract tags you defined in the taglib under abstract group of tags

8) Once taglibs and injectors are created then you can use these in your test configuration under /data/module/Frontend with a new folder usinf new Test configuration file. See the example in /module/frontend//apigee/Apigee.xml

9) For each test you can define your variable in two groups like common or local variable. For example, loginname, password are common variable. Apiname, basepath etc are local variable. Whatever variable you are defining in the action, those varaibles can be invoked by tags in runtime using @<variable name>. For detail, please see loginToApigee tag how loginname and password is passed.

How to use dataset in the test?

1) Open ApigeeGlobal.xml under /data/module/apigee folder

2) Look at TC1 with attribute setglobal=”yes” and it fetches 3 records from the Test table

3) In TC2 this 3 rows are used as global dataset with attribute dataset=”global:query.q1” . q1 is the name of query in TC1

4) If you pass similar static dataset using XLS or XML. Look at the dataset example in /data/global/sample.xls or testing.xml. To pass the sample.xls data as dataset you can use attribute as dataset=”global.xls.sample”. Similarly for XML dataset you can use attribute as dataset=”global:<XML Global tag name attribute>.<tag>” . For example: in testing.xml dataset=”global:testing. CIFunctional”.

5) While using dataset, you can directly use the column name as variable in the target source without defining the variable in your action config file under validator as local or common varaible. For example in testing.xml , the attributes msg,localmsg, order_no can directly be used using @msg, @localmsg or @order_no in injectors, taglib or action config. Similarly, for dynamic dataset using database or XLS, you can use column name directly with “@”symbol in your injector, taglib or action config.

Where to find Test results and reports?

1) Select the folder output in TestMaxProject and click refresh

2) Find the latest bottom most sub folder created under output

3) Open up report folder and see the HTML report file

4) Open other sub folders and see respective logs

How to write Java injectors for frontend test?

1) You can write java injectors if your XML injectors can not implement your logic which is very complex.

2) Createapackagenamesayscm.myinjector.myapp.us (llok at the src folder for this example)

3) Write a Java class in this package say MyPage.java extending from BaseTrack class

4) Add below method into your java class which will give you access to handler and driver object

```
public void setSeleniumRunner(SeleniumBaseHandler handler){
this.handler=handler; this.driver=this.handler.getDriver(); }
```

5) Write your own java method see detail in the src folder in Mypage.java

6) In the config file located at /data/config folder change properties TRACK=us that is same as last part of the package name scm.myinjector.myapp.us

7) Invoke the method by passing mypage.deleteAPI as injector ID in the XML taglib where mypage is the lower case of MyPage.java. See this example in /data/taglib/apigee/Apigee.xml

8) See example how these tag libs are invoked during runtime in /data/module/Frontend/apigee/ApigeeGlobal.xml

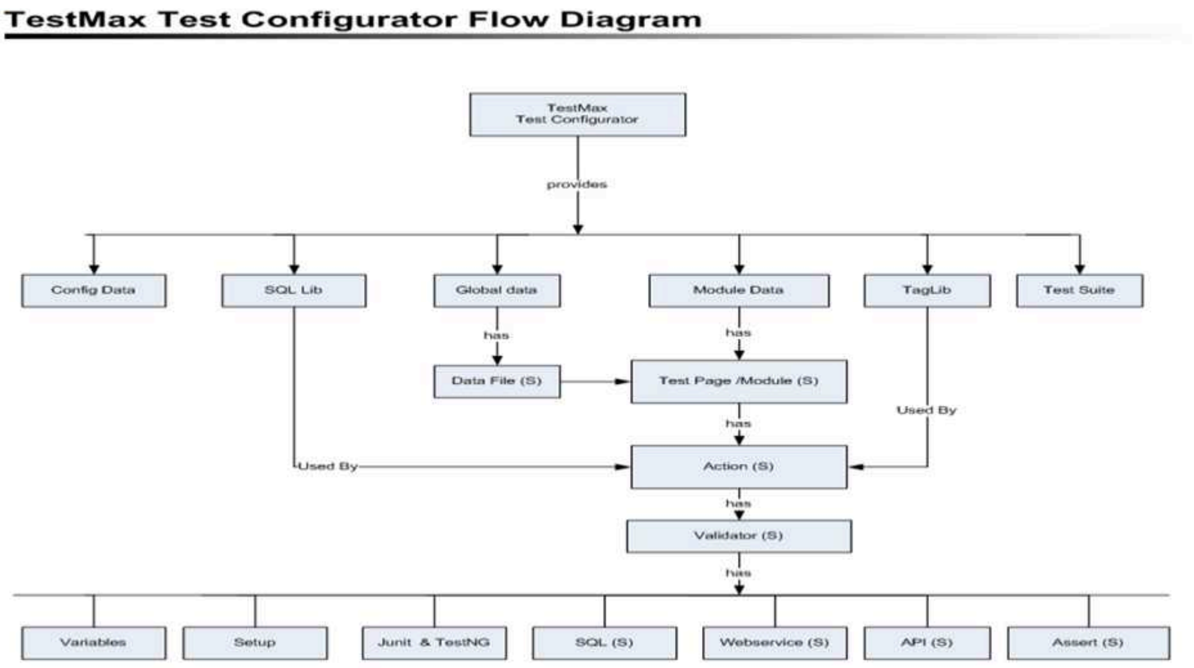
Where to report bug?

1) If you find any bug , please file under https://sourceforge.net/p/testmax/tickets/

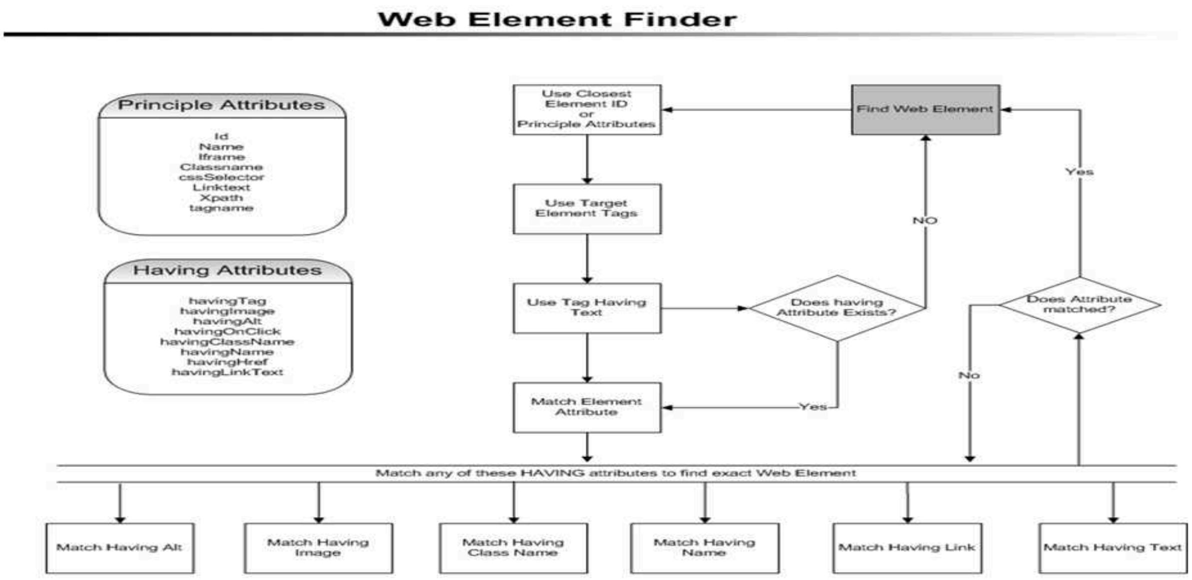
2) If you need to discuss any topics, please put your topic under discussion forum.

What is TestMax configuration flow?

Below diagram shows TestMax configuration flow. Within a validator you can have combination of below tasks configuration.



What are Principle and Having attributes in XML injectors?



Principle attributes are those which selenium webdriver recognizes directly. The most recommended principle attributes are html ID, Name and sometimes classname. In the TextMax injector you should try to find html ID of the target element or any closest ID that contains the target element.

Using having attributes along with havingTag you can find your target element. For example:

If you are directly using selenium click() , sendKeys(), selectByVisibleText() etc. and you do not have ID of the target element but you have aa ID for the nearest element then you can use below tags like below within your html injectors.

Element	Tag Examples
ID with Operation=click	<tag description="Click Advance Search" id="advanced-tab" timeout="3000" method="click"></tag>
ID with Operation Typing input @startdate is dataset column or variable within validators	<tag description="Enter Start Date" id="post_date_start" timeout="3000" method="sendKeys">@startdate</tag
Name with Operation=click	<tag description="Click Advance Search" name="advanced-tab" timeout="3000" method="click"></tag>
Name with Operation Typing input @startdate is dataset column or variable within validators	<tag description="Enter Start Date" name="post_date_start" timeout="3000" method="sendKeys">@startdate</tag
Not having ID but contained within div which is having ID	<tag description="Click Search button" id="viewIdeasInput" havingTag="a" havingText="Search" timeout="3000" method="click"></tag>
You can choose any of the having attribute in diagram combination with havingTag together along with principle ID	<tag description="Click Evaluation File Task" id="viewAllTaskForm" havingTag="a" havingHref="EvaluationFile?taskid=" timeout="3000" method="click"></tag>
Not having ID but contained within div which is having Name	<tag description="Click Search button" name="viewIdeasInput" havingTag="a" havingText="Search" timeout="3000" method="click"></tag>
You can choose any of the having attribute in the diagram combination with havingTag together along with principle attribute=Name	<tag description="Click Evaluation File Task" name="viewAllTaskForm" havingTag="a" havingHref="EvaluationFile?taskid=" timeout="3000" method="click"></tag>
Click an element which nearest element class and target element having link	<tag description="Select Team Member" classname="ui-menu-item" havingTag="a" havingText="@memberlogin" timeout="3000" method="click"></tag>

text =variable	

Can I write injector without using WebDriver methods?

Yes you can. There is a TestMax method which encapsulates any operation with a attribute name=operation. If you do not set this attribute property than by default will perform click() operation.

You can set below html element attributes within the havingAttribute

Id, title, text, href, class

The values need to be set within havingAttribute=<Attribute>:<Value>. You can also use comma separated multiple attributes and their values together.

TestMax Methods	Tag Examples
executeOperation method without operation attribute in tag (click a link that having title attribute)	<tag description="Click Post Evaluation Tab" tagname="a" havingAttribute="title:Evaluation" timeout="3000" method="executeOperation"></tag>
executeOperation method with operation attribute in tag (click a link that having ID)	<tag description="Click Post Evaluation Tab" tagname="a" havingAttribute="id:myid" timeout="3000" method="executeOperation"></tag>
Click a span that having link text	<tag description="Click Invite Button" tagname="span" havingAttribute="text:Invite" timeout="3000" method="executeOperation"></tag>
Click a link that having class attribute	<tag description="Click Invite Team Member" tagname="a" havingAttribute="class:inviteMember" timeout="3000" method="executeOperation"></tag>
Write value in a input which having name Value passed as variable=@imagepath	<tag description="Select Image File" tagname="input" havingAttribute="name:logoFile" operation="write" timeout="3000" method="executeOperation">@imagepath</tag>
Click an element which is having href (default operation=click)	<tag description="Click Post New Challenge" tagname="a" havingAttribute="href:/Page/PostChallenge" operation="click" timeout="5000" method="executeOperation"></tag>
Pass variable as value to any attribute	<tag description="Enter Login Name" tagname="a" havingAttribute="href:@actionuri" timeout="500" operation="click" method="executeOperation"></tag>

How to use conditional switch statements in injectors or taglib XML function?

Below is the example of using switch statements

```
<clickPostIdeaWithoutRandom description="use this tag to click Post Idea tab">
                <tag description="Click Post Idea Tab"  id="navWrap"  havingTag="a"
havingHref="/Page/PostIdea"  timeout="3000" method="click"></tag>
                <switch name="verify if idea title field exists" description="verify if permission
for post idea available for community user">
                <switch-element description="validate if idea title field exists"  id="title"
timeout="1000"></switch-element>
                        <case  condition="true">
                          <injector id="taglib:postidea-injector.ideaWithoutramdom" ></injector>
                        </case>
                        <case condition="false">
                                <tag description="verify message" id="post_idea_fresh"
havingTag="div" havingClassName="content" timeout="3000" method="asserttext">@ideatitle</tag>
                        </case>
        </switch>
         <injector id="taglib:viewideas-injector.inviteMember" ></injector>
         <injector id="taglib:viewideas-injector.postComment" ></injector>
         <injector id="taglib:viewideas-injector.getLatestIdeaStats"></injector>
                 <injector id="taglib:viewideas-injector.getGraduationTasks"></injector>

                 <injector id="taglib:viewideas-injector.getGraduationReviewTasks"></injector>
   </clickPostIdeaWithoutRandom>
```

Can I call Injector with another injector?

Yes you can. See example above.

How to invoke iframe within rich editors?

You can call any javascript using *executeScript* method. Id attribute is not important. You can pass any garbadge value to id. You can use your own java script also even jQuery.

```
<tag description="Enter Description"  id="tinymce"   timeout="5000"
method="executeScript">

        if(navigator.userAgent.toLowerCase().indexOf('msie')>=0||navigator.userAgent.toLowerCase().indexOf('safari')>=0||naviga
        tor.userAgent.toLowerCase().indexOf('firefox')>=0){

        document.getElementsByTagName('iframe')[0].contentWindow.document.getElementById('tinymce').click();

        document.getElementsByTagName('iframe')[0].contentWindow.document.getElementById('tinymce').innerHTML='@detail
s';
        }else{
                document.getElementsByTagName('iframe')[0].document.getElementById('tinymce').click();
                document.getElementsByTagName('iframe')[0].document.getElementById('tinymce').innerHTML='@details';
        }

</tag>
```

How to use conditional if in injector?

Below is the sample.

```
<if name="verify if summary exists" description="verify if id=apiList">
  <if-element description="validate if condition for id=summary"  id="summary"  timeout="1000"></if-element>
   <tag description="Enter Summary" id="summary"  type="input" timeout="3000"
method="sendKeys">@summary</tag>
</if>
```

How to disable events while executing test?

```
<tag description="Member User Name" id="inviteTeamMemberUserName"   timeout="3000"
method="executeScript">
                                        var keyboardEvent = document.createEvent("KeyboardEvent");
                                        var initMethod = typeof keyboardEvent.initKeyboardEvent !== 'undefined' ?
"initKeyboardEvent" : "initKeyEvent";

                                        keyboardEvent[initMethod](
                                                  "keydown", // event type : keydown, keyup, keypress
                                                   true, // bubbles
                                                   true, // cancelable
                                                   window, // viewArg: should be window
                                                   false, // ctrlKeyArg
                                                   false, // altKeyArg
                                                   false, // shiftKeyArg
                                                   false, // metaKeyArg
                                                   40, // keyCodeArg : unsigned long the virtual key code, else 0
                                                   0 // charCodeArgs : unsigned long the Unicode character associated with the
depressed key, else 0
                                        );
                                        document.dispatchEvent(keyboardEvent);

</tag>
```

How to assert an HTML element value against user defined variables or dataset value?

See below example. @category and @postedcategory are variables from dataset

```
<tag description="verify category"  id="idea_stats" havingTag="a" havingText="@category" timeout="3000"
method="asserttext">@postedcategory</tag>
```

How to remove certain portion of html text and than assert the value?

See below example

```
<tag description="verify idea count from view idea page"  id="ideas-result" havingTag="p"
havingText="ideas" timeout="3000" stripchar="ideas" method="asserttext">@ideacount</tag>
```

How to execute my own java script within the browser?

Same as before, use executeScript TestMax method.

```
<tag description="Enter File Name"  id="uploadfileNameId" timeout="3000" method="executeScript">

                        var arr = document.getElementsByTagName("input");
                         for (var i = 0; i &lt;arr.length; i++) {
                                if(arr[i].getAttribute('id')!=null){
                                        if(arr[i].getAttribute('id').indexOf("uploadfileNameId")>=0){
                                           arr[i].setAttribute('value','@evaluation_file');
                                        }
                                }
                        }
</tag>
```

Can I use while in the injector?

Yes, you can. See this example

```
<verifyChallengeStarted description="Verify if the challenge has started">
        <while name="verify if challenge started" description="verify if challenge started for community user">
                <while-element description="Loop until challenge Starts"  id="site_challenge_countdown"
timeout="3000">Starting</while-element>
                        <tag description="Click Home Tab"  id="navWrap"  havingTag="a"  havingHref="/Page/Home"
timeout="3000" method="click"></tag>
                        <tag description="Extract Site Challenge Count down status"  id="site_challenge_countdown"
timeout="3000" method="asserttext">@challenge_countdown</tag>
                </while>
</verifyChallengeStarted>
```

Can I change my Excel sheet and run same test using another Excel?

Yes you can. You can use datasetextension attribute in your testsuite configuration file, which is optional. For example if you are using dataset=”global:xls:mytestdata” in your action configuration, then you can create another excel name as api_mytestdata under global folder. In the runtime global:xls:mytestdata will be replaced as global:xls:api_mytestdata. The name “api_” is user defined and you can use anything like myapi_ etc.

Caution: Don’t use this optional attribute if you do not need to use different excel for same test suite.

```
Test name="SetupCommunity"

   page="SetupCommunity"

   browsers="firefox"

   env="qe36api"

    datasetextension="api_"

   overrideattributes="password=Spigit1234;adminuser=admin "

   groupbythread="community"
```

```
baseurl="https://cappilot.[env].spigit.com"

action="unit"

threads="1"

timeout="60" />
```

How can I use latest Selenium Web Driver?

Go to the site http://docs.seleniumhq.org/download/ and download latest selenium standalone jar and IE drivers. Replace these under TestmaxProject\lib folder. Follow the naming convention of IE driver for 32 and 64 bit windows.

What are these method names supported in TagLib?

We support all Selenium methods and also TextMax provides some custom methods. Please look below table to find method names to pass using <tag> method attribute.

Selenium Method Name	Use
sendKeys	To send input to textbox
click	To Click any element
selectByValue	To select dropdown by value
selectByVisibleText	To select dropdown by visible text
deselectByVisibleText	To de select dropdown against visible text
clear	To clear input
movemouse	To move mouse
submit	To submit form
get	To open url

TestMax custom methods	Uses
asserttext	To assert and extract text value from element.
executescript	To execute any java script including jQuery or other scripts
executeoperation	Use this method to perform some operation on a selected html element including operation=click, read or write
urlextract	To extract certain value from the url after

	subtracting certain portion of URL from beginning. Use additional attribute in the <tag> stripchar=<URL to subtract>
download	To download a file from link.
uploadfile	To upload a file using upload input control

How to use random numbers in the variable input?

TestMax supports to pre-defined variables random and smallrandam. Use these variables without declaring anywhere in your taglib , injector, action or validator configuration as @random and @smallrandom. In the runtime the value will automatically be replaced within the tag.

Ex: <tag description="Enter Idea Title" id="title" timeout="3000" method="sendKeys">@title @random @smallrandom</tag>

In the above example @random and @smallrandom value will be replaced automatically to create a new unique title. You can use together or either of these.

How to use pre-defined timestamps within tag automatically?

You can use below pre-defined timestamps variables similar to @random and @smallrandom which will be replaced iruntime in mili sec within the tag

Timestamp	Uses (all in mili sec)
onehourlate	@onehourlate
fivehourlate	@fivehourlate
fiftyhourlate	@fiftyhourlate
thousandhourlate	@thousandhourlate
fivethousandhourlate	@fivethousandhourlate
now	@now
oneminlate	@oneminlate
twominlate	@twominlate
threeminlate	@threeminlate
fourminlate	@fourminlate
fiveminlate	@fiveminlate
tenminlate	@tenminlate
thirtyminlate	@thirtyminlate

How to use pre-defined date format (MM/dd/yyyy)?

Use below variables within the tag or pass value using @ within var.

Date (MM/dd/yyyy)	Uses

today	@today
yesterday	@yesterday
tomorrow	@tomorrow
oneweekbefore	@oneweekbefore
onemonthbefore	@onemonthbefore
oneyearbefore	@oneyearbefore
oneweekafter	@oneweekafter
onemonthafter	@onemonthafter
oneyearafter	@oneyearafter