LISA SDK: Create Your Own Filter

February 25, 2015

Revision History

Date	Version	Change History	Author	Reviewer
February 25, 2015	1.0	LISA SDK: Create Your Own Filter	Monika Mehta	Abhishek Mohan

Table of Contents

Description

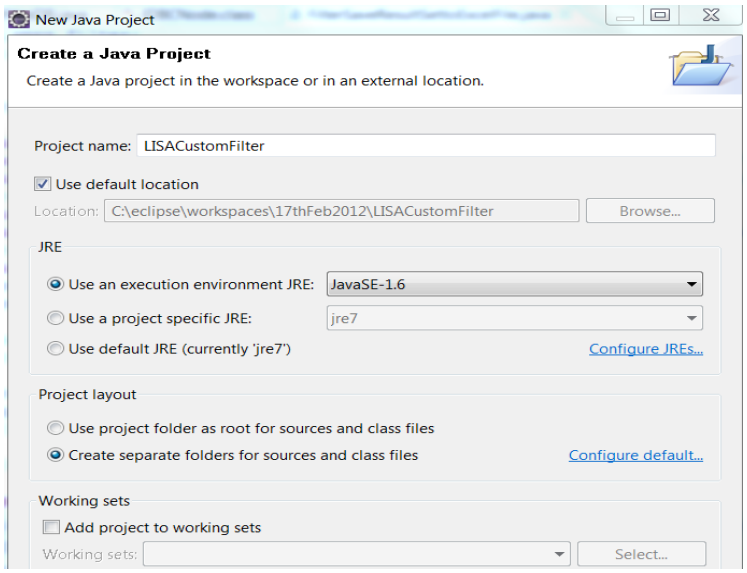
This document is intended to be used by any individual who wishes to create their own Filter to handle a specific situation. The LISA software provides built-in support for custom filters.

Pre-requisite

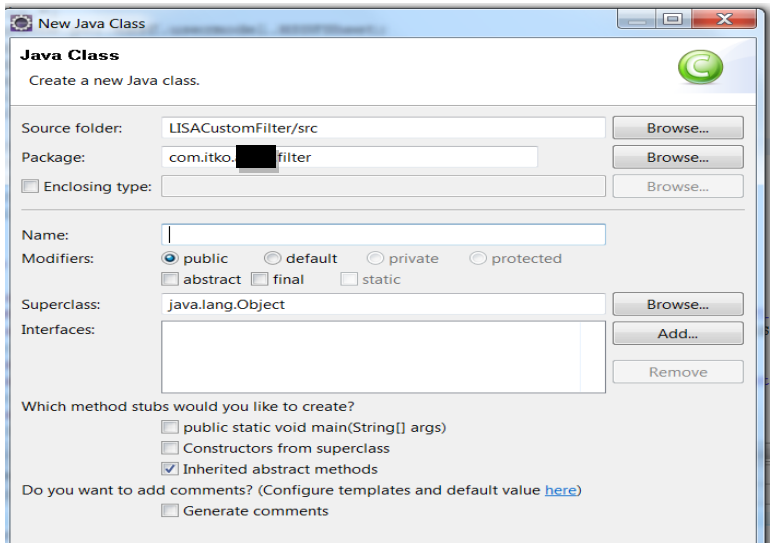
- Java IDE must be installed on machine.
- LISA must be installed on machine.

Steps to Create a Custom Filter

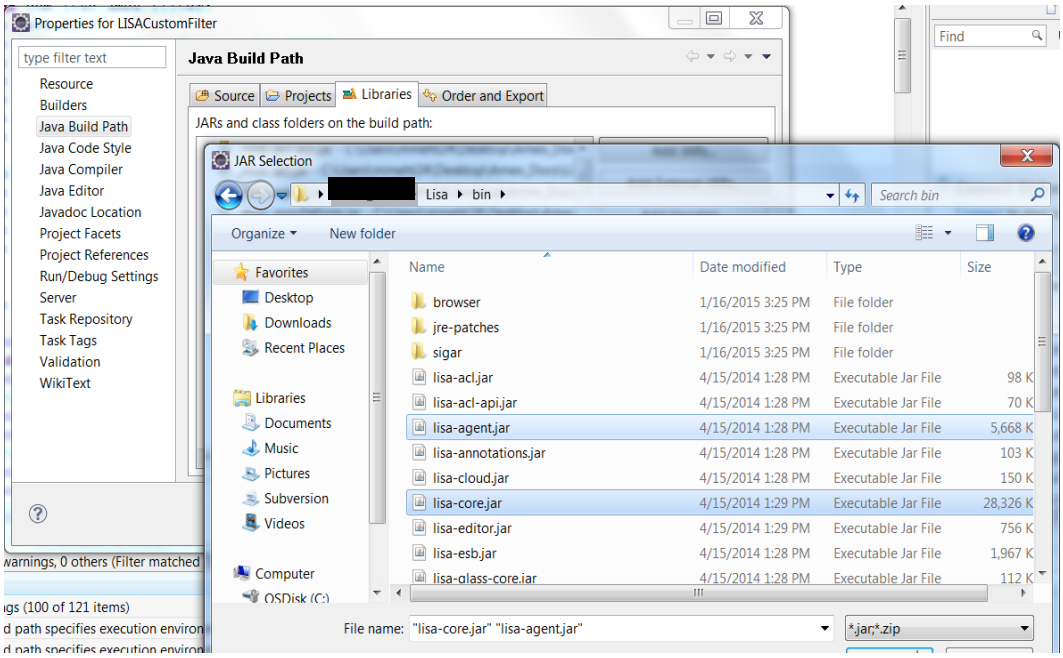
1. Open IDE and create a new Java Project.



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 **"FilterBaseImpl".**
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 filter in the model editor.

```
public String getTypeName()
{
        return "SQL to Result Set Filter";
}
```

   b. **getParameters** method: For each item in the Filter Attributes section of the Filters tab in the model editor, add a **Parameter** to the ParameterList for the filter.

```
public ParameterList getParameters()
{
        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 filter object with the value of the DOM Element.

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

d. **subPostFilter/subPreFilter** method: Because the filters execute before and after the test step, you get two chances to implement the filter logic. Implement the filter logic before node execution with the **subPreFilter** method. Implement the filter logic after node execution with the **subPostFilter** method.

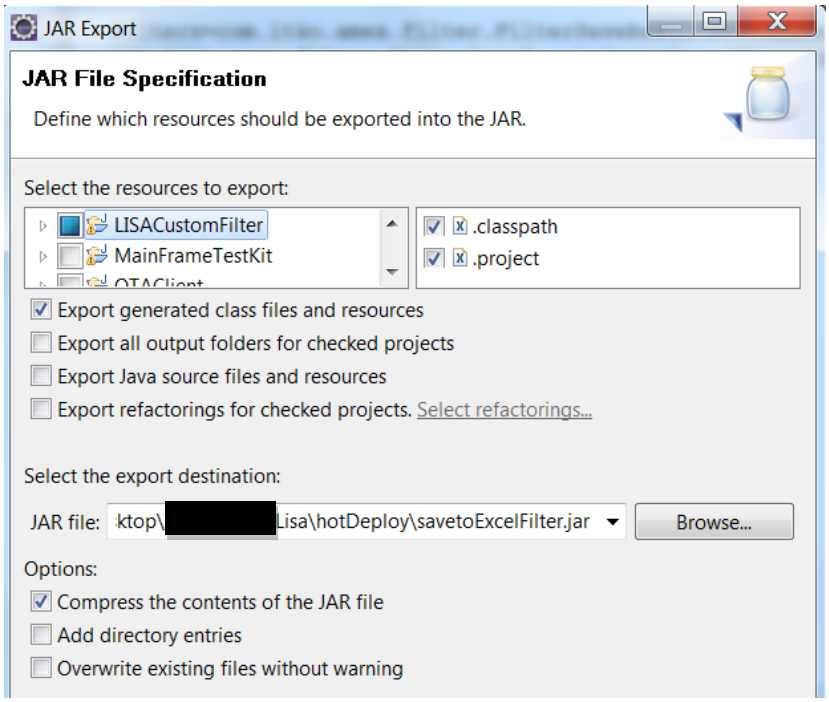
```
public boolean subPostFilter(TestExec testExec)
{
//Provide main Logic here
}
```

```
package com.itko.[redacted]filter;
import java.io.*;

/*     */
public class FilterSaveResultSettoExcelFile extends FilterBaseImpl{

    private static final long serialVersionUID = 1L;
    public static final String FILTER_TITLE = "Save ResultSet Value to an Excel File";
    private static final Logger LOGGER = LoggerFactory.getLogger(FilterSaveResultSettoExcelFile.class
    private static final String FILE_PARAM = "file";
    private static final String FILE_PARAM_DESC = ModuleLegacy.resources.get("test.fsavep2f.filedesc"
    private static final String Sheet_PARAM = "sheet_name";
    private String sheet_name;
    private String file;

    //private static final String FILE_PATH = "filePath";
    //private String filePath;

    public FilterSaveResultSettoExcelFile() {}
    public String getFile()

    public void setFile(String file)

    public String getTypeName()

     public ParameterList getParameters()

   public void initialize(Element e)

    public boolean subPostFilter(TestExec testExec)  {

    public boolean subPreFilter(TestExec arg0) throws TestRunException {
}
```

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

```
# LISA Extensions file

filters=com.itko.[redacted]ilter.FilterSaveResultSettoExcelFile
com.itko.[redacted].filter.FilterSaveResultSettoExcelFile=com.itko.lisa.editor.FilterController,com.itko.lisa.editor.DefaultFilterEditor
```

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



Instructions to Deploy a New Filter

1. Copy the JAR file that contains custom filter and lisaextensions file to the **LISA_HOME/hotDeploy** directory. If your custom filter 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 Filters section of the file and provide the class name with package name as shown below:

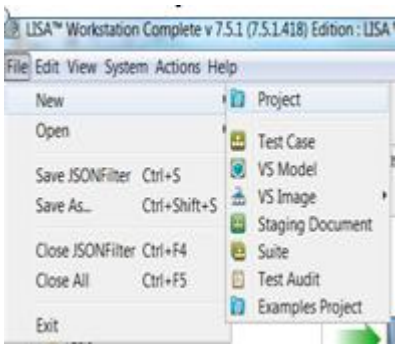
typemap.properties

```
634
635
636 ####################
637 #Filters
638 filters=com.itko.lisa.test.FilterParseOutByExpression\
639 ,com.itko.lisa.test.FilterSaveResponse\
640 ,com.itko.lisa.dynexec.FilterConvertObjToLastResponse\
641 ,com.itko.lisa.test.FilterSavePropToFile\
642 ,com.itko.citi.filter.JSON_to_XML\
643 ,com.itko.[redacted]x.filter.FilterSaveResultSettoExcelFile\
```

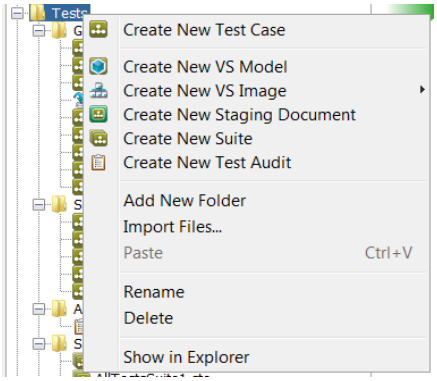
3. Restart LISA, if it is in running state.

Steps for Implementation:

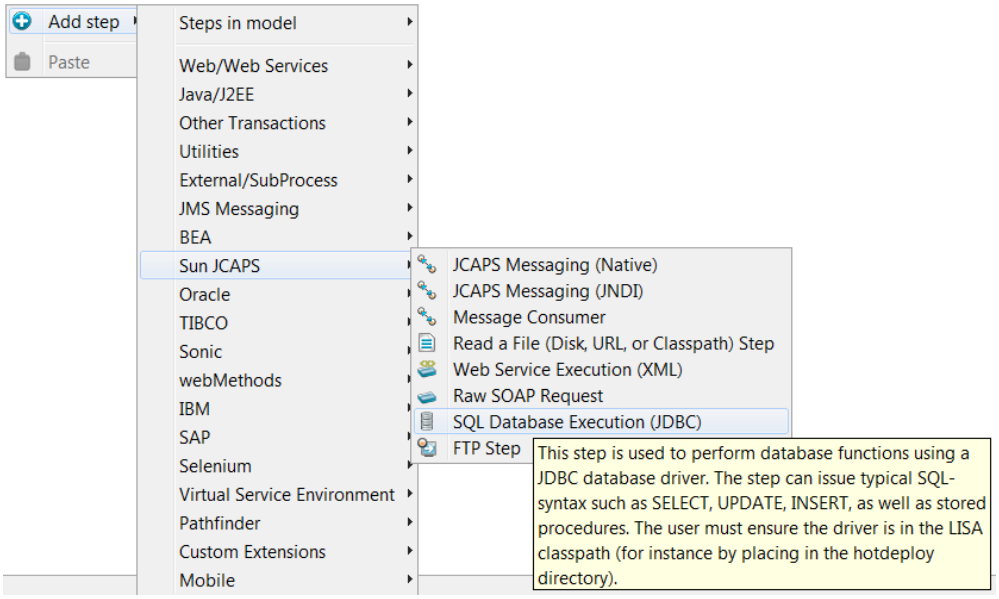
1. Create a Project in LISA workstation.



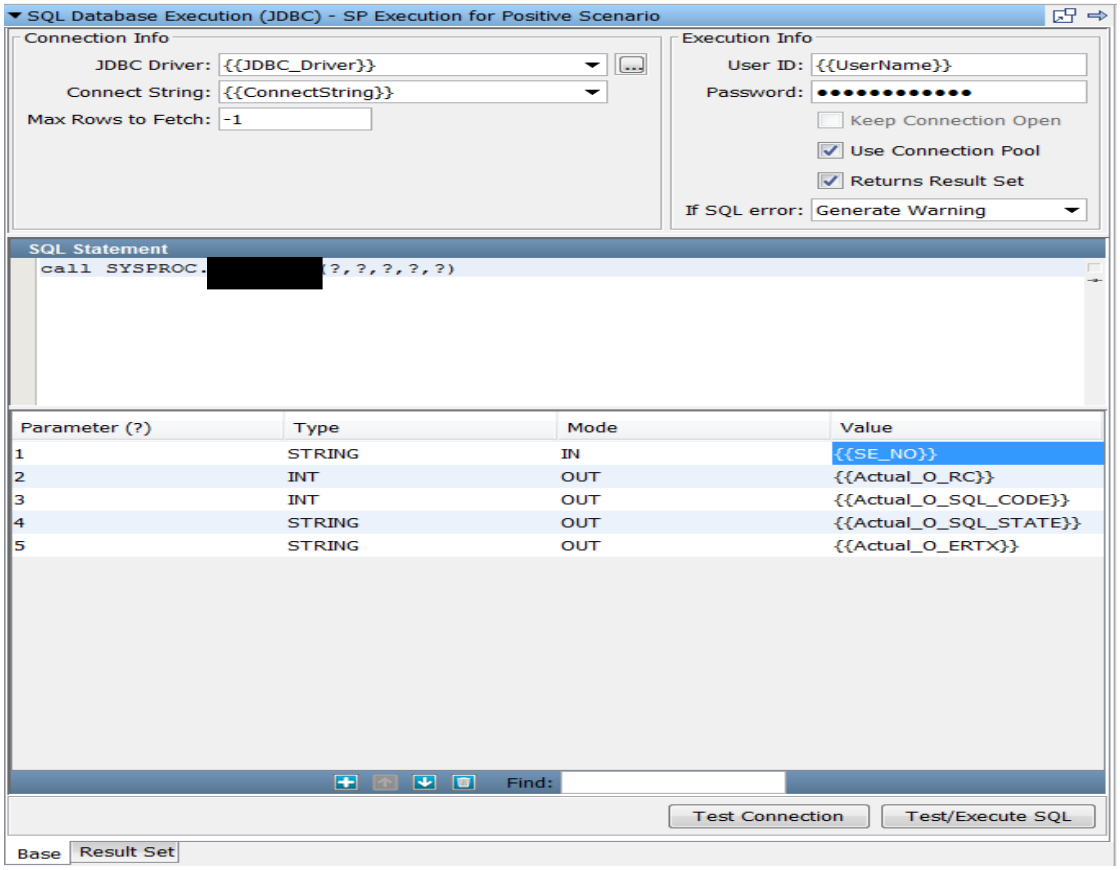
2. Create a Test Case.



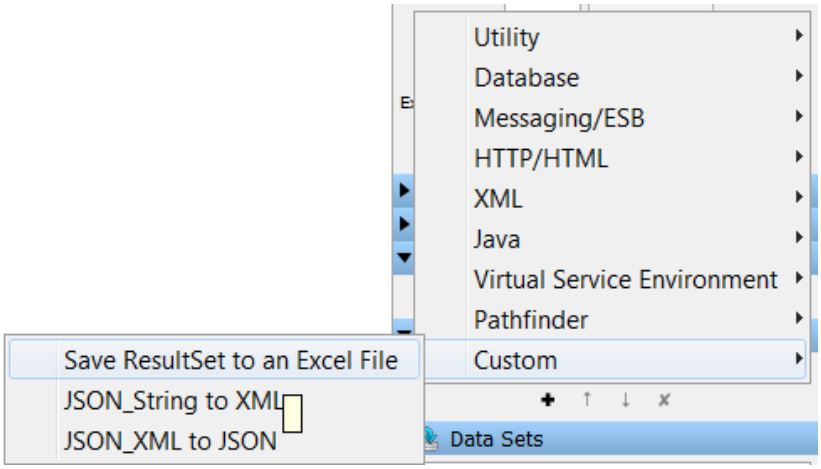
3. Add a Test Step



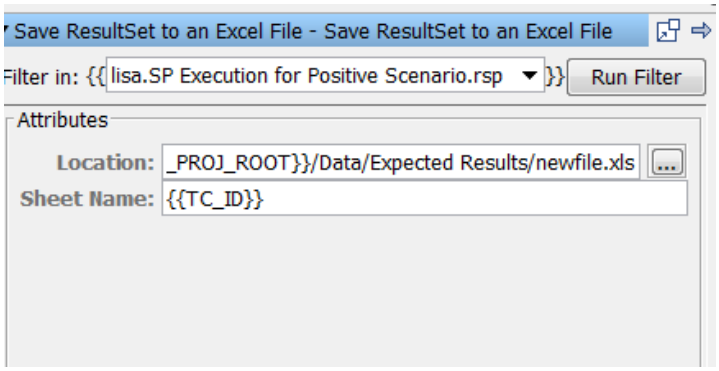
4. Open the step and provide the required Details.



5. On the right side, under Step Information, Click on ✚ button under Filters Section and select the filter created under Custom Filters Submenu.



6. Open the Filter and provide the values for parameters.



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

References:

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