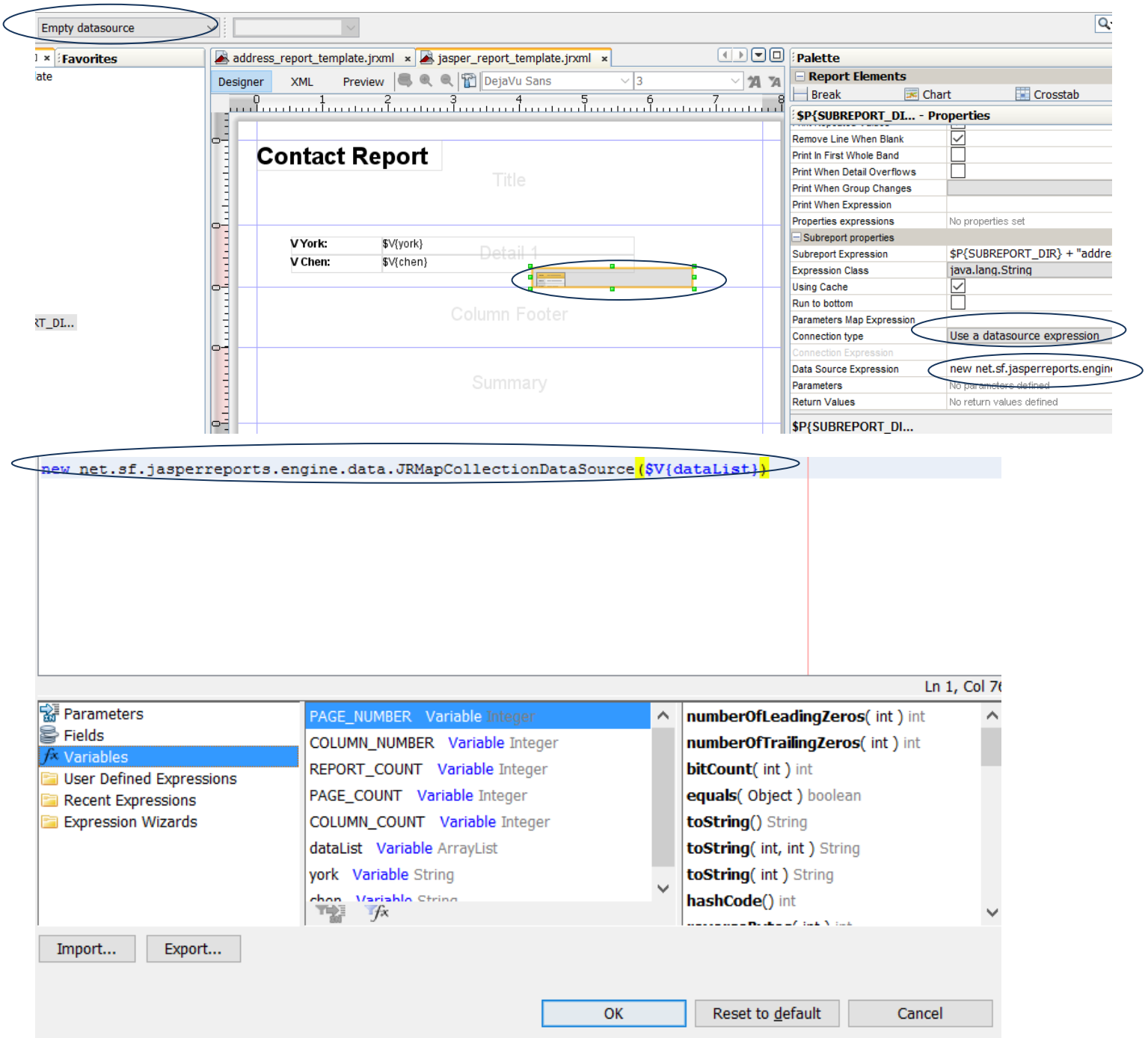
1. How to pass Parameters and Datasource from Master report to subreport:

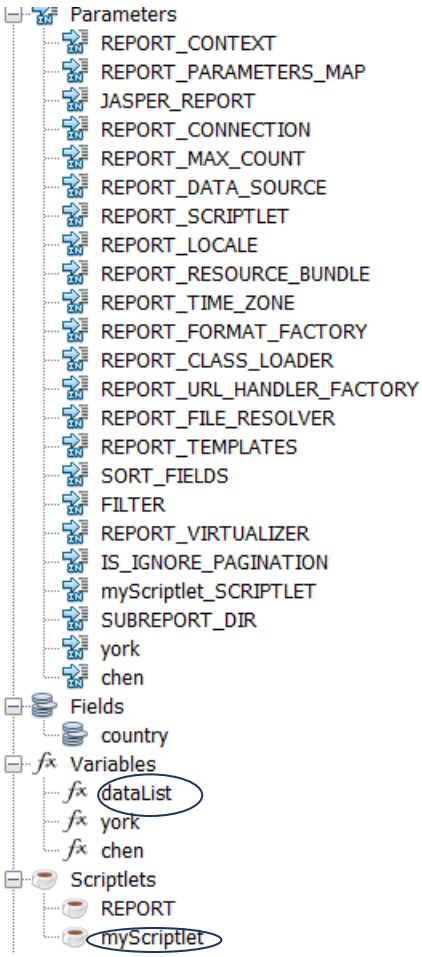
1.1. Subreport Expression:

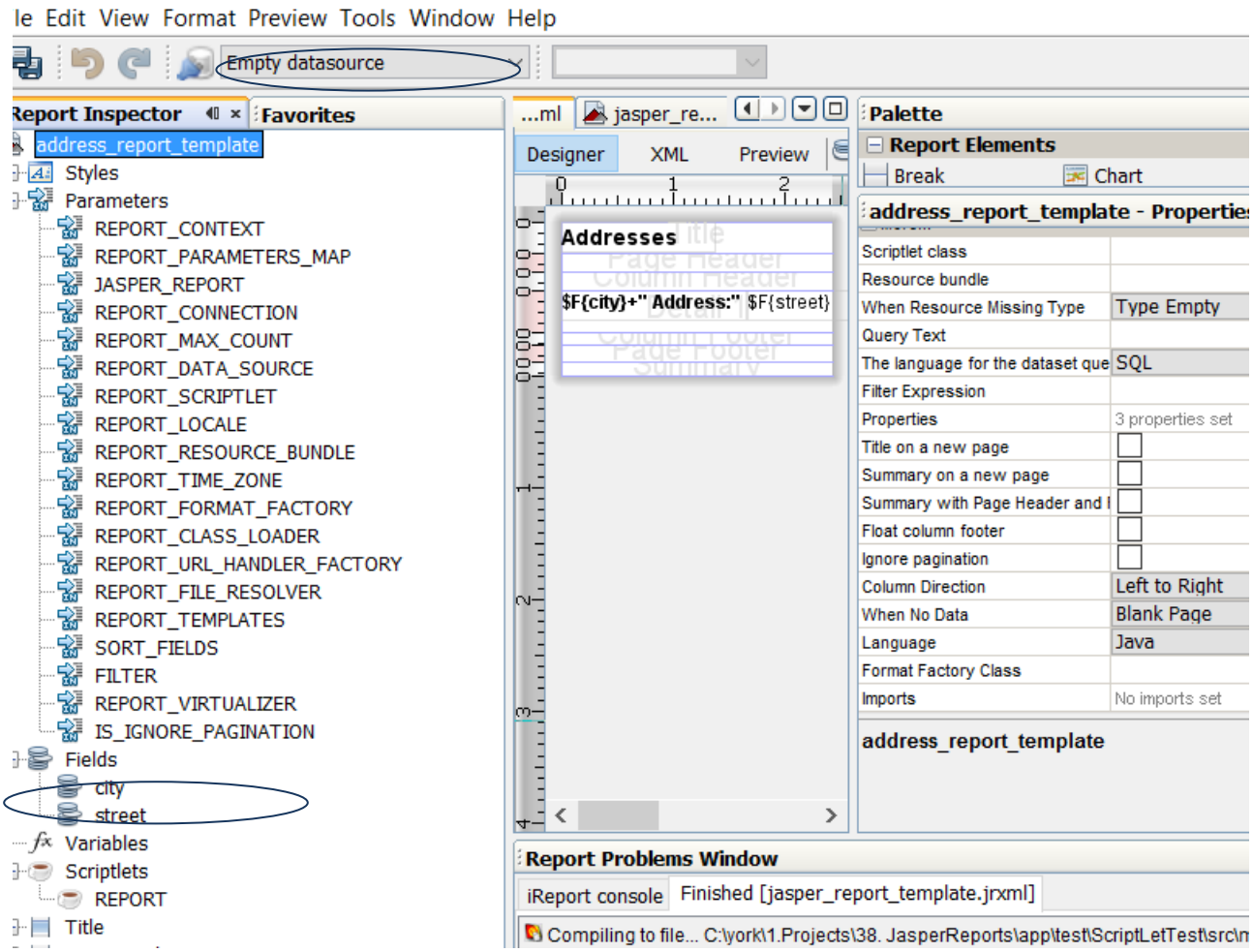
$P{SUBREPORT_DIR} + "address_report_template.jasper"

1.2. Data Source Expression:

new net.sf.jasperreports.engine.data.JRMapCollectionDataSource($V{dataList})




Parameters
REPORT_CONTEXT
REPORT_PARAMETERS_MAP
JASPER_REPORT
REPORT_CONNECTION
REPORT_MAX_COUNT
REPORT_DATA_SOURCE
REPORT_SCRIPTLET
REPORT_LOCALE
REPORT_RESOURCE_BUNDLE
REPORT_TIME_ZONE
REPORT_FORMAT_FACTORY
REPORT_CLASS_LOADER
REPORT_URL_HANDLER_FACTORY
REPORT_FILE_RESOLVER
REPORT_TEMPLATES
SORT_FIELDS
FILTER
REPORT_VIRTUALIZER
IS_IGNORE_PAGINATION
myScriptlet_SCRIPTLET
SUBREPORT_DIR
york
chen
Fields
country
Variables
dataList
york
chen
Scriptlets
REPORT
myScriptlet



1.3. Follow the steps below to pass parameters into subreports:

1. Create main report parameter, such as DATE_PARAM.
2. Open sub report and Create parameter with the same and and same type.
3. Go back to main report
4. Right-click on sub report, select properties
5. Choose parameter
6. Add parameter from main report to sub report with parameter name same parameter name

The parameter is passed from the main report to the subreport.

2. Use ScriptLets (ScriptLets can change variables.

(1) Set the sub report data source to variable:

new net.sf.jasperreports.engine.data.JRMapCollectionDataSource($V{dataList})

(2) Make sure the master datasource has at least one record.

For example, use following into SQL, but only use variables in the report

select 'name' as name, 'country' as country

(3) Put subreprot into detail section and re-set all variables in afterDetailEval()

2.1.

```
package org.york.jasper.jasperapp;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import net.sf.jasperreports.engine.JRDefaultScriptlet;
import net.sf.jasperreports.engine.JRScriptletException;

public class MyScriptlet extends JRDefaultScriptlet {
	// List<Map<String, ?>> dataList = new ArrayList<Map<String, ?>>();

	public void getDataList() {
		List<Map<String, ?>> maps = new ArrayList<Map<String, ?>>();
		Map<String, Object> map = new HashMap<String, Object>();
		map.put("city", "Beijing99");
		map.put("street", "11 Wulidian00");
		maps.add(map);

		map = new HashMap<String, Object>();
		map.put("city", "Beijing299");
		map.put("street", "22 Wulidian200");
		maps.add(map);

		try {
			this.setVariableValue("dataList", maps);
		} catch (Exception e) {
			e.printStackTrace();
		}

	}

	public void updateVariables() throws JRScriptletException {
		System.out.println("call afterReportInit()");

		String york = (String) this.getParameterValue("york");
		String chen = (String) this.getParameterValue("chen");

		System.out.println("updateVariables, v york=" + york);
		System.out.println("updateVariables, v chen=" + chen);
		this.setVariableValue("york", new String("updateVariables, V york="
				+ york));
		this.setVariableValue("chen", new String("updateVariables, V chen="
				+ chen));

		getDataList();
	}

	public String hello() throws JRScriptletException {
		return "Hello! I'm the report's scriptlet object.";
	}

	public void beforeReportInit() throws JRScriptletException {
		System.out.println("call beforeReportInit()");
	}
```

```
	public void afterReportInit() throws JRScriptletException {
		System.out.println("call afterReportInit()");
	}

	public void beforePageInit() throws JRScriptletException {
		System.out.println("call beforePageInit()");
	}

	public void afterPageInit() throws JRScriptletException {
		System.out.println("call afterPageInit()");
	}

	public void beforeColumnInit() throws JRScriptletException {
		System.out.println("call beforeColumnInit()");
	}

	public void afterColumnInit() throws JRScriptletException {
		System.out.println("call afterColumnInit()");
	}

	public void beforeDetailEval() throws JRScriptletException {
		System.out.println("call beforeDetailEval()");
	}

	public void afterDetailEval() throws JRScriptletException {
		System.out.println("call afterDetailEval()");
		updateVariables();
	}

}
```

2.2. Put subreport into detail section

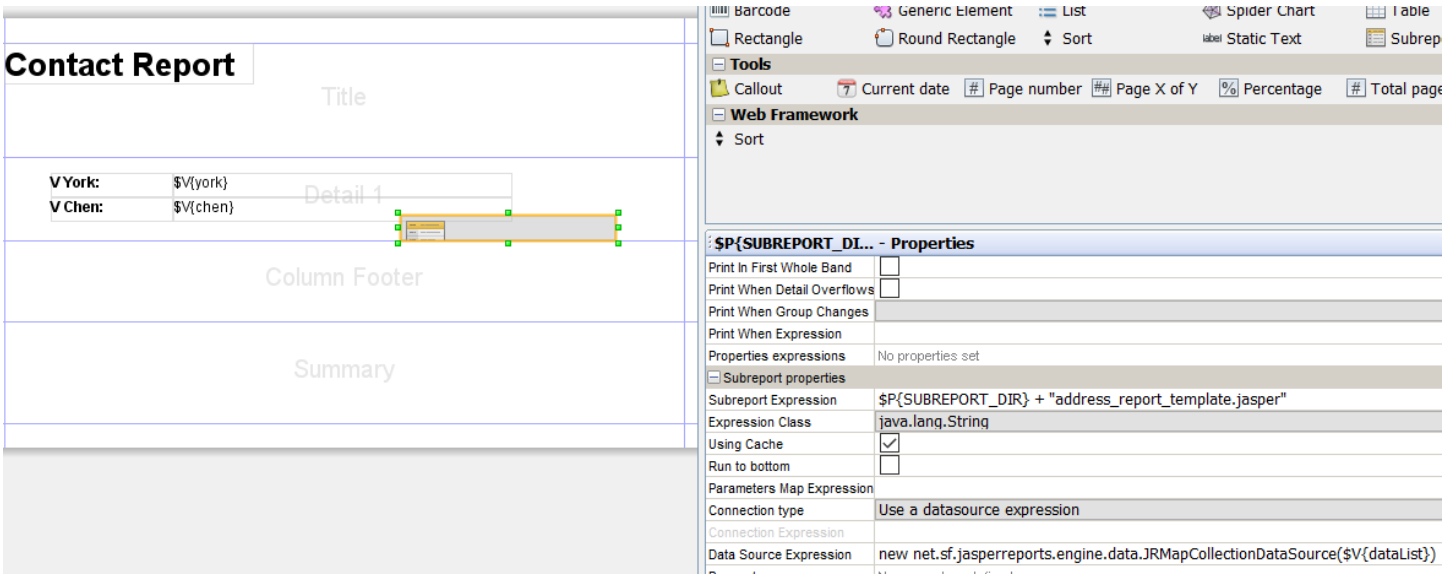
```
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import org.york.jasper.jasperapp.*;

import net.sf.jasperreports.engine.JRDataSource;
import net.sf.jasperreports.engine.JRException;
import net.sf.jasperreports.engine.JasperCompileManager;
import net.sf.jasperreports.engine.JasperExportManager;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperReport;
import net.sf.jasperreports.engine.data.JRBeanCollectionDataSource;
import net.sf.jasperreports.engine.data.JRMapCollectionDataSource;

public class CreateReport {

	public static void main(String[] args) {
		String path = "C:\\york\\1.Projects\\38.
JasperReports\\app\\test\\ScriptLetTest\\src\\main\\resources";
```

```
		// String masterReportFileName = path + "/jasper_report_template.jrxml";
		// String subReportFileName = path + "/address_report_template.jrxml";
		String destFileName = path + "/jasper_report_template.jrprint";
		String jasperMasterReport = path + "/jasper_report_template.jasper";
		String pdfName = path + "/jasper_report_template.pdf";

		try {
			/* Compile the master and sub report */
			// JasperReport jasperMasterReport =
			// JasperCompileManager.compileReport(masterReportFileName);
			// JasperReport jasperSubReport =
			// JasperCompileManager.compileReport(subReportFileName);

			Map<String, Object> parameters = new HashMap<String, Object>();
			// parameters.put("subreportParameter", null);
			parameters.put("chen", "chen1");
			parameters.put("york", "york1");

			List<Map<String, ?>> maps = new ArrayList<Map<String, ?>>();
			for (int i = 0; i < 1; i++) {
				Map<String, Object> map = new HashMap<String, Object>();
				map.put("name", "Shoud not be here");
				map.put("country", "Shoud not be here");
				maps.add(map);
			}
			JRMapCollectionDataSource dataSource = new JRMapCollectionDataSource(
					maps);

			JasperFillManager.fillReportToFile(jasperMasterReport,
					destFileName, parameters, dataSource);

			export(destFileName, pdfName);

			System.out.println("PDF created...");
		} catch (JRException e) {

			e.printStackTrace();
		}
		System.out.println("Done filling!!! ...");
	}

	public static void export(String sourceFileName, String pdfName) {
		try {
			// 1- export to PDF
			JasperExportManager.exportReportToPdfFile(sourceFileName, pdfName);
			// exporter.exportReport();
```

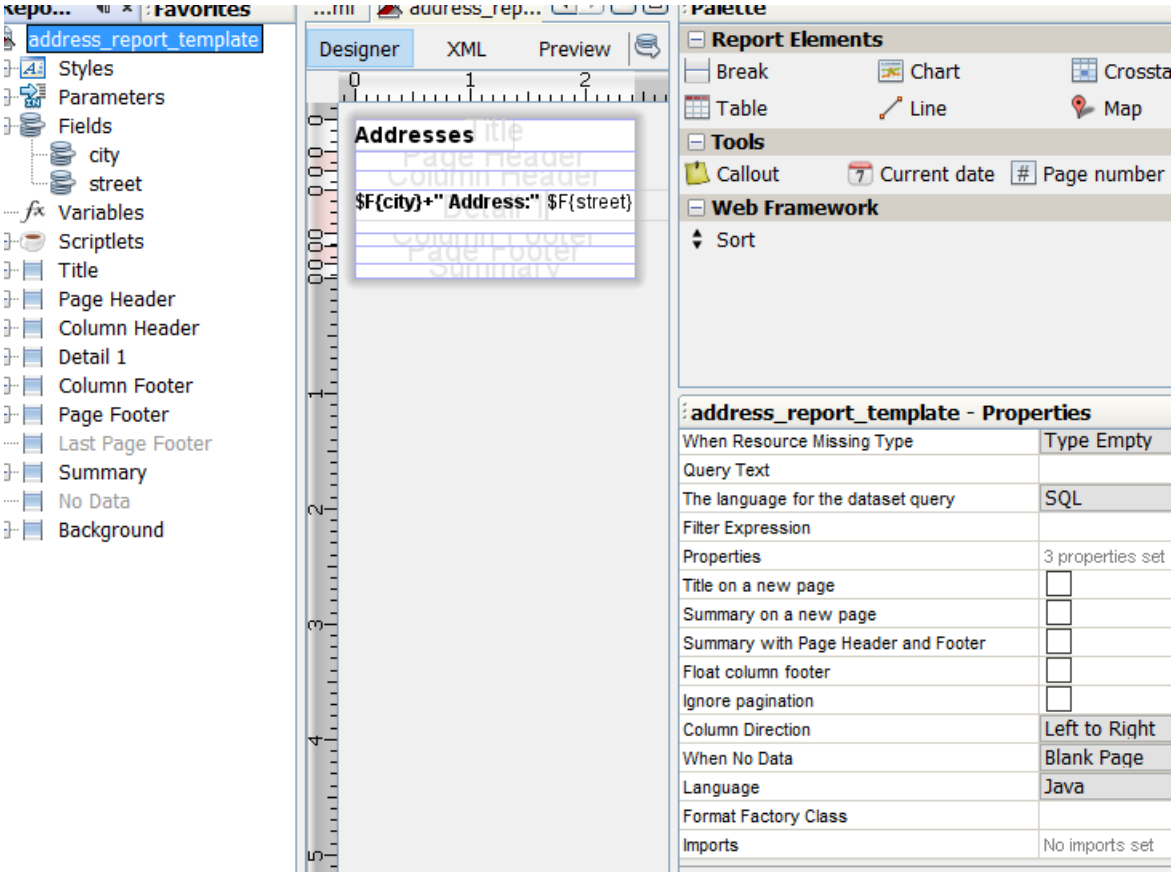
```
                } catch (JRException e) {
                        e.printStackTrace();
                }
        }

}
```

2.3.

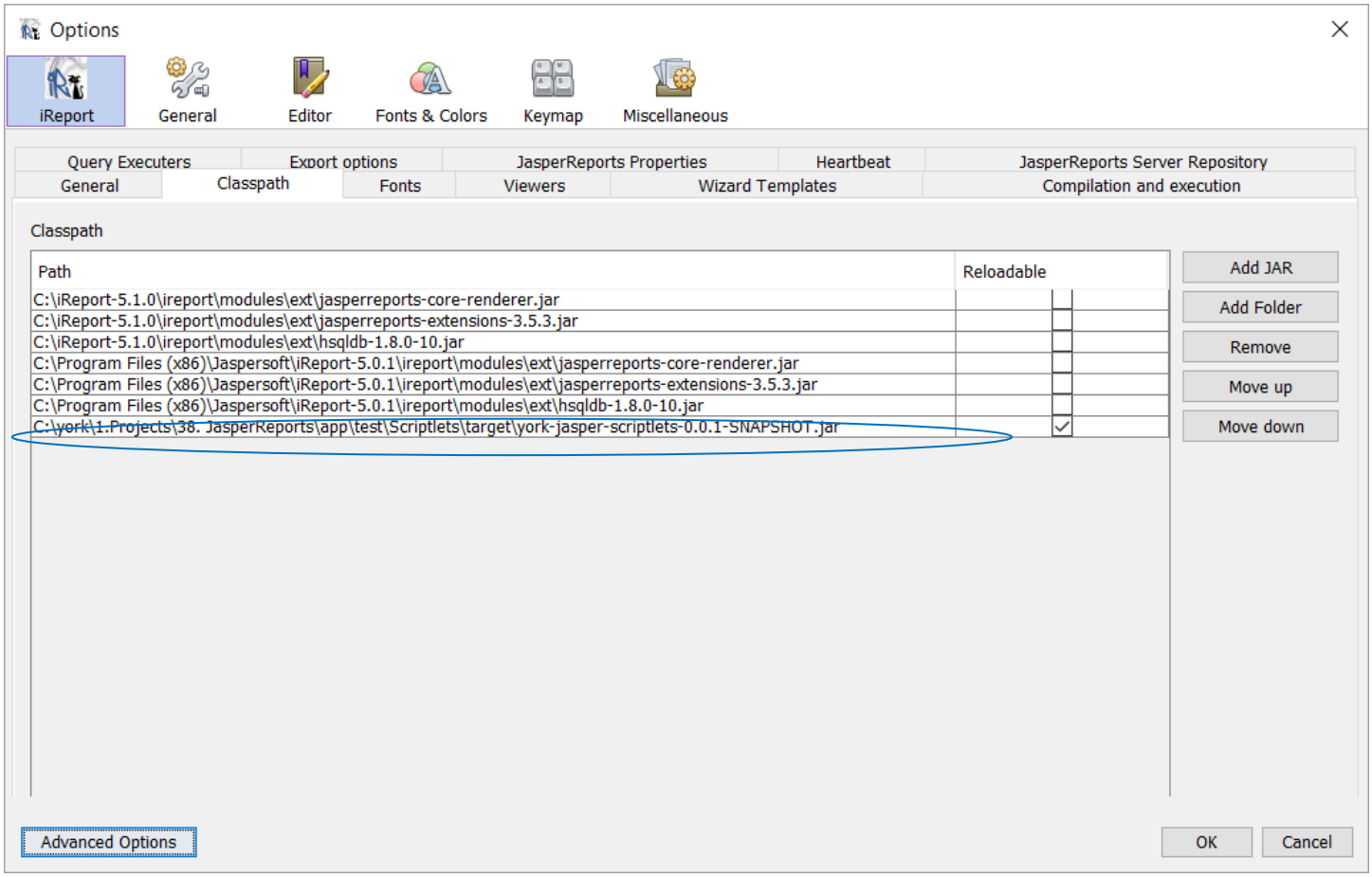
Master report:



2.4. Subreport:



For testing in iReport, need to add the Jar file into class path:


Options
iReport
General
Editor
Fonts & Colors
Keymap
Miscellaneous
Query Executers
Export options
JasperReports Properties
Heartbeat
JasperReports Server Repository
General
Classpath
Fonts
Viewers
Wizard Templates
Compilation and execution
Classpath
Path
Reloadable
C:\iReport-5.1.0\ireport\modules\ext\jasperreports-core-renderer.jar
C:\iReport-5.1.0\ireport\modules\ext\jasperreports-extensions-3.5.3.jar
C:\iReport-5.1.0\ireport\modules\ext\hsqldb-1.8.0-10.jar
C:\Program Files (x86)\Jaspersoft\iReport-5.0.1\ireport\modules\ext\jasperreports-core-renderer.jar
C:\Program Files (x86)\Jaspersoft\iReport-5.0.1\ireport\modules\ext\jasperreports-extensions-3.5.3.jar
C:\Program Files (x86)\Jaspersoft\iReport-5.0.1\ireport\modules\ext\hsqldb-1.8.0-10.jar
C:\york\1.Projects\38. JasperReports\app\test\Scriptlets\target\york-jasper-scriptlets-0.0.1-SNAPSHOT.jar
Add JAR
Add Folder
Remove
Move up
Move down
Advanced Options
OK
Cancel