

## JasperReports介绍及VOP集成分析

目前 VOP 集成 JasperReports 现状：

1. 可以较轻松的进行复杂报表（具有行合并列合并的报表）的功能开发。
2. 支持变参查询。
3. 对于报表的个性化样式，html 展示时需要在 JS 手动控制，导出 Excel 时的样式也需要在导出时的 POI 对象中手动控制。
4. 不支持分页。

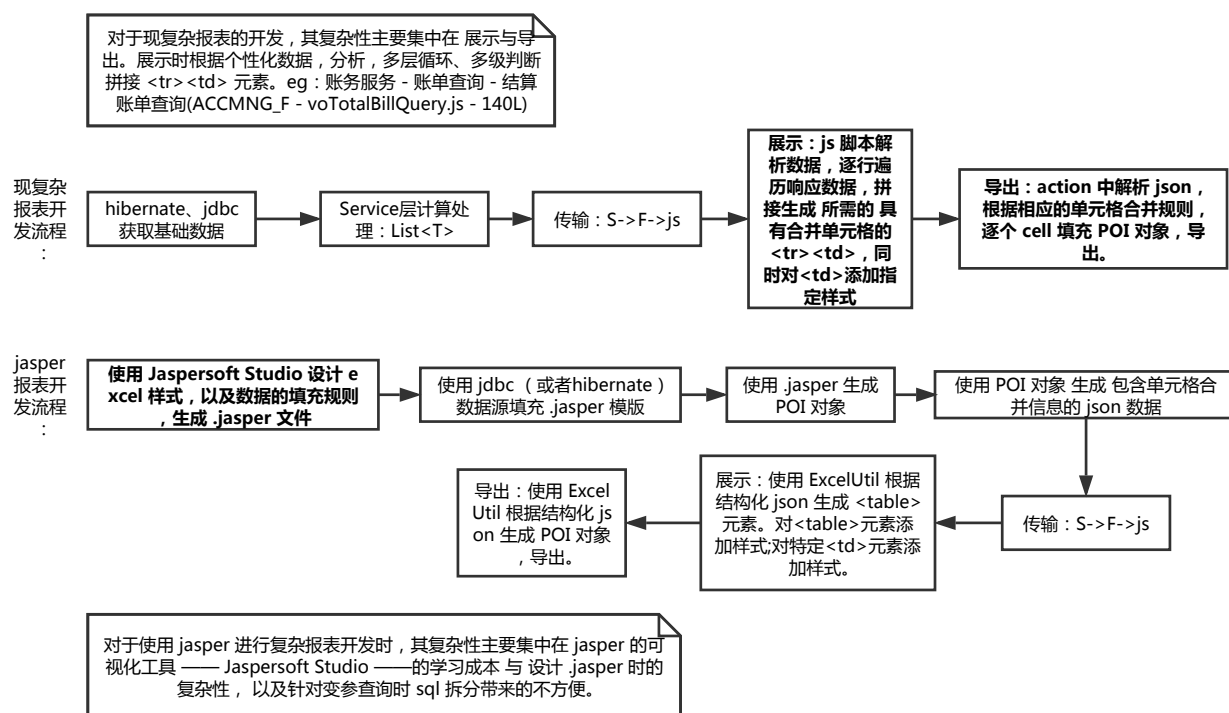
下一步优化完善思路：

1. 对样式进行封装，避免在 JS 中操作报表。建议分析一下 jasperreports HtmlExporter 源码，service 层的开发过程可能也需要进一步封装。
2. 对分页功能进行统一的封装，考虑到行合并 和 列合并，此功能的实现可能会略显复杂。而目前对于没有行合并的报表，data-table.js 对其已经有较好支持。

## JasperReports介绍：

**JasperReports**是一个用Java开发的开源的程式库，使用户能够透过它，利用Java语言来开发具有报告功能的程式。JasperReports的模板采用XML格式，从JDBC数据库中撷取合适的资料，并把资料在屏幕、打印机显示，或以PDF、HTML、XLS、CSV及XML等各种格式储存。

## VOP 集成 JasperReports 分析：



### 现复杂报表开发示例：

账务服务 -> 账单查询 -> 结算账单查询  
ACCMNG\_F -> voTotalBillQuery.js -> 140L ...

### jasper复杂报表开发示例：

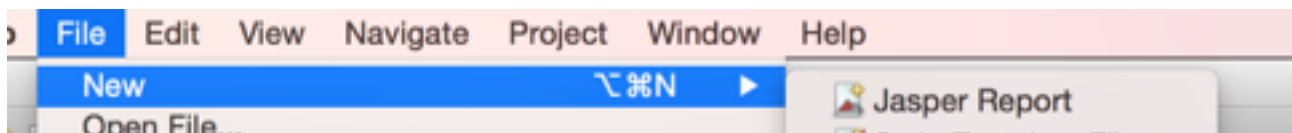
mvno\_inst\_subscribe\_analyse.jrxml  
mvno\_inst\_subscribe\_analyse\_S\_inst.jrxml

运营服务 -> 转售企业管理-Test -> 转售企业指令统计报表  
VOPMNG\_F -> mvnoInstStatistics.js -> 59L ...

使用 Jasper 设计具有行合并、列合并报表：

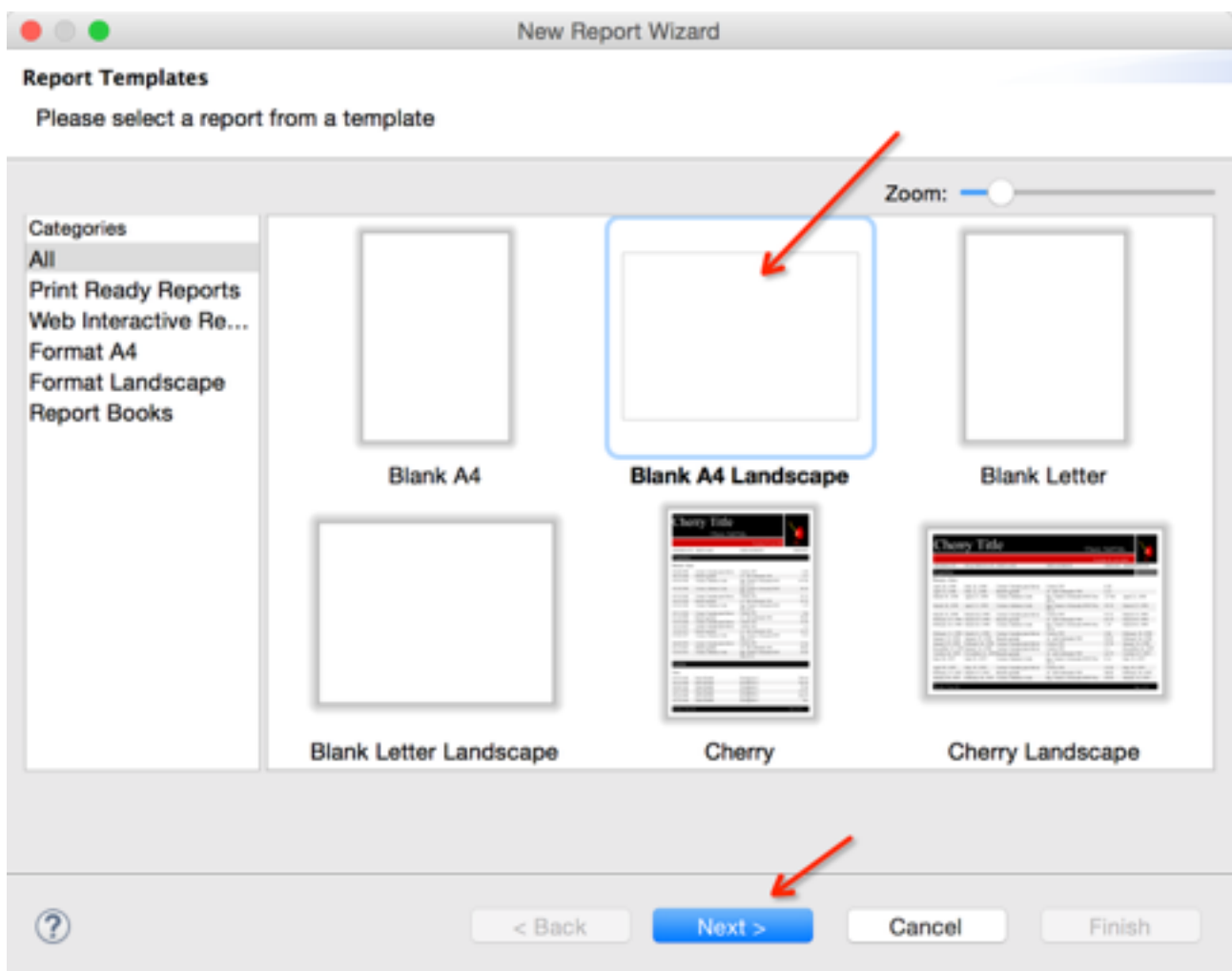
这里以实例的方式介绍一下使用jasper制作报表并在VOP中集成的过程。  
eg: “统计转售企业的指令订购数”

#### 1. 安装JasperReports

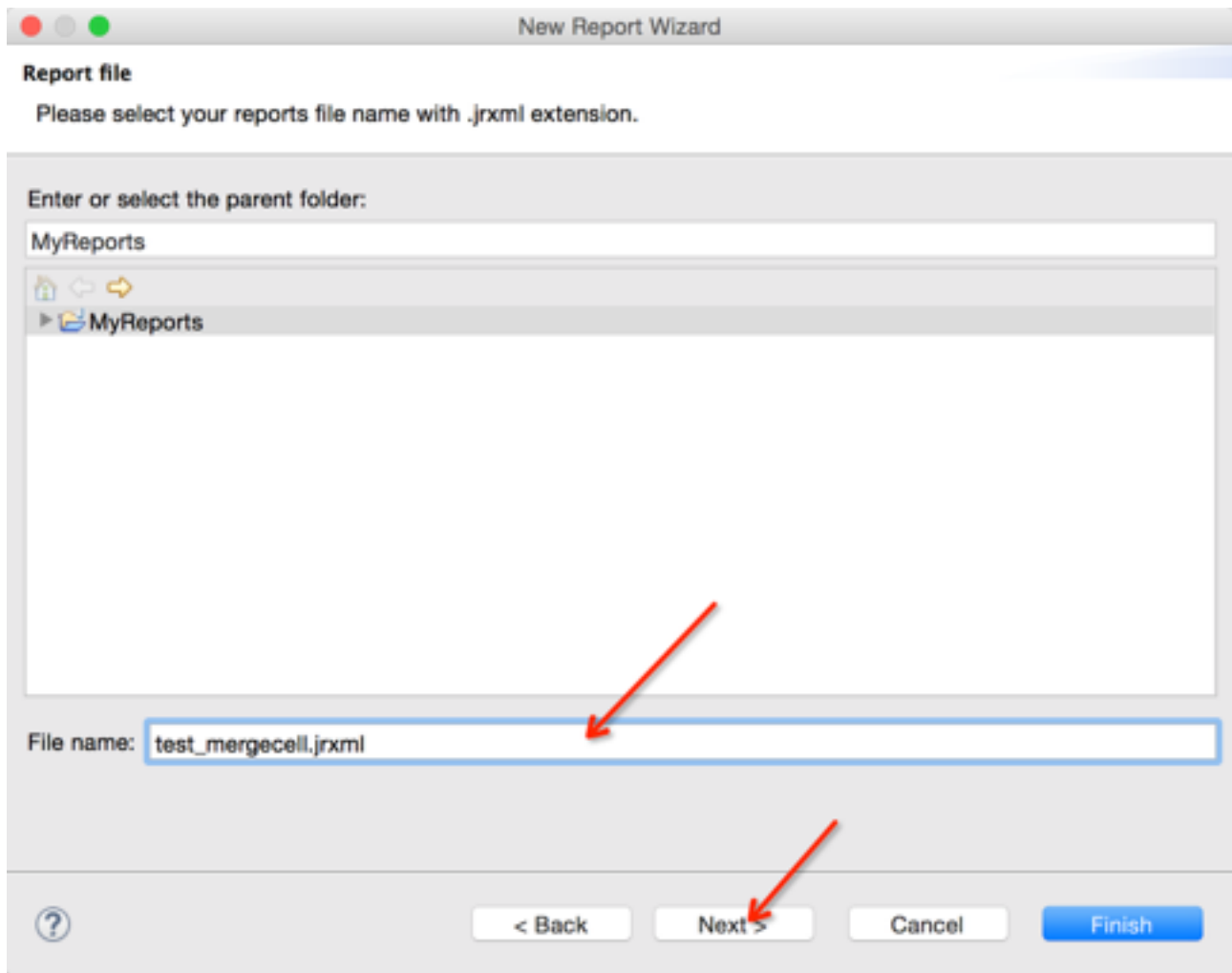


#### 2. 新建 Jasper Report

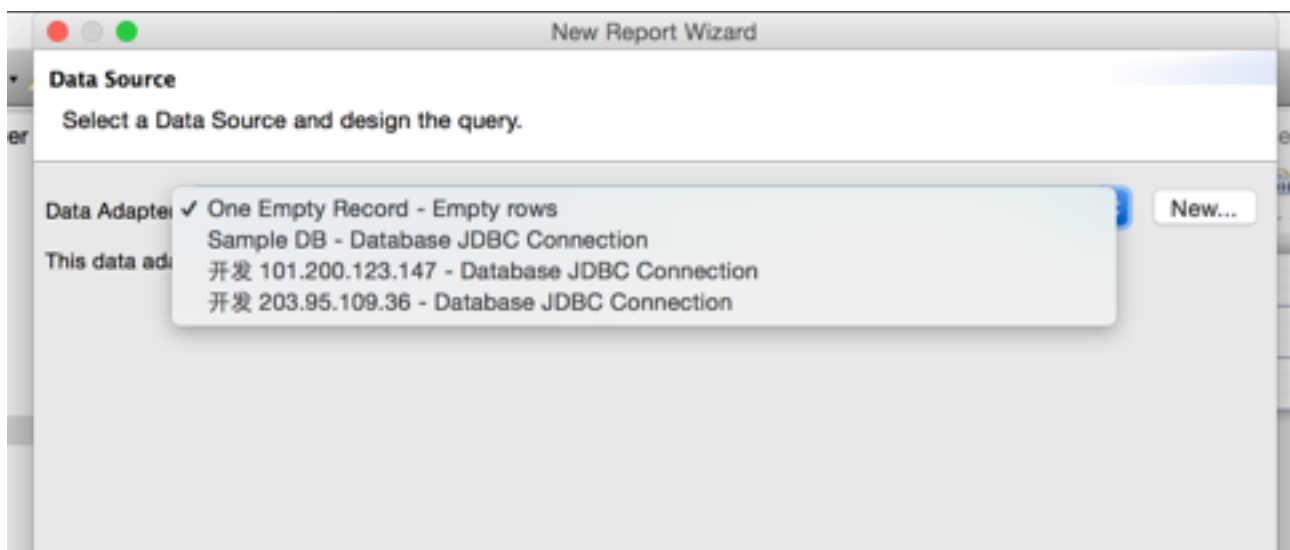
File -> New -> JasperReport



-> Next  
-> Next



选择数据源 101.200.123.147



详见：Jaspersoft Studio如何添加数据源

New Report Wizard

**Data Source**

Select a Data Source and design the query.

Data Adapter 开发 101.200.123.147 - Database JDBC Connection New...

☐ APEX\_PUBLIC\_USER  
☐ APPQOSSYS  
☐ BILL  
☐ BILLING  
☐ BSBSYNC  
☐ CHENNAN  
☐ DBBPM  
☐ DBBSDM  
☐ DBSNMP  
☒ DBVOP (CURRENT)  
☐ DBVOPLOG  
☐ DBVOPSTSC  
☐ DIP  
☐ EXPUSER  
☐ MDDATA  
☐ ORACLE\_OCM  
☐ OUTLN

1 select \* from mvno\_business

Texts Outline Diagram

? < Back Next > Cancel Finish

查询 sql : select \* from mvno\_business

SQL过会儿可以根据需要再调整，注意结尾处不能加分号“;”

New Report Wizard

**Fields**

Please select dataset fields

Dataset Fields

- ☐ MVNO\_NO
- ☐ CONTACTOR
- ☐ CONTACT\_PHONE
- ☐ EMAIL
- ☐ MVNO\_ADDR
- ☐ UPD\_TIME
- ☐ DESCRIBES
- ☐ IMIS\_RATIO
- ☐ IMSI\_RATE\_BEGIN\_TIME
- ☐ IMSI\_RATE\_END\_TIME
- ☐ VALID\_FLAG

> >> < <<

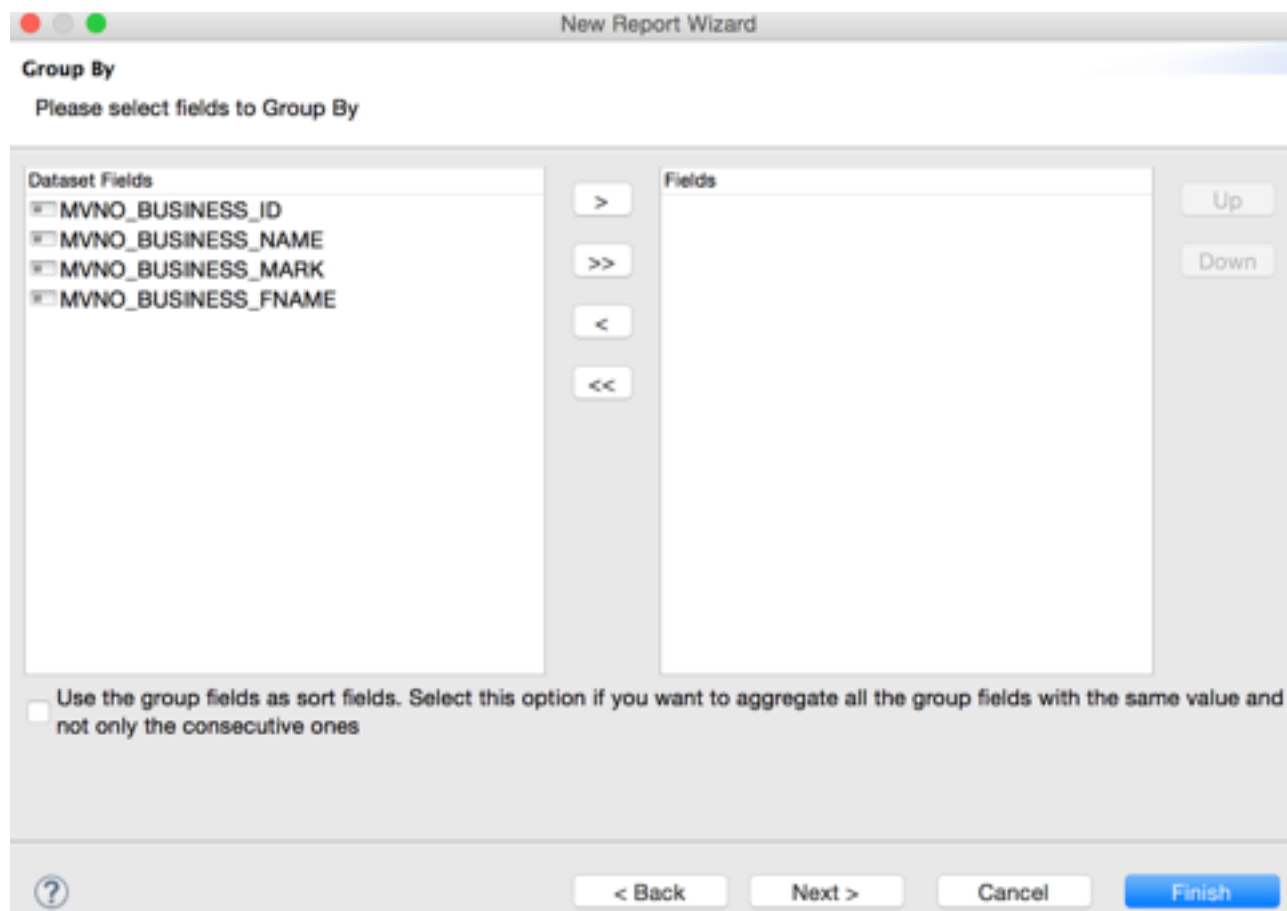
Fields

- ☐ MVNO\_BUSINESS\_ID
- ☐ MVNO\_BUSINESS\_NAME
- ☐ MVNO\_BUSINESS\_MARK
- ☐ MVNO\_BUSINESS\_FNAME

Up Down

? < Back Next > Cancel Finish

字段过会儿根据需要再调整。



The image shows a 'New Report Wizard' window with a 'Group By' section. The title bar says 'New Report Wizard'. Below the title, it says 'Group By' and 'Please select fields to Group By'. There are two main panes: 'Dataset Fields' on the left and 'Fields' on the right. The 'Dataset Fields' pane contains a list of fields: MVNO\_BUSINESS\_ID, MVNO\_BUSINESS\_NAME, MVNO\_BUSINESS\_MARK, and MVNO\_BUSINESS\_FNAME. Between the panes are four buttons: '>', '>>', '<', and '<<'. The 'Fields' pane is currently empty. To the right of the 'Fields' pane are two buttons: 'Up' and 'Down'. Below the panes, there is a checkbox labeled 'Use the group fields as sort fields. Select this option if you want to aggregate all the group fields with the same value and not only the consecutive ones'. At the bottom of the window, there is a row of buttons: a help button (question mark icon), '< Back', 'Next >', 'Cancel', and a blue 'Finish' button.

New Report Wizard

**Group By**

Please select fields to Group By

Dataset Fields

- ☐ MVNO\_BUSINESS\_ID
- ☐ MVNO\_BUSINESS\_NAME
- ☐ MVNO\_BUSINESS\_MARK
- ☐ MVNO\_BUSINESS\_FNAME

> >> < <<

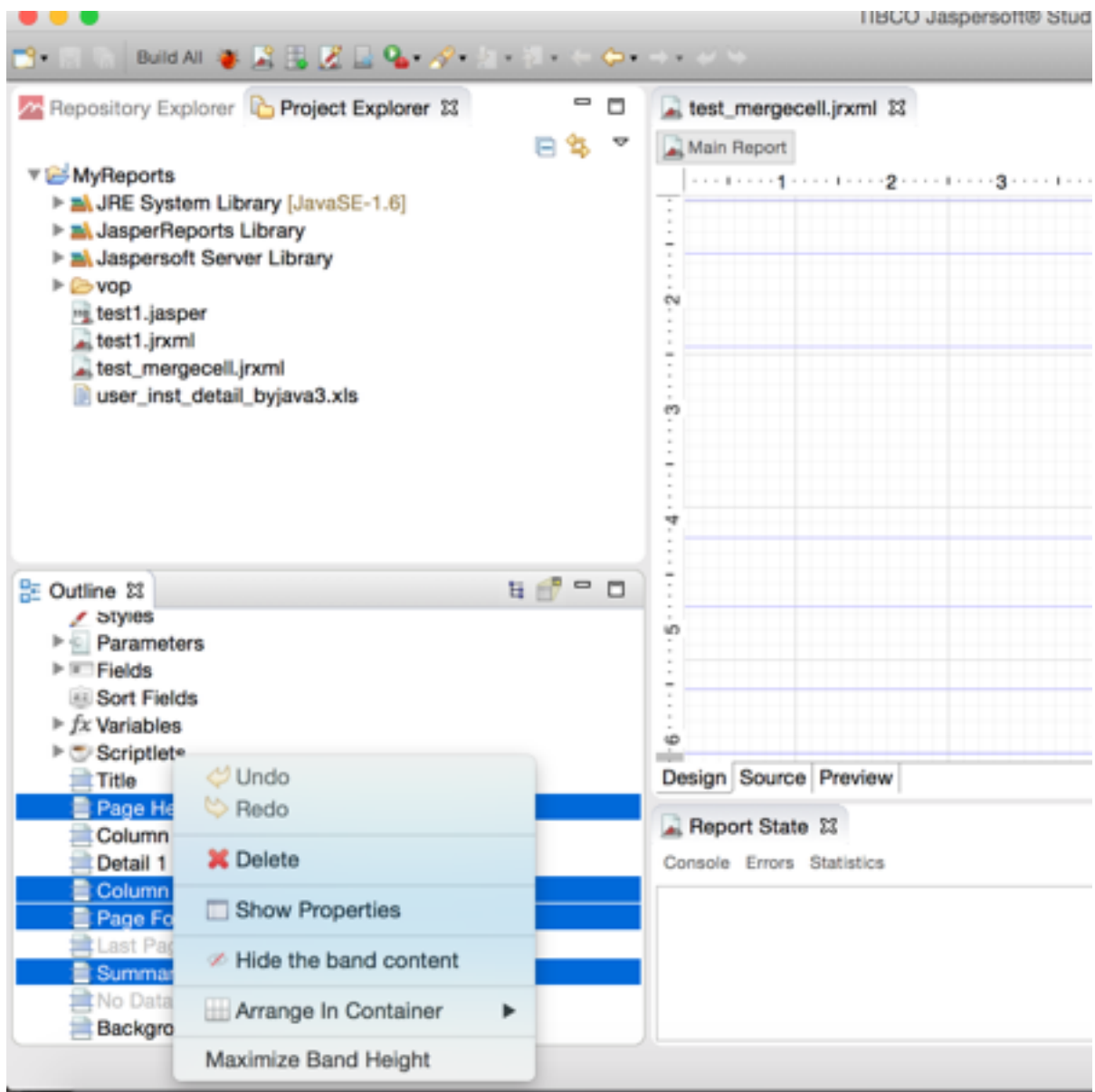
Fields

Up Down

☐ Use the group fields as sort fields. Select this option if you want to aggregate all the group fields with the same value and not only the consecutive ones

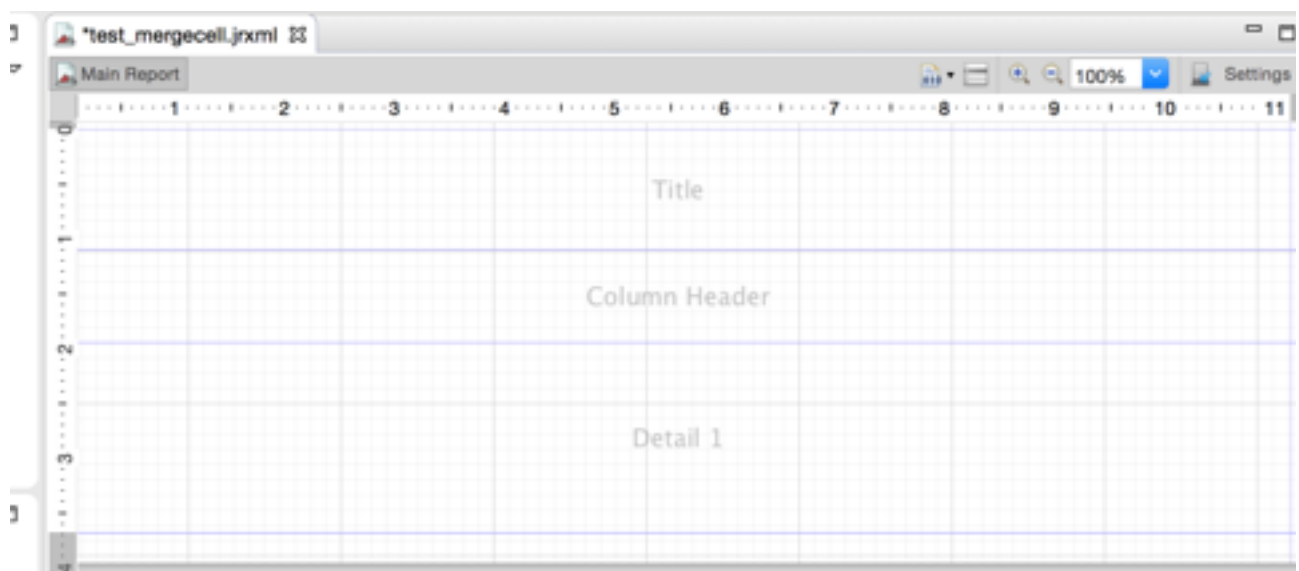
? < Back Next > Cancel Finish

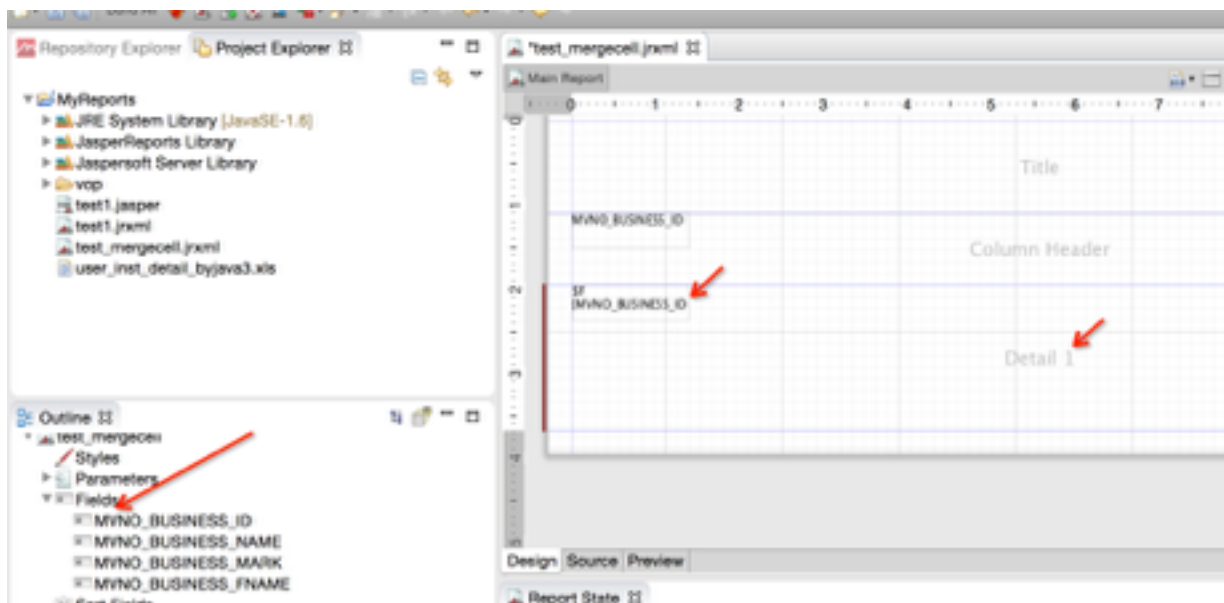
3. 设计：为了导出excel更好看一点， 删除用不到的局域



delete

删除后效果图：

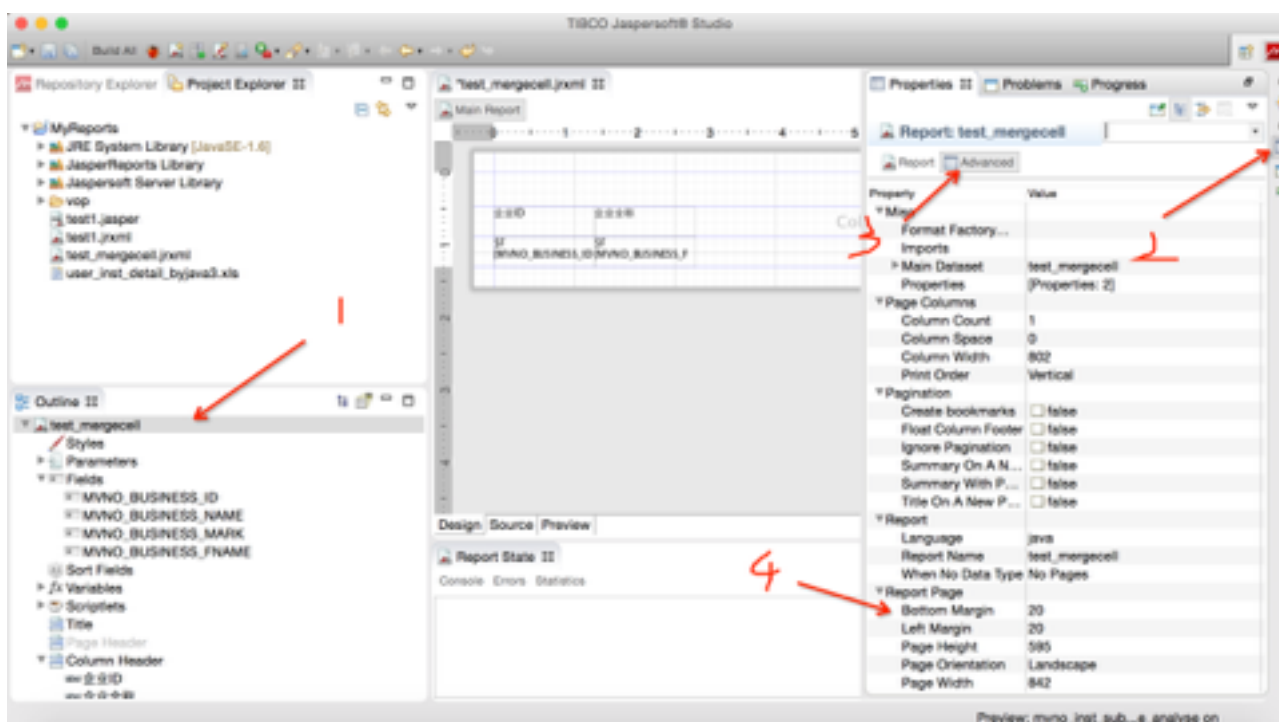




从 Fileds 区域 拖拽所需属性到 右侧 Detail 内

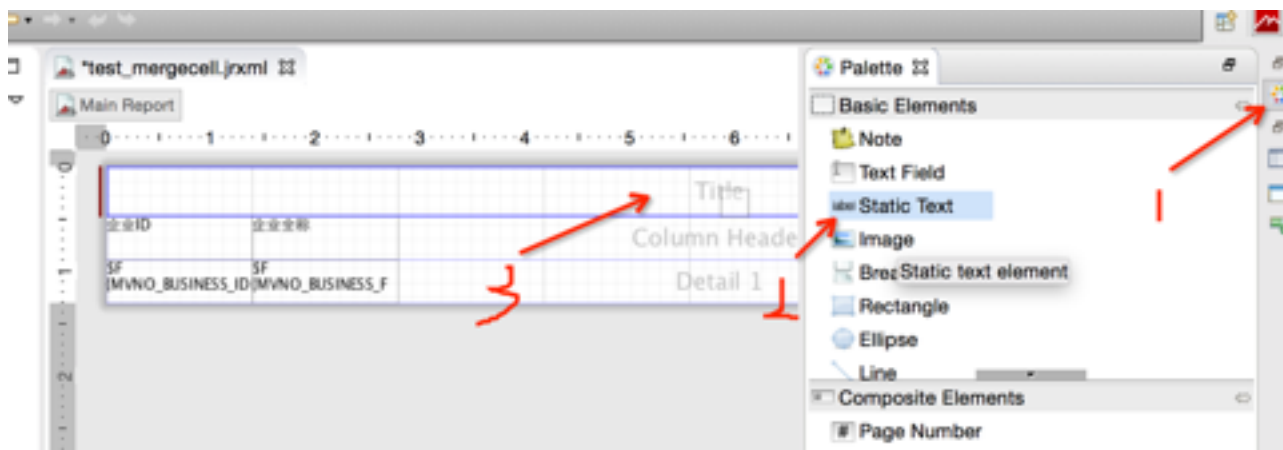
注意 Column Header 中生成的 MVNO\_BUNSINESS\_ID 为静态文本， 可以随意修改，  
Detail 1 中生成的 `$F{MVNO_BUSINESS_ID}` 为查询的结果集的字段， 最后生成的报表会用sql查询的结果集中一行一行的记录 填充 detail。

现在，简单调整报表样式，选中左侧“test\_mergecell”，选中右侧属性图标，“advanced”， 4中边框0。





增加标题：拖拽 StaticText, 然后修改下文本 和 “单元格”范围



简单调整下样式：

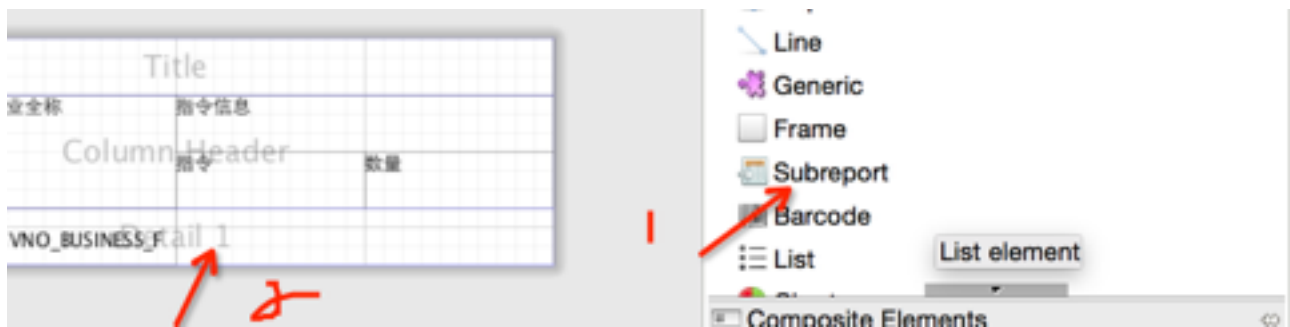
转售企业指令统计		Title	
企业ID	企业全称	指令信息	
SF		指令	数量
{MVNO_BUSINESS_ID}{MVNO_BUSINESS_F		Detail 1	

然后，  
重点，  
一个转售企业对应多个指令，

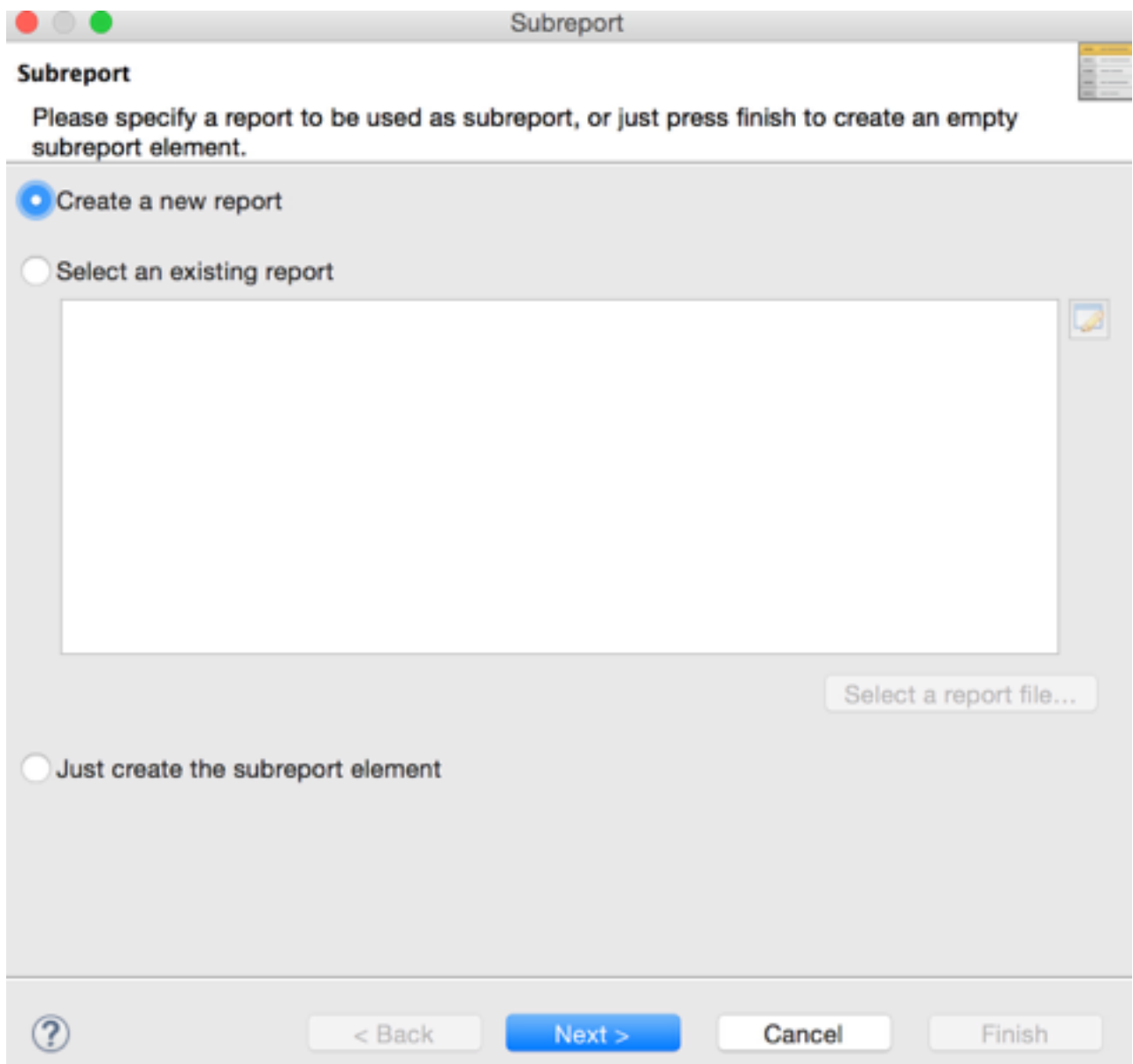
企业名	企业标示	指令详情		
		网别	指令	指令状态
北京国美电器有限公司	GOME	0	V0001	1
		0	V0002	1
		4G	V0001	1
		4G	V0001	2
		4G	V0023	1
		4G	V0024	2

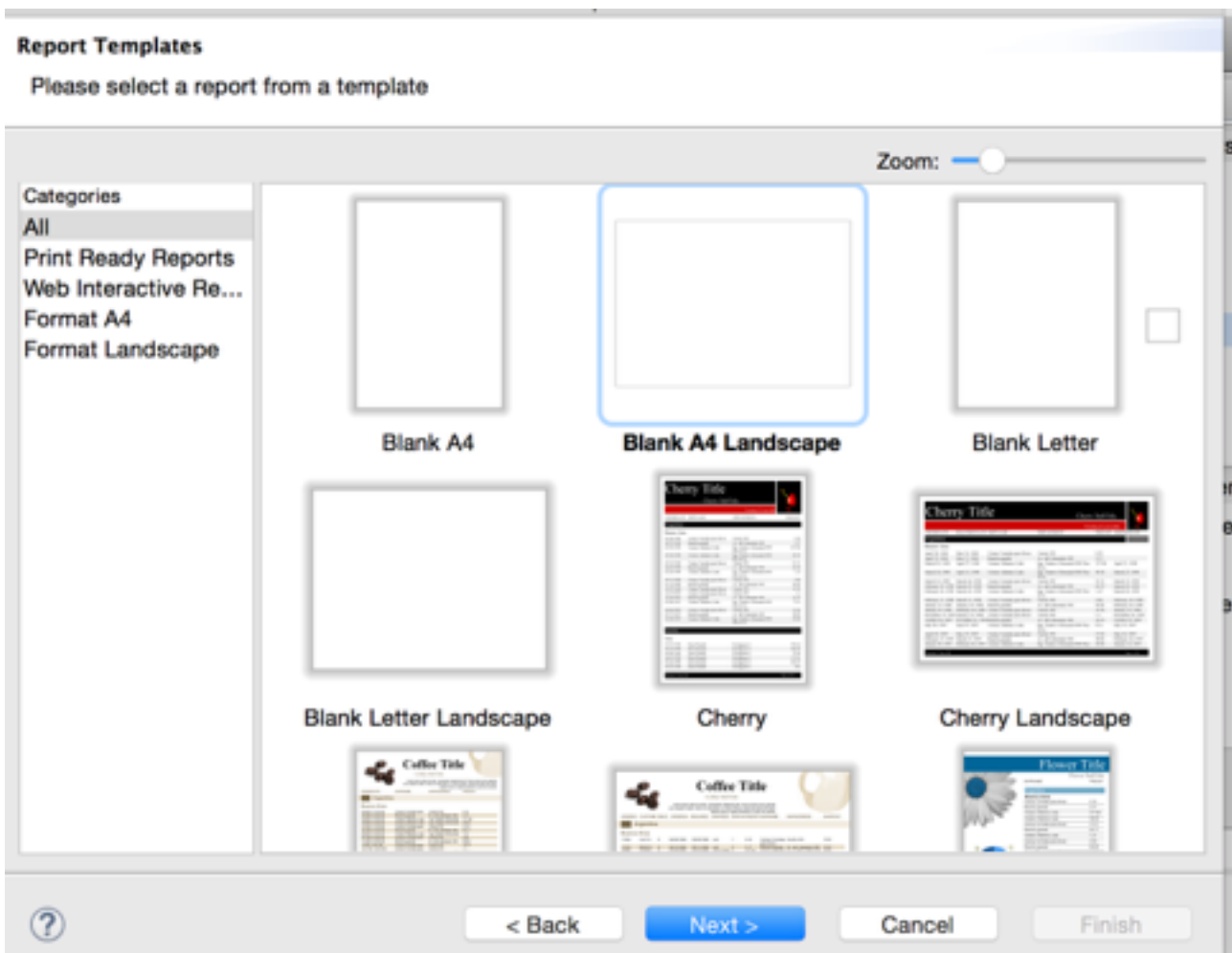
我们预期的是类似于  
这样的行合并效果，  
现在， 添加子表，“一个转企”对应“一个子表” 而一个子表 包含多个指令， 以此实现excel行合并的效果

拖拽 subreport 到detail

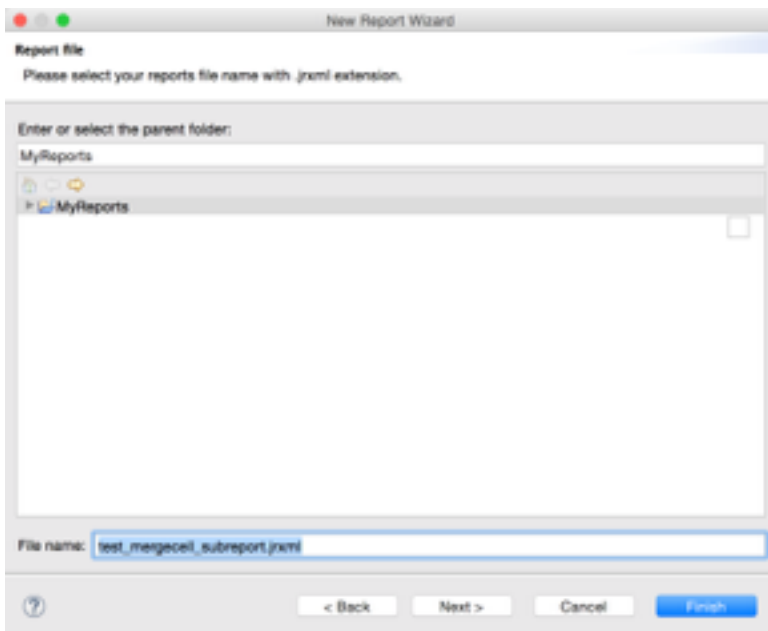


create a new report





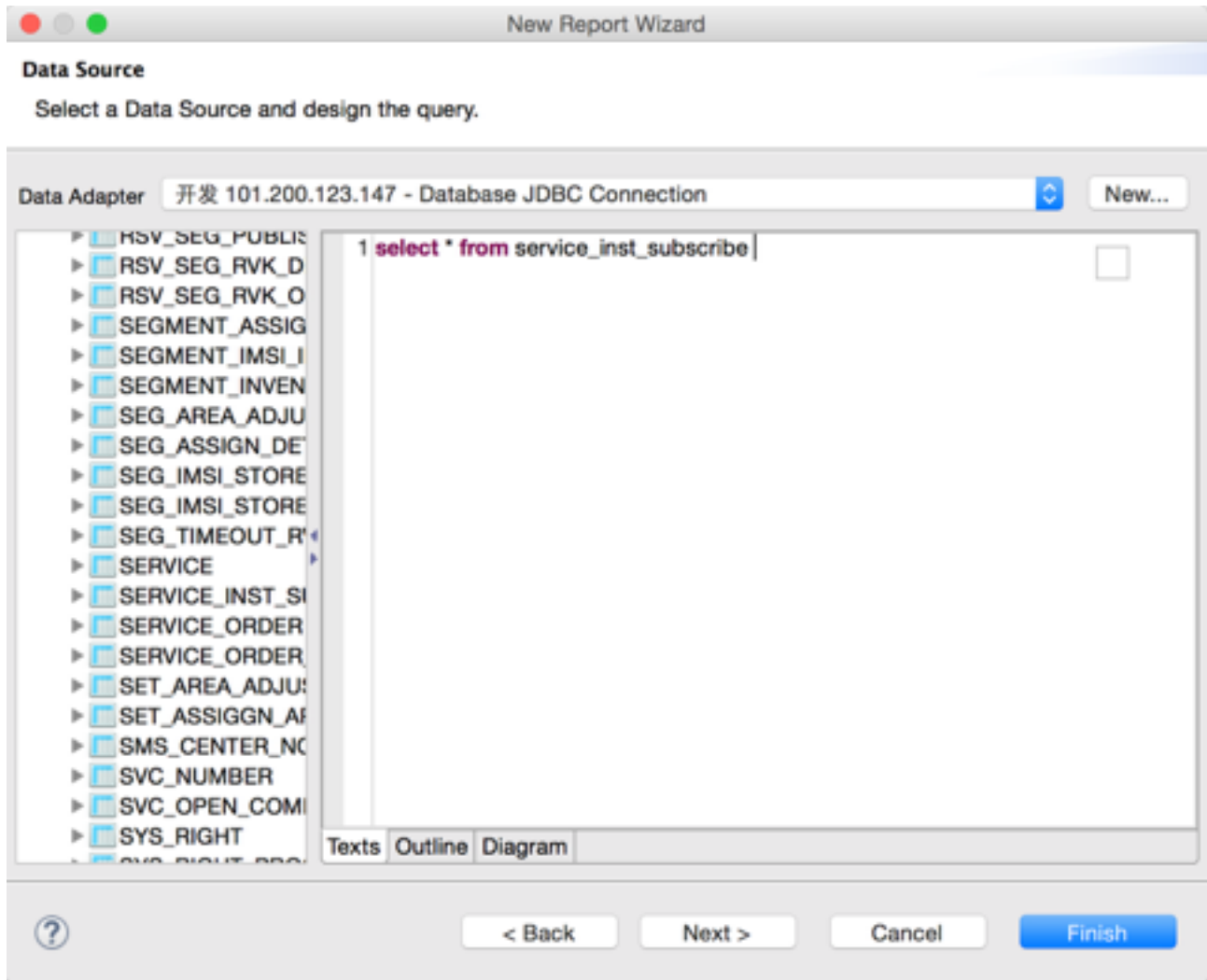
->next



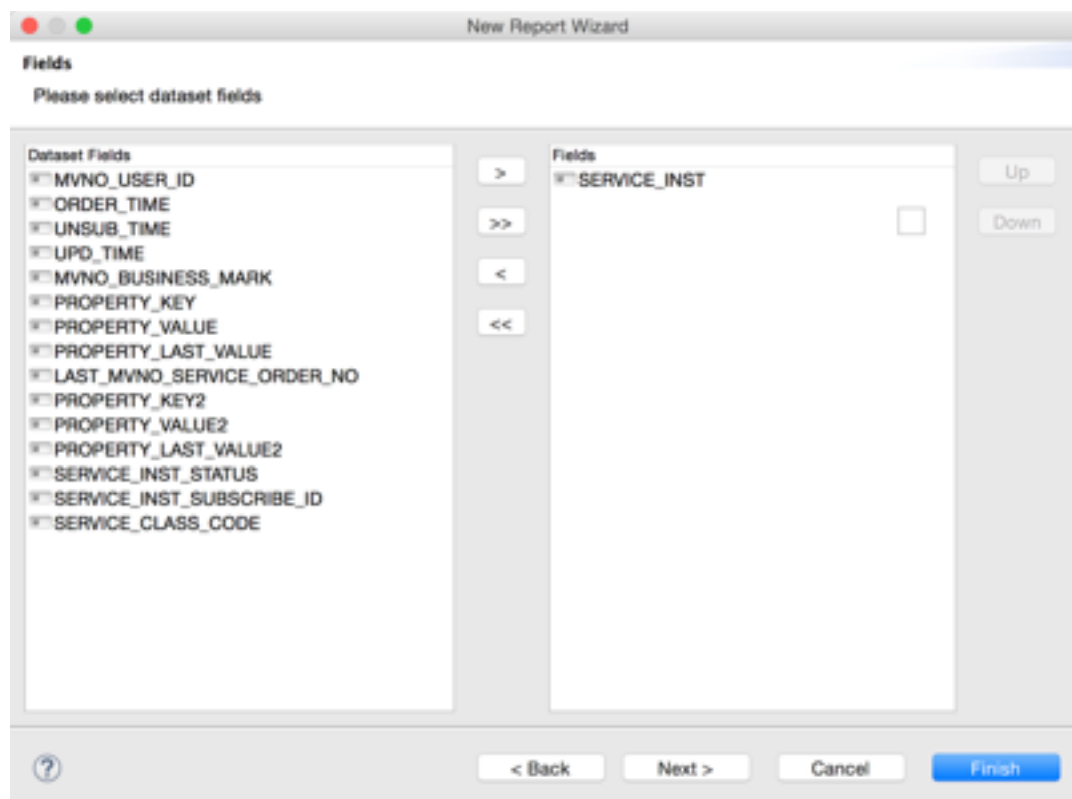
命名(随你喜欢..)  
test\_mergecell\_subreport.jrxml

-> next

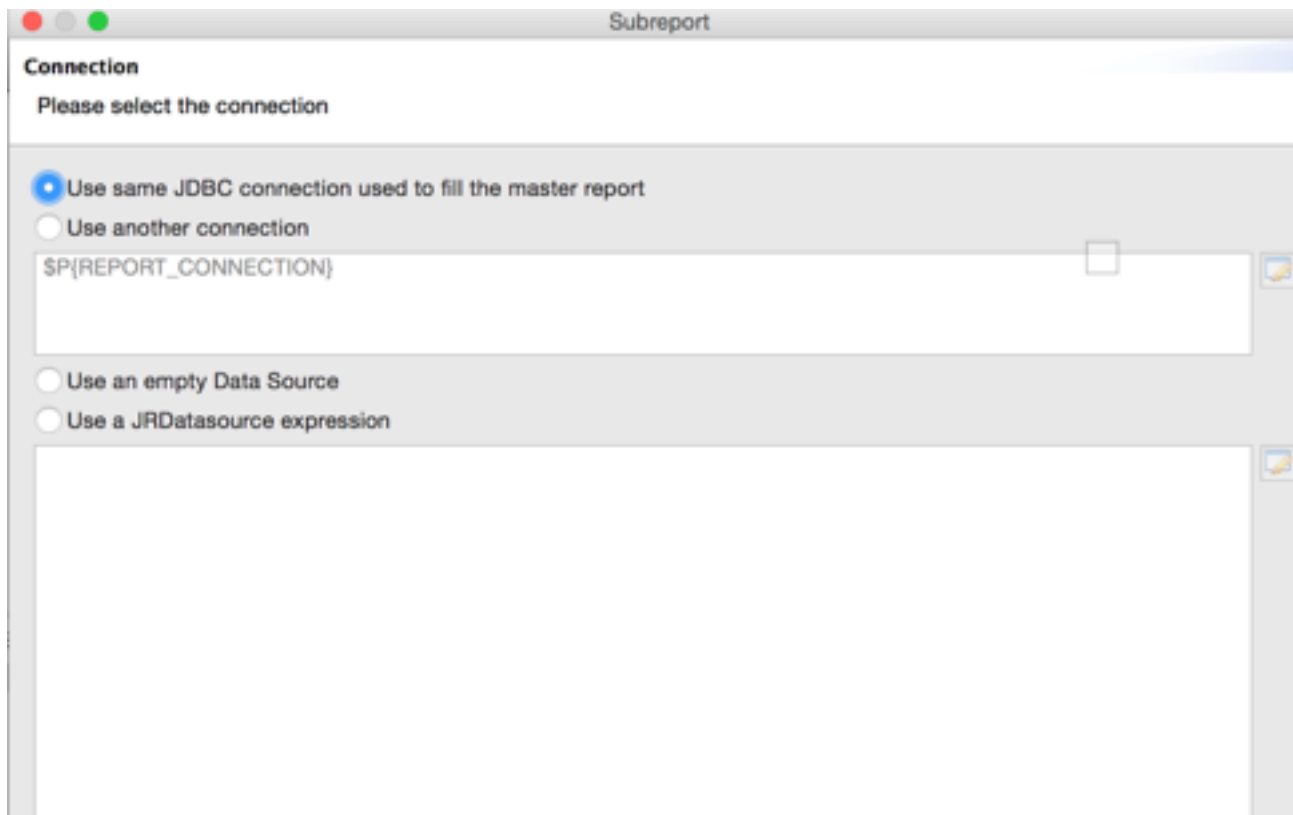
然后,



过会儿再根据需要调整。



next  
next



The image shows a 'Subreport' dialog box with a 'Connection' section. It prompts the user to 'Please select the connection'. There are four radio button options: 'Use same JDBC connection used to fill the master report' (which is selected), 'Use another connection', 'Use an empty Data Source', and 'Use a JRDataSource expression'. Below the 'Use another connection' option is a text field containing the expression '\$P{REPORT\_CONNECTION}'. There are also two empty text fields below the 'Use an empty Data Source' and 'Use a JRDataSource expression' options. A small square checkbox is located to the right of the first two text fields.

Use same JDBC connection used to fill the master report  
使用和父表相同的数据源

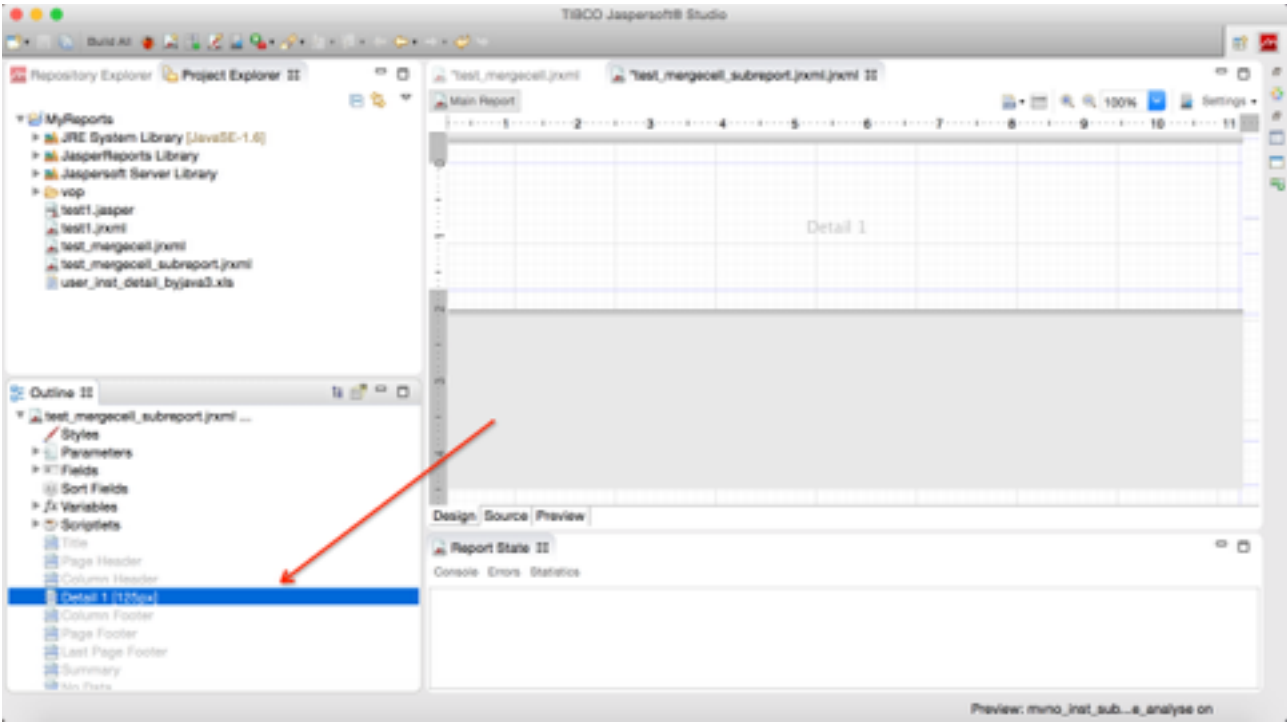


The image shows a 'Dataset Parameters' dialog box. It has a title bar 'Dataset Parameters' and a subtitle 'Dataset Parameters.'. Below this is a table with two columns: 'Name' and 'Expression'. The table is currently empty. To the right of the table are four buttons: 'Add', 'Edit', 'Delete', and 'Up'. A small square checkbox is located to the right of the table.

过会儿再加参数。

finish.

根据需要，



只留detail，

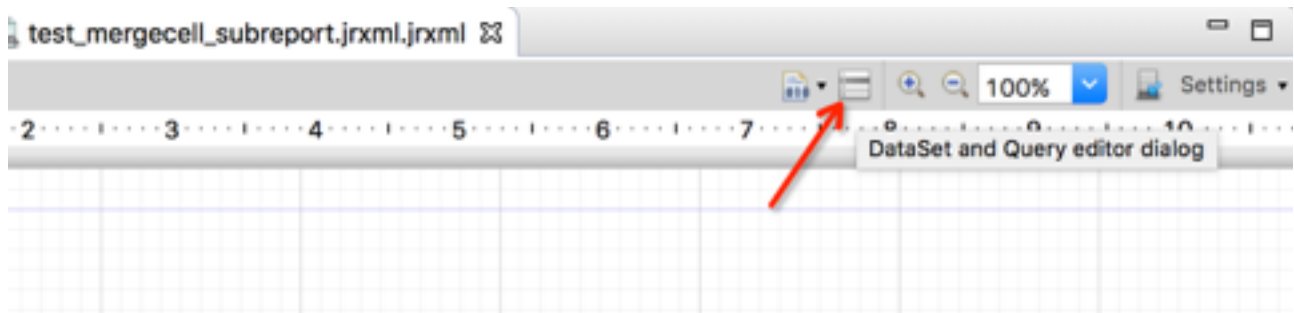
值得注意的是，为了使得导出的excel 较为符合大众审美观，在设计jasper 的时候 各个数据格一定要对齐，除了数据格 不要留没有意义的空白，jasper 中每一丝“空白”，导出excel 后都会以空单元格的形式出现在你眼前。

预期子报表示例：

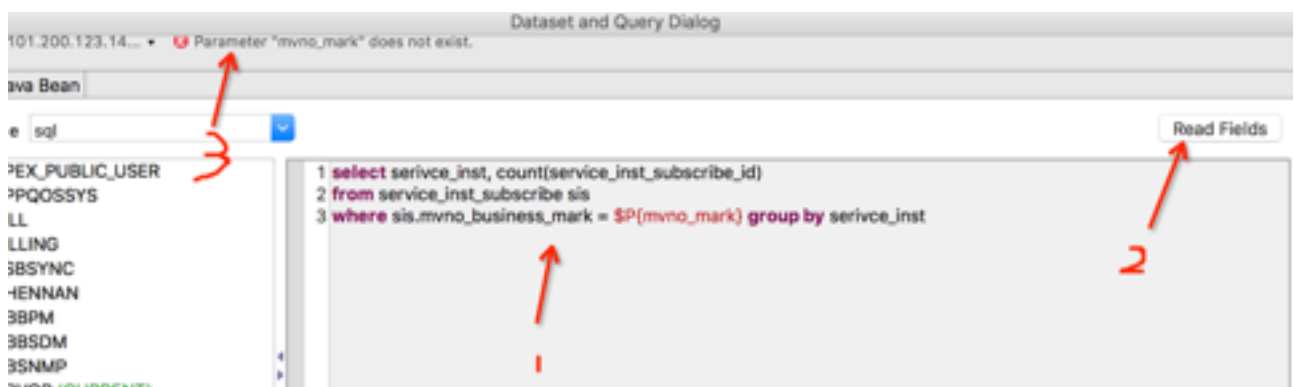
企业	指令	总数
转企1	指令1	11
	指令2	12
转企2	指令1	21
	指令2	22
	指令3	23

“子报表”

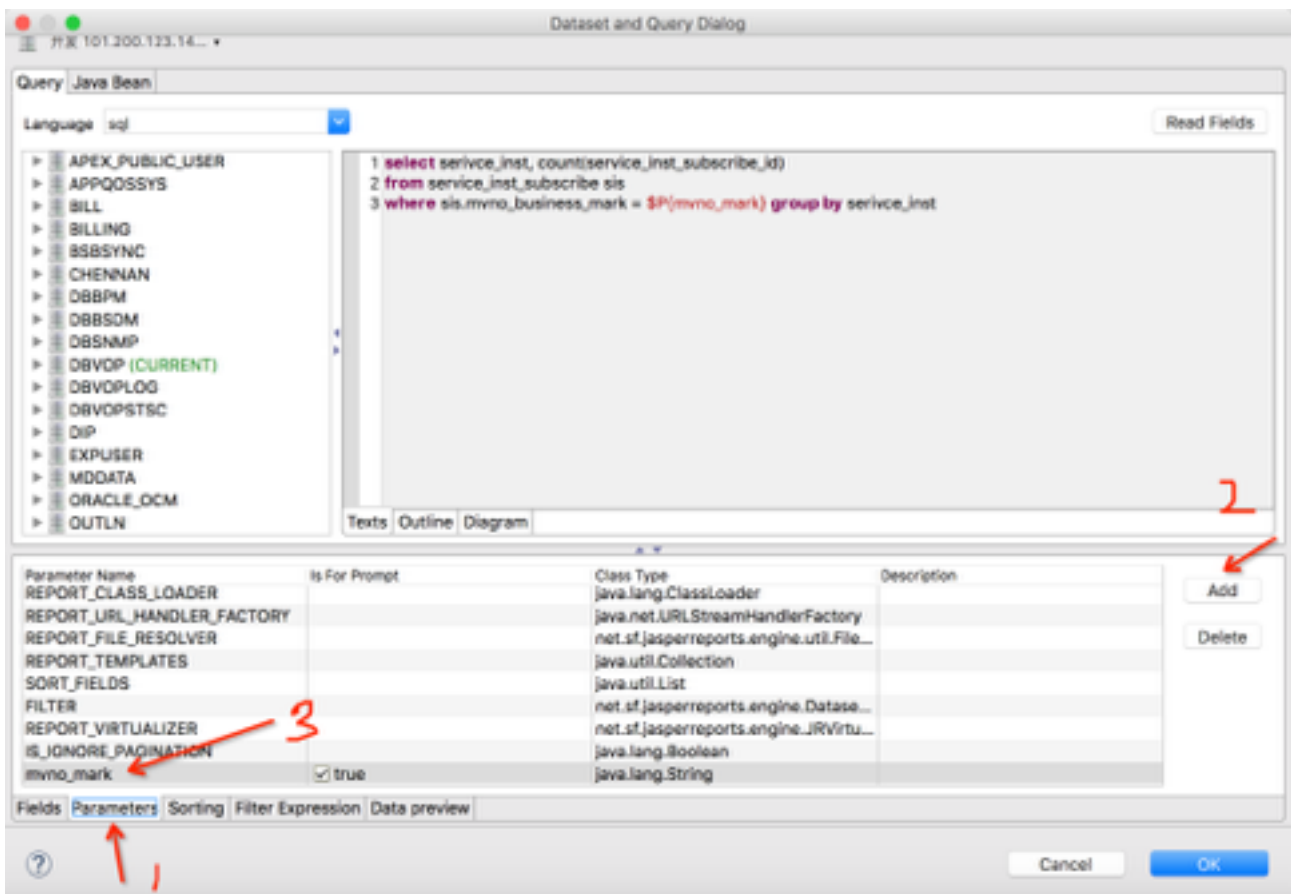
每一个子报表 对应 主报表 的一条记录,  
调整子报表查询sql:



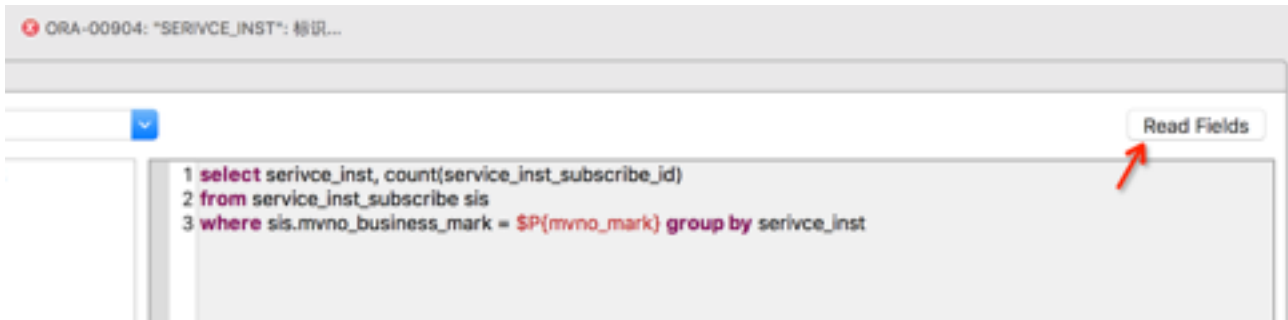
点击 编辑查询语句



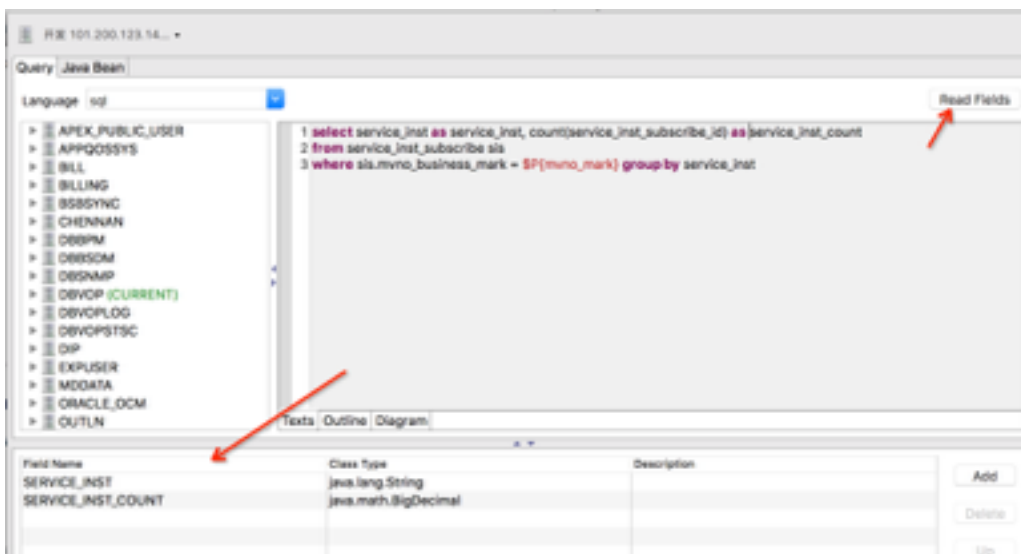
1. 修改查询sql ——其中 \${mvno\_mark} 表示参数,
2. 点击 Read Fields 简单测sql语法
3. Parameter "mvno\_mark" does not exist



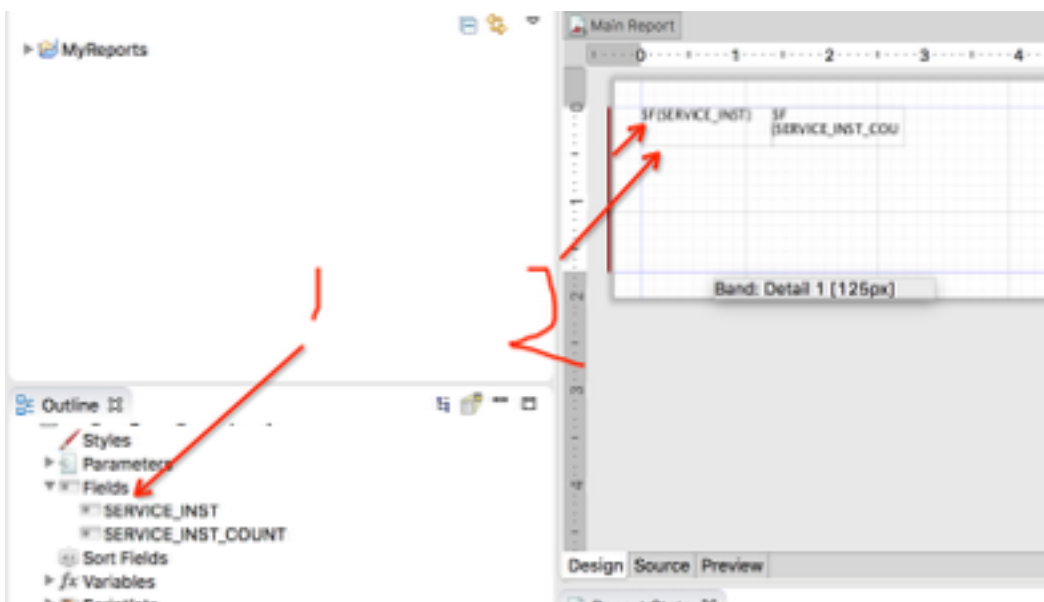
添加过参数变量后，回到 Fields 页， Read Fields ，



标示符拼写错误，  
修正， Read Fields



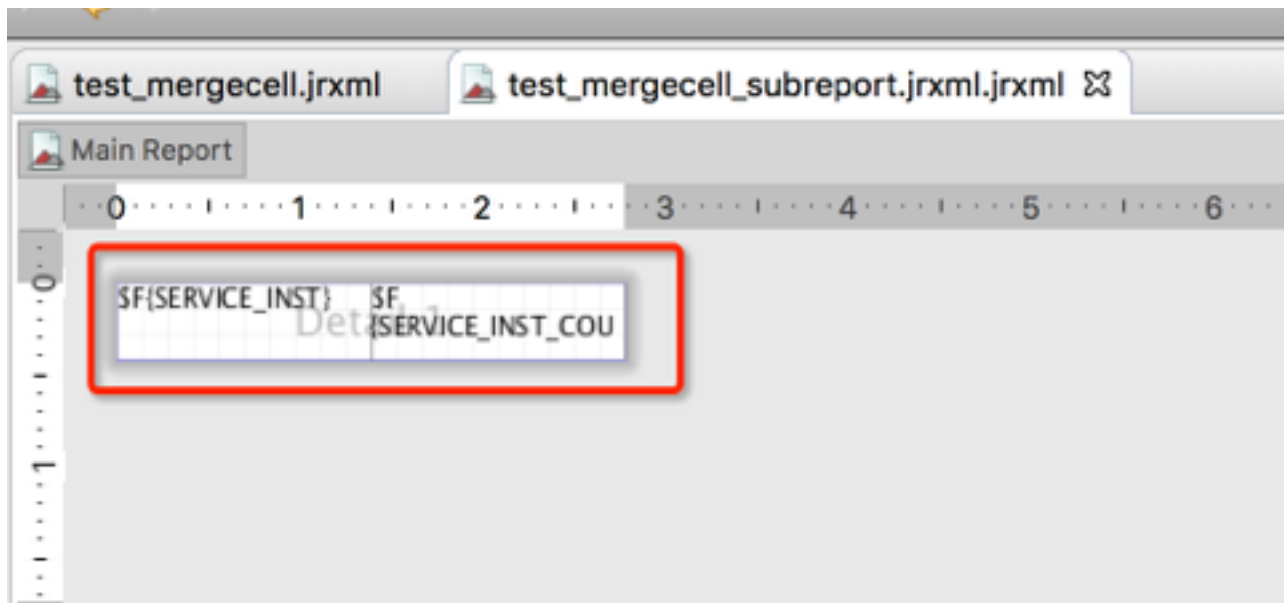
在 Detail 域添加新读出的字段， 删掉静态文本 留下字段变量——\$F{...}



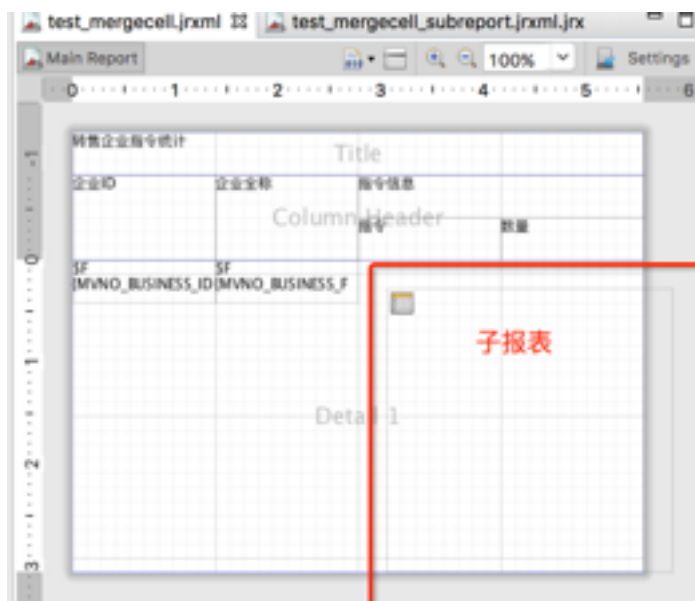


简单调整子报表布局

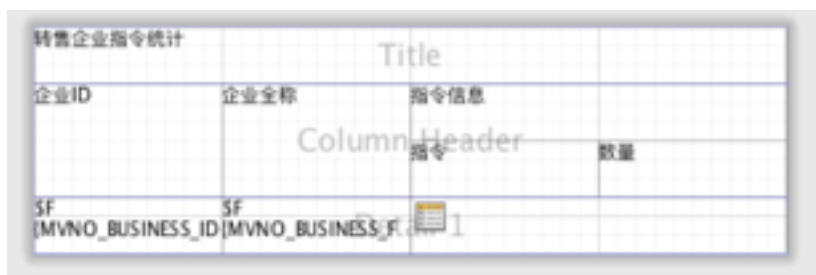
最终效果图：Page Height: 30 Page Width: 200 与主报表预留空间一致



主报表：

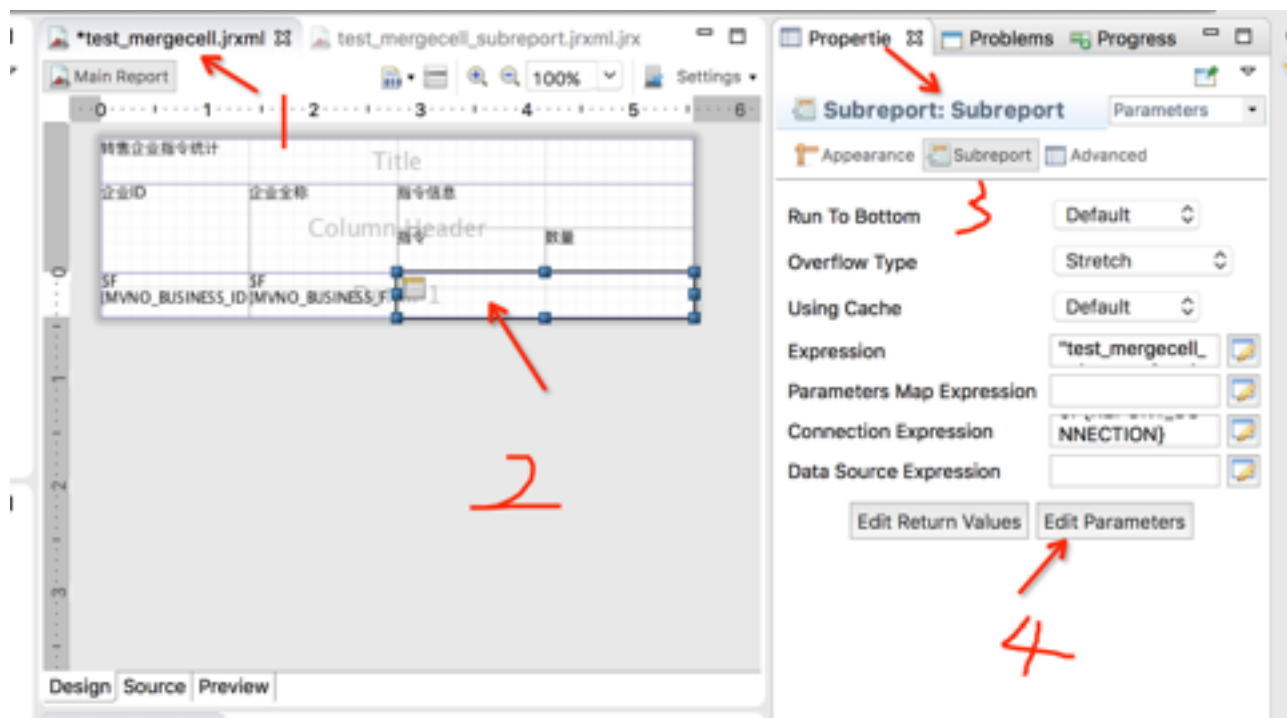


... 调整主报表布局 ->

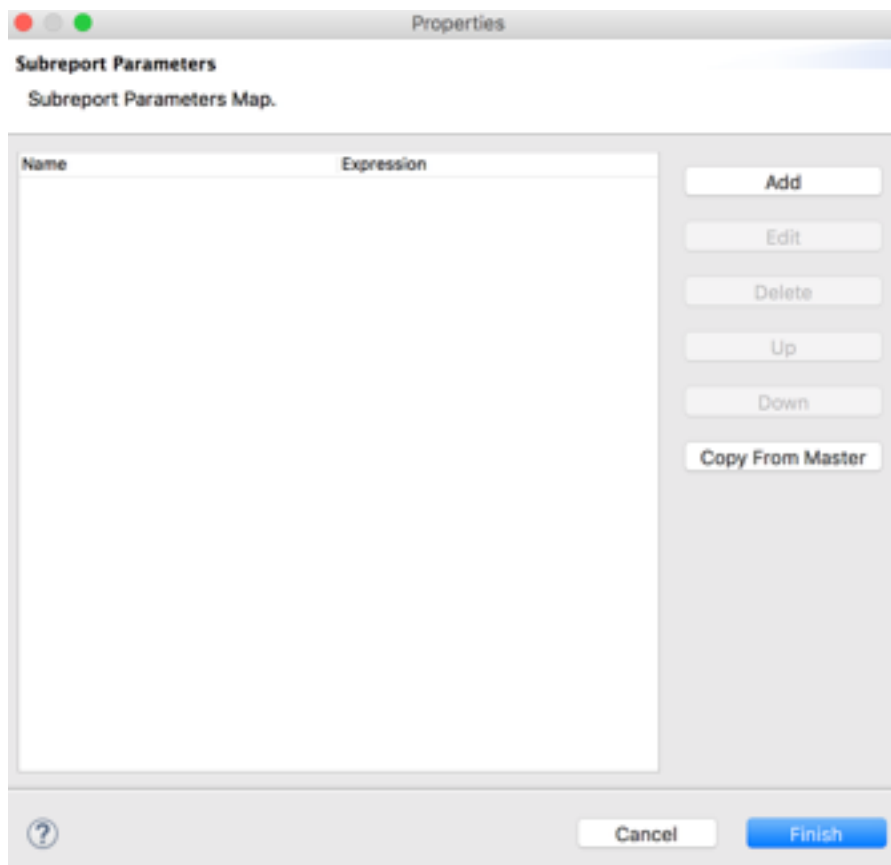


在主报表中 点击 子表， 向其设置参数。

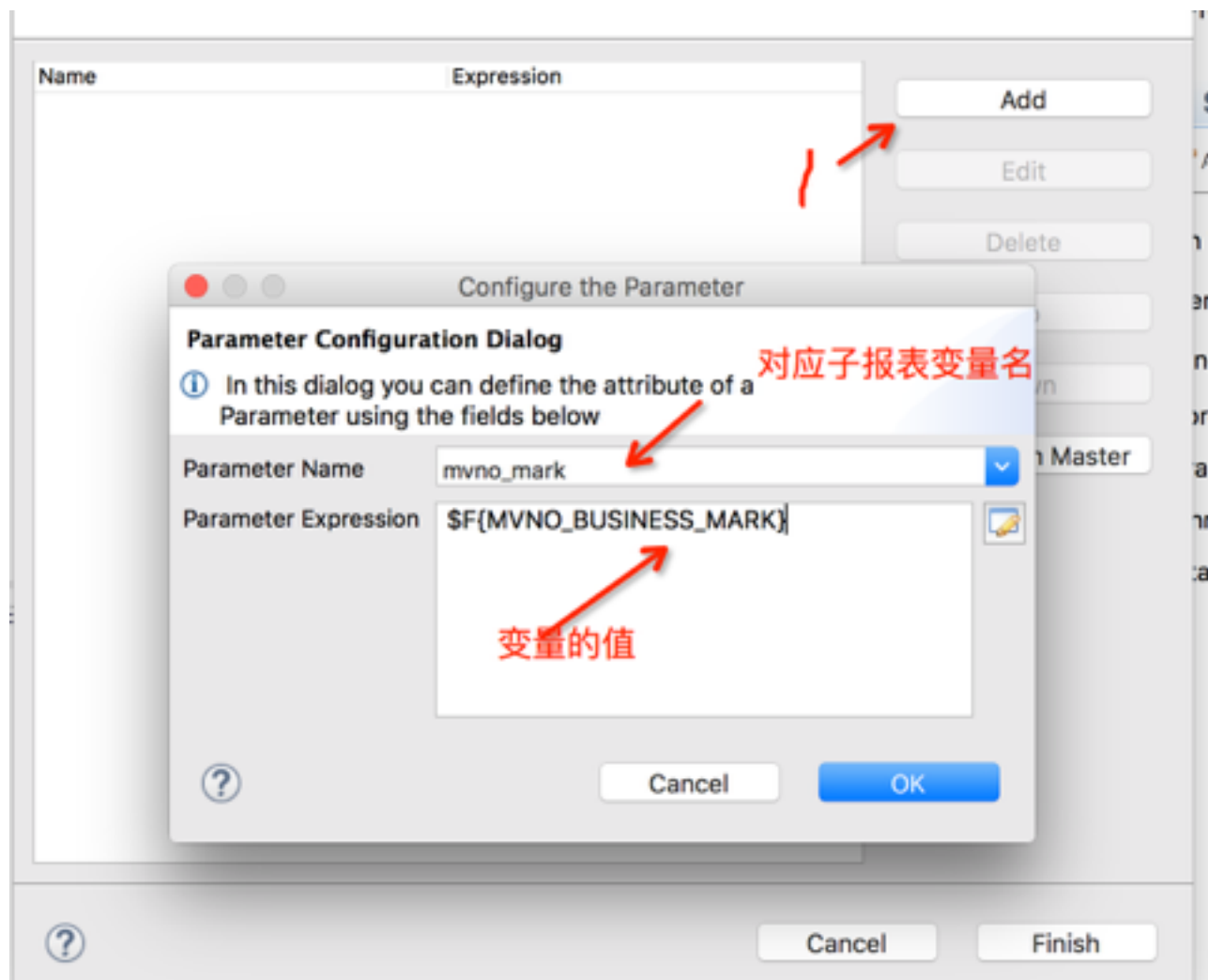
\* 当子报表需要使用 mvno\_mark 参数时， 1. 在子报表中“声明” mvno\_mark 参数， 2. 父报表“调用” 子报表时 添加 mvno\_mark 参数：



按上图操作 打开编辑页

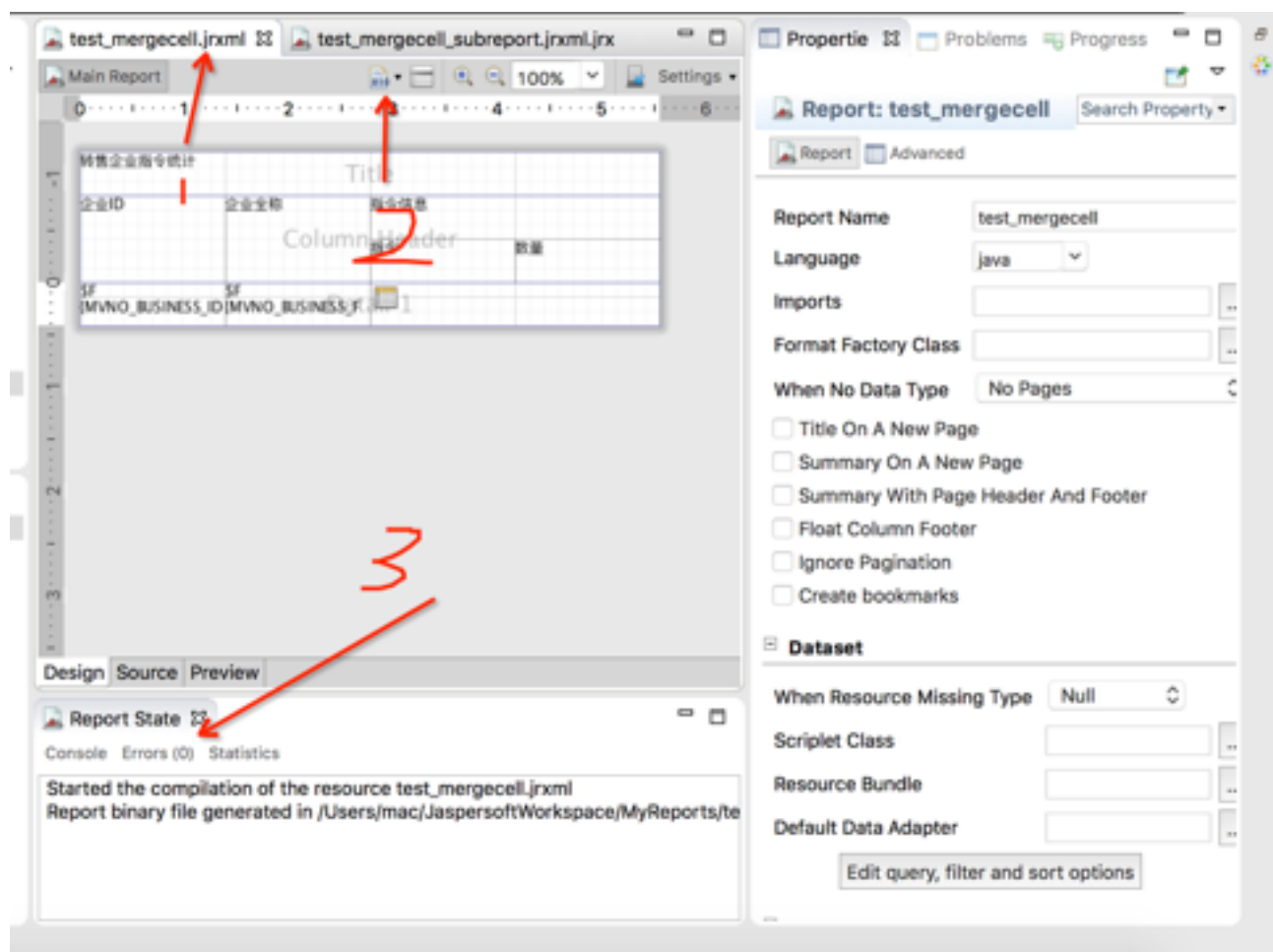


添加参数：

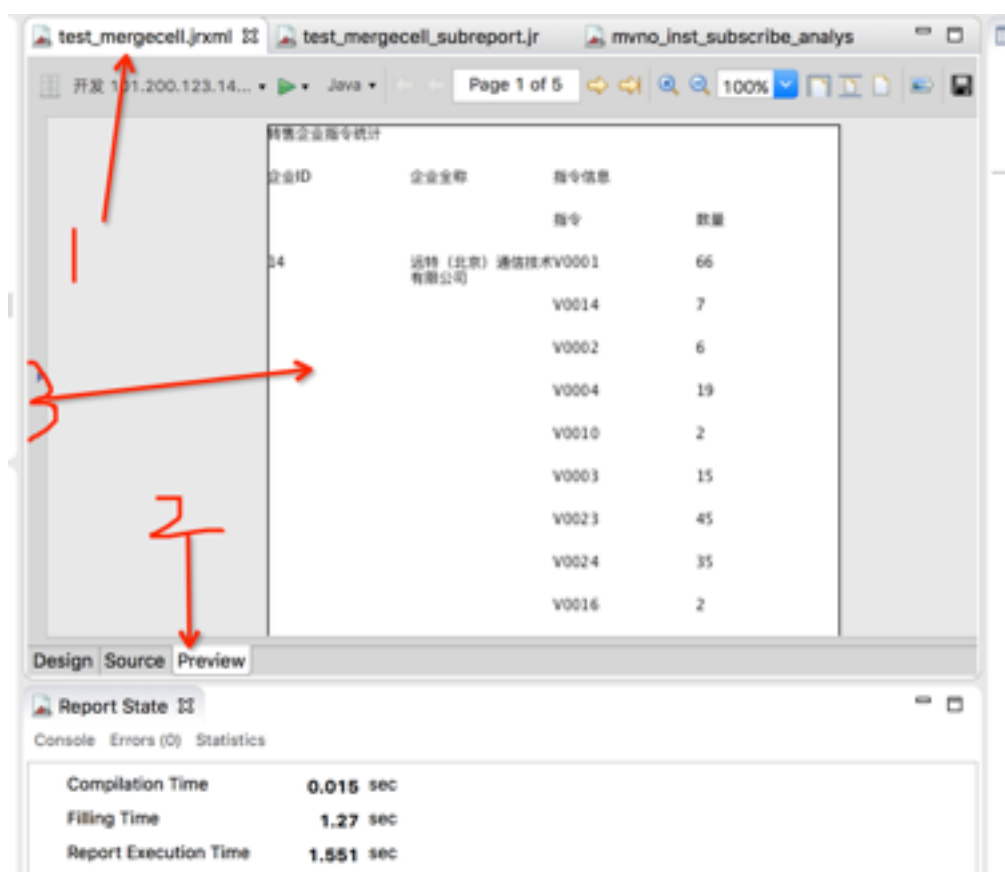


如图，  
然后 ok， finish

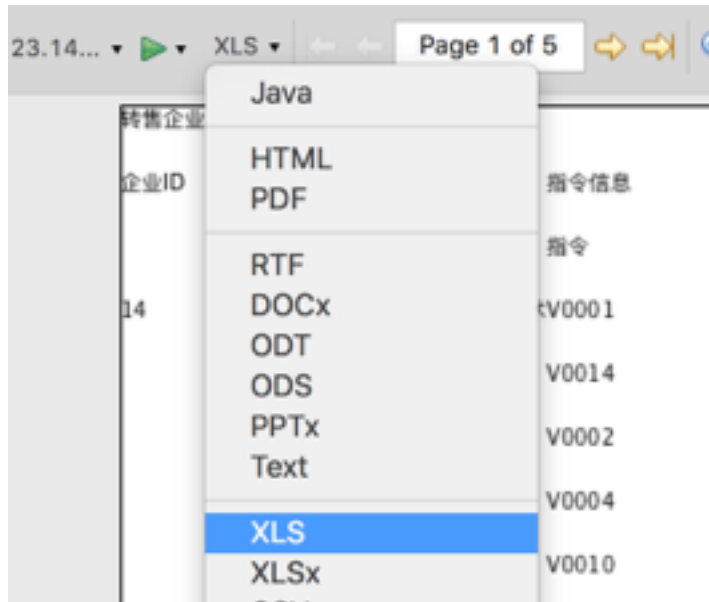
保存，编译测试一下：



点击 Preview 预览一下，



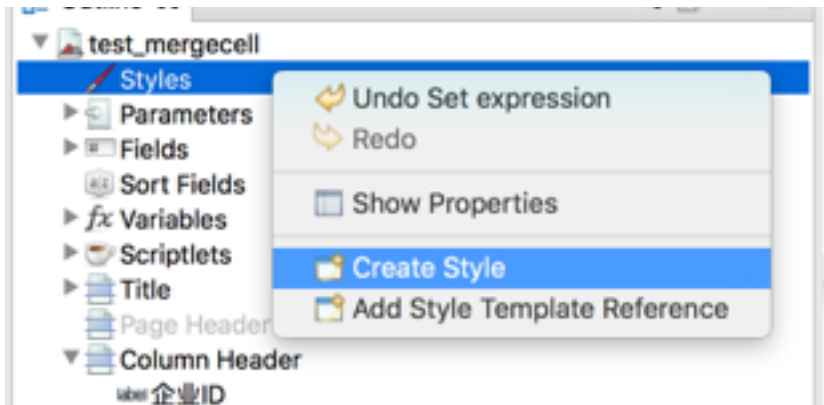
额，以 xls 格式打开试下看，



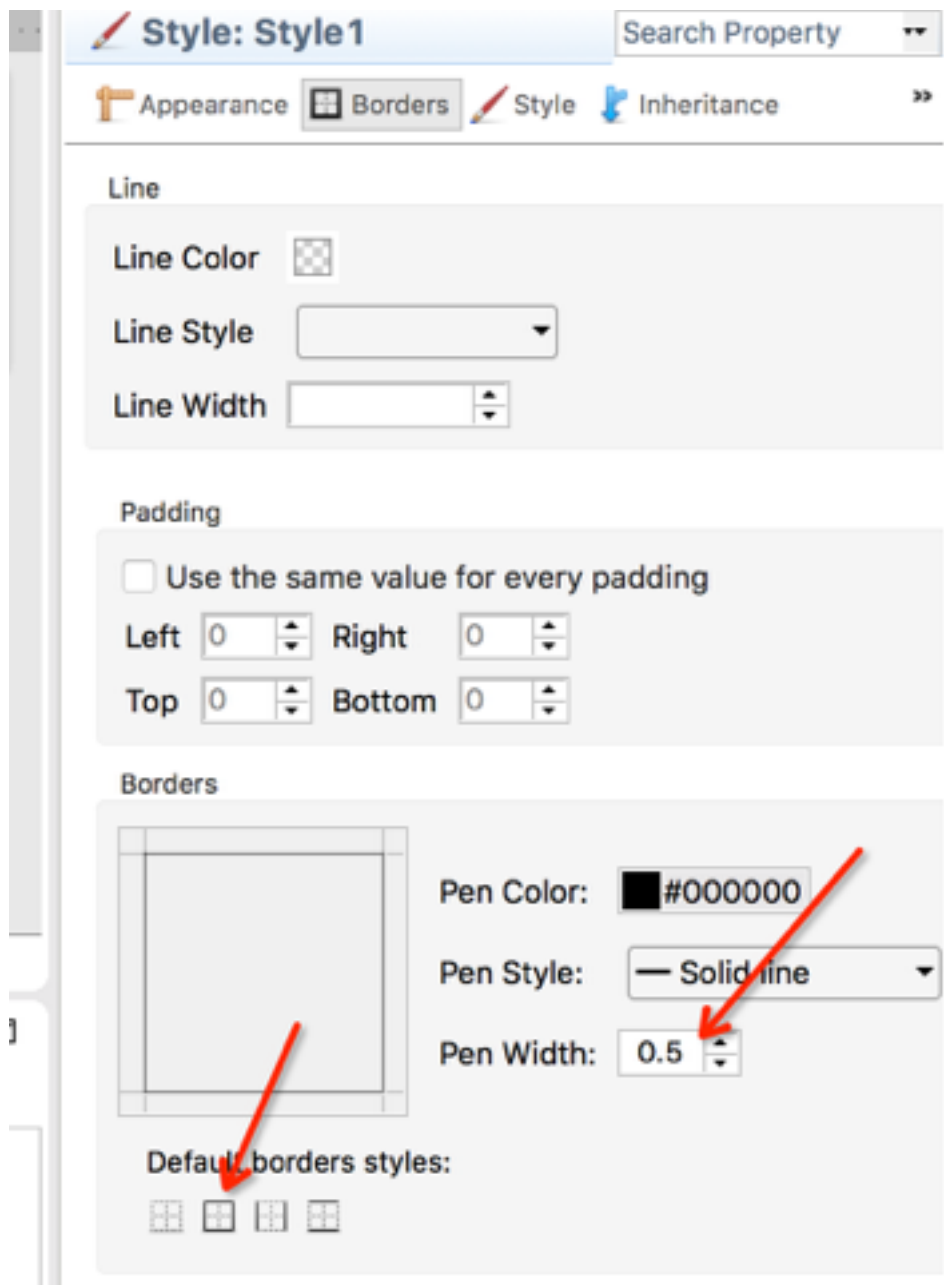
点击那个小下拉框，选XLS

test_mergecell				
A	B	C	D	E
转售企业指令统计				
企业ID	企业全称	指令信息		
		指令	数量	
14	远特（北京）通信技术有限公司	V0001	66	
		V0014	7	
		V0002	6	
		V0004	19	
		V0010	2	
		V0003	15	
		V0023	45	
		V0024	35	
3	苏宁云商集团股份有 限公司	V0016	2	
		V0001	50	
		V0014	15	
		V0002	45	

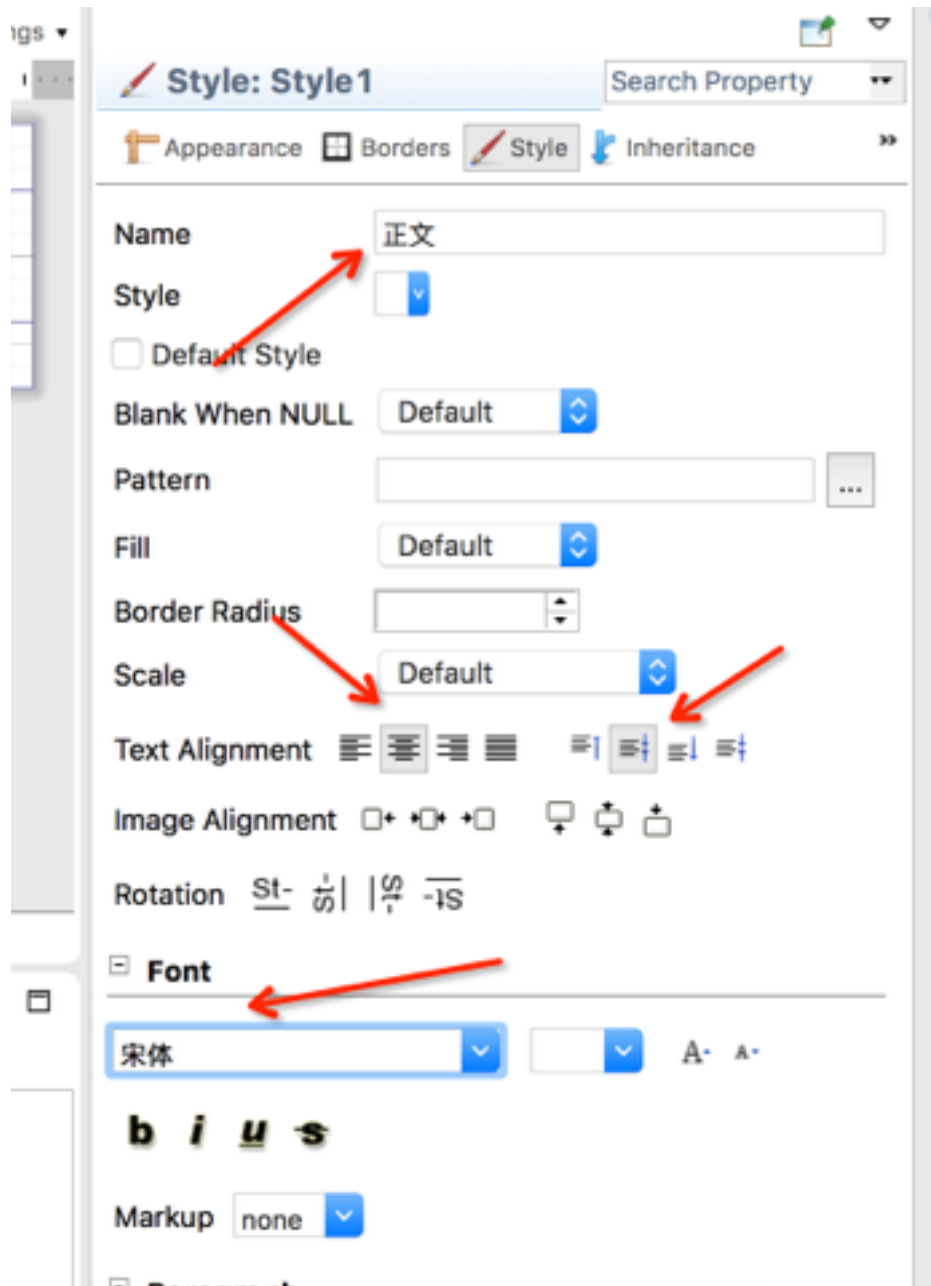
调整下样式：



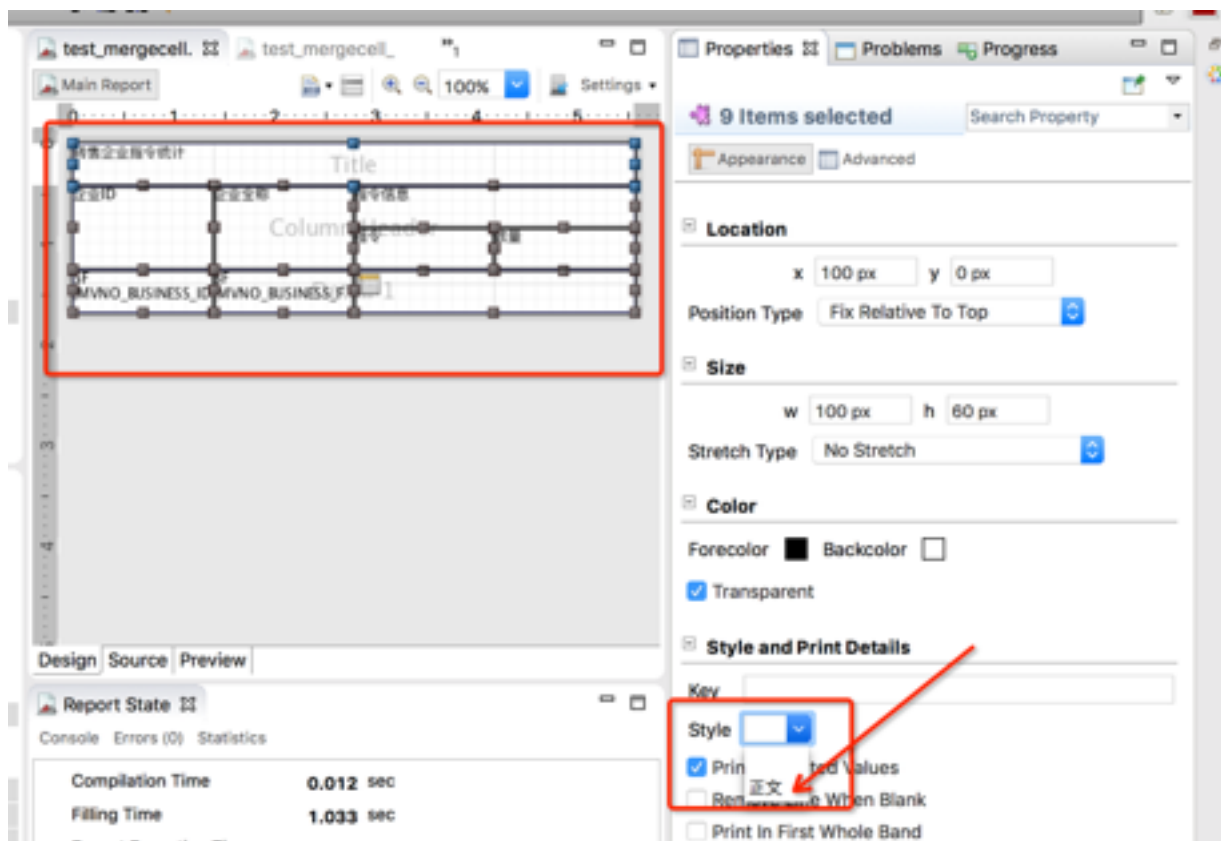
Styles -> Create Style  
边框：



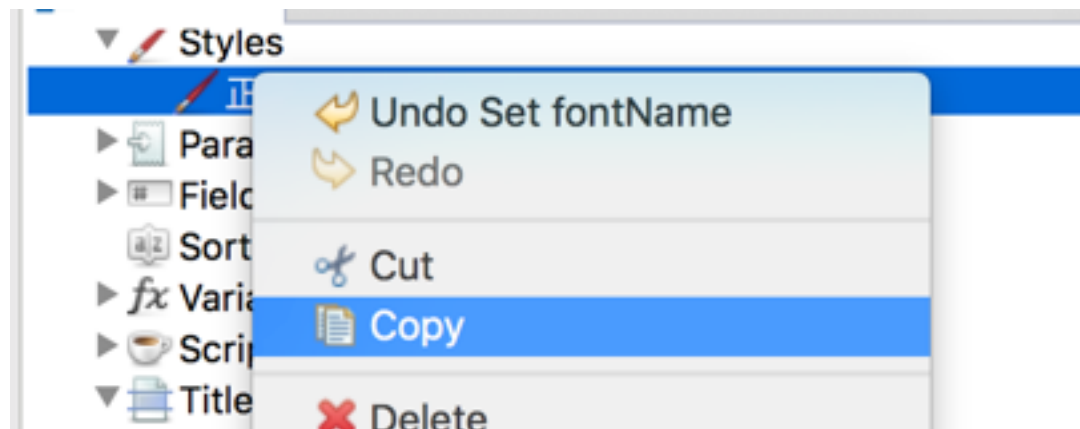
样式重命名，内容居中，字体：宋体（需要安装）



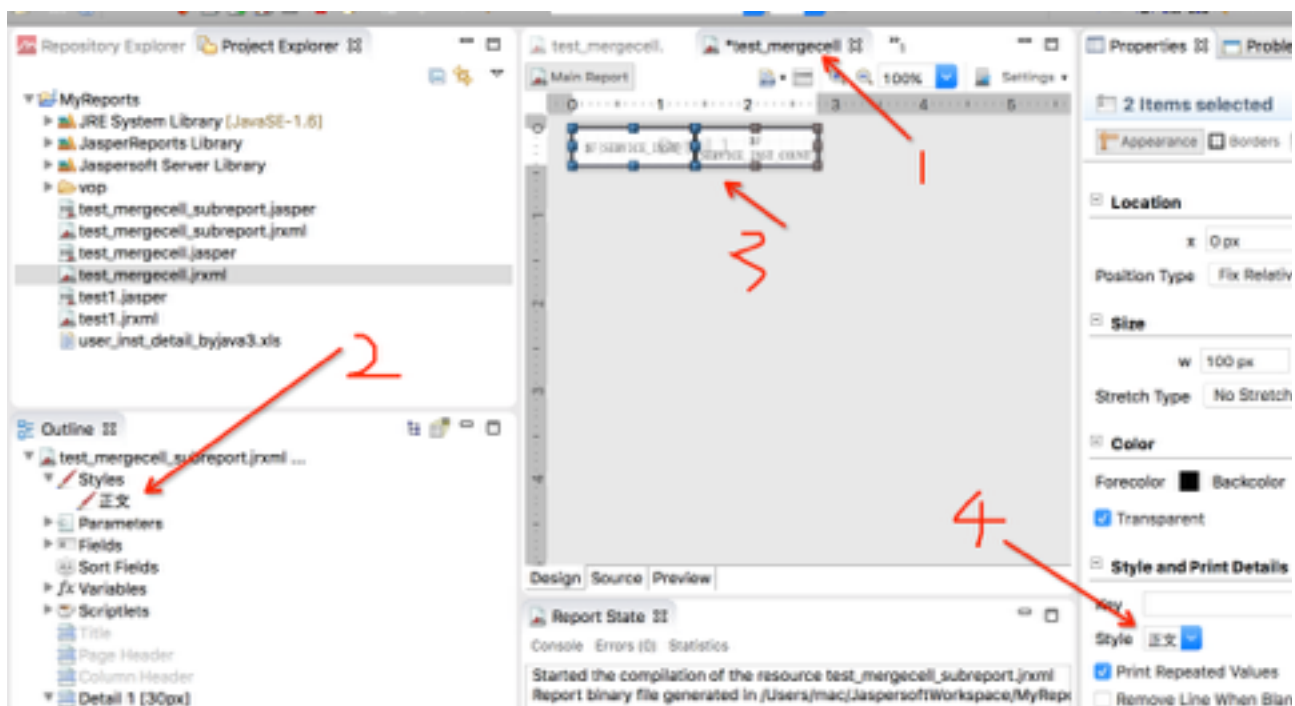
选中所有， 设置 Style



然后， 复制主表“正文”样式， 粘贴到子报表：



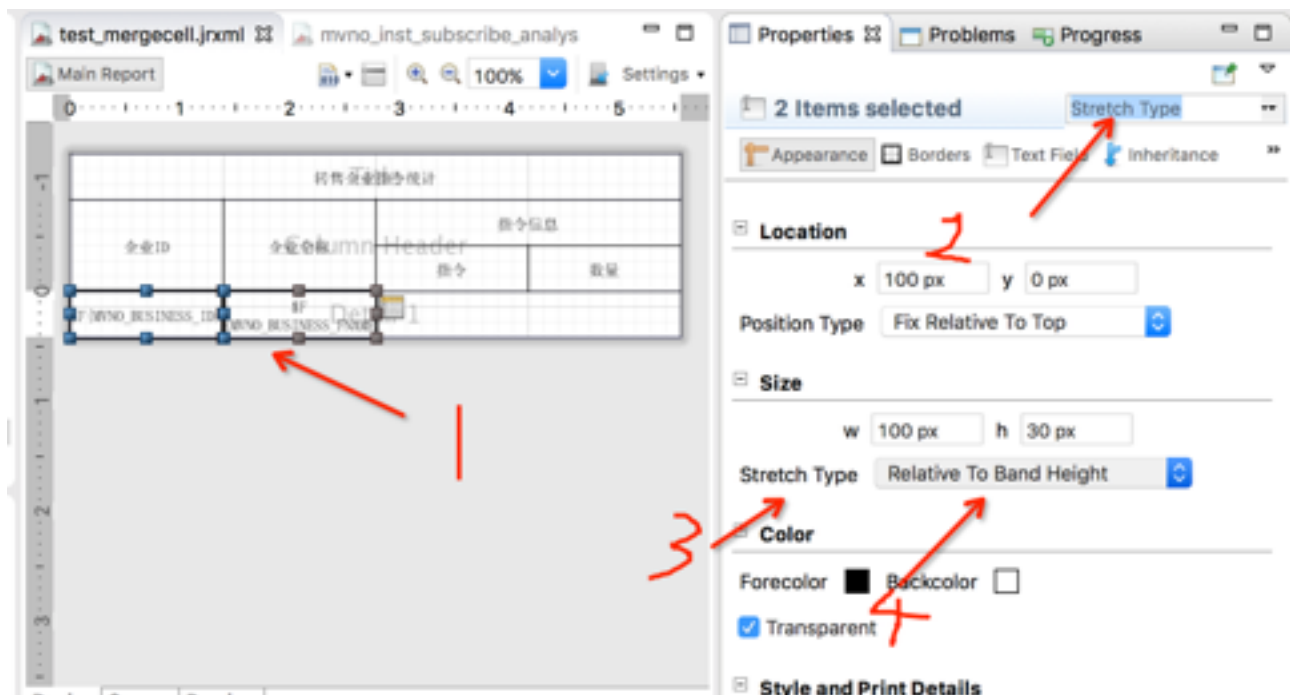




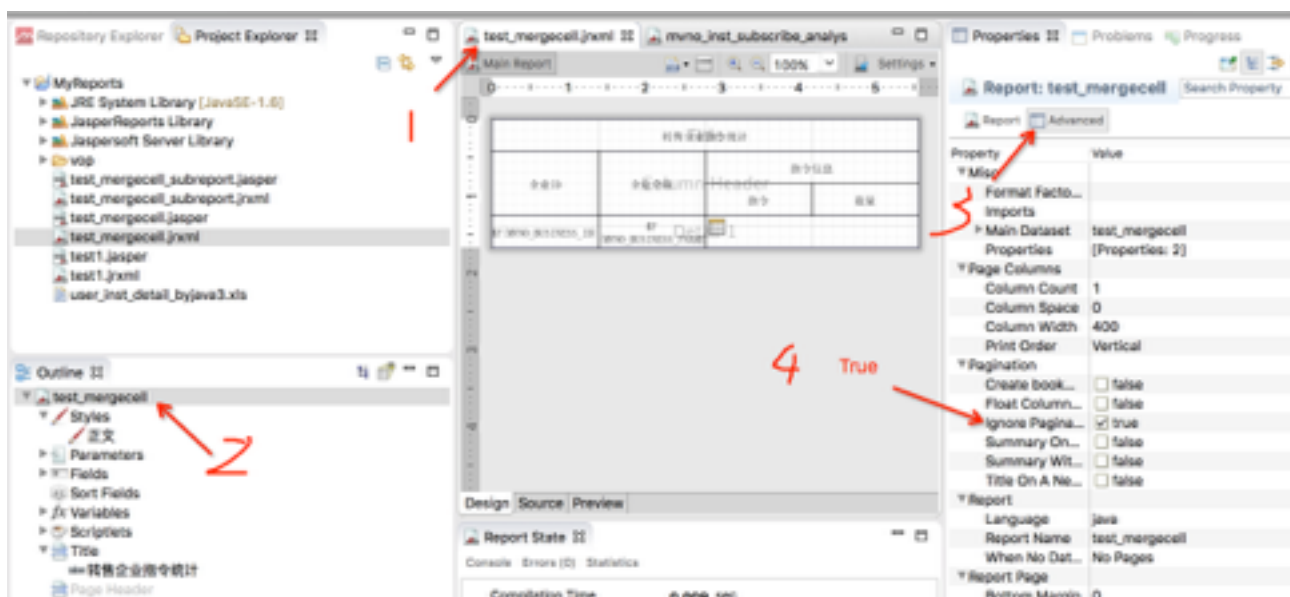
保存，编译，XLS 预览：

	A	B	C	D
1	转售企业指令统计			
2	企业ID	企业全称	指令信息	
3			指令	数量
4	14	远特（北京）通信技术 有限公司	V0001	66
5			V0014	7
6			V0002	6
7			V0004	19
8			V0010	2
9			V0003	15
10			V0023	45
11			V0024	35
12			V0016	2

列：企业ID， 企业全称 预期为合并单元格，  
现在，主表设置 对应单元格 Stretch：



然后，取消分页：



保存， 编译， 预览 XLS：

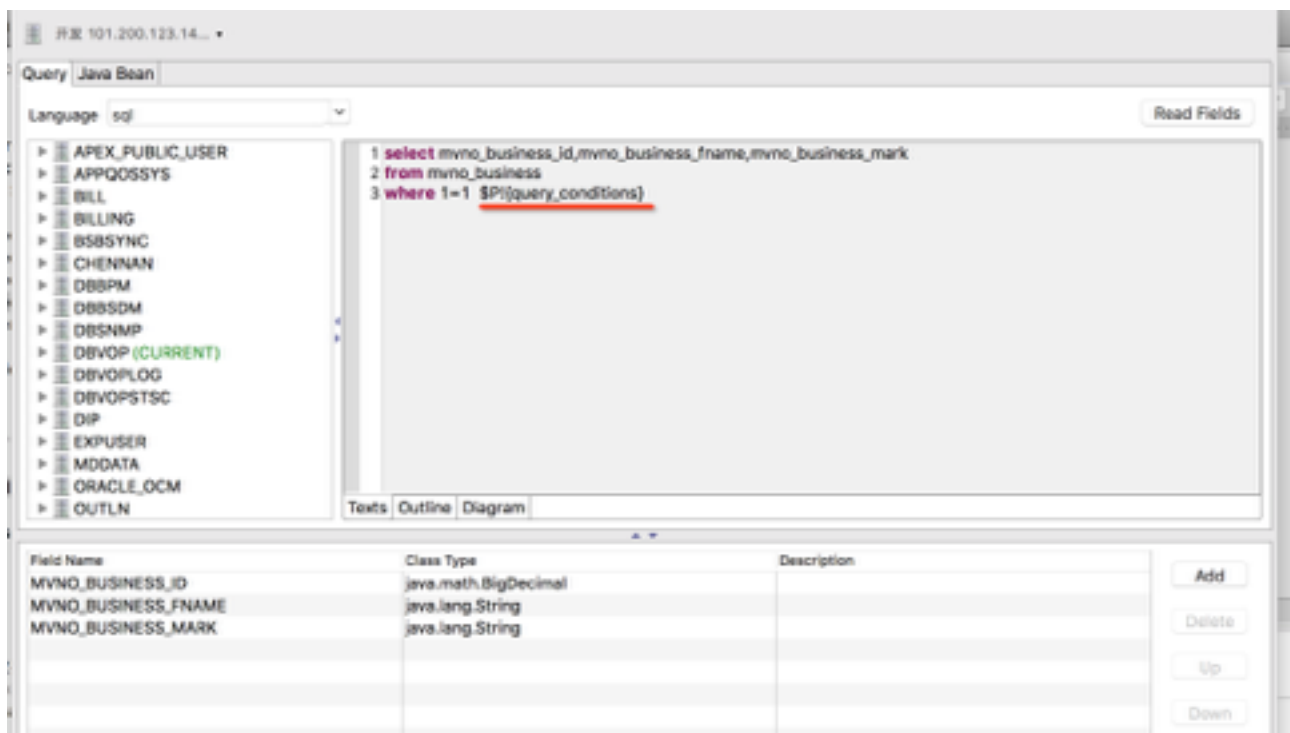
效果：

转售企业指令统计			
企业ID	企业全称	指令信息	
		指令	数量
14	远特（北京）通信技术有限公司	V0001	66
		V0014	7
		V0002	6
		V0004	19
		V0010	2
		V0003	15
		V0023	45
		V0024	35
		V0016	2
		V0001	50
		V0014	15
3	苏宁云商集团股份有限公司	V0002	13
		V0004	20
		V0010	26
		V0003	23
		V0024	22
		V0023	25
		V0016	8
4	北京迪信通通信服务有限公司		
5	苏州蜗牛数字科技股份有限公司		
6	话机世界数码连锁集团股份有限公司		
7	深圳市爱施德股份有限公司		
8	北京乐语世纪通讯设备连锁有限公司		
9	天音通信有限公司		
10	北京万网志成科技有限公司		
		V0001	95

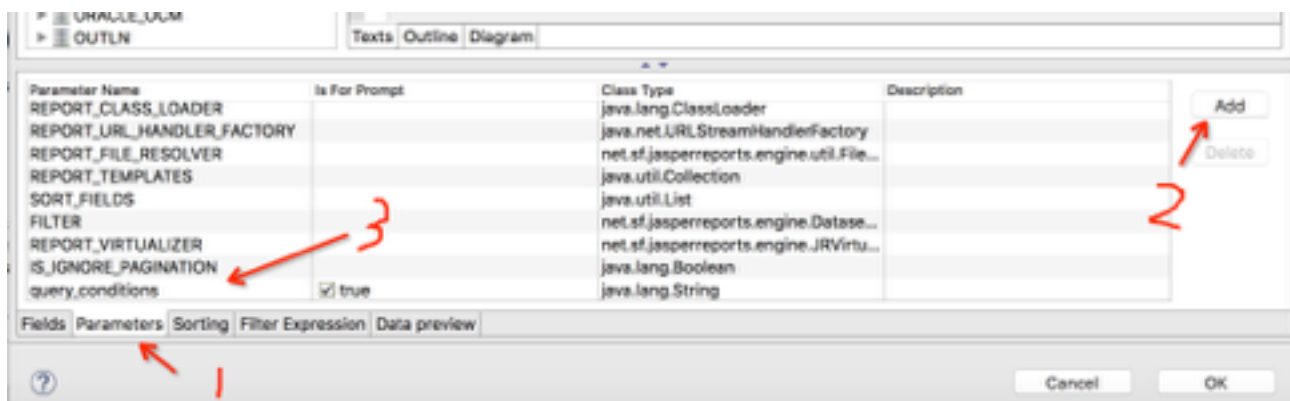
其中4~10 是确实没有数据

对于变参查询：

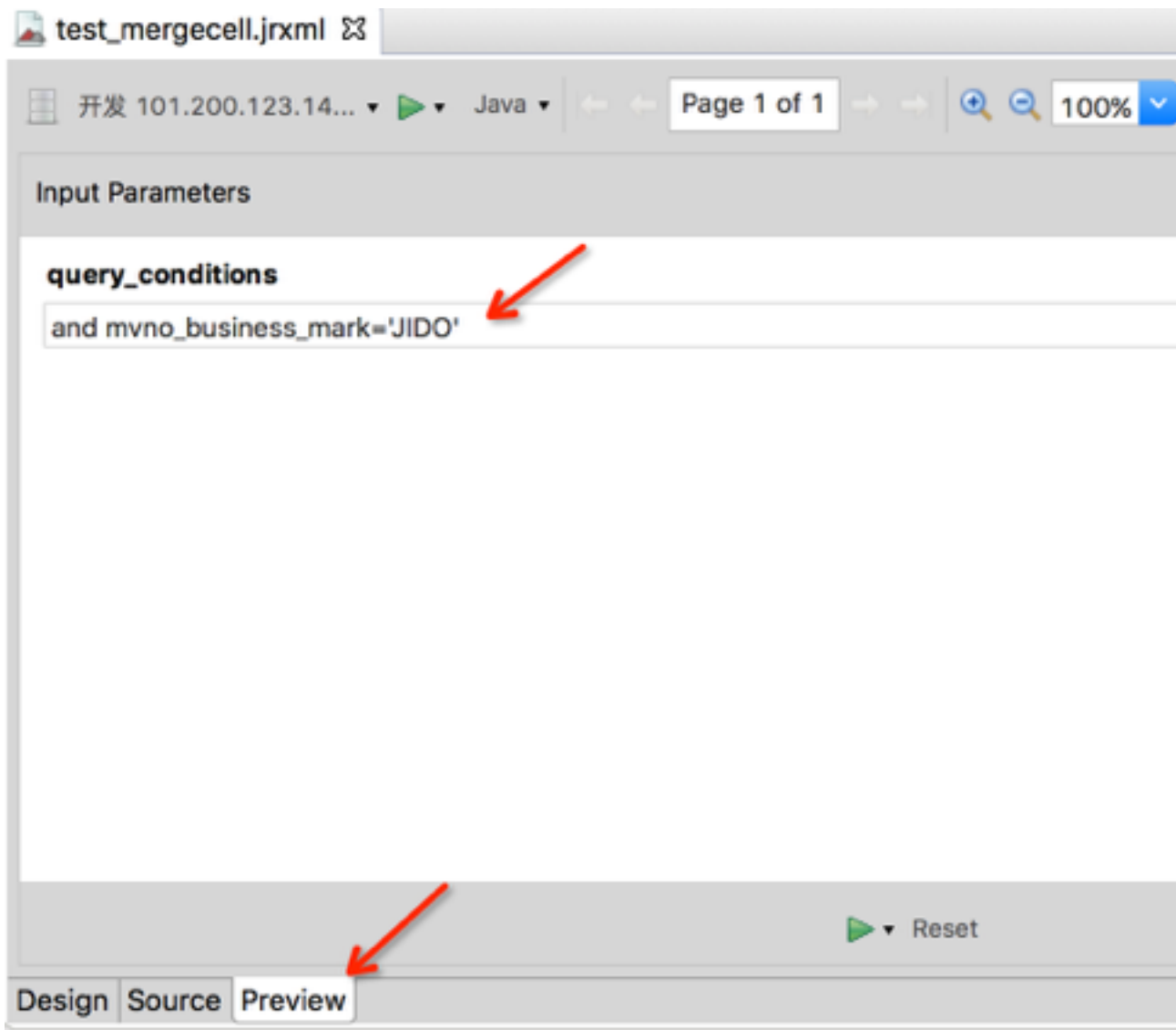
### 1. 调整 SQL ， 添加参数



使用 `$P{...}` 拼接sql



保存, 编译, 预览



设置参数： and mvno\_business\_mark='JIDO'

即：

```
select mvno_business_id,mvno_business_fname,mvno_business_mark
from mvno_business
where 1=1 $P!{query_conditions}
=》
```

```
select mvno_business_id,mvno_business_fname,mvno_business_mark
from mvno_business
where 1=1 and mvno_business_mark='JIDO'
```

效果：

显示 隐藏 公式 表格 图表 文本 形状 媒体 批注

+	test_mergecell				
转售企业指令统计					
企业ID	企业全称	指令信息			
		指令	数量		
2	北京京东叁佰陆拾度电子商务有限公司	V0022	9		
		V0019	7		
		V0018	60		
		V0001	624		
		V0014	34		
		V0002	144		
		V0010	106		
		V0004	106		
		V0003	258		
		V0024	316		
		V0023	252		
		V0016	45		
		V0017	11		

关于Jasper 可能遇到的问题：

1. JasperSoft 没有 JDBC Oracle 驱动，需要添加一下 ojdbc Jar包。
2. Jasper导出中文不显示，需要JasperSoft中安装一下宋体，并导出宋体Jar包引入到项目中。

JasperSoft 更多资料，见官方文档：

<http://community.jaspersoft.com/documentation?version=33331>

-> TIBCO Jaspersoft Studio User Guide

省份地市编码表 结果实例

结果展示：

1.

运营服务 > 智慧企业管理-Test > H5mi VOP Style

省份地市编码

省份:  查询

省份地市编码表			
省份编码	省份名	地市信息	
		地市编码	地市区名
10	内蒙古	100	海拉尔
		100	通辽市
		101	锡林郭勒盟
		102	包头市
		103	兴安盟
		104	阿拉善盟
		105	呼伦贝尔市
		此省份地市区: 14	
11	北京	110	北京市
		此省份地市区: 1	

· 联系方式 · 帮助中心

2014 中国移动通信有限公司

One

2.

省份:  查询

省份地市编码表			
省份编码	省份名	地市信息	
		地市编码	地市区名
10	内蒙古	101	呼和浩特市
		102	包头市
		103	乌兰察布市
		104	鄂尔多斯市
		105	巴彦淖尔市
		106	乌海市
		107	赤峰市
		108	海拉尔
		109	通辽市
		111	锡林郭勒盟
		112	包头市
		113	兴安盟
		114	阿拉善盟
		---	---

3.

省份地市编码

省份:

查询

省份地市编码表

省市编码	省份名	地市信息	
		地市编码	地市名
10	内蒙古	101	呼和浩特市
		102	包头市
		103	乌兰察布市
		104	鄂尔多斯市
		105	巴彦淖尔市
		106	乌海市
		107	赤峰市
		108	海拉尔
		109	通辽市
		111	锡林郭勒盟
		112	额济纳盟



F层代码示例：

```
public String queryProvincialparchyCode() throws Exception{
    String reqStr = "",serviceName="",methodName="",reqStr="";
    // 获取主表数据: 省份编码 [100]-> InputStream
    serviceName = "codeService";
    methodName = "getListProvInfo";
    reqStr = "";
    reqStr = reqStr + reqHeader(serviceName,methodName,reqStr);
    InputStream masterInputStream = new ByteArrayInputStream(reqStr.getBytes());
    // 获取子表数据: 城市编码 [100]-> InputStream
    serviceName = "codeService";
    methodName = "getListParchyInfo";
    reqStr = "";
    reqStr = reqStr + reqHeader(serviceName,methodName,reqStr);
    InputStream subreportInputStream = new ByteArrayInputStream(reqStr.getBytes());
    // JSPREPORT 参数:
    String jasperPath = this.getClass().getResource("jasperFile/test_ProvinceCode.jasper").getPath();
    Map<String,Object> paramsMap = new HashMap<String,Object>();
    paramsMap.put("REPORT_DATA_SOURCE", new JsonDataSource(masterInputStream));
    paramsMap.put("SUBREPORT_DATA_SOURCE", new JRDataSourceResettable(new JsonDataSource(subreportInputStream)));
    paramsMap.put("queryProvince", province==null?"":province);
    List<Integer> colsWidth = new ArrayList<Integer>();
    colsWidth.add(100);
    colsWidth.add(100);
    colsWidth.add(200);
    colsWidth.add(300);
    colsWidth.add(200);
    String theReport = HmUtil.processReport(jasperPath, paramsMap, colsWidth, 3, HmUtil.STYLE_SCROLL_VOP_STYLE);
    String theReport = HmUtil.processReport(jasperPath, paramsMap, reportWidth, 3, HmUtil.STYLE_SCROLL_VOP_STYLE);
    resultMap.put("theReport", theReport);
    return SUCCESS;
}
```

数据源

数据源

参数

调用工具类,生成html字符串

/\*\*

\* 处理 jasper 到 html 文本

\* @param jasperPath : jasper文件路径

\* @param paramsMap : jasper参数

\* @param colsWidth : html-table 列总宽度 (经浮点运算后 实际结果可能有细微偏差)

\* @param tHeadRowNum : 表头行数

\* @param reportStyle : 样式模版

\* @return theReport

\* @throws JRException

\*/

public static String processReport(String jasperPath, Map<String,Object> paramsMap, int colsSumWidth, int tHeadRowNum, String reportStyle)

```
<body>[
<div class="titlePanel">[
... 报表名字输入</div>
... <!-- [
    <button id="exportExcelBtn" class="btn button exportBtn" style="margin-right: 10px;">导出Excel</button>
    ... ]-->[
    <form id="exportForm" action="#" method="post">[
        <input id="sheetArrayJsonStr" name="sheetArrayJsonStr" type="hidden"/>[
        </form>[
    </div>[
    <input type="hidden" id="sysMainMenuNo" value="VOP_OPKS_000001"/>[
    <input type="hidden" id="subSysMenuNo" value="VOP_MWNO_SFAT_0005"/>[
    <div id="queryPanel" class="queryPanel">[
        <div class="queryPanelRow">[
            <span class="inlineLabel">省份:</span>[
            <div class="inlineInputGroup">[
                <div class="input-group">[
                    <input id="province" class="form-control" type="text" placeholder="请输入省份中文名"/>[
                    </div>[
                </div>[
            <button id="queryBtn" class="btn button btn-primary">查询</button>[
            <i class="fa fa-spinner fa-spin" id="loading"></i>[
        </div>[
    </div>[
    <div id="theReport">[
    </div>[
... ]
]

```

**查询条件、等元素**

**报表展示位置**

```

1 $(function(){
2     var queryBtn = $("#queryBtn");
3     var province = $("#province");
4
5     // 查询
6     $(queryBtn).on('click', function(e) {
7         $.tydicAjax({
8             url: "/VOPMNG_F/operSer/queryProvinceEparchyCodeHtmlVopStyle",
9             type: 'POST',
10            data: {
11                province: $(province).val(),
12                reportWidth: $("#theReport").width()
13            },
14            eventBtn: $(this),
15            isSuccessDupl: true
16        });
17        .done(function (data){
18
19            var theReport = data.theReport;
20            $("#theReport").html(theReport);
21
22        });
23    });
24 });

```

把action中处理完成的报表，填充到html页面中

导出示例：

```
84 > }
85 > // 导出
86 > public String exportMvnoInstStatistics() throws Exception{
87 >     String[] sheetArray = JsonUtil.fromJsonToObject(sheetArrayJsonStr, new TypeReference<String[][]> {});
88 >     Workbook wb = new HSSFWorkbook();
89 >     ExcelUtil.addSheetArrayToWorkbook(wb, sheetArray, "this is Sheet Name");
90 >     // fileName = URLLEncoder.encode("销售转售企业指令统计.xlsx", "UTF-8");
91 >     fileName = new String("销售转售企业指令统计.xlsx").getBytes("UTF-8");
92 >     FileOutputStream fileOut = new FileOutputStream(fileName);
93 >     wb.write(fileOut);
94 >     fileOut.close();
95 >     String[] tempSheetArray = ExcelUtil.analyzeExcelFile(wb);
96 >     return "exportReportToExcel";
97 > }
98 > }
99 > public List<Map> getSelectMvnos() {
100 >     return selectMvnos;
101 > }
102 > public void setSelectMvnos(List<Map> selectMvnos) {
103 >     this.selectMvnos = selectMvnos;
104 > }
```

获取特定格式的数据

使用工具类把数据解析到POI对象中

POI对象 写入输出流

其他 VOP 集成 Jasper 示例效果图：

运营服务 > 转售企业管理-Test > 转售企业指令统计报表

转售企业管理-Test

转售企业查询\_Test

转售企业指令统计报表

万号段开户数配置

信息管理

测试号卡管理

接入管理

承诺信息管理

转售企业指令统计报表

转售企业: 全部

导出Excel

转售企业指令统计报表						
企业名称	企业标识	指令详情			总数	
		网别	指令	指令状态		
巴士在线股份有限公司	GOME	4G	V0004	1	43	
		4G	V0004	2	8	
		巴士在线股份有限公司指令发生总数:				271
		0	V0001	1	1	
		0	V0002	1	1	
		4G	V0001	1	5	
		4G	V0001	2	6	
		4G	V0023	1	4	
北京国美电器有限公司	GOME	4G	V0024	2	5	
		北京国美电器有限公司指令发生总数:				22
		0	V0001	0	1	

运营服务 > 转售企业管理-Test > 转售企业指令统计报表

转售企业管理-Test

转售企业查询\_Test

转售企业指令统计报表

万号段开户数配置

信息管理

测试号卡管理

接入管理

承诺信息管理

转售企业指令统计报表

转售企业: 全部

导出Excel

转售企业指令统计报表					
企业名称	企业标识	指令详情			总数
		网别	指令	指令状态	
达特(北京)通信技术有限公司	GOME	4G	V0010	2	1
		4G	V0014	1	3
		4G	V0014	2	1
		4G	V0015	1	2
		4G	V0023	1	33
		4G	V0023	2	12
		4G	V0024	0	1
		4G	V0024	1	25
达特(北京)通信技术有限公司	GOME	4G	V0024	2	9
		达特(北京)通信技术有限公司指令发生总数:			
转售企业已发生指令总数:					2762

联系方式 · 帮助中心

2014 中国联通网络通信有限公司

Jasper 设计图：  
主表：

mvno\_inst\_subscribe\_analyse.jrxml   mvno\_inst\_subscribe\_analyse\_S\_inst.jrxml

Main Report

0 1 2 3 4 5 6 7 8 9

转售企业指令统计报表					
企业名	企业标识	指令详情			
		Column Header		指令状态	总数
		网别	指令		
Detail 1					
\$F {MVNO_BUSINESS_FNAME} \$F {MVNO_BUSINESS_MARK}					
\$F {MVNO_BUSINESS_FNAME}		指令发生次数		\$F {MVNO_INST_COUNT}	
Footer 1					
"转售企业已发生指令总数" \$F {total_inst}					

子表：

mvno\_inst\_subscribe\_analyse.jrxml   mvno\_inst\_subscribe\_analyse\_S\_inst.jrxml

Main Report

0 1 2 3 4 5 6 7 8

\$F {SERVICE_CLASS_CODE}	\$F {SERVICE_INST}	\$F {SERVICE_INST_STATUS}	\$F {INST_COUNT}
--------------------------	--------------------	---------------------------	------------------