
LISA SDK: Create Your Own Assertion

Table of Contents

Description	3
Pre-requisite	3
Steps to Create a Custom Assertion	3
Instructions to Deploy a New Assertion	6
Steps for Implementation:	6
References:.....	9

Description

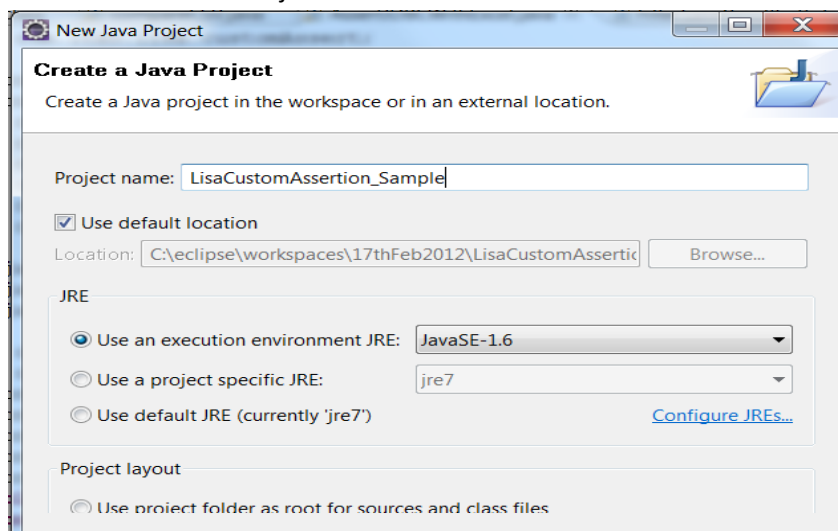
This document is intended to be used by any individual who wishes to create their own Assertion to handle a specific situation. The LISA software provides built-in support for custom assertions.

Pre-requisite

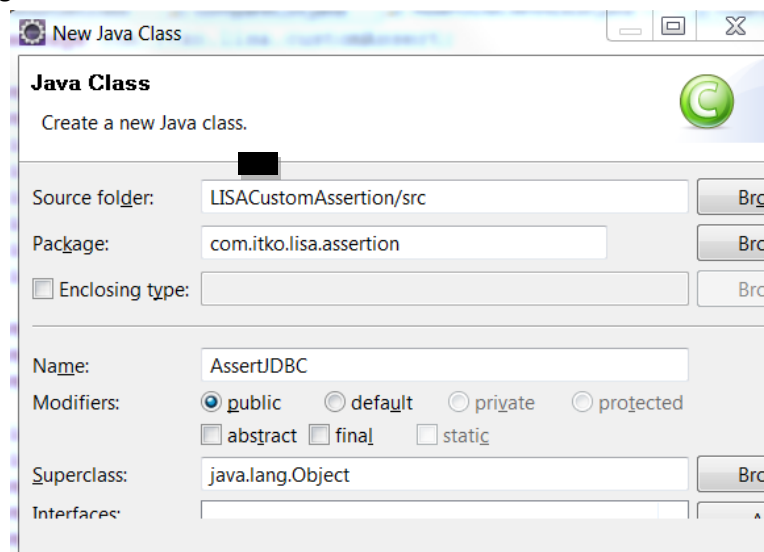
- Java IDE must be installed on machine.
- LISA must be installed on machine.

Steps to Create a Custom Assertion

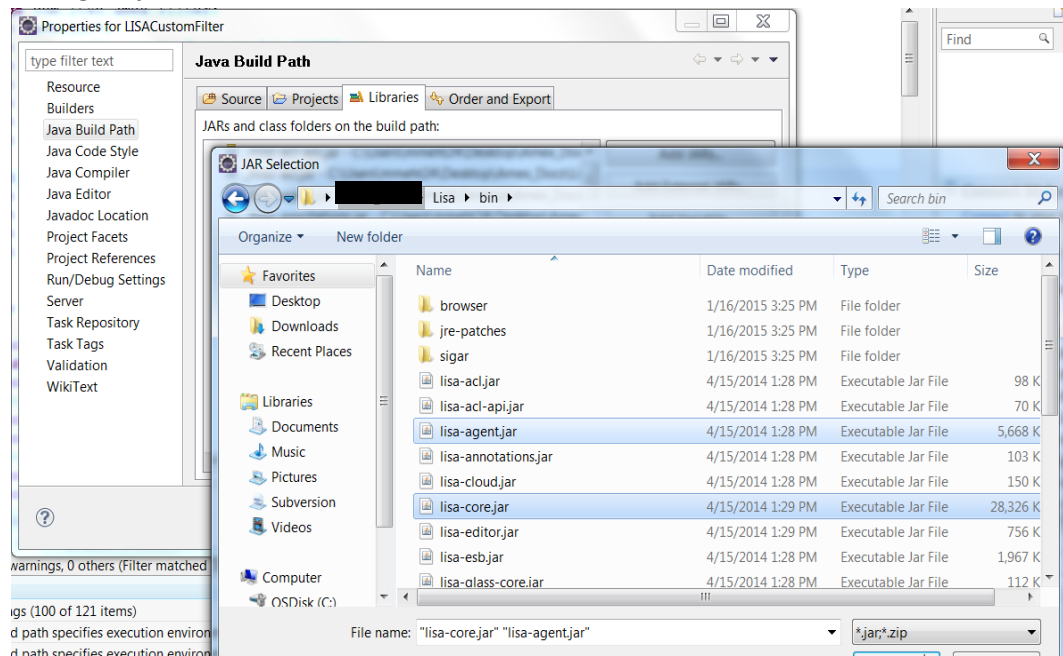
1. Open IDE and create a new Java Project.



2. Provide a Package name and Java Class name.



3. Add External Libraries in build Path from LISA_HOME/bin directory as shown below
 - a. Lisa-core.jar
 - b. Lisa-agent.jar



4. Java Class created in Step 2 must extend **"CheckResult"**.
5. Implement all mandatory methods. Below are the methods to implement:
 - a. **getTypeName** method: This method provides the name that is used to identify the custom assertion in the model editor.

```
public String getTypeName()
{
    return "Assert JDBC Result Set";
}
```
 - b. **getCustomParameters** method: In this method, you create a ParameterList and add a Parameter for each parameter to the assertion., add a **Parameter** to the ParameterList for the assertion.

```
public ParameterList getCustomParameters ()
{
    ParameterList p = new ParameterList();
    p.addParameter( new Parameter( "Is FTP", ISFTP_PARAM, new
    p.addParameter(new Parameter(FILE_PARAM_DESC, "file", this.file,
    OutputStream.class));
}
```
 - c. **initialize** method: Initialize the custom assertion object with the value of the DOM Element.

```
public void initialize(Element e)
{
    this.file = XMLUtils.getAttributeOrChildText(e, "file");
}
```

- d. **evaluate** method: The TestExec parameter provides access to the test environment, such as logs and events. The Object parameter provides access to results returned from executing the node. The Boolean return type returns true if the assertion is true. Otherwise it returns false.

```
public boolean evaluate(TestExec testExec)
```

```
{
    //Provide main Logic here
}

package com.itko.lisa.customAssert;

import com.itko.lisa.core.ModuleLegacy;

public class AssertJDBCWithExcel extends CheckResult {
    private static final String WARNING_FOUND = ModuleLegacy.resources.get("rset.chkresset.");

    protected static Log cat = LogFactory.getLog("com.itko.lisa.customAssert.AssertJDBCWithExcel");
    private static final String FILE_PARAM = "file";
    private static final String FILE_PARAM_DESC = ModuleLegacy.resources.get("test.fsavexp2f.fi");
    private static final String Sheet_PARAM = "sheet_name";
    private String sheet_name;
    private String file;

    private boolean errorWhenNotAppropriate = true;
    public AssertJDBCWithExcel() {}

    public boolean isErrorWhenNotAppropriate() {}
    public void initialize(Element rNode) {}
    public void setErrorWhenNotAppropriate(boolean error) {}
    public String getFile() {}

    public void setFile(String file) {}

    public String getTypeName() {}

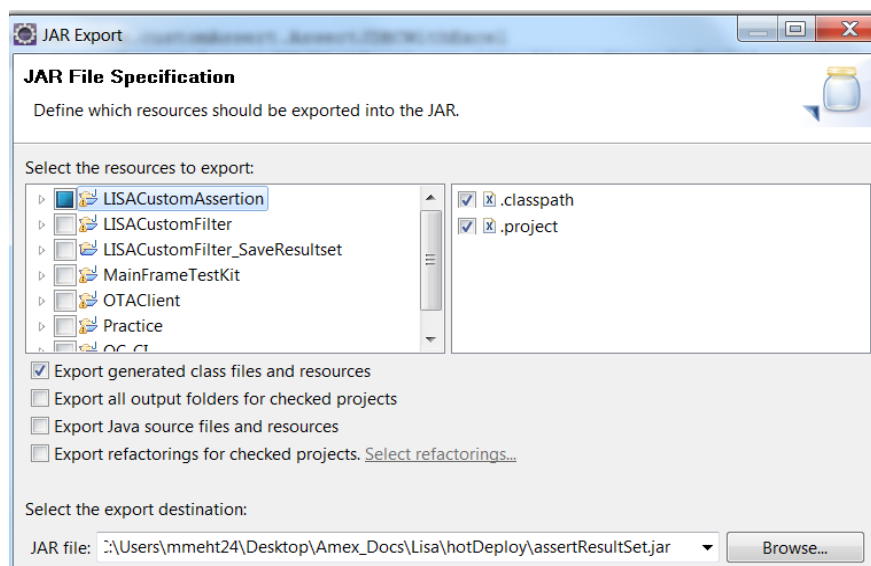
    public ParameterList getCustomParameters() {}

    protected boolean evaluate(TestExec ts, Object oresult) {}
}
```

6. Create .lisaextensions file in the same Project Folder and provide the assertion details as shown below:

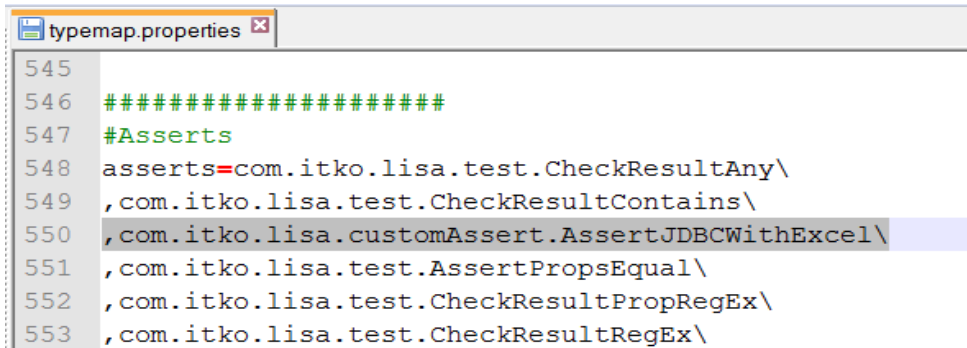
```
1 # LISA Extensions file
2
3 asserts=com.itko.lisa.customAssert.AssertJDBCWithExcel
4 com.itko.lisa.customAssert.AssertJDBCWithExcel=com.itko.lisa.editor.DefaultAssertController,com.itko.
```

7. Export the project into a jar file on your local system.



Instructions to Deploy a New Assertion

1. Copy the JAR file that contains custom assertion and lisaextensions file to the **LISA_HOME/hotDeploy** directory. If your custom assertion depends on any third-party libraries, copy those libraries to the LISA_HOME/hotDeploy directory.
2. Navigate to LISA_HOME and open the file “typemap.properties” with notepad. Navigate to Assertions section of the file and provide the class name with package name as shown below:

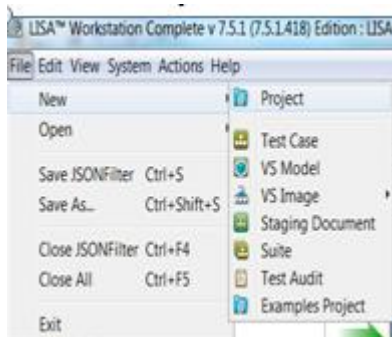


```
545
546 #####
547 #Asserts
548 asserts=com.itko.lisa.test.CheckResultAny\
549 ,com.itko.lisa.test.CheckResultContains\
550 ,com.itko.lisa.customAssert.AssertJDBCWithExcel\
551 ,com.itko.lisa.test.AssertPropsEqual\
552 ,com.itko.lisa.test.CheckResultPropRegEx\
553 ,com.itko.lisa.test.CheckResultRegEx\
```

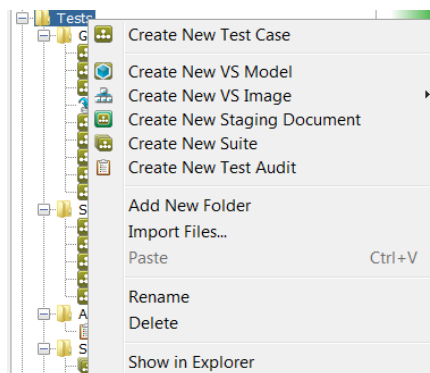
3. Restart LISA, if it is in running state.

Steps for Implementation:

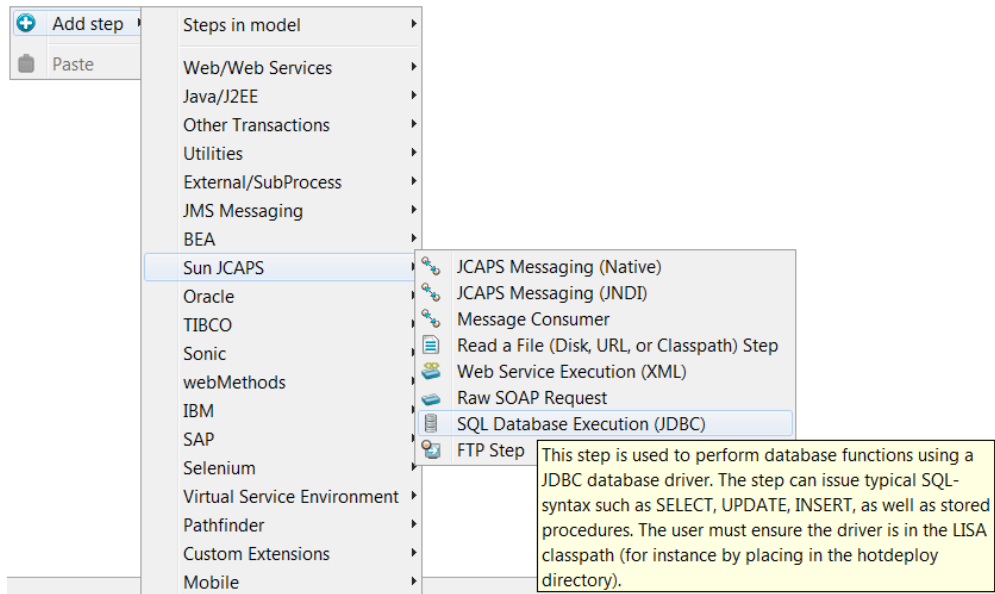
1. Create a Project in LISA workstation.



2. Create a Test Case.



3. Add a Test Step



4. Open the step and provide the required Details.

SQL Database Execution (JDBC) - SP Execution for Positive Scenario

Connection Info

JDBC Driver:

Connect String:

Max Rows to Fetch:

Execution Info

User ID:

Password:

☐ Keep Connection Open

☒ Use Connection Pool

☒ Returns Result Set

If SQL error:


SQL Statement

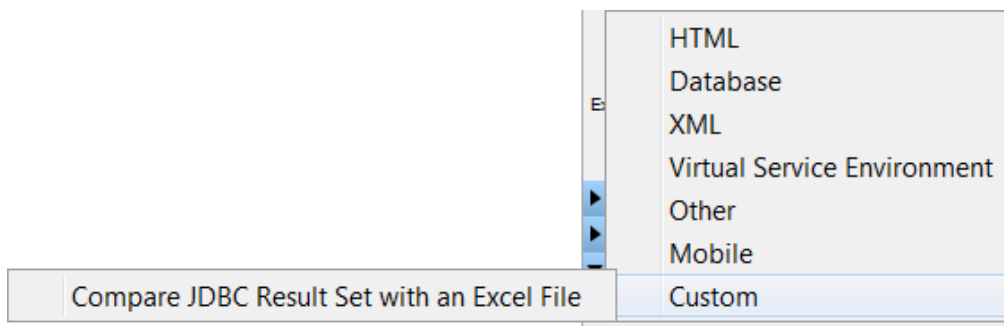
```
call SYSPROC.UTP.UTPRUN(? , ? , ? , ? , ?)
```

Parameter (?)	Type	Mode	Value
1	STRING	IN	{{SE_NO}}
2	INT	OUT	{{Actual_O_RC}}
3	INT	OUT	{{Actual_O_SQL_CODE}}
4	STRING	OUT	{{Actual_O_SQL_STATE}}
5	STRING	OUT	{{Actual_O_ERTX}}

Find:

Base Result Set

- On the right side, under Step Information, Click on  button under Assertions Section and select the assertion created under Custom Submenu.



- Open the Assertion and provide the values for parameters.

▼ Compare JDBC Result Set with an Excel File - Compare JDBC Result Set with an Excel File

Name: Compare JDBC Result Set with an Excel File If False then Fail the Test

Log: JDBC Result Set with an Excel File checks for: false is of type: Compare JDBC Result Set with an Excel File .

Run Assertion

Location: {{LISA_PROJ_ROOT}}/Data/Baseline Results/JZSEADDR/JZSEADDR_ResultSet.xls

Sheet Name: {{TC_ID}}

7. Click Start a new ITR and execute the Test Case to Test the Assertion.

References:

1. https://support.ca.com/cadocs/7/CA%20LISA%207%205%202-ENU/Bookshelf_Files/PDF/LISA_Developer_ENU_r7.5.2.pdf