**JasperReport 6.17.0**

**使用说明**

**V1.0**

**目 录**

[1. 简介 1](#_Toc105935387)

[1.1. 报表生命周期 1](#_Toc105935388)

[1.2. 界面 2](#_Toc105935389)

[2. 基本使用 3](#_Toc105935390)

[2.1. 创建列表式报表 3](#_Toc105935391)

[2.1.1. 绘制报表 3](#_Toc105935392)

[2.1.2. 通过JSON生成报表 6](#_Toc105935393)

[2.1.2.1. 配置数据适配器（JSON） 6](#_Toc105935394)

[2.1.2.2. 读取字段列表 9](#_Toc105935395)

[2.1.2.3. 报表各部分组成 10](#_Toc105935396)

[2.1.2.4. 绘制报表 11](#_Toc105935397)

[2.1.2.4.1. 添加标题 11](#_Toc105935398)

[2.1.2.4.2. 添加字段 12](#_Toc105935399)

[2.1.2.4.3. 设定PDF字体（重要！） 14](#_Toc105935400)

[2.1.2.4.4. 设定表格边框 14](#_Toc105935401)

[2.1.2.4.5. 设定各区域高度 15](#_Toc105935402)

[2.1.2.4.6. 设置字段组件 16](#_Toc105935403)

[2.1.3. 通过查询数据库生成报表 17](#_Toc105935404)

[2.1.3.1. 配置数据适配器（DB） 17](#_Toc105935405)

[2.1.3.2. 设置查询语句、读取字段列表 20](#_Toc105935406)

[2.1.3.3. 参数的使用 21](#_Toc105935407)

[2.1.3.3.1. 添加参数 21](#_Toc105935408)

[2.1.3.3.2. 程序中传递参数 23](#_Toc105935409)

[2.1.3.3.3. 报表模板接收参数 23](#_Toc105935410)

[2.1.3.4. 自定义变量的使用 23](#_Toc105935411)

[2.1.3.4.1. 顺序号 23](#_Toc105935412)

[2.1.3.4.2. 小计 25](#_Toc105935413)

[2.1.3.4.3. 合计 27](#_Toc105935414)

[2.1.3.4.3.1. 表头合计 27](#_Toc105935415)

[2.1.3.4.3.2. 表尾合计 27](#_Toc105935416)

[2.2. 创建卡片式报表 29](#_Toc105935417)

[2.3. 编译、发布 29](#_Toc105935418)

[3. 报表设计器进阶 29](#_Toc105935419)

[3.1. 加入图片 29](#_Toc105935420)

[3.2. 页码相关 33](#_Toc105935421)

[3.2.1. 添加页码 33](#_Toc105935422)

[3.2.2. 根据页码进行打印控制 34](#_Toc105935423)

[3.3. 参数 35](#_Toc105935424)

[3.3.1. 添加参数 35](#_Toc105935425)

[3.3.2. 程序中传递参数 36](#_Toc105935426)

[3.3.3. 报表模板接收参数 37](#_Toc105935427)

[3.4. 自定义变量 37](#_Toc105935428)

[3.4.1. 顺序号 37](#_Toc105935429)

[3.4.2. 小计 39](#_Toc105935430)

[3.4.3. 合计 40](#_Toc105935431)

[3.4.3.1. 表头合计 40](#_Toc105935432)

[3.4.3.2. 表尾合计 41](#_Toc105935433)

[3.5. 分栏报表（横向循环） 42](#_Toc105935434)

[3.6. 加入条码/二维码 45](#_Toc105935435)

[3.7. 分组报表 47](#_Toc105935436)

[3.8. 图形报表 50](#_Toc105935437)

[3.9. 子报表 52](#_Toc105935438)

[3.9.1. 制作父报表 52](#_Toc105935439)

[3.9.2. 制作子报表 53](#_Toc105935440)

[3.9.3. 子报表自适应行高 58](#_Toc105935441)

# 简介

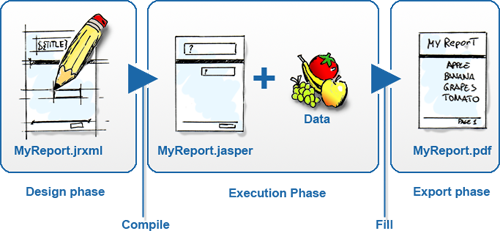
Jaspersoft Studio是一个专为JasperReports报表引擎而开发的报表设计器，基于Eclipse平台实现。它能够让你创建包含图表、图片、子报表、交叉表等非常复杂的布局。可以通过JSON、JDBC、TableModels、JavaBeans、XML、Hibernate、CSV或自定义数据源来读取数据。可以将报表输出成PDF、RTF、XML、XLS、CSV、HTML、XHTML、TEXT、DOCX或OpenOffice。

## 报表生命周期

当你使用JasperSoft Studio做报表时，要创建一个JRXML文件，该文件是包含报表布局定义的XML文档。布局是完全可视化，这样你就可以忽略JRXML文件的基础结构。执行报告之前，JRXML必须编译成名为Jasper的二进制文件。Jasper文件就是应用程序生成报表所需要的。

有许多数据源类型。您可以从SQL查询、HQL（Hibernate查询语言）查询，传入JSON、XML、CSV，也可以传入JavaBeans集合等。如果你没有一个合适的数据源，JasperReports的允许你编写自己的自定义数据源。通过Jasper文件和数据源，JasperReports 就能生成最终用户想要的文档格式。

JasperSoft Studio允许您配置数据源，并使用它们来测试您的报表。在许多情况下，“数据驱动向导”功能可以帮助您快速设计您的报表。JasperSoft Studio包括JasperReports引擎本身，可以让您预览报表输出，测试和优化您的报表。

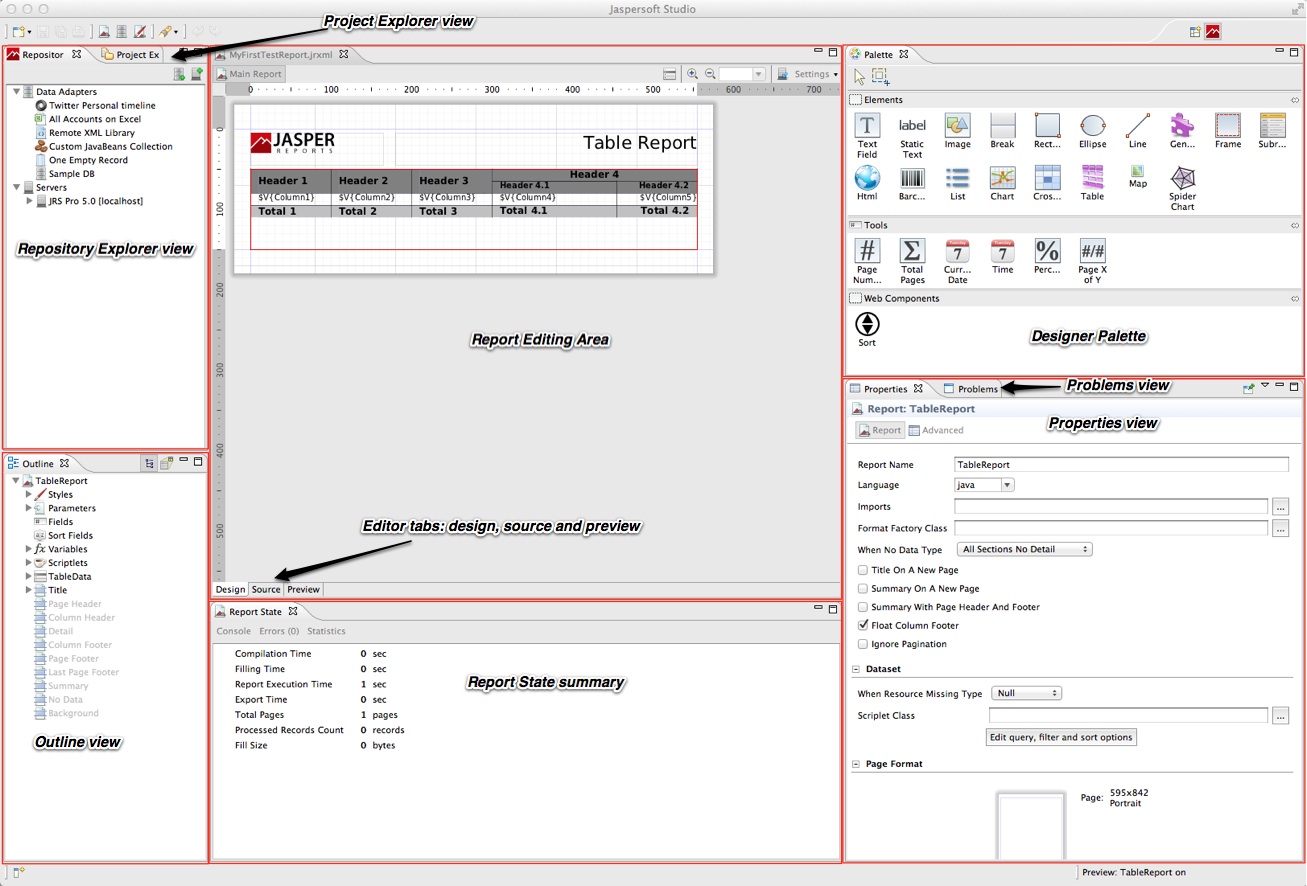


## 界面

Jaspersoft Studio安装完毕后，会在桌面放置一个图标。



双击此图标，即可运行报表设计器。下面可以看到JasperSoft Studio界面：



**Report editing area（主编辑区域）**中，您直观地通过拖动，定位，对齐和通过**Designer palette（设计器调色板）**对报表元素调整大小。

JasperSoft Studio有一个多标签编辑器，**Design**,**Source**和 **Preview**：

**Design tab**：当你打开一个报告文件，它允许您以图形方式创建报表选中

**Source tab**： 包含用于报表的JRXML源代码。

**Preview tab**： 允许在选择数据源和输出格式后，运行报表预览。

很多页面可以查看数据：

**Repository Explorer view**：包含JasperServer生成的连接和可用的数据适配器列表

**Project Explorer view**：包含JasperReports的工程项目清单

**Outline view**：在大纲视图中显示了一个树的形式的方式报告的完整结构。

**Properties view**：通常是任何基于Eclipse的产品/插件的基础之一。它通常被填充与实际所选元素的属性的信息。这就是这样，当你从主设计区域（即：一个文本字段）选择一个报表元素或从大纲，视图显示了它的信息。其中一些属性可以是只读的，但大部分都是可编辑的，对其进行修改，通常会通知更改绘制的元素（如：元素的宽度或高度）。

**Problems view**：显示的问题和错误，例如可以阻断报告的正确的编译。

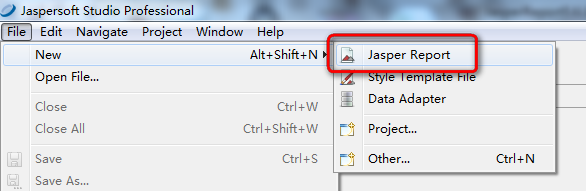
最后，**Report state summary**提供了有关在报表编译/填充/执行统计用户有用的信息。错误会显示在这里。

# 基本使用

## 创建列表式报表

### 绘制报表

Let’s begin！！！先来创建一个新的空白报表开练！点击菜单中的【文件】-【New】，系统会创建一个新的空白报表模板。

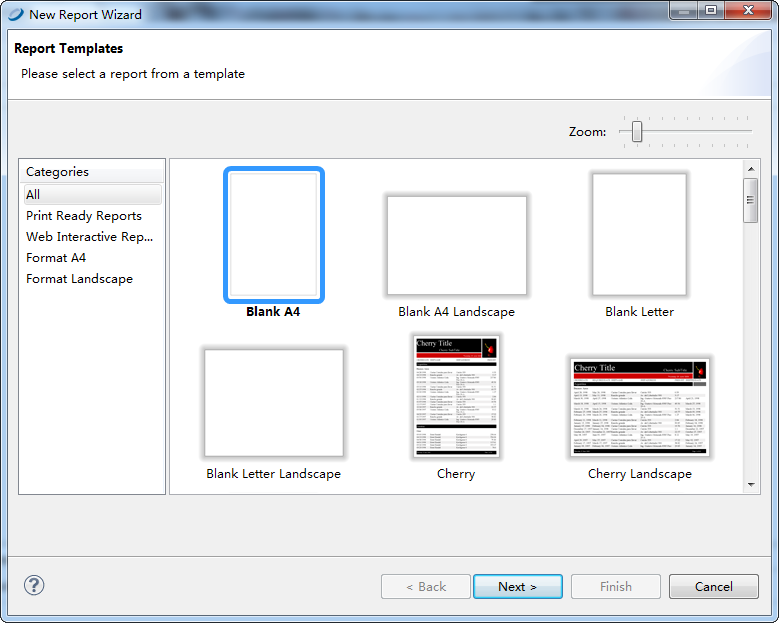


或者点击工具栏上的图标。



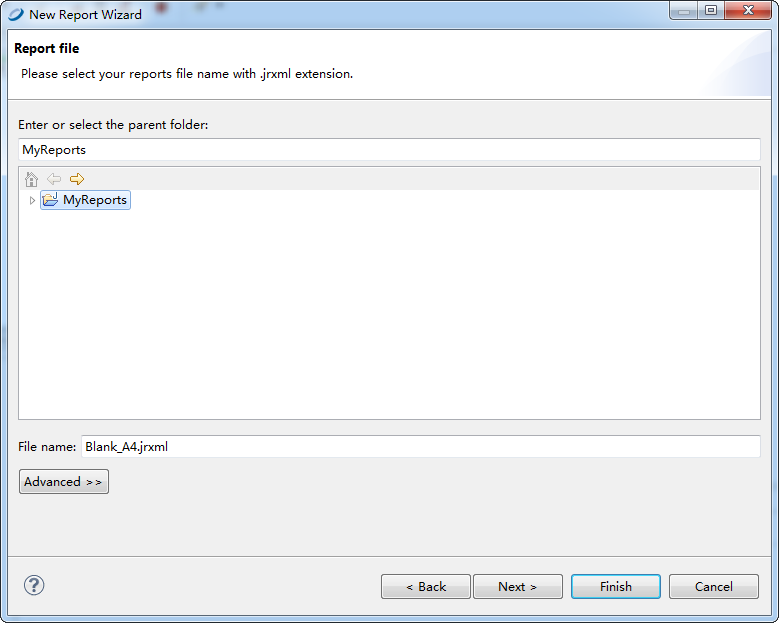
Jaspersoft Studio提供了丰富多样的报表和图表，几乎可以满足日常工作的各种需要。

为了便于讲解，我们选择最标准的A4竖向空白报表。

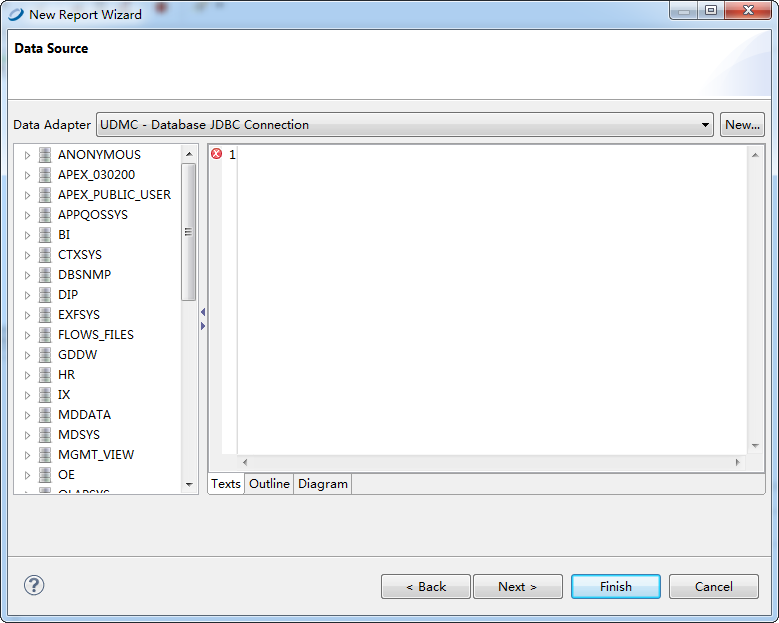


选择完毕，点击【Next】。

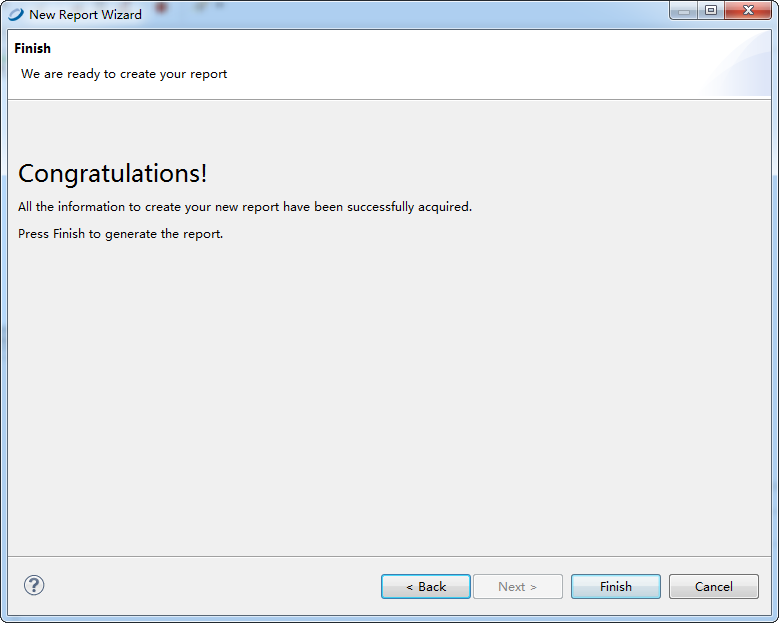
系统显示“名称和位置”界面，可以在这里设置报表的名称，以及报表文件存储的位置。



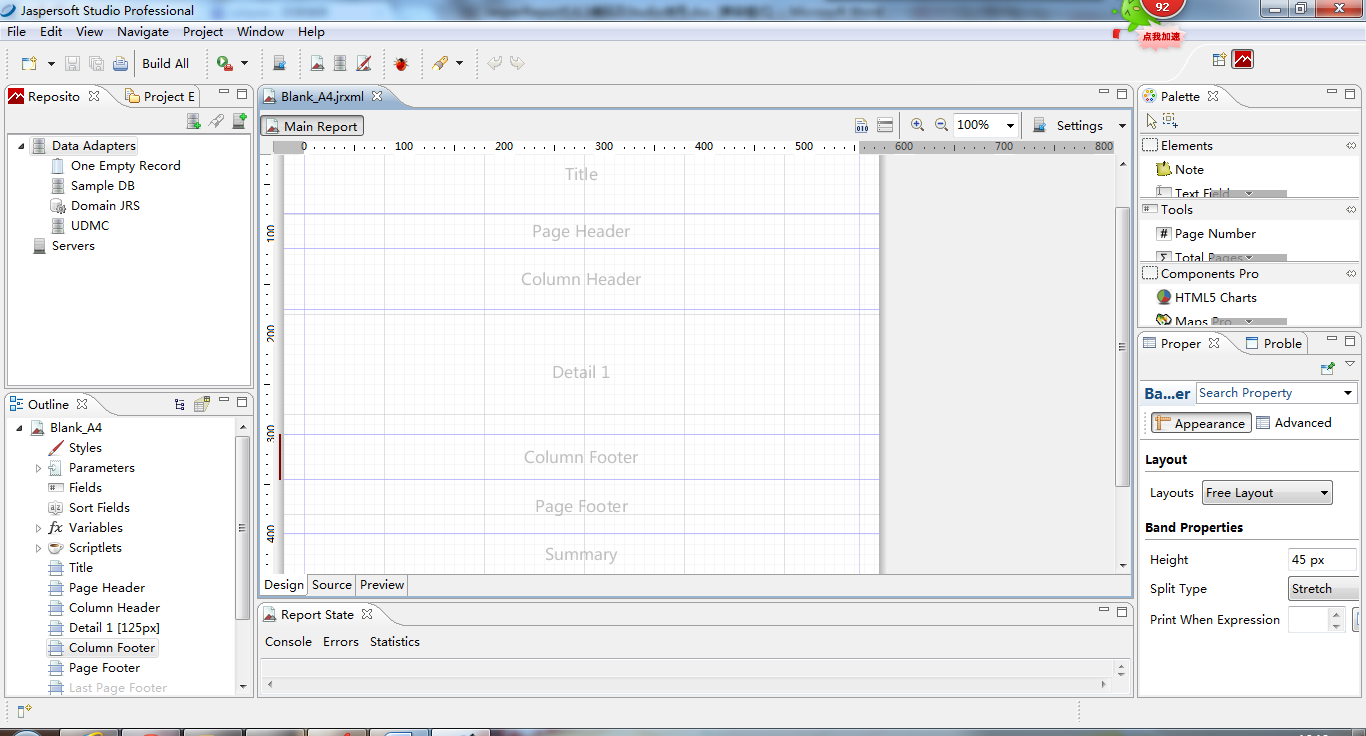
设置完毕，点击【Next】，系统进入“数据源”界面。可以选择已经配置在系统中的数据源，或者点击【New】按钮，创建新的数据源连接。可以在此处设置查询语句，查询需要编制报表的原始数据。



设置完毕，点击【Next】，系统进入“完成”界面。点击【Finish】按钮，即完成了报表的生成。



空白报表的界面如下。

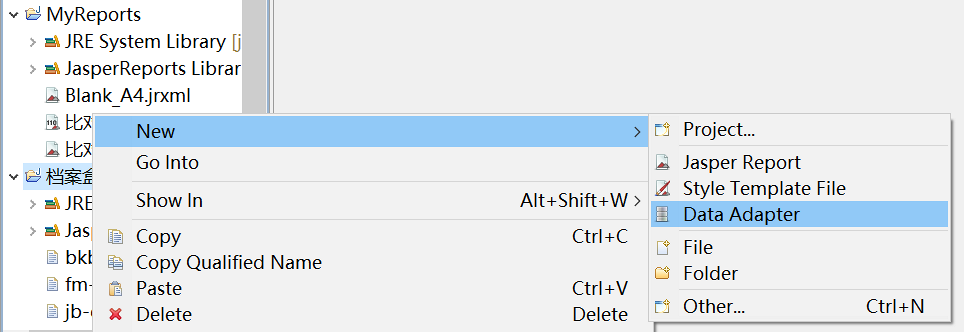


### 通过JSON生成报表

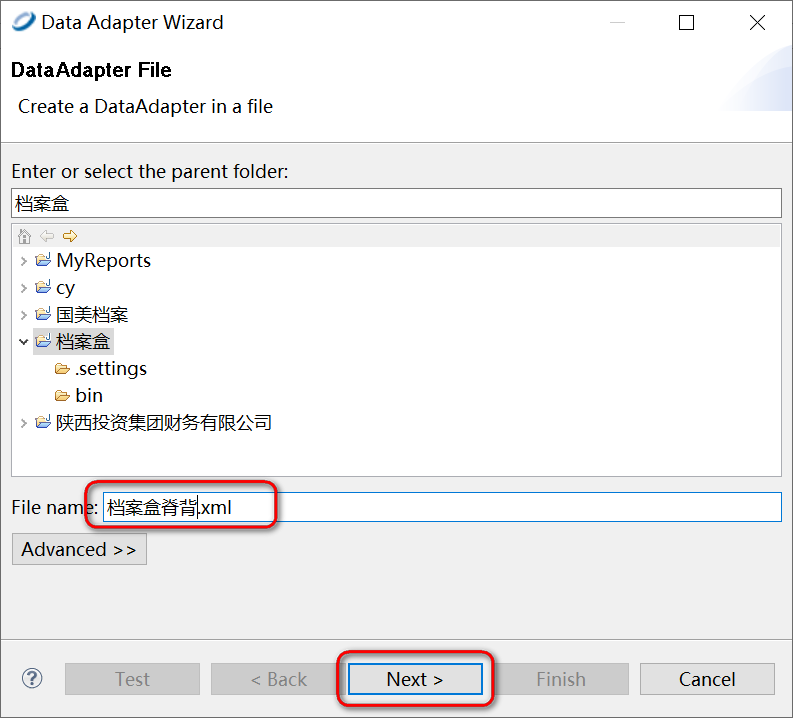
#### 配置数据适配器（JSON）

要做报表，先要配置一个数据连接，否则咱们报表里的数据从哪里来涅？

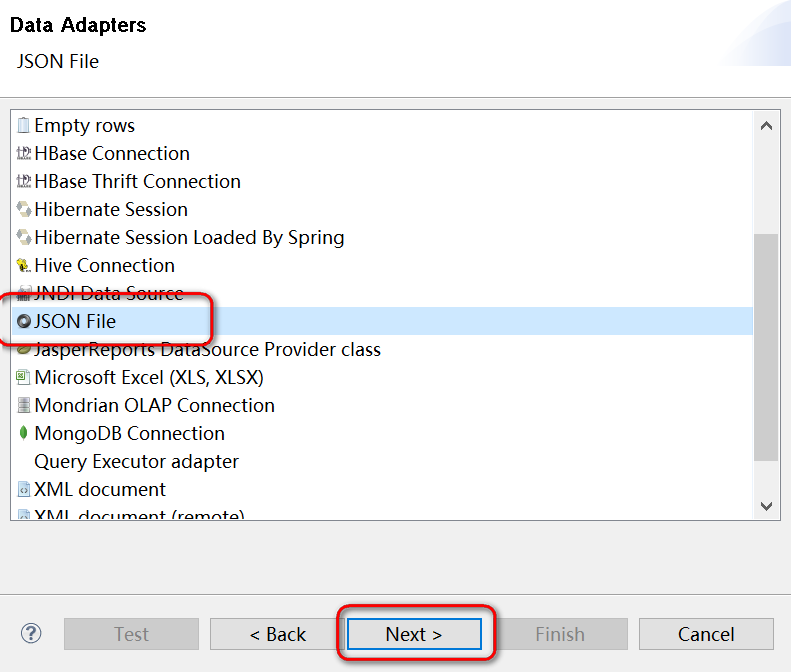
在左侧Repository Explorer区中，在当前操作的报表工程上点击鼠标右键，选择【New】-【Data Adapter】，创建“数据适配器”。



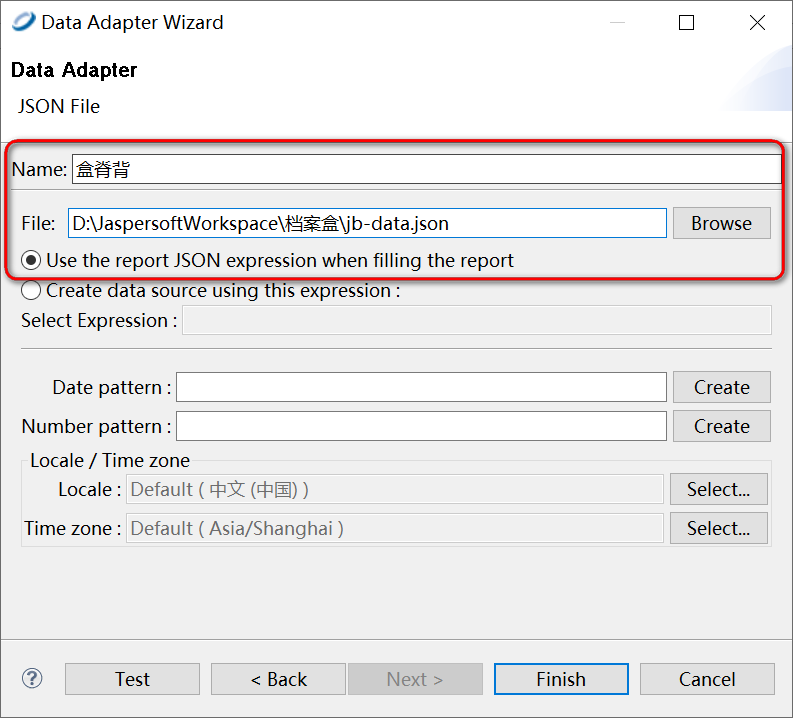
系统显示如下界面：



此处可以对数据适配器配置文件进行重命名。比如：“档案盒脊背.xml”。点击【Next】按钮。



本例，最终需要通过接口传入JSON，所以在编制报表时，使用示例的JSON文件。此处选择【JSON File】，点击【Next】按钮。



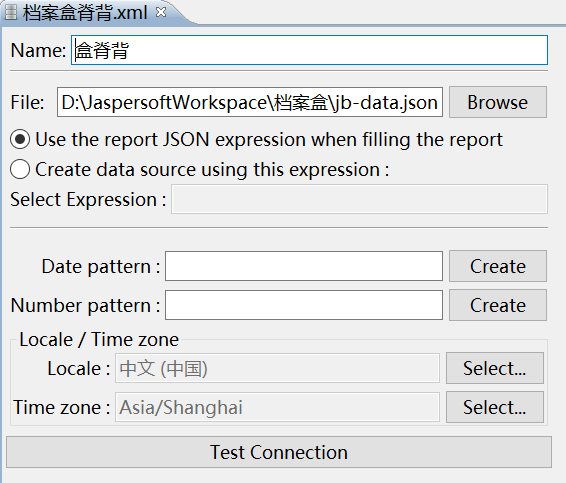
这一步，需要：

（1）Name：给创建的这个数据连接起个名字，这里是在编制报表、预览报表时选择数据源时显示用的内容，本例输入“盒脊背”。

（2）File：选择示例JSON文件。选择时，一定要复制一下文件所在的文件夹，然后粘贴过来，形成绝对地址。

（3）类型：选择第一项“Use the report JSON expression when filling the report”。

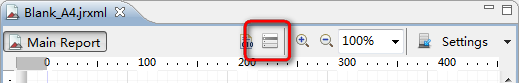
选择完毕，点击【Finish】按钮，即完成数据源设置。



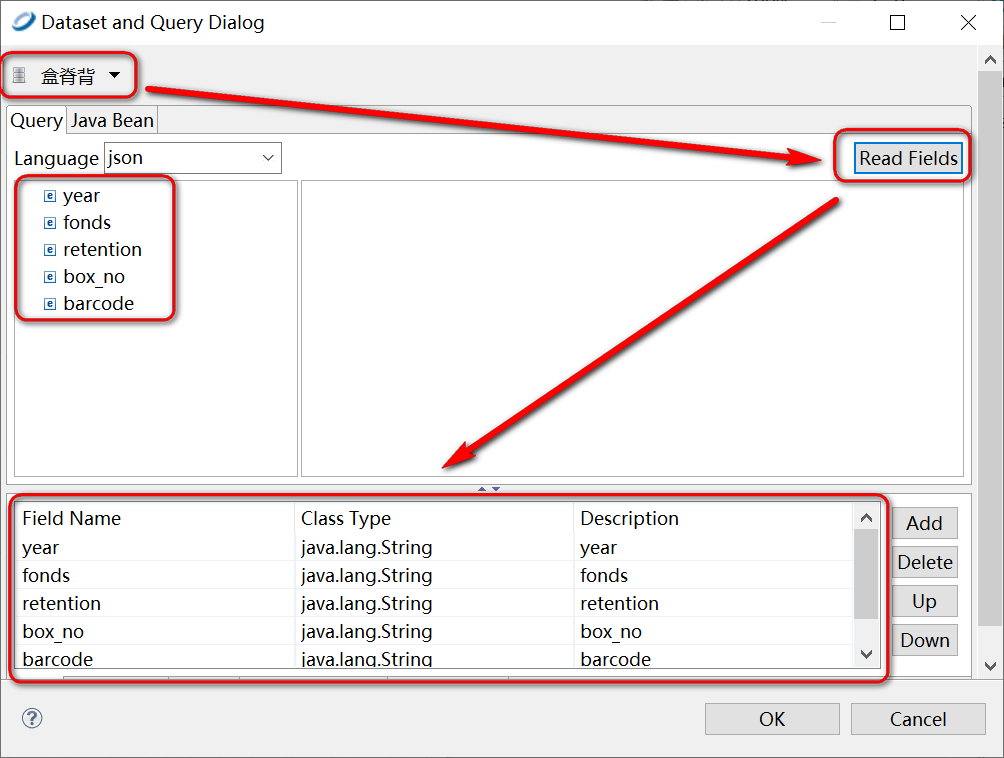
#### 读取字段列表

数据适配器设定完毕后，就可以读取JSON中的结构，查询字段列表，用以后续设置、填充报表了。

点击报表设计工作区中，工具栏上的“DataSet and Query editor dialog”按钮，如下图。



点击后，系统会显示“DataSet and Query Dialog”界面。

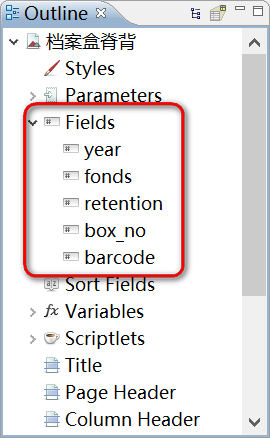


点击左上角的【数据适配器】，选择刚刚创建的“盒脊背”，下方的字段列表自动刷新为JSON文件中的key。

点击右侧的【Read Fields】按钮，系统将JSON中的key作为字段，读入下方的字段列表中。

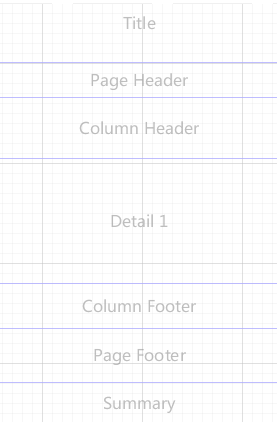
点击【OK】按钮，完成配置。

此时，在报表的“Fields”中，即可看到本次加载的所有字段。后续可以直接选择使用。



#### 报表各部分组成

下面，就可以正式开始制作报表模板了。首先，需要了解一下报表各部分的组成：

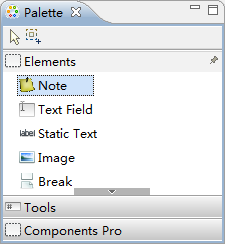


1. Title : 报表的表头。只在首页打印一次。
2. PageHeader : 报表的页首。每页都打印。
3. ColumnHeader : 报表的行首，通常用来定义行的字段名称。每页都打印。
4. Detail : 报表的内容。根据数据数量，自动循环输出。
5. ColumnFooter : 报表的行尾，可用来告知报表的一些参数，诸如页码等。每页都打印。
6. PageFooter : 报表的页尾。每页都打印。
7. LastPageFooter : 最后页的页尾。只在最后页打印一次。
8. Summary : 合计、计算用。

#### 绘制报表

##### 添加标题

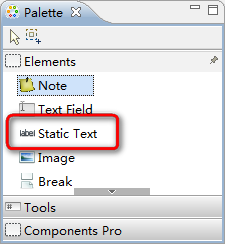
我们以制作列表式的报表为例。我们可以在右侧【Palette】标签的【Elements】组中找到需要的组件。



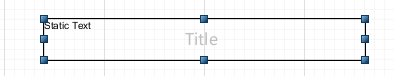
首先需要设定标题。标题中，一般都是报表的名称，所以添加不可修改的文字即可。

如果标题只在第一页输出，则需要将标题文字放在“Title”区；如果标题每页都要输出，则应该放在“pageHeader”，即“页首”区。

点击“Elements”上的“Static Text”组件，如下图。

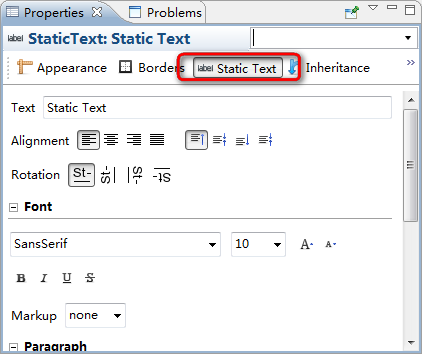


然后将组件拖拽到“Title”区或“pageHeader”区中即可。



在此组件中输入标题文字即可。输入完毕，需要设置文字属性，对文字进行排版。

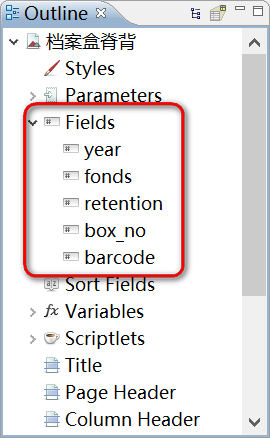
在右侧的【Properties】标签中，可以设置静态文本框的位置、边框、文字属性等等。点击相应的页签，即可进行对应的设置。本例中，点击【Static Text】页签，即可设置文字属性。



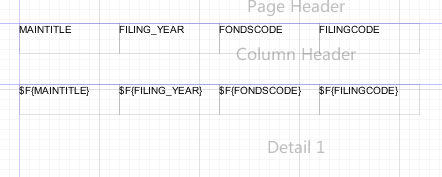
##### 添加字段

设置完标题，即可设置报表需要输出的内容。首先可以设定需要输出的字段。

在左侧“Outline”区中，展开“Fields”项，即可看到此时报表模板中所能访问的所有字段。



选中一个需要打印的字段，然后拖动至报表中，放在“Detail”区中。拖动所有需要输出的字段，在“Detail”区中排好位置。



系统会自动在“Column Header”区中放置与字段对应的“表头文本”，我们只需调整位置、修改文字、设置格式即可。



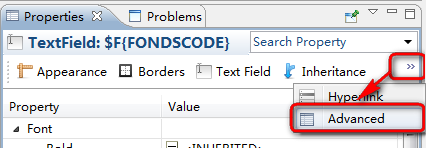
***Tips:***

*如果使用Excel作为设置的模板，Excel中，行高的高度与Studio中设置基本相同，而列宽单位与行高单位不同，1单位（列宽）=6.48磅（行高），可以根据此公式进行换算。*

##### 设定PDF字体（重要！）

为了让报表在PDF格式输出时可以正常输出中文，需要设定每个组件的文字字体。

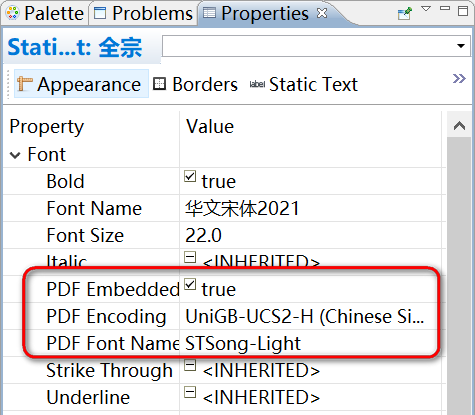
【文字组件】和【字段组件】均需要设置。选择组件后，在右侧的【Properties】页，选择【Advanced】页，即可在此处修改相关字体的设置。



以下3项必须修改，否则生成的PDF无法正常输出中文。*（多个组件无法一起批量修改，只能一个一个改。建议设置好一组，剩下的复制、粘贴）*

具体设置如下：

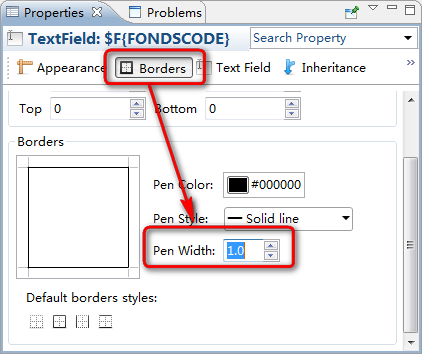
1. PDF Embedded：True
2. PDF Encoding：UniGB-UCS2-H（Chinese Simplified）
3. PDF Font Name：STSong-Light



##### 设定表格边框

报表中可以设定每个组件的边框。一般我们会选择四边均有边框。

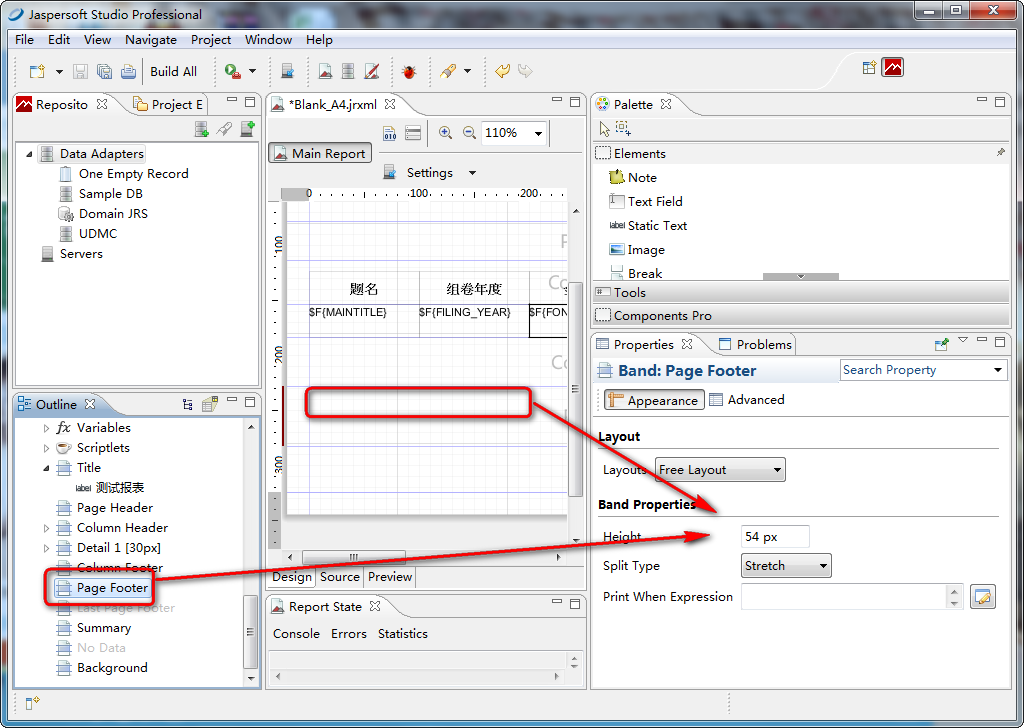
点击需要设定边框的字符或者字段域，在右侧的【Properties】页，选择【Borders】页，先点选“Borders”中模拟图的4个边，修改【Pen Width】（笔宽度），例如设置为“1”，即可显示此域的边框。



##### 设定各区域高度

为了让报表输出的更美观，我们可以设定报表各个区域高度。

点击左侧树形图中各个区域的名称，或者直接点击工作区中的区域，在右侧【Properties】页中，即可修改当前区域的高度。不需要显示的区域，可以将高度设置为0。



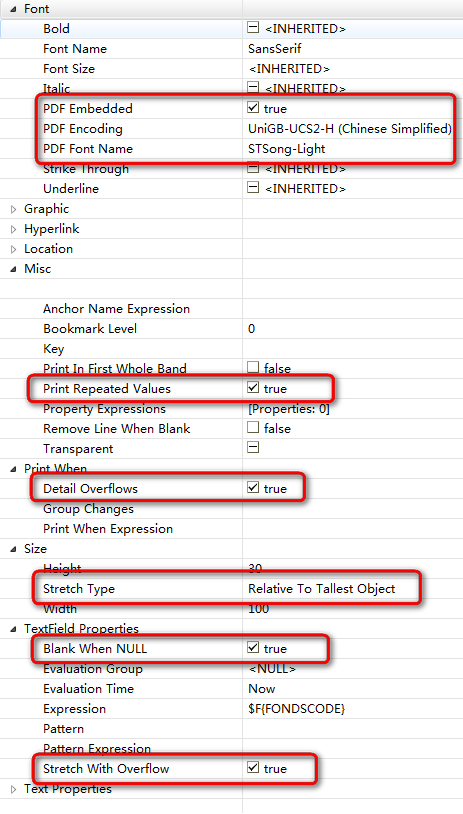
为了使Detail区中的表格可以连续输出，需要将Detail区的高度设为与字段组件高度相同。本例中，字段组件高度为30，则Detail区域的高度也设定为30。

##### 设置字段组件

以下，说明一下报表中“字段组件”的一些常用设置。

在打印报表时，有的时候会遇到字段中的内容很长，标准的表格打印不下的情况。此时，用户就会希望表格能够自动适应内容的多少而自动增加行高。在JasperReport中可以通过设置实现。

选中所有需要设置的字段组件，在右侧【Properties】页中可以进行如下设置。



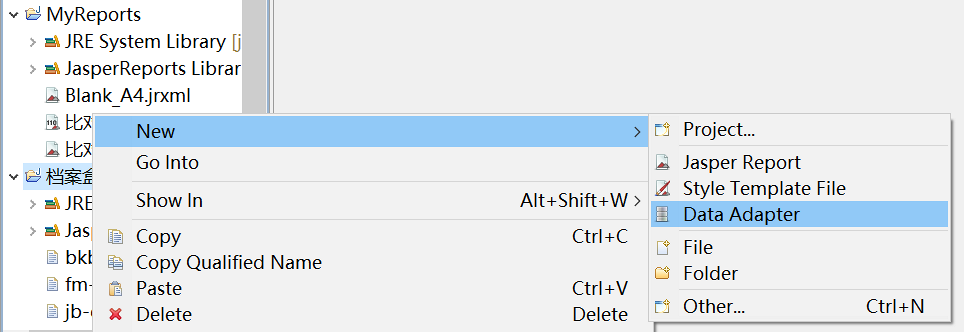
1. **“Font”-“PDF embedded”：选择“true”，可以将报表输出到PDF中。**
2. **“Font”-“PDF Encoding”：选择“UniGB-UCS2-H（Chinese Simplified）”，保证PDF中的中文可以正确编码。**
3. **“Font”-“PDF Font Name”：选择“STSong-Light”，保证PDF报表中的中文可以正确显示。**
4. “Misc”-“Print repeated values”：选择“true”。在打印时，可以输出相同的值。
5. “Print When”-“Detail Overflow”：选择“true”。在数据当页没有打印完毕，需要在第二页打印时，可以将表格的内容，包括边框，在第二页中输出。
6. “Size”-“Stretch Type”：选择“Relative to Tallest Object”。则所有被选中的字段域可以自动适应条目拉伸。
7. “TextField Properties”-“Blank when null”：选择“true”，是当字段的值是Null时，输出空白格。
8. “TextField Properties”-“Stretch with overflow”：选择“true”，当文本域的内容不能完全被显示在模板定义的区域内时，允许文本域拉伸。

### 通过查询数据库生成报表

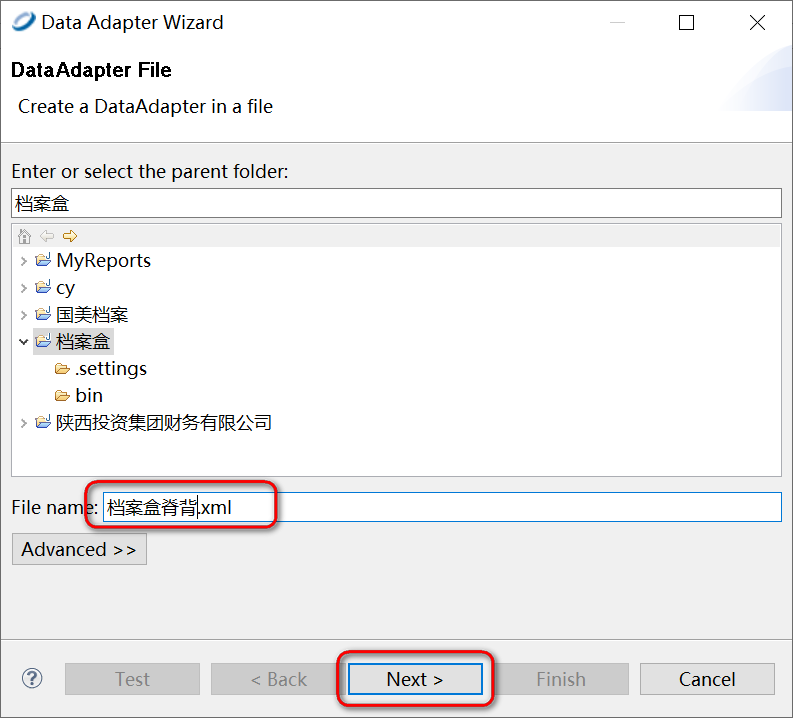
报表的绘制过程、方法，与前述章节相同。本章节只说明不同部分。

#### 配置数据适配器（DB）

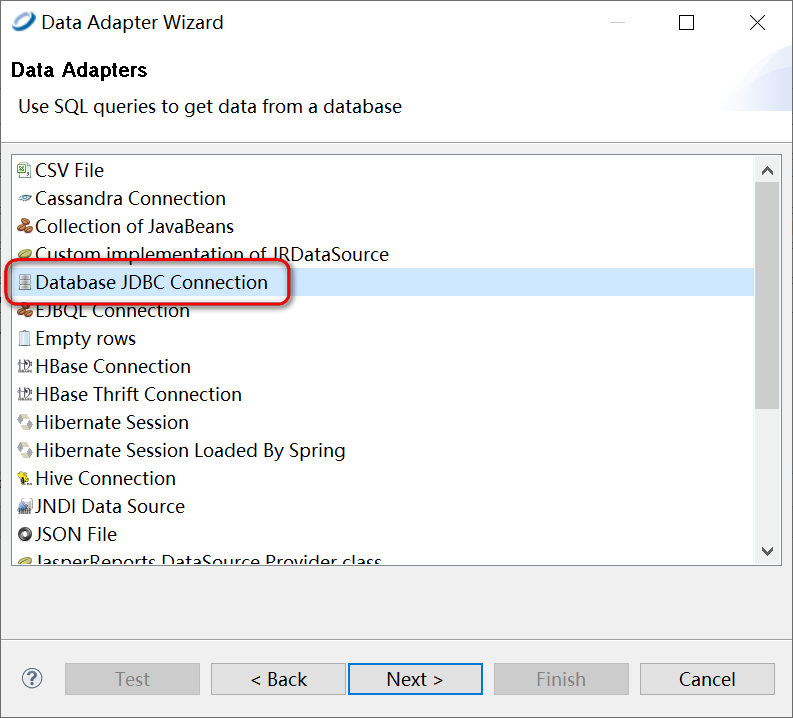
在左侧Repository Explorer区中，在当前操作的报表工程上点击鼠标右键，选择【New】-【Data Adapter】，创建“数据适配器”。



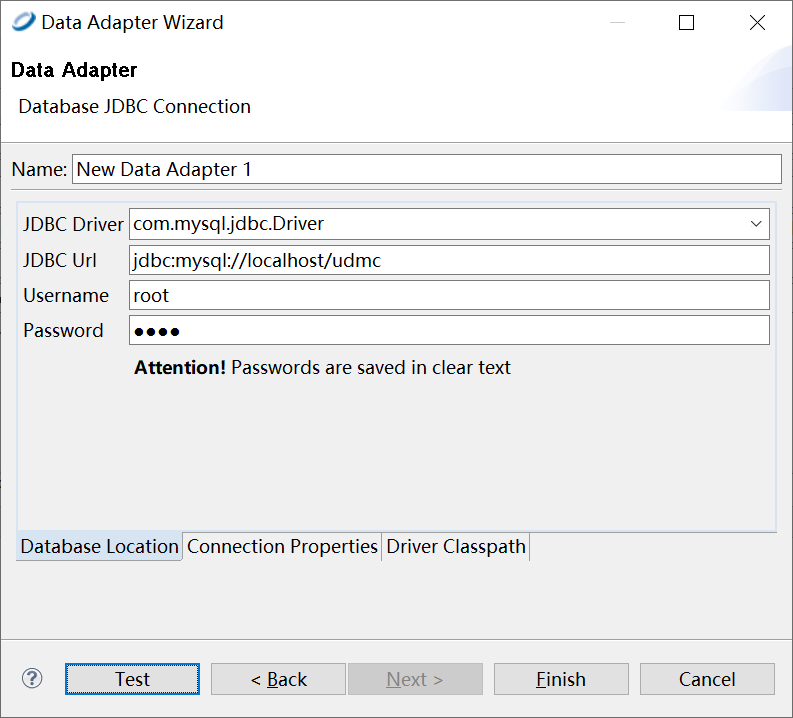
系统显示如下界面：



此处可以对数据适配器配置文件进行重命名。比如：“档案盒脊背.xml”。点击【Next】按钮。



这一步，选择数据源的类型。这里有各种你喜欢的或者你不喜欢的数据源。我是个传统的人……我用关系型数据库，所以我选择“Database JDBC Connection”。【Next】！



这一步，需要：

（1）给创建的这个数据连接起个名字；

（2）根据数据库选择驱动类型，本例我们使用的是MySQL；

（3）修改一下数据连接URL，要符合你实际的数据库IP、服务名什么的；

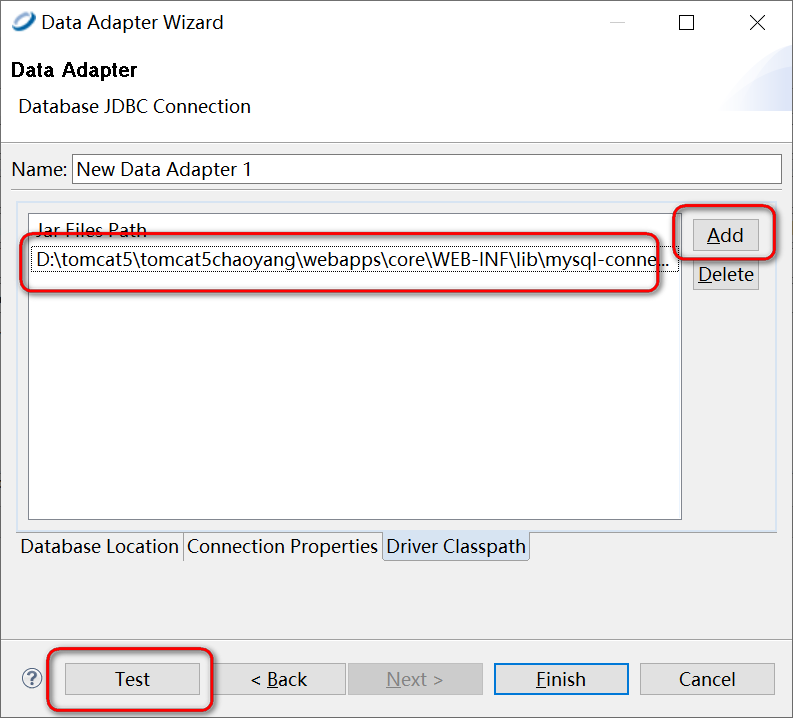
（4）输入用户名、密码；

（5）测试一下，如果能连接成功，就可以做后面的工作了。

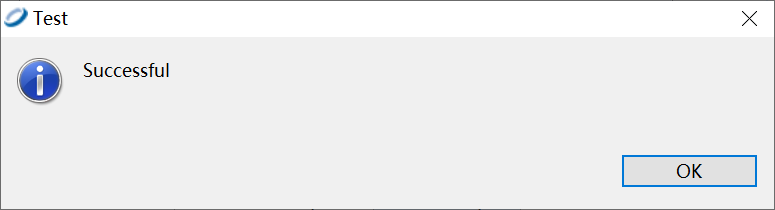
可惜，连接失败，提示没有找到JDBC驱动类…………看来这个数据库的驱动并没有内置。

我们需要手动添加数据库的驱动jar包。

点击【Driver Classpath】页，选择本地的MySQL的JDBC驱动jar包。



添加完毕后，一定要点击【Test】按钮测试一下，成功了才算完毕。

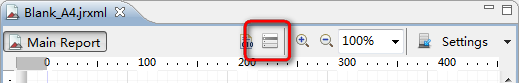


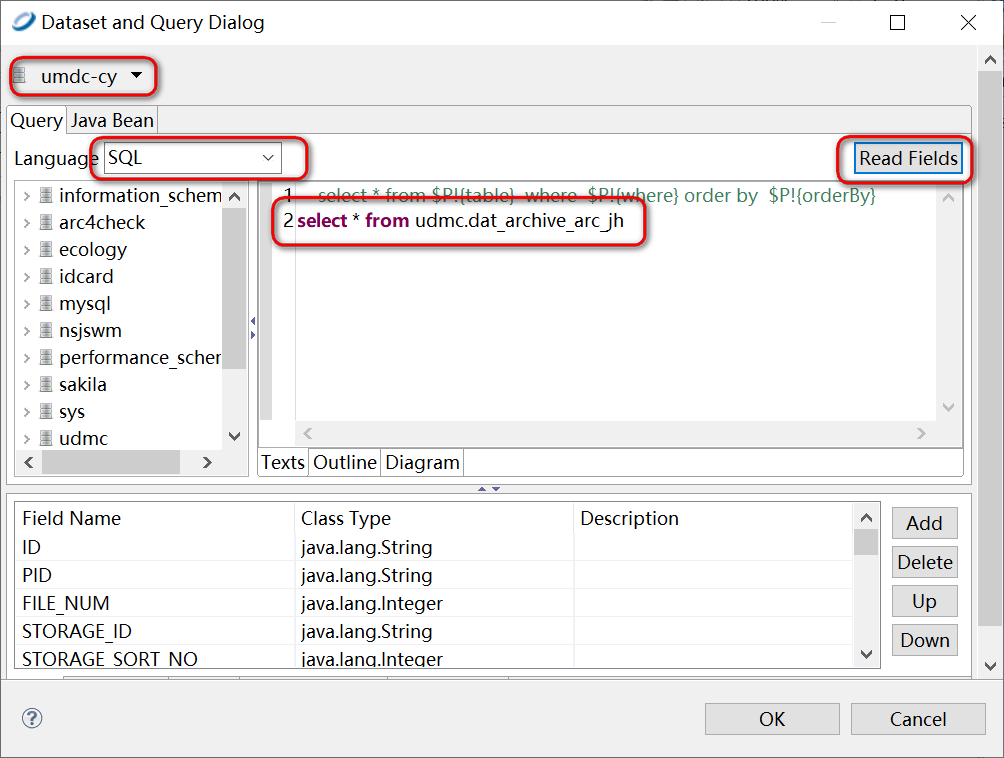
搞定之后，点击“Finish”，把我们的劳动成果保存下来。

#### 设置查询语句、读取字段列表

数据库连接设定完毕后，就可以通过SQL语句查询出数据，用以设置、填充报表了。

点击报表设计工作区中，工具栏上的【DataSet and Query editor dialog】按钮，如下图，系统会显示“DataSet and Query Dialog”界面。





“数据适配器”选择刚刚创建好的JDBC数据源；“Language”选择“SQL”；在SQL查询语句区中，输入SQL查询语句，点击右上角的【Read Fields】按钮，系统会执行查询，并将查询到的字段加载到界面下方的字段列表中。在后面的报表设计中，我们就可以直接使用这些字段了。

**注意：在“Fields”列表中，建议只保留报表中使用的字段，其他用不到的字段最好用“Delete”删掉，防止由于数据表变化，导致报表模板中的字段设置与数据表对应不上，导致报表报错。**

输入完毕后，点击“OK”按钮，系统即会把查询语句保存在报表模板中。

这里要说明一下SQL查询语句。此处可以在报表模板中设置“参数”，并在SQL语句中使用“$P!{参数名}”的方式进行引用。后续章节详细说明“参数的使用”。

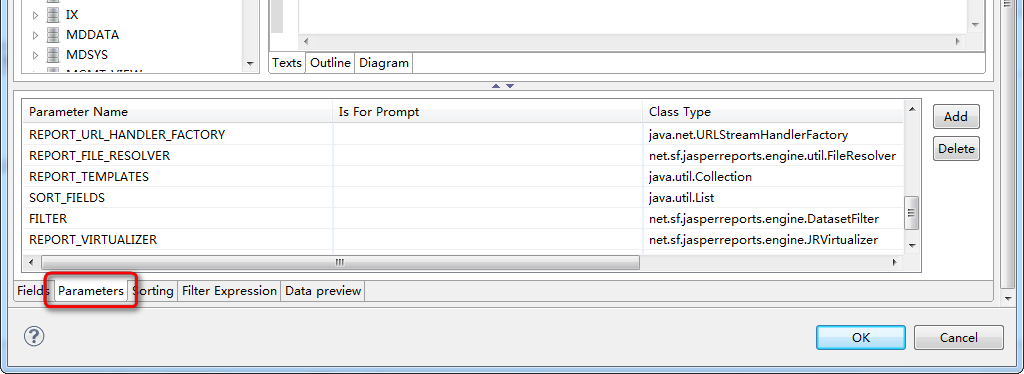
#### 参数的使用

##### 添加参数

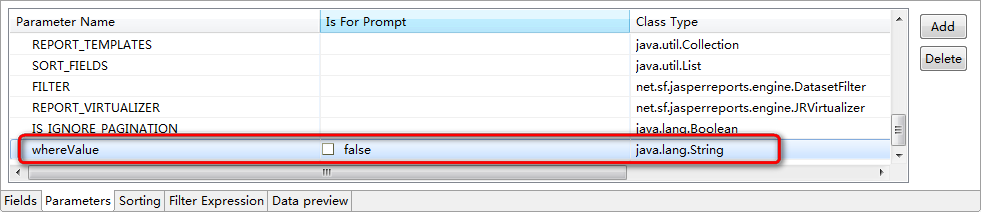
上面的例子中，报表模板使用的SQL查询语句是固定的，不能随着程序的运行而改变。

使用Studio制作报表模板，可以通过使用“参数”功能，接收程序传递过来的参数表，并将参数传递到报表模板中，从而实现根据程序运行的情况，动态组合SQL查询语句、动态生成报表数据。

在“查询设置”界面的下方，点击【Parameters】页，就可以切换到参数列表界面。

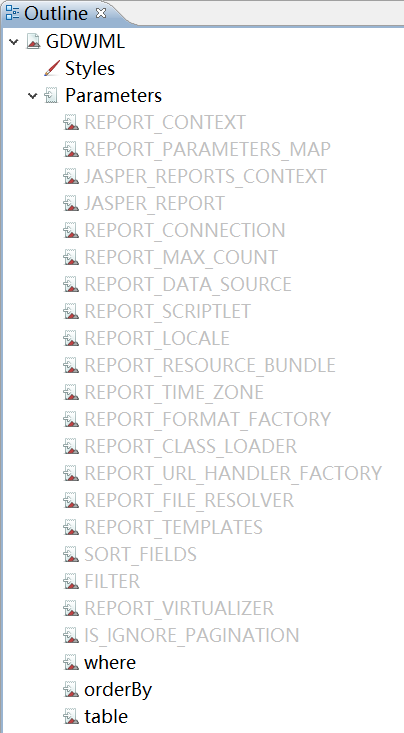


点击【Add】，就可以添加新的参数。



在“Parameter Name”中添加变量名称，在“Class Type”选择变量的数据类型。

也可以在主界面左侧的【Outline】页签中，点击“Parameters”节点，通过鼠标右键菜单维护。



##### 程序中传递参数

添加完毕的变量，既可以在变量列表中浏览到，可以在程序中通过Map传递过来。

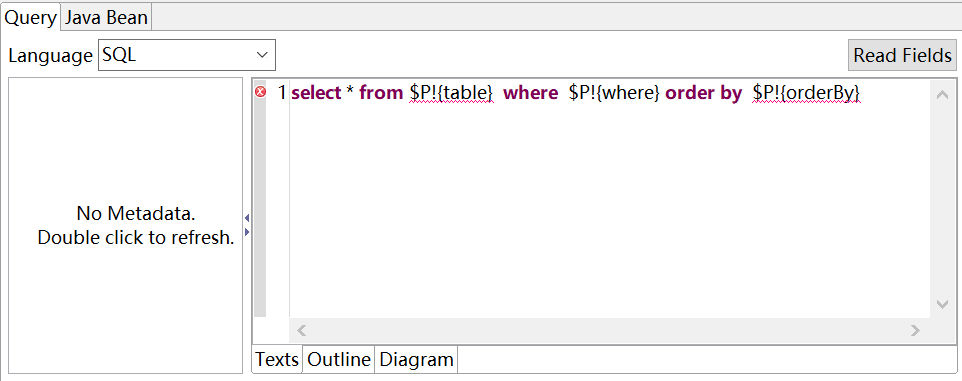
本例中共传递了3个参数：

1. table：数据表名。如果使用此参数，系统会自动将当前所选择的数据表的表名传递到此参数中，代替变量位置的字符串。
2. where：查询子句。如果使用此参数，系统会自动将当前结果集的查询子句传递到此参数中，代替变量位置的字符串。
3. orderBy：排序字段。如果使用此参数，系统会自动将当前激活的排序方案的排序字段设置，传递到此参数中，代替变量位置的字符串。

##### 报表模板接收参数

要想在报表模板中使用变量，需要在SQL查询语句中输入“$P!{变量名}”，如此格式进行调用。

最简单的办法是，直接在参数列表中选中需要加入的参数，然后拖拽到SQL语句中。系统自动添加的标签为：“$P{变量名}”，改为“$P!{变量名}”即可。

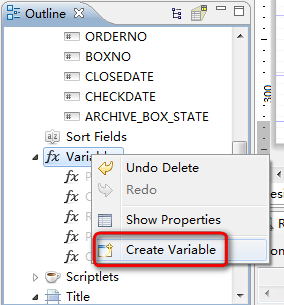


#### 自定义变量的使用

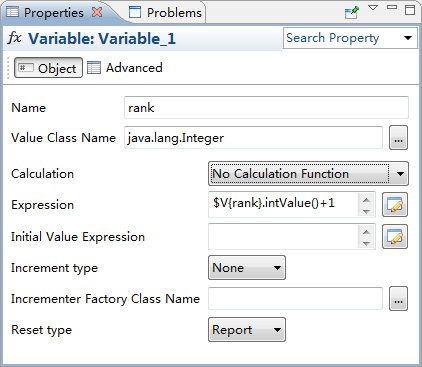
##### 顺序号

如果需要设置顺序号（计数器），可以按照以下方式设置。设置完毕后在字段域中引用此变量即可。

在【Outline】页中的“Variable”节点上点击鼠标右键，选择【Create Variable】，新增加一个变量。



在右侧的【Properties】页中，可以设置新增的变量的属性。



各属性解释如下：

Name：变量的名称改为“rank”。

Value Class Name：变量的数据类型。此处根据数据库的不同，字段类型的不同，选择不同的数据类型。例如，Oracle的数值型字段均定义为Number型，则此处需要选择“java.math.BigDecimal”；若为SQL Server或MySQL数据库，整型为Integer，则此处选择“java.lang.Integer”。如此类推。

Calculation：计算函数。此处选择“No Calculation Function”。

Expression：变量计算公式。可以在此处输入变量统计时计算的公式，可以使用系统中的变量、字段等资源，通过调用不同的方法，实现公式的设置。本例中，公式设置为：“$V{rank}.intValue()+1”

Initial Value Expression：变量重置时初始值。可以留空，系统可以自动根据字段类型赋予0或者空值。也可以手工赋予指定类型的数据。例如，本例中留空。

Increment type：增长类型。表示变量增长的计算范围。如果在分组报表中使用，此处选择“group”，即在分组范围内进行累加；如果在普通报表中使用，此处选择None。

Incrementer Factory Class：增长工厂类。一般用不到。

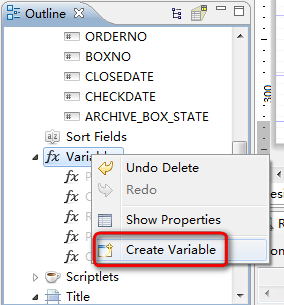
Reset type：重置类型。表示变量在什么时候做重置操作。如果是在分组报表中使用，此处选择Group，当分组变化时，即重新合计；如果在普通报表中使用，此处选择Report。

##### 小计

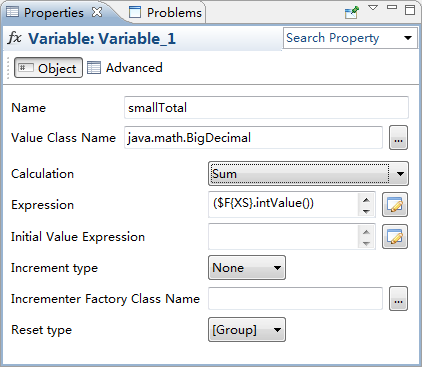
**小计需要与分组报表配合使用，一般情况下，小计行放在Group Footer区域中。**

如果需要设置合计，可以按照以下方式设置。设置完毕后在“Group Footer”区域中引用此变量即可。

在【Outline】页中的“Variable”节点上点击鼠标右键，选择【Create Variable】，新增加一个变量。



在右侧的【Properties】页中，可以设置新增的变量的属性。



各属性解释如下：

Name：变量的名称。

Variable Class Name：变量的数据类型。此处根据数据库的不同，字段类型的不同，选择不同的数据类型。例如，Oracle的数值型字段均定义为Number型，则此处需要选择“java.math.BigDecimal”；若为SQL Server或MySQL数据库，整型为Integer，则此处选择“java.lang.Integer”。如此类推。

Calculation：计算函数。此处选择求和，即Sum。

Expression：变量计算公式。可以在此处输入变量统计时计算的公式，可以使用系统中的变量、字段等资源，通过调用不同的方法，实现公式的设置。本例中，公式设置为“new java.math.BigDecimal($F{XS}.intValue())”，即累加XS字段的整型值。

Initial Value Expression：变量重置时初始值。不可以留空。例如，本例中使用“new java.math.BigDecimal(0)”赋予0。

Increment type：增长类型。表示变量增长的计算范围。此处选择“None”，即不继续增长。

Incrementer Factory Class：增长工厂类。一般用不到。

Reset type：重置类型。表示变量在什么时候做重置操作。此处选择“Group”，当分组变化时，即重新合计。

##### 合计

###### 表头合计

表头合计，即在Detail区之前需要读取合计值，此时需要使用SQL语句预先求出数值。

可以采用在Select语句中嵌套子语句实现，如下：

select

d.maintitle as maintitle,

(Select sum(s.cs) From SonData s Where s.pid=d.id) as t\_cs

From

FatherData d

说明：

FatherData是父表；SonData是子表，在这个Sql语句中，通过嵌套的Select语句，动态求出当子表的pid值等于父表id值时，子表cs字段的和，填充到父表记录列表中的t\_cs字段中。

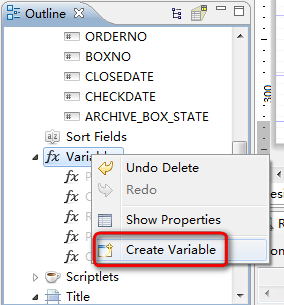
通过SQL语句取出合计值后，添加Field后，即可在模板中调用此值。

###### 表尾合计

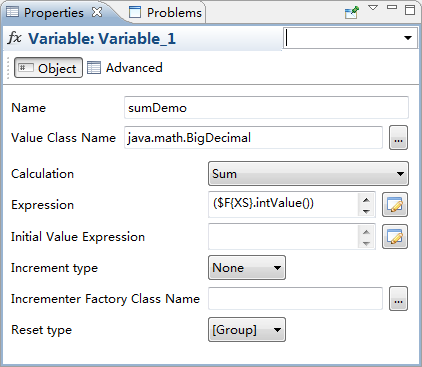
如果需要设置合计，可以按照以下方式设置。设置完毕后在“Summary”区域中引用此变量即可。

**注意：合计类变量只有在Summary区域中才有效！！**

在【Outline】页中的“Variable”节点上点击鼠标右键，选择【Create Variable】，新增加一个变量。



在右侧的【Properties】页中，可以设置新增的变量的属性。



各属性解释如下：

Name：变量的名称。

Variable Class Name：变量的数据类型。此处根据数据库的不同，字段类型的不同，选择不同的数据类型。例如，Oracle的数值型字段均定义为Number型，则此处需要选择“java.math.BigDecimal”；若为SQL Server或MySQL数据库，整型为Integer，则此处选择“java.lang.Integer”。如此类推。

Calculation：计算函数。此处选择求和，即Sum。

Expression：变量计算公式。可以在此处输入变量统计时计算的公式，可以使用系统中的变量、字段等资源，通过调用不同的方法，实现公式的设置。本例中，公式设置为

“new java.math.BigDecimal($V{sum\_cs}.intValue()+$F{mx\_cs}.intValue())”

Initial Value Expression：变量重置时初始值。不可以留空。例如，本例中使用“new java.math.BigDecimal(0)”赋予0。

Increment type：增长类型。表示变量增长的计算范围。此处选择“group”，即在分组范围内进行累加。

Incrementer Factory Class：增长工厂类。一般用不到。

Reset type：重置类型。表示变量在什么时候做重置操作。此处选择Group，当分组变化时，即重新合计。

上例中的“合计”，为每个分组之间的合计，也可以理解为“小计”。合计还有其他几种方式：

1. 累计合计，即总计。Reset type和Increment type均选择Report。
2. 本页合计，即合计当前页的值。Reset type选择Page，Increment type选择Report。

## 创建卡片式报表

卡片式报表，即为每页只输出一条记录。

处理方式非常简单：

* 只保留“detail”区域。
* “detail”区域的高度设置为与纸张高度相同（可自定义规格）。
* 将所有字段按照卡片的内容格式排布在工作区中。

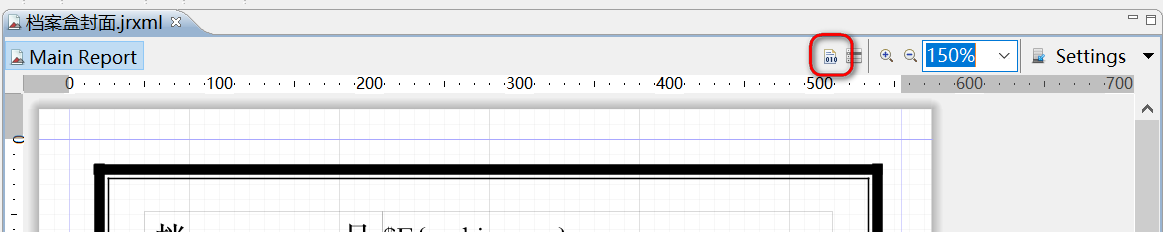
系统即会按照数据的条数，自动进行循环，每页生成一条数据。

## 编译、发布

报表编制完毕，想要在程序里使用，还需要【编译】。

我们之前操作的模板文件的扩展名是“jrxml”，是一个使用xml来记录模板设置的文件；“编译”后的文件扩展名是“jasper”，就可以使用JasperReport的java包加载、填充数据了。

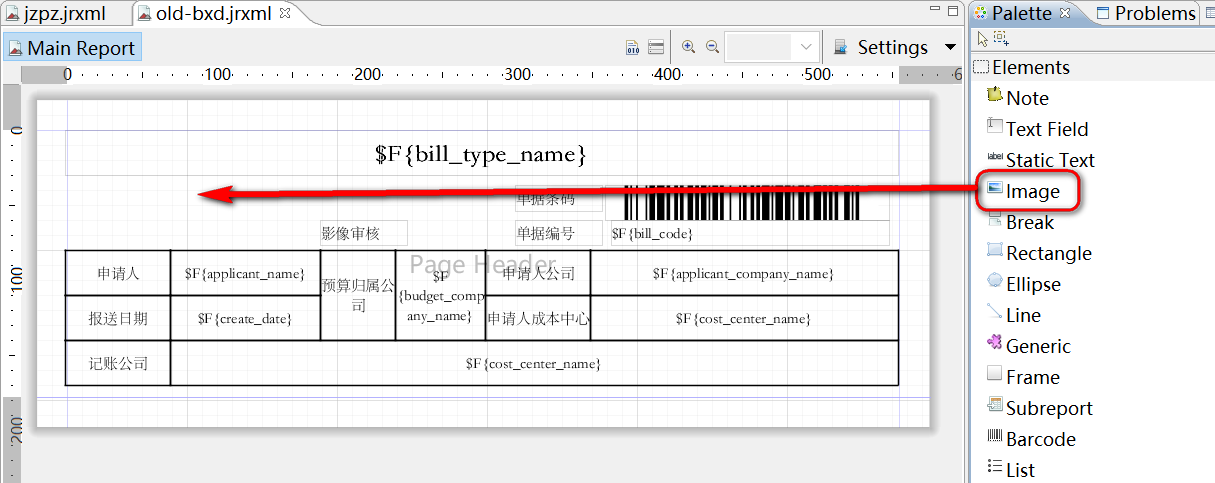
点击工具栏中的【编译】图标，系统即会自动对报表模板进行编译，如果编译过程有错误，会在下方的控制台中显示相关信息。



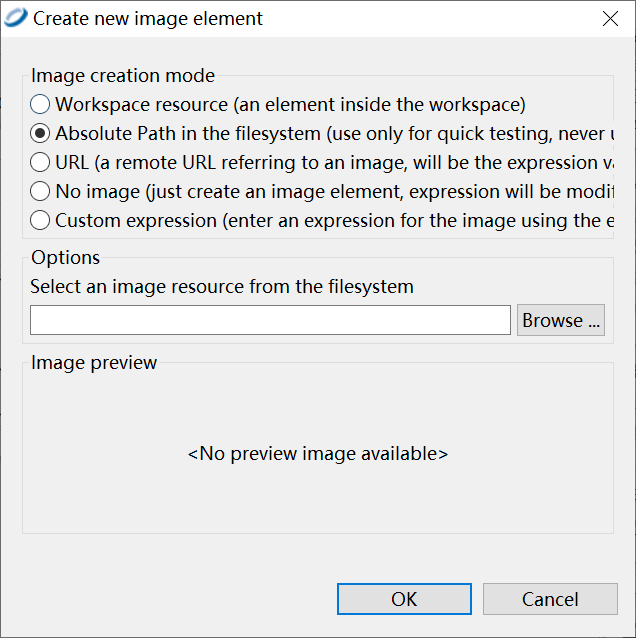
# 报表设计器进阶

## 加入图片

当报表中需要加入Logo等图片时，可以使用“Image”组件。在组件列表中，找到“Image”，拖拽到报表画布的适当位置。



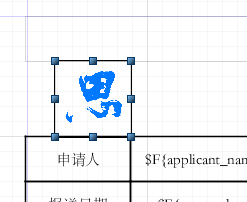
Image组件自动显示属性选择窗口，需要确认此图片组件的图像创建模式。



依次为：

* 工作区资源：在工作区中已经存在的元素，包括字段、变量、参数、资源文件等。
* 文件系统中的绝对路径：直接指定文件系统中的图片，用于快速测试。后续在实际发布时，可以用参数将路径传入，拼装完整的图片文件路径。
* URL：可以访问URL地址中的图片资源。
* 无图片：创建一个空的图片元素，后续通过条件控制图片显示内容。
* 定制条件：通过条件控制显示的图片。

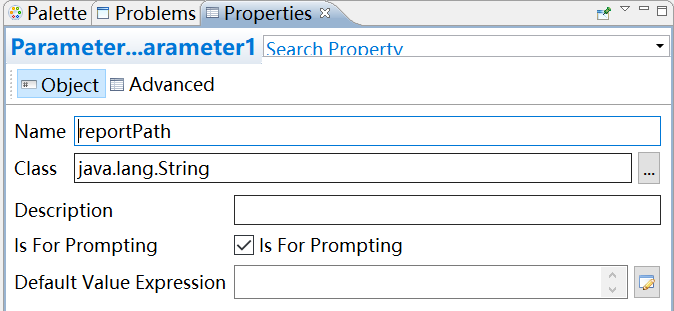
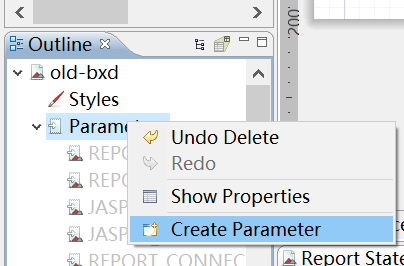
本例中，选用第二项“文件系统中的绝对路径”，直接指定图片即可。



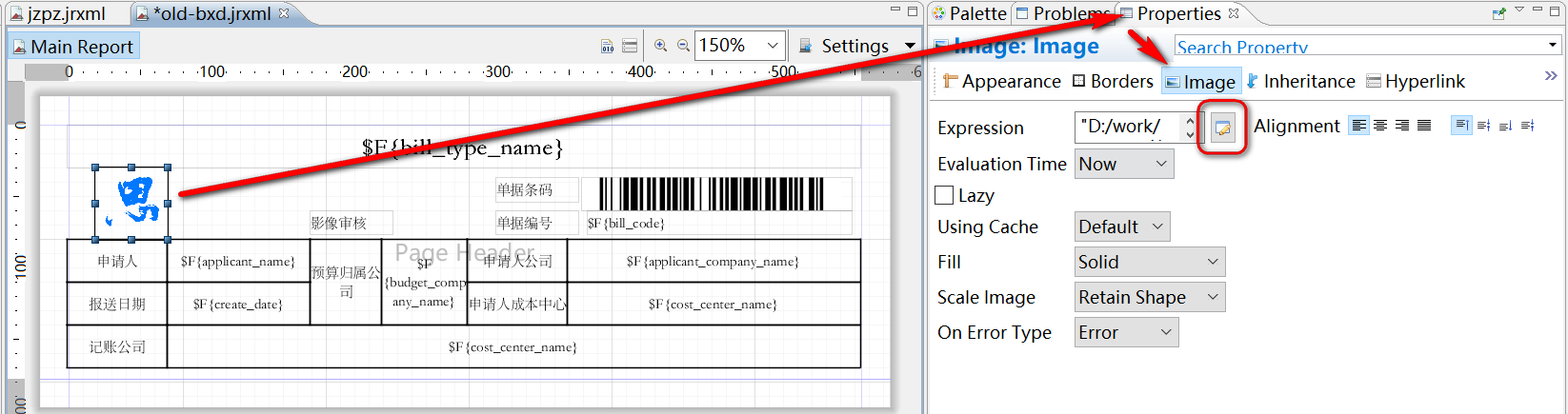
但是，此时的图片路径是固定，如果编译发布之后，图片不在现在设置的位置，就会导致无法显示图片。为了解决这个问题，可以通过在参数中加入“路径”，通过外部程序传入图片的相对路径，即可完成。

为了保证图片便于访问，我们可以将图片放置到与报表文件相同的文件夹中。

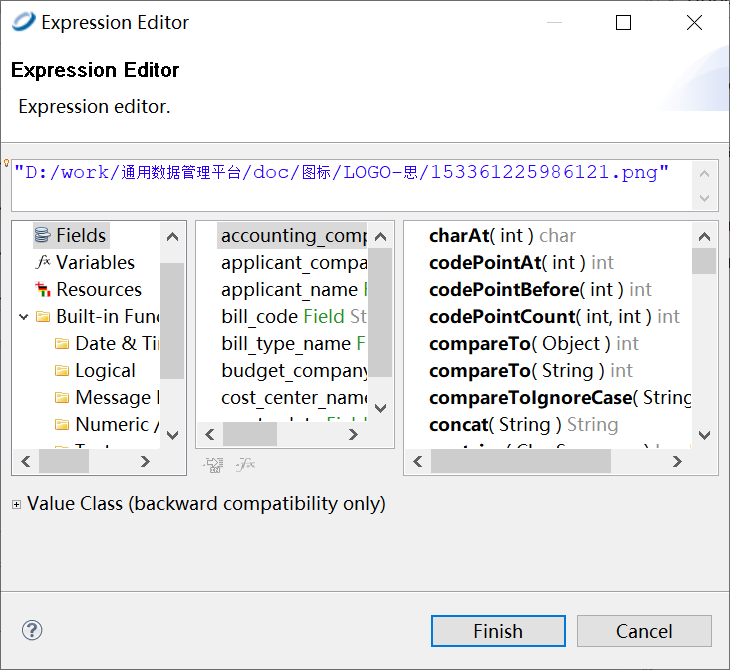
首先，在参数列表中加入“reportPath”参数。



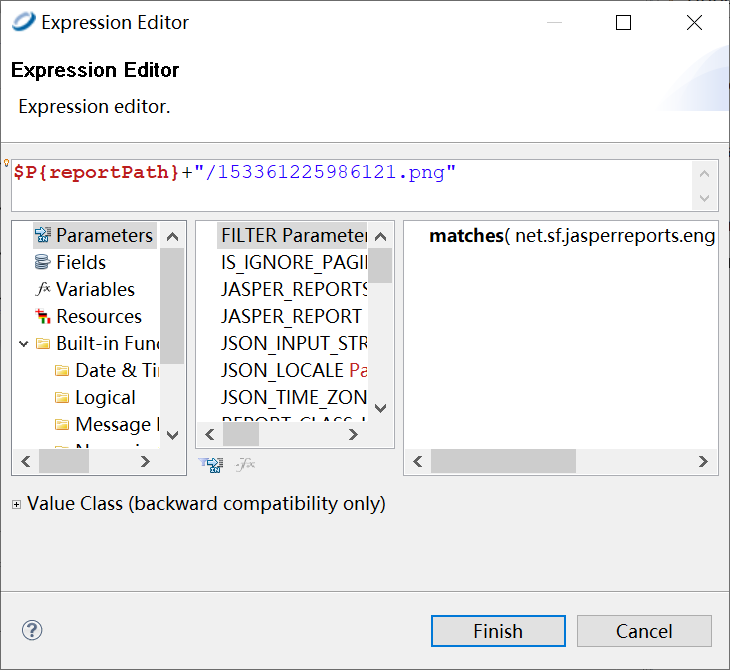
然后修改“Image”组件的文件路径设置。



在窗口中，能看到直接指定的文件的全路径和文件名。



左侧选择“Parameters”，选择列表最后的“reportPath”，双击加入到条件框中；添加“+”与后续文件名连接；原有的路径名删掉，只保留文件名。如下图。

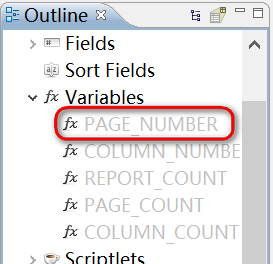


后续在通过程序调用此报表时，在参数列表中，将当前报表所在的路径名传入“reportPath”参数即可。

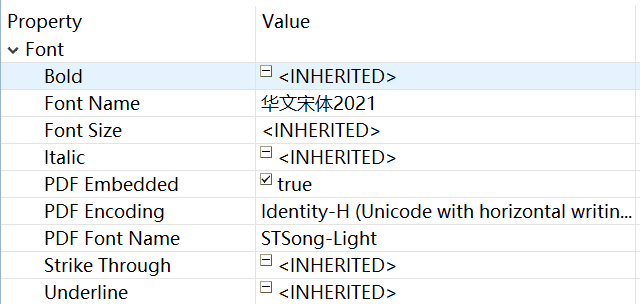
## 页码相关

### 添加页码

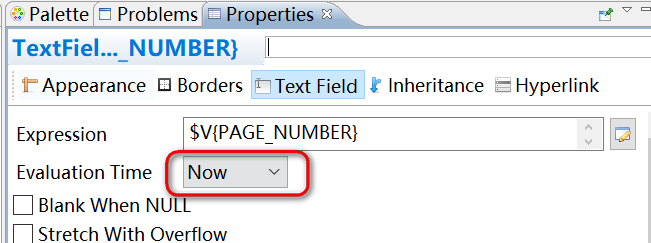
在制作报表时，经常需要在页脚或页头加入页码。一般情况下，只需显示当前页，可按照如下方式操作。在“变量”（Variables）列表中，找到“PAGE\_NUMBER”，拖拽到报表对应的区域中即可。



需要注意，此时的变量组件是个标准的文本框，如果需要打印PDF格式的报表，则需要设置PDF相关的3个参数。



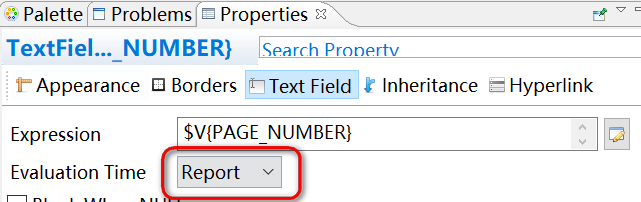
注意：此时变量的默认取值范围是“Now”，即当前页的页码。



如果我们需要在页面输出“第X页/共Y页”这种格式，则需要再添加一个“PAGE\_NUMBER”对象，如下：（此处，“第”、“页/共”、“页”均为静态文字对象）



第二个“PAGE\_NUMBER”对象的取值范围改为“Report”，即获取当前报表的总页数。



注意：获取总页数，只能通过此种方式得到。也就是说，无法通过“报表总页数”在组件中进行条件判断！！切记。

### 根据页码进行打印控制

在实际的报表需求中，有类似与“只有最末页最后一行打印合计，前面的页面均打印空格”的需求。

此时需要外部参数和输出条件互相配合。

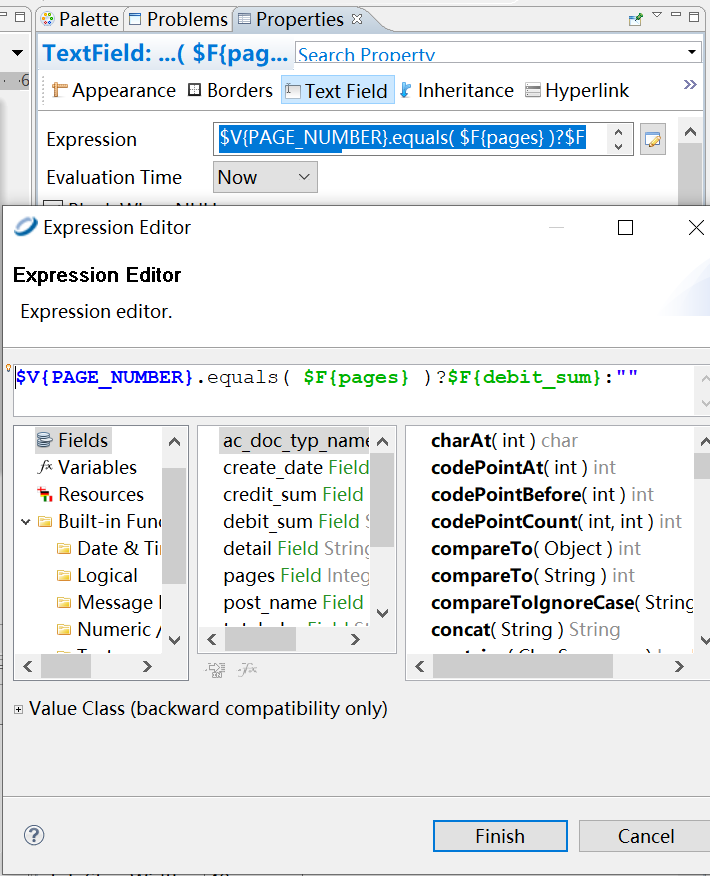
首先，需要在设置报表时，尽量固定每页打印的行数（此种模式下不要使用行高自动扩展，否则难以计算），然后根据记录数计算报表总页数（记录数/每页条数，如果有余数即加1页）。将此计算出来的总页数，做为参数传递到报表中。

报表增加“pages”参数，用于接收总页数。

其次，在报表中需要进行此类打印控制的文本框的“Expression”框中输入如下内容：

$V{PAGE\_NUMBER}.equals( $F{pages} )?$F{debit\_sum}:""

即，当前页码等于传入的最大页数时，打印此字段的值，否则打印空。



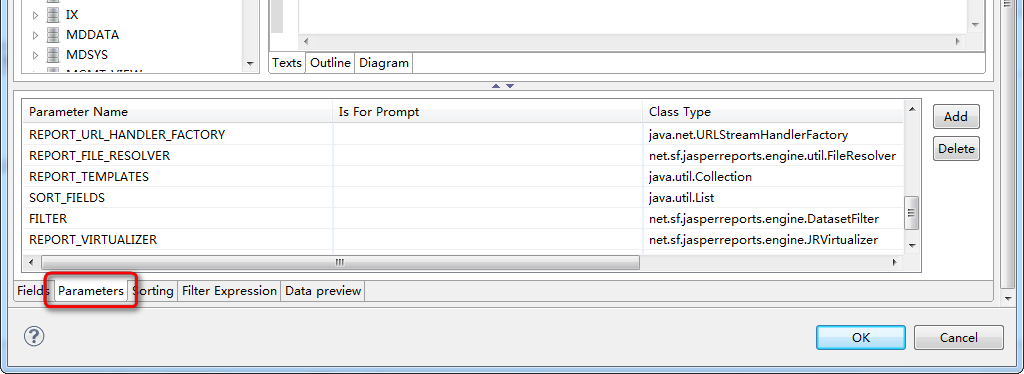
## 参数

### 添加参数

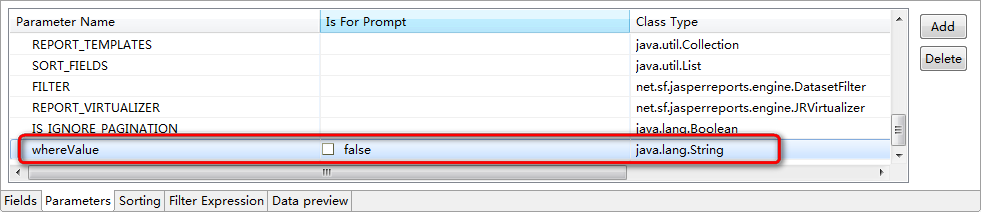
上面的例子中，报表模板使用的SQL查询语句是固定的，不能随着程序的运行而改变。

使用Studio制作报表模板，可以通过使用“参数”功能，接收程序传递过来的参数表，并将参数传递到报表模板中，从而实现根据程序运行的情况，动态组合SQL查询语句，动态生成报表数据。

在“查询设置”界面的下方，点击“Parameters”，就可以切换到参数列表界面。



点击“Add”，就可以添加新的参数。



在“Parameter Name”中添加变量名称，在“Class Type”选择变量的数据类型。

### 程序中传递参数

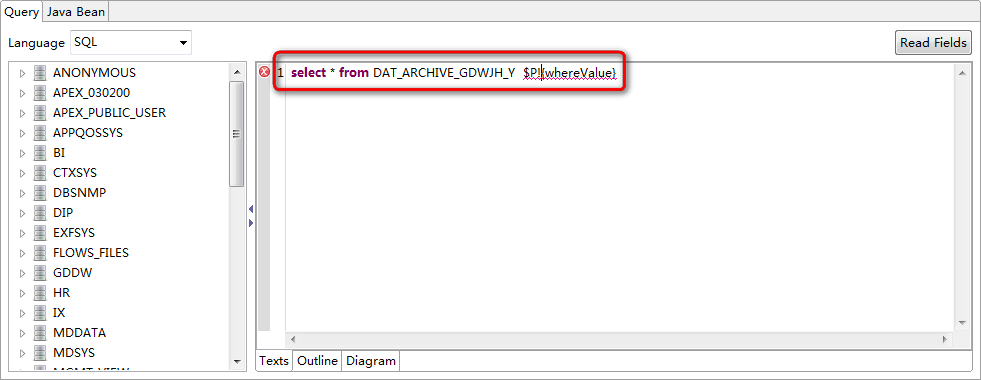
添加完毕的变量，既可以在变量列表中浏览到，可以在程序中通过Map传递过来。

程序中共传递了3个参数：

1. tableName：数据表名。如果使用此参数，系统会自动将当前所选择的数据表的表名传递到此参数中，代替变量位置的字符串。系统会自动将表名后面加入数据表缩写名“d”，写sql语句时需要注意。
2. whereValue：查询子句。如果使用此参数，系统会自动将当前结果集的查询子句传递到此参数中，代替变量位置的字符串。
3. orderBy：排序字段。如果使用此参数，系统会自动将当前激活的排序方案的排序字段设置，传递到此参数中，代替变量位置的字符串。
4. PtableName：父数据表名。如果使用此参数，系统会自动将当前数据表的父表名传递到此参数中，代替变量位置的字符串。系统会自动将表名后面加入数据表缩写名“p”，写sql语句时需要注意。
5. SstableName：唯一子数据表名。当前数据表如果有唯一的一个子表时，此参数有效。如果使用此参数，系统会自动将当前数据表的唯一的子表名传递到此参数中，代替变量位置的字符串。系统会自动将表名后面加入数据表缩写名“ss”，写sql语句时需要注意。

### 报表模板接收参数

要想在报表模板中使用变量，需要在SQL查询语句中输入“$P{变量名}”，如此格式进行调用。最简单的办法是，直接在参数列表中选中需要加入的参数，然后拖拽到SQL语句中。

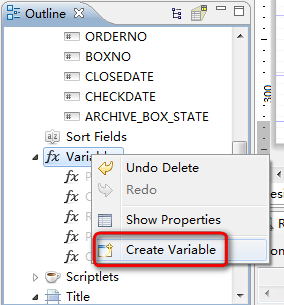


## 自定义变量

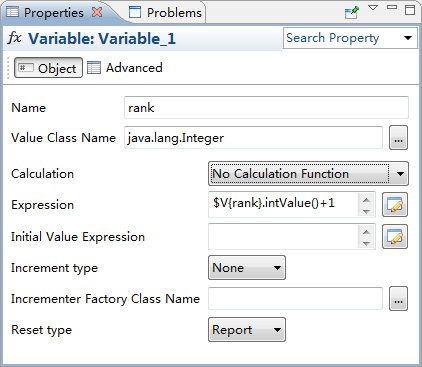
### 顺序号

如果需要设置顺序号（计数器），可以按照以下方式设置。设置完毕后在字段域中引用此变量即可。

在“Outline”区中的“Variable”节点上点击鼠标右键，选择“Create Variable”，新增加一个变量。



在右侧的“属性”区中，可以设置新增的变量的属性。



各属性解释如下：

Name：变量的名称改为“rank”。

Value Class Name：变量的数据类型。此处根据数据库的不同，字段类型的不同，选择不同的数据类型。例如，Oracle的数值型字段均定义为Number型，则此处需要选择“java.math.BigDecimal”；若为SQL Server或MySQL数据库，整型为Integer，则此处选择“java.lang.Integer”。如此类推。

Calculation：计算函数。此处选择“No Calculation Function”。

Expression：变量计算公式。可以在此处输入变量统计时计算的公式，可以使用系统中的变量、字段等资源，通过调用不同的方法，实现公式的设置。本例中，公式设置为

“$V{rank}.intValue()+1”

Initial Value Expression：变量重置时初始值。可以留空，系统可以自动根据字段类型赋予0或者空值。也可以手工赋予指定类型的数据。例如，本例中留空。

Increment type：增长类型。表示变量增长的计算范围。如果在分组报表中使用，此处选择“group”，即在分组范围内进行累加；如果在普通报表中使用，此处选择None。

Incrementer Factory Class：增长工厂类。一般用不到。

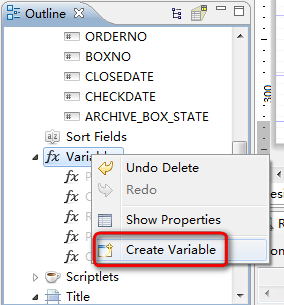
Reset type：重置类型。表示变量在什么时候做重置操作。如果是在分组报表中使用，此处选择Group，当分组变化时，即重新合计；如果在普通报表中使用，此处选择Report。

### 小计

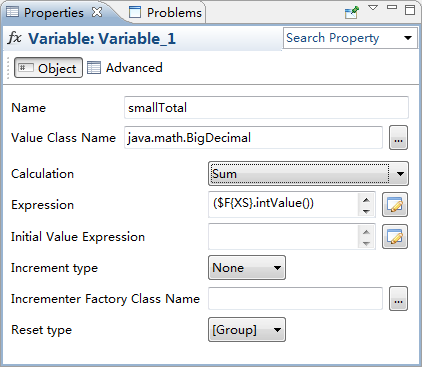
**小计需要与分组报表配合使用，一般情况下，小计行放在Group Footer区域中。**

如果需要设置合计，可以按照以下方式设置。设置完毕后在“Group Footer”域中引用此变量即可。

在“Outline”区中的“Variable”节点上点击鼠标右键，选择“Create Variable”，新增加一个变量。



在右侧的“属性”区中，可以设置新增的变量的属性。



各属性解释如下：

Name：变量的名称。

Variable Class Name：变量的数据类型。此处根据数据库的不同，字段类型的不同，选择不同的数据类型。例如，Oracle的数值型字段均定义为Number型，则此处需要选择“java.math.BigDecimal”；若为SQL Server或MySQL数据库，整型为Integer，则此处选择“java.lang.Integer”。如此类推。

Calculation：计算函数。此处选择求和，即Sum。

Expression：变量计算公式。可以在此处输入变量统计时计算的公式，可以使用系统中的变量、字段等资源，通过调用不同的方法，实现公式的设置。本例中，公式设置为“new java.math.BigDecimal($F{XS}.intValue())”，即累加XS字段的整型值。

Initial Value Expression：变量重置时初始值。不可以留空。例如，本例中使用“new java.math.BigDecimal(0)”赋予0。

Increment type：增长类型。表示变量增长的计算范围。此处选择“None”，即不继续增长。

Incrementer Factory Class：增长工厂类。一般用不到。

Reset type：重置类型。表示变量在什么时候做重置操作。此处选择“Group”，当分组变化时，即重新合计。

### 合计

#### 表头合计

表头合计，即在Detail区之前需要读取合计值，此时需要使用SQL语句预先求出数值。

可以采用在Select语句中嵌套子语句实现，如下：

select

d.maintitle as maintitle,

(Select sum(s.cs) From SonData s Where s.pid=d.id) as t\_cs

From

FatherData d

说明：

FatherData是父表；SonData是子表，在这个Sql语句中，通过嵌套的Select语句，动态求出当子表的pid值等于父表id值时，子表cs字段的和，填充到父表记录列表中的t\_cs字段中。

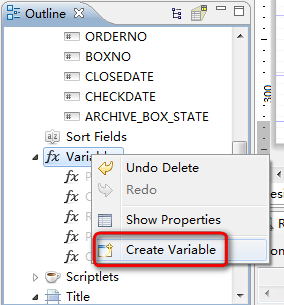
通过SQL语句取出合计值后，添加Field后，即可在模板中调用此值。

#### 表尾合计

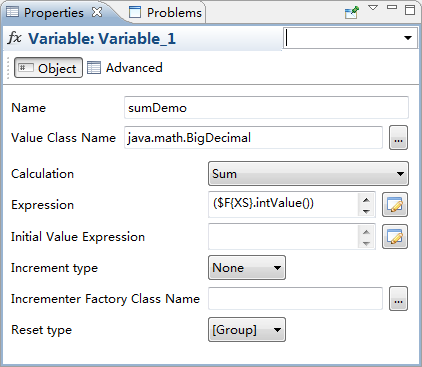
如果需要设置合计，可以按照以下方式设置。设置完毕后在“Summary”域中引用此变量即可。

**注意：合计类变量只有在Summary域中才有效！！**

在“Outline”区中的“Variable”节点上点击鼠标右键，选择“Create Variable”，新增加一个变量。



在右侧的“属性”区中，可以设置新增的变量的属性。



各属性解释如下：

Name：变量的名称。

Variable Class Name：变量的数据类型。此处根据数据库的不同，字段类型的不同，选择不同的数据类型。例如，Oracle的数值型字段均定义为Number型，则此处需要选择“java.math.BigDecimal”；若为SQL Server或MySQL数据库，整型为Integer，则此处选择“java.lang.Integer”。如此类推。

Calculation：计算函数。此处选择求和，即Sum。

Expression：变量计算公式。可以在此处输入变量统计时计算的公式，可以使用系统中的变量、字段等资源，通过调用不同的方法，实现公式的设置。本例中，公式设置为

“new java.math.BigDecimal($V{sum\_cs}.intValue()+$F{mx\_cs}.intValue())”

Initial Value Expression：变量重置时初始值。不可以留空。例如，本例中使用“new java.math.BigDecimal(0)”赋予0。

Increment type：增长类型。表示变量增长的计算范围。此处选择“group”，即在分组范围内进行累加。

Incrementer Factory Class：增长工厂类。一般用不到。

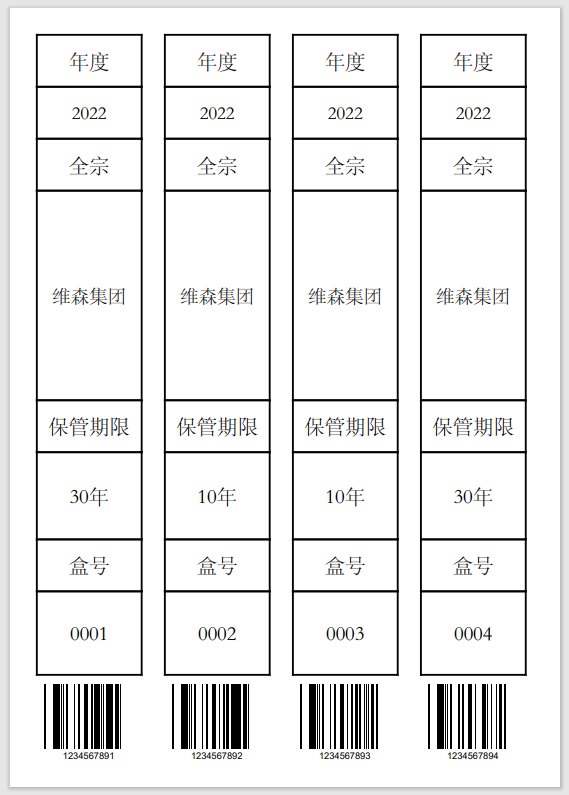
Reset type：重置类型。表示变量在什么时候做重置操作。此处选择Group，当分组变化时，即重新合计。

上例中的“合计”，为每个分组之间的合计，也可以理解为“小计”。合计还有其他几种方式：

1. 累计合计，即总计。Reset type和Increment type均选择Report。
2. 本页合计，即合计当前页的值。Reset type选择Page，Increment type选择Report。

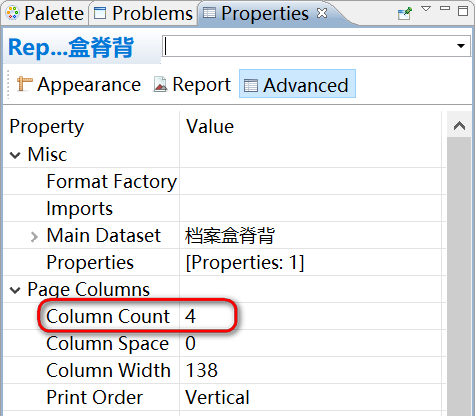
## 分栏报表（横向循环）

大多数报表都是自上向下循环输出的，例如“列表式”报表；但是也有部分场景需要报表横向循环输出，比如“盒脊背”就是每条竖向打印，然后横向输出充满整张报表。如下图。

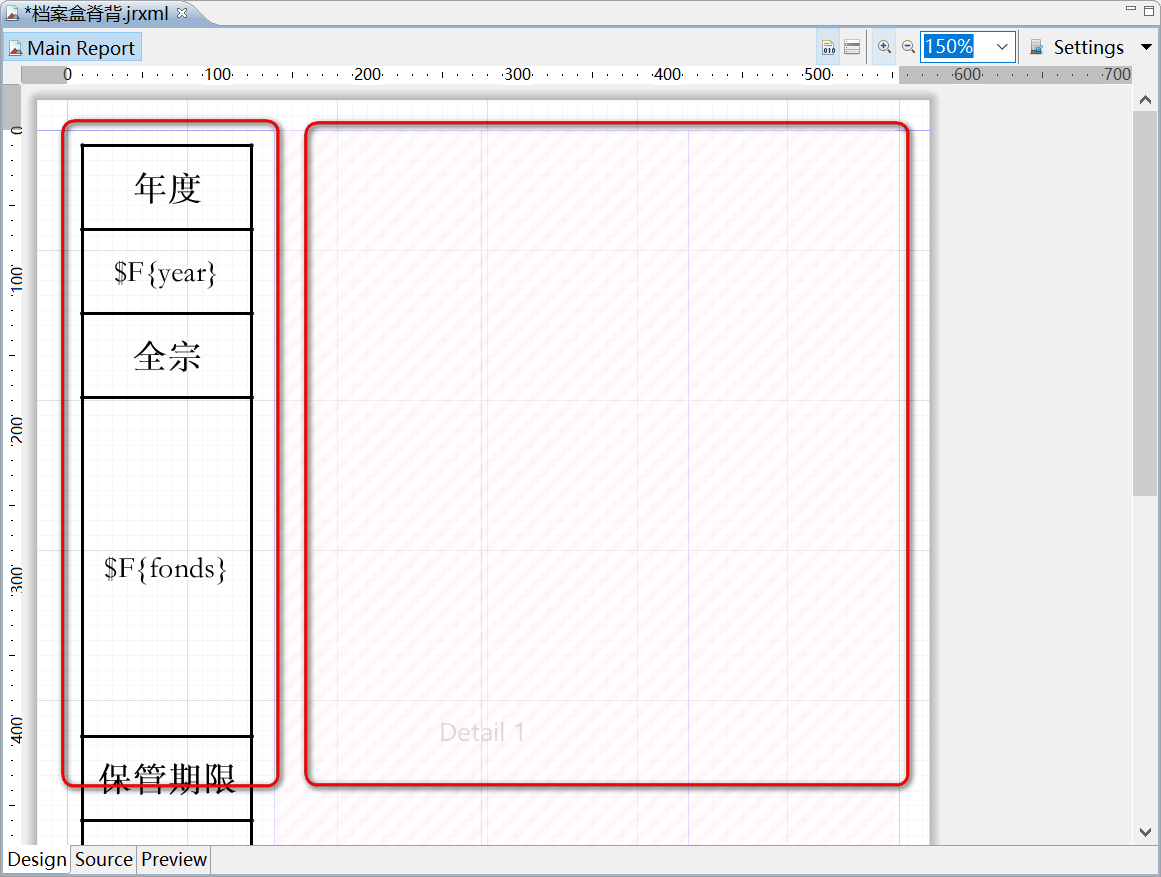


JasperReport提供了“报表分栏”的功能，可以实现此类需求。

首先，点击报表边缘的空白部分，使之变为“报表选中”模式，右侧【Properties】页中显示的是当前报表的属性。



点击【Advanced】页，修改【Page Columns】-【Column Count】。默认为1，即1栏；本例改为4，即分为4栏。效果如下。



左侧空白区域即是当前需要加入组件的编辑区；右侧有斜条纹的区域为循环区，无需放置组件。

参照前述章节的说明编制报表，填充数据进行预览，效果如下。

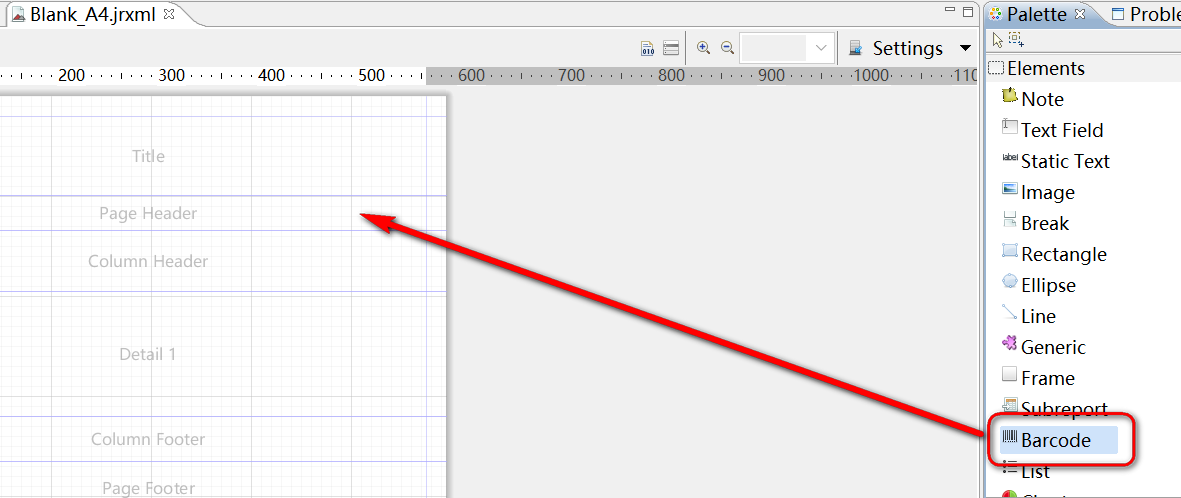


## 加入条码/二维码

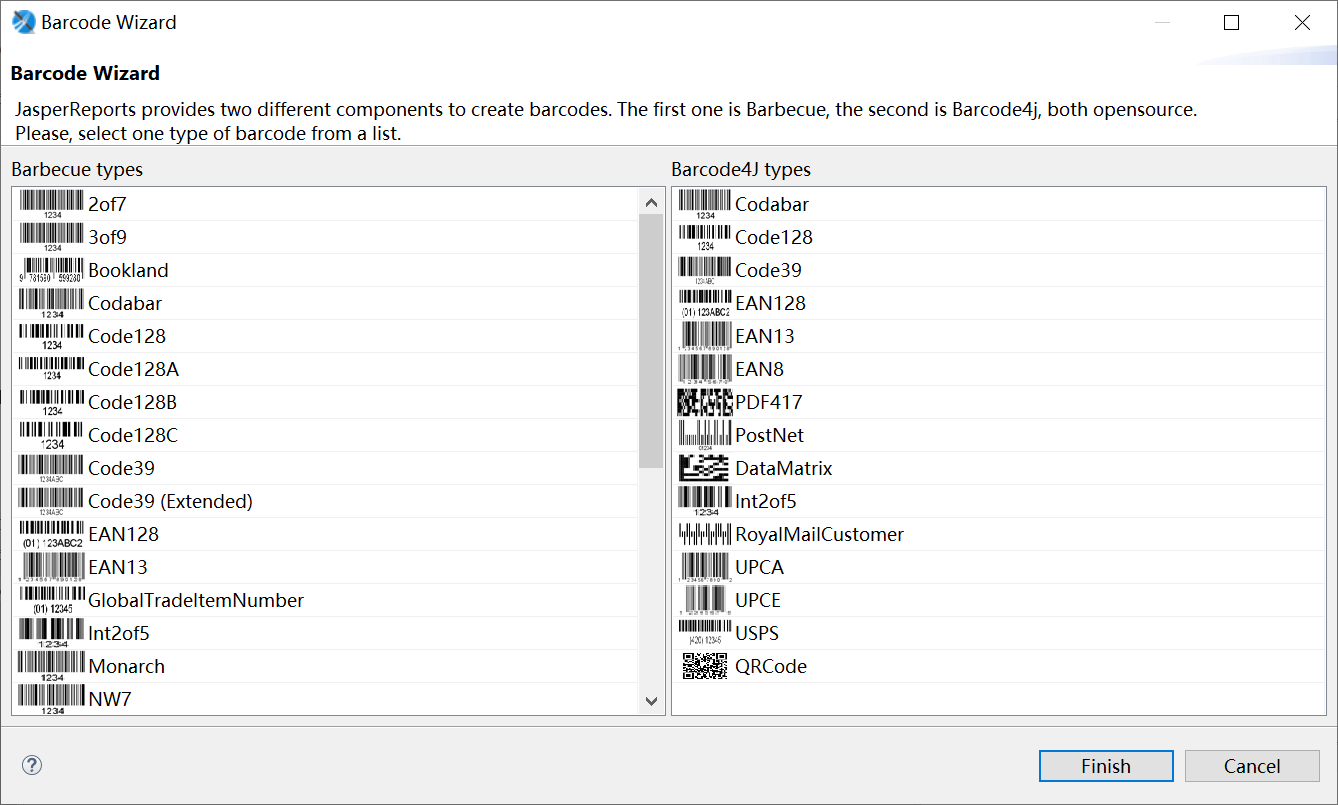
在打印报表的时候，经常会需要在报表中加入条码或二维码，方便打印为纸质报表后，通过扫码枪、App等直接进行扫码查询。

JasperReport使用Barbecue或Barcode4j组件生成条码或二维码，开发者可以选择任意一种集成到自己的项目中，然后在设置报表模板的时候，选择对应的包即可。

在编制报表的时候，可以在组件列表中找到【Barcode】组件，然后拖拽到画布中即可。一般条码/二维码都放置在报表顶部，所以可以拖动到【Page Header】区域中。



放置之后，系统自动弹出如下界面，可以选择此条码使用的工具包和编码格式。



**特别说明：如果要用“二维码”，就只能选择“Barcode4j”这个包，只有它支持“QRCode”！**

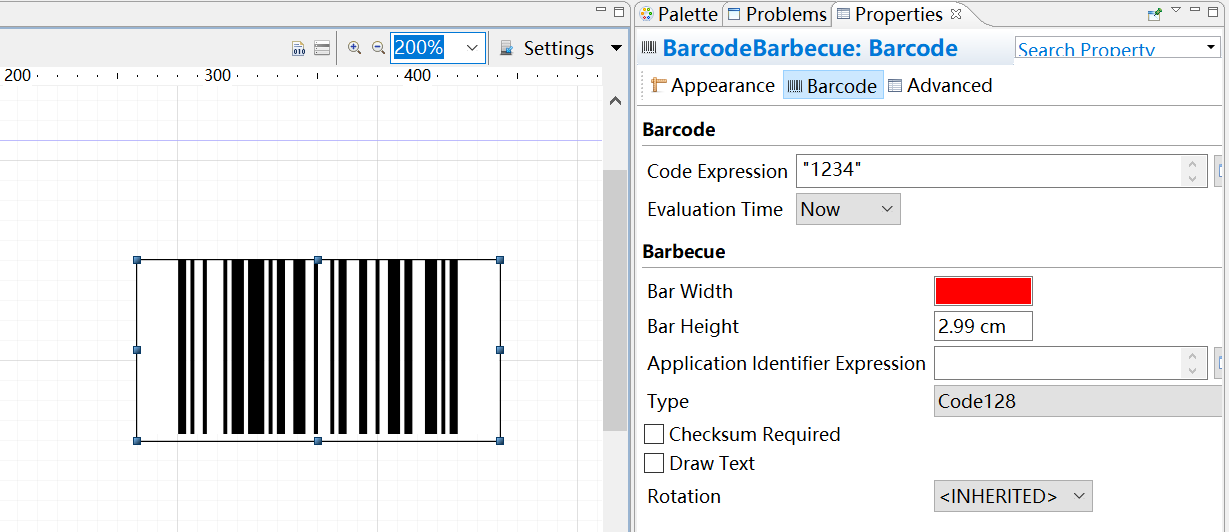
选择好工具和编码之后，点击【Finish】按钮，即可在画布上看到这个条码组件了。

不过这个组件比较丑，需要调整一下参数。

点击条码，在右侧选择【Properties】-【Barcode】页，可以通过【Code Expression】设置条码的取值字段；调整【Bar Height】调整条码的输出高度；修改【Type】可以切换编码格式。

注意：这里面的【Draw Text】启用后，会在Studio画布中显示编码对应的文字，但是通过程序生成的PDF中无法显示文字……这东西也没有PDF相关设置项……不知道是啥问题。

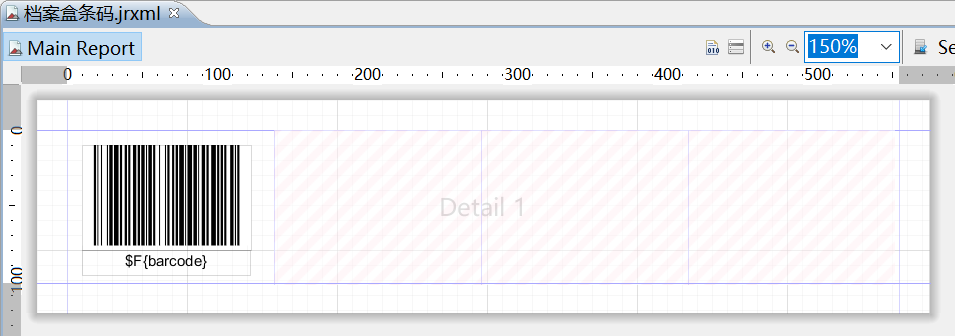
我提供的解决方案：在条码下面再放一个“字段组件”，绑定到与条码相同的字段上。



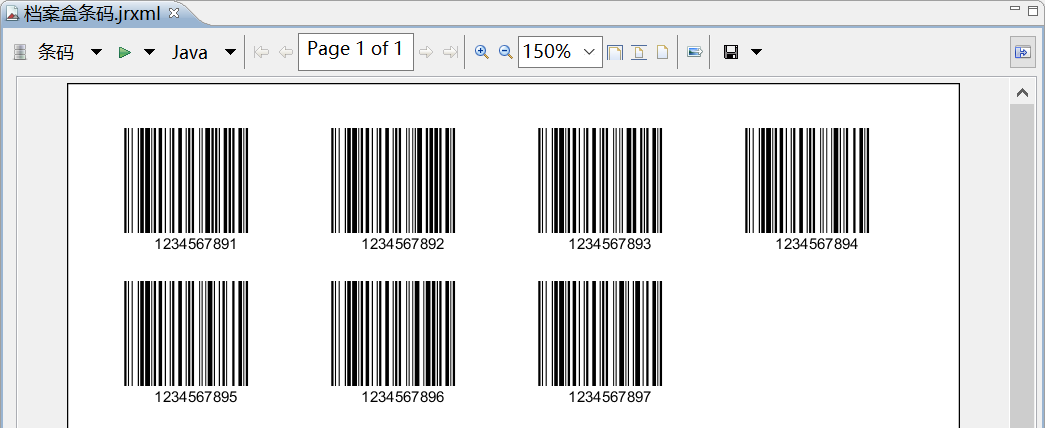
**特殊场景：生成一整页条码/二维码，可以打印在不干胶贴纸上，降低粘贴条码的成本。**

结合上一章节的“分栏报表”可以很容易的实现这个需求。

首先，报表只保留“Detail”区域；其次，设置报表栏数，可以根据条码尺寸计算；最后，在“Detail”区域的左侧放置条码组件和字段组件，调整字段绑定和显示设置。



通过预览，可以看到实现效果如下。



## 分组报表

当数据分为两层表时，经常需要批量打印子表的数据。打印时，常常需要按照父表的外键或关联值进行自动分组，即每一条父表记录所属的子表记录打印到一组报表中，每组报表都单独计数及计算页数。

在应用中，可以通过选择需要打印的父表记录，将父表记录的ID传入，由报表自动进行分组。

示例：

1. 设置报表查询语句，输入如下SQL语句：

select

wjh.filingcode as filingcode,

wj.note as note

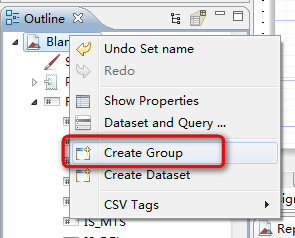
from DAT\_ARCHIVE\_GDWJ wj left join DAT\_ARCHIVE\_GDWJH wjh

on wj.pid=wjh.id order by wj.pid

此处一定要用分组的字段进行排序，否则生成的报表是错误的。

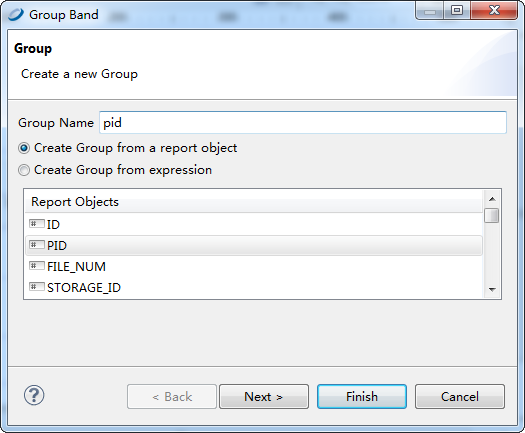
1. 新建报表群组

选中报表名称点击右键，选择菜单中的“Create Group”。



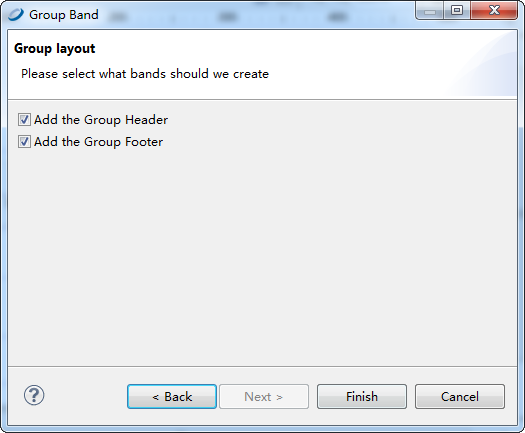
系统显示分组创建向导界面。

第一个界面中，需要设置分组的名称、分组字段。也可以设置按照指定的函数、方法处理后进行分组。



本例将“Group name”设置为“pid”，按照字段“pid”进行分组。设置完毕，点击“Next”。

系统显示细节设置界面。此处可以设置是否加入“group header”和“group footer”区。建议保持默认选中，加入这两个区域，这样可以控制在每组报表的结尾，打印相应的信息，例如统计信息或者父表所属的字段内容等。



设置完毕，点击“Finish”即可。

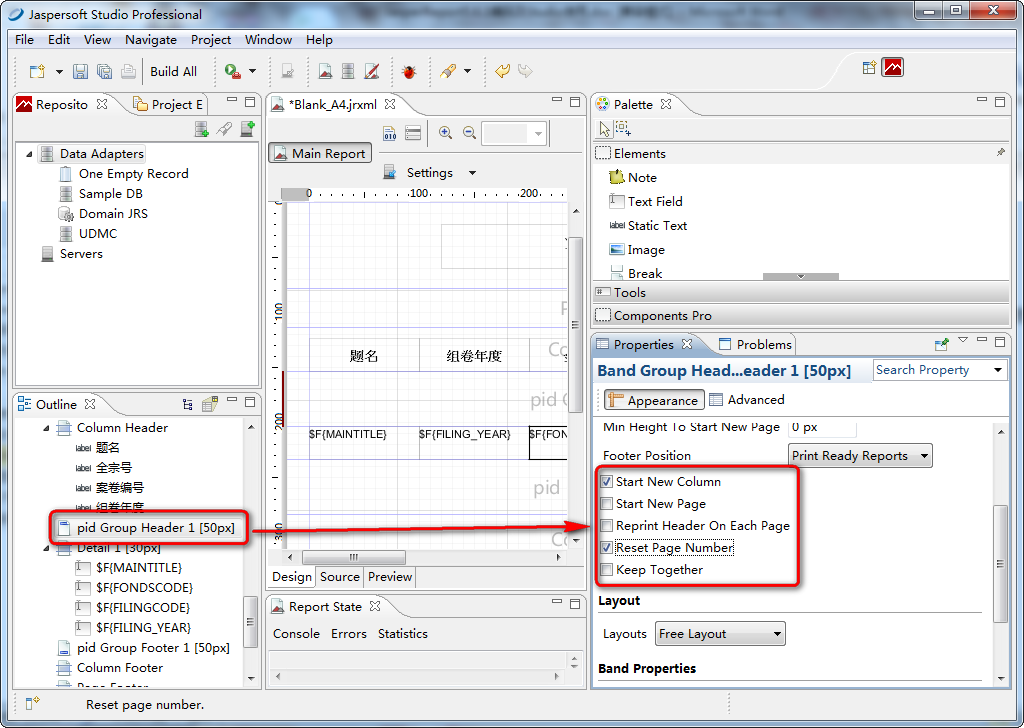
1. 放置报表数据

将需要作为表头打印的内容拖入pid Group Header1栏，将字段拖入detail栏，将每个分组结尾需要打印的内容放入pid group footer1栏，将页脚需要打印的内容放入Page Footer栏，如下图。



1. 设置分组分页打印

选择分组表头，在右侧的属性栏中可以设置分页打印的选项，如下图。



“Start on a new page”，即在新页面打印分组头和分组尾。

“Reset page number”，即新页面中重置页码。

## 图形报表

1. 设置报表查询。例如，在查询设置界面中输入如下SQL语句。

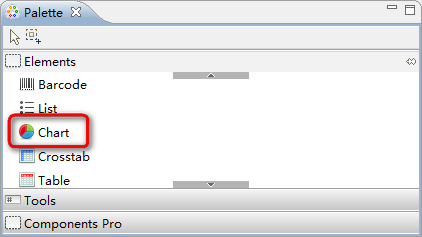
Select t2.deptName,count(t1.empId) as empCount

From employee t1

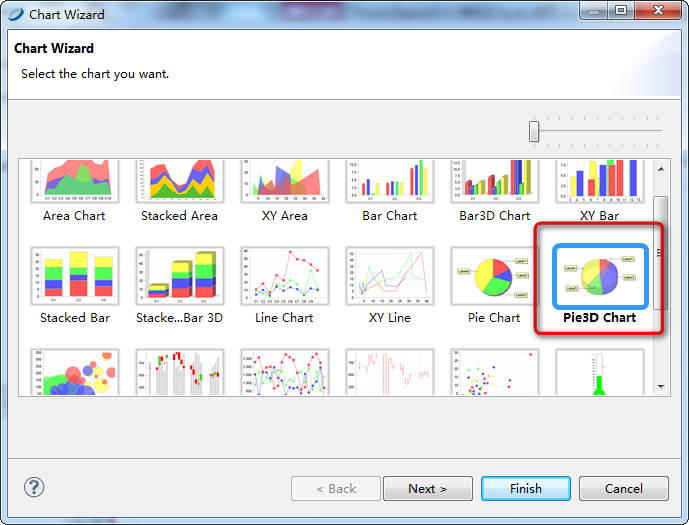
Left join department t2 on t1.deptId=t2.deptId

Group by t2.deptName

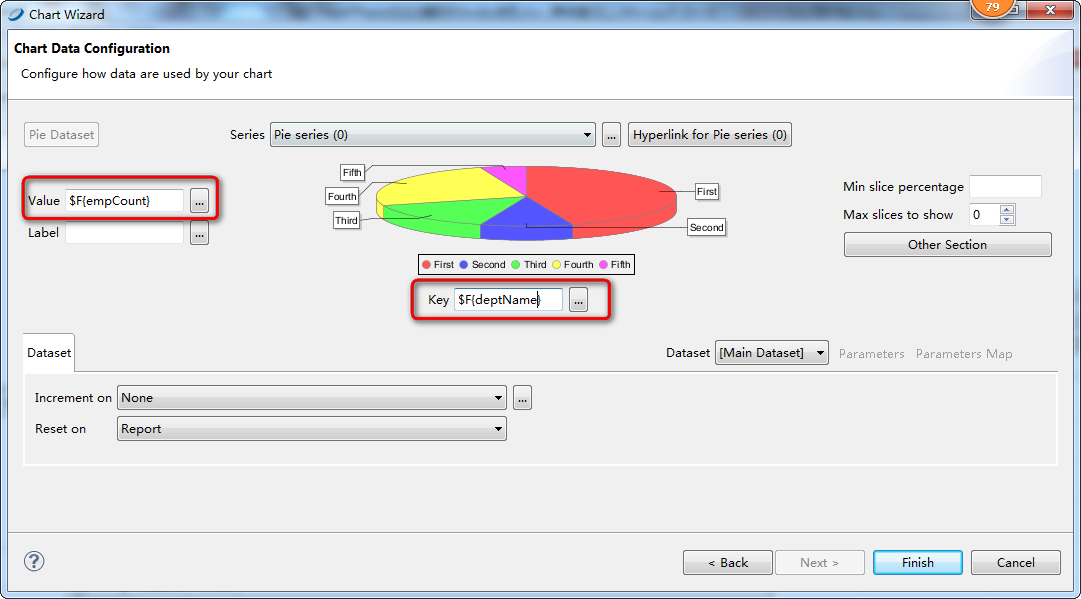
1. 删除一些报表栏。将columnHeader.detail.colmnFooter等栏高度设置为0。
2. 增加Summary栏的高度，设为400。
3. 插入饼图，放置在Summary栏。选择如图所示的按钮，插入图形报表。



选择三维饼图。



1. 点击“Next”，可设置饼图的参数。



1. 查看结果。



## 子报表

### 制作父报表

首先制作父报表，就是调用子报表的一个基础报表。

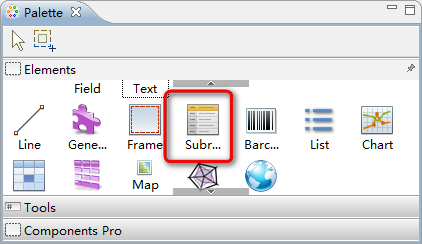
如果父报表中需要显示数据，在SQL编辑器中输入SQL语句，查询出需要的字段，将字段加入到报表中即可。

如果父报表不需要显示任何数据，只是作为子报表的载体，那么也需要在SQL编辑器中输入语句，并且需要能查询出记录，系统才会在生成报表时显示父报表，否则，系统会显示一个空白报表。

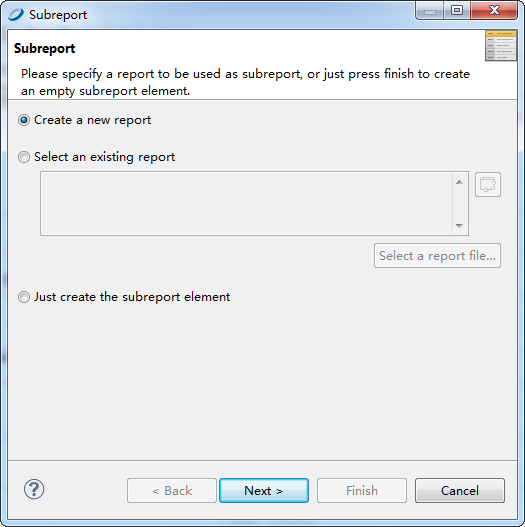
本系统中，建议输入“select id from sys\_user where user\_name='admin'”，只查询出一条记录即可使报表正常生成。

### 制作子报表

点击组件面板上的“Subreport”按钮，拖动到报表工作区上。

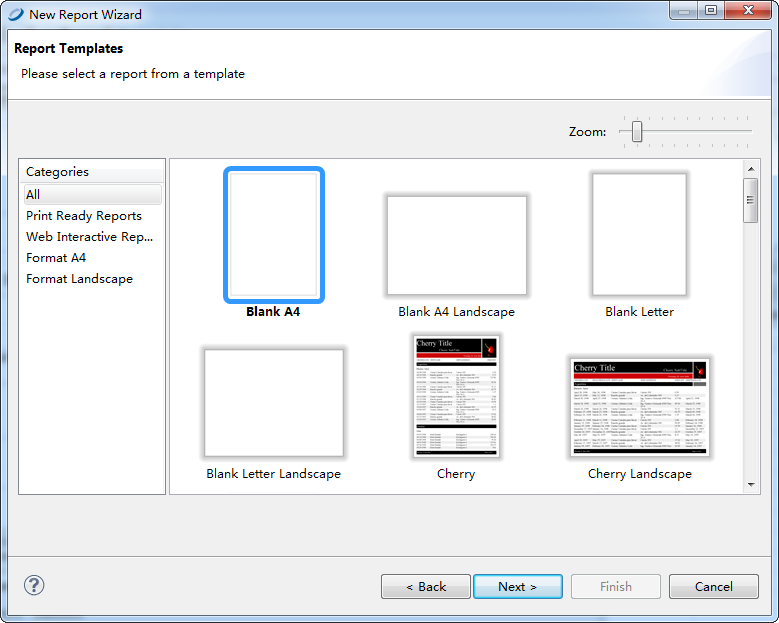


系统会自动弹出子报表选择窗口。可以选择创建一个新报表，还是使用一个已有的报表作为子报表。

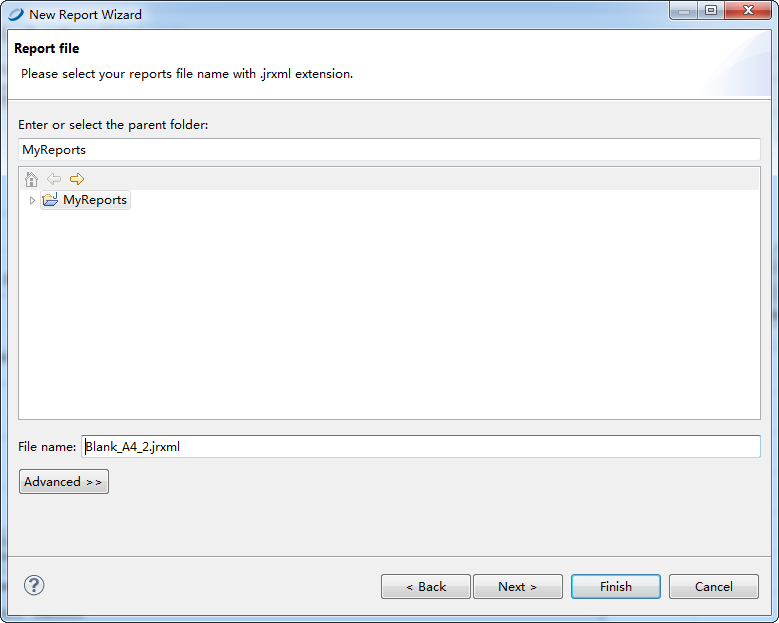


选择“Create a new report”，可以立即制作新的子报表；如果选择“Select an existing report”，则可以调用已经有的报表作为子报表；如果选择“Just create the subreport element”，系统会生成一个子报表区，可以在之后挂接需要的子报表。

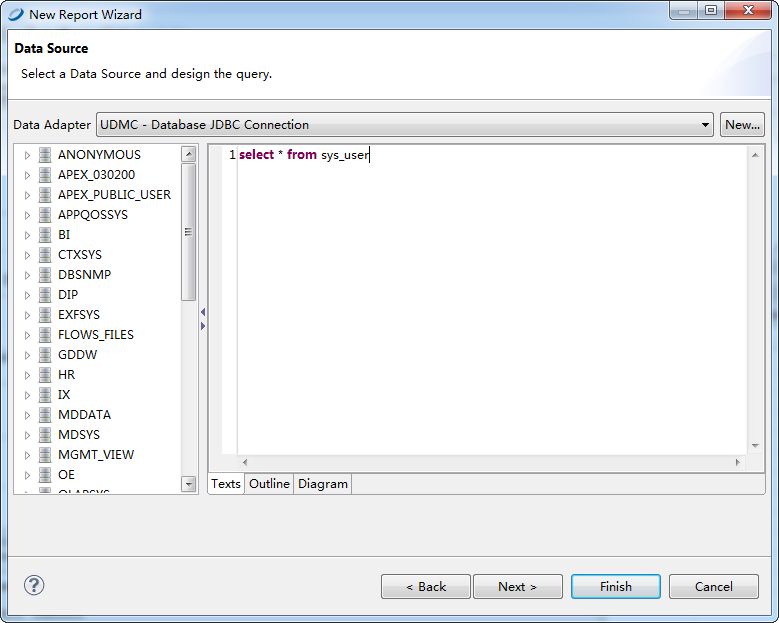
选择“Create a new report”的，点击 “Next”会出现“选择纸型”界面。本例中选择纵向的A4纸，即“Blank A4”。



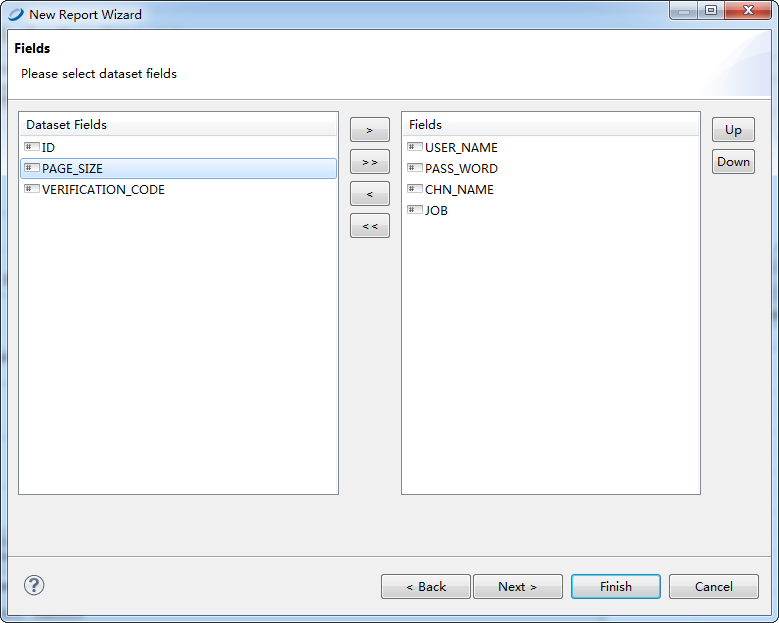
选择完毕，点击“Next”，系统显示文件存储设置页面，在此处设置子报表文件存放在何处。



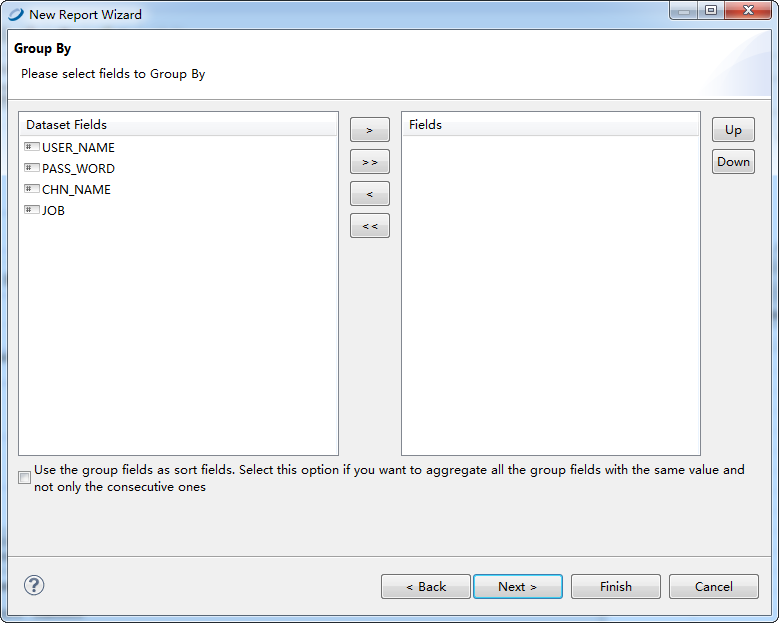
选择完毕，点击“Next”，系统显示“Data Source”界面。需要在此处选择数据连接，并输入SQL语句，来控制子报表数据的查询范围。



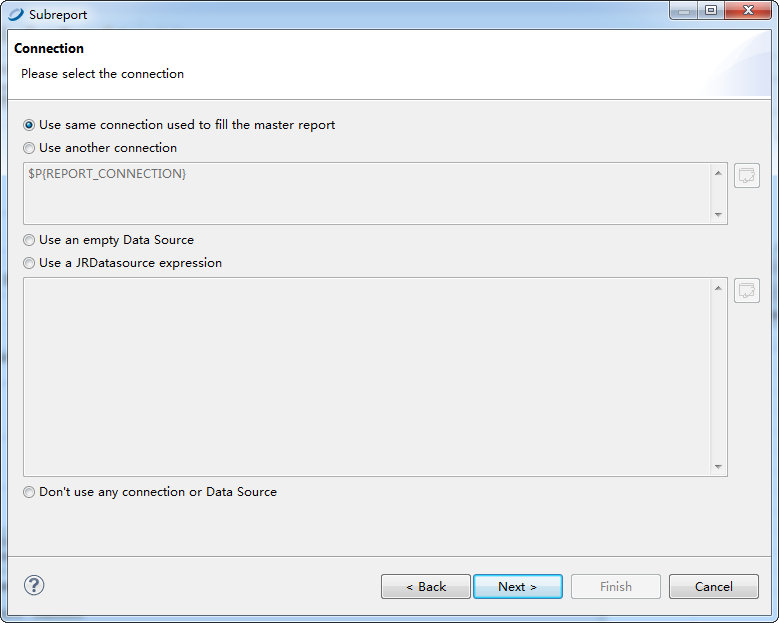
点击“Next”按钮，系统显示“Fields”界面。可以在此处选择子报表中显示的字段。



点击“Next”按钮，系统显示“Group By”界面。可以在此处设置子表的分组字段。

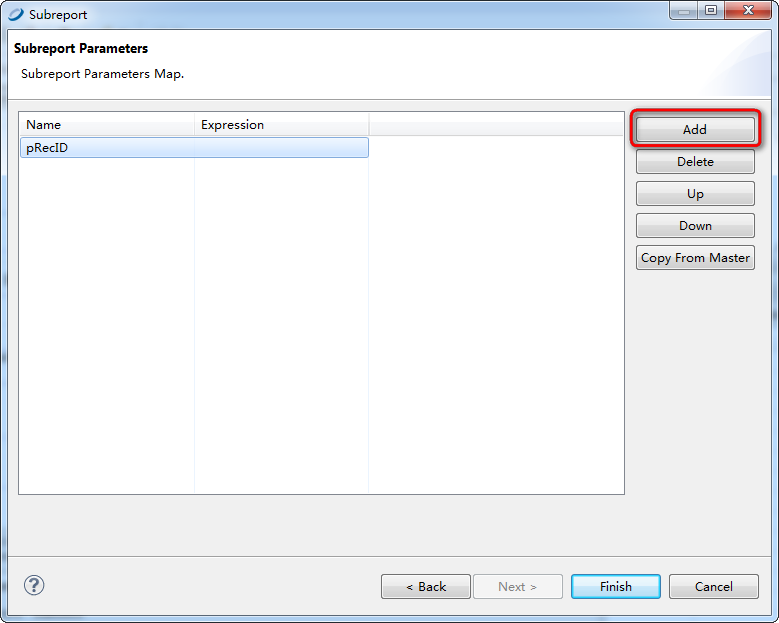


点击“Next”按钮，系统显示“Connection”界面。默认使用与父报表相同的数据连接，也可以选择使用其他指定的数据连接。



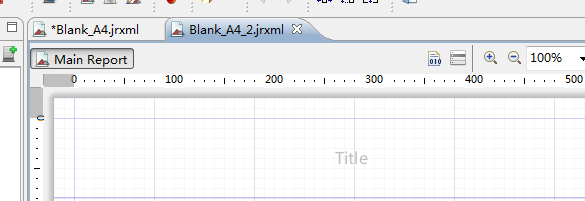
点击“Next”按钮，系统显示“Subreport Parameters”界面，在此处可以设置从父表需要传入子表的变量，一般都是关联字段的值。

子报表如果想与父报表联动工作，需要通过参数传递，将父报表中的值传递给子报表。



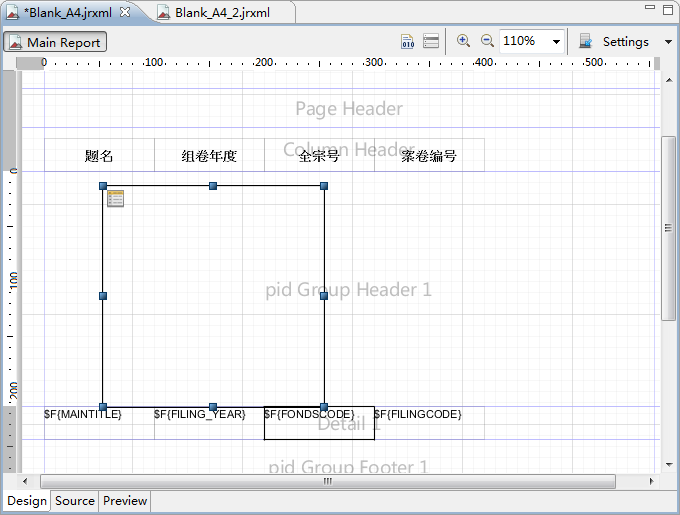
在“Name”中输入参数的名字，在“Expression”中输入参数的公式。公式可以使用系统的公式生成器自动生成。设定好后。点击“Finish”按钮即可。

系统自动显示子报表界面，如下所示。



可以在此处修改子报表的格式。修改完毕后，保存即可。

设置完毕后，主报表显示如下。



### 子报表自适应行高

在准备放置子报表的区域，先放置一个Field域，删除关联的字段值，即放置一个空白域。设置此域为自动扩展行高。设置如下。

其中，“Stretch Type”选项，选择“Relative to Band Height”。

子报表的field域设置为自动扩展行高。

父报表中的子报表域设置为取当前行最高值，即“Stretch Type”选项，选择“Relative to Tallest Object”。如此设置即可。