

特别声明：本文版权为原文作者李克喜所有。

作者 BLOG: <http://blog.csdn.net/jemlee2002/>。

本人是在使用 JasperReport 作报表时在网上找到这篇文章的，可能是经过多次转载的原因，显得有些凌乱和不全，故把资料重新整理了一下，以供学习者参考，勿作商业用途。特别感谢原文作者。

文章将会涉及 3 个方面的内容：

第一部分：使用 iReport 制作报表的详细过程（Windows 环境下）

第二部分：使用 Jasperreport 作为报表控件开发胖客户端报表应用

第三部分：使用 Jasperreport 作为报表控件开发 Web 报表应用

## 1 使用 iReport 制作报表的详细过程（Windows 环境下）

### 1.1 前言

在网络上可以搜索到很多使用 iReport 和 Jasperreport 配合实现各种报表任务的文章，但是我觉得很少有一篇（几乎没有）做一个比较详细的介绍如何使用 iReport 制作报表的全过程，我所看过的文章的基本思想是覆盖面广，很多内容都是提及即过，并不是开发人员都有时间为每个实践花费时间自己实现，如果能有更详细的资料，那岂不是一件乐事。出于这个念头促使我写这篇文章，希望能对那些使用 iReport 和 Jasperreport 朋友有所帮助，特别是需要亲身去实现报表的朋友，希望能给你们带来一点帮助。本文不是对 iReport 的每个细节进行介绍，关于 iReport 的每个细节可以参考 iReport 网站的资料，但是可能得花费一些费用。一般用户没有必要付出这些费用，因为我们关心的是如何制作我们需要的报表，而不是去扩展 iReport，比如制作 iReport 的插件。

### 1.2 准备

### 1.2.1 下载 JDK

地址: <http://www.sun.com>, 选择 1.3 以上版本 (建议 1.4.2 以上版本), 安装 JDK, 默认安装即可; 如果你的系统已经有安装过 JDK 或是有 JRE 即可省略这一步骤, 验证 JDK 或是 JRE 是否可以默认运行, 在命令行 (CMD) 打入 X:>java 如果出现: Usage: java 开头的一堆信息既是通过验证。否则必须进行配置, 配置信息如下, 在 windows 的环境变量设置:

path: 在最后面加入 “;java 的安装目录”

JAVA\_HOME : “java 的安装目录”

CLASSPATH: “java 的安装目录\bin”

重新验证 JDK 或是 JRE 是否可以在 CMD 任何位置运行

### 1.2.2 下载 iReport

地址: <http://ireport.sourceforge.net/>, 选择 0.3.2 版本 (发稿之前为止建议使用的版本) 解压 iReport 在任意目录, 解压后的文件里面有一个 iReport.bat, 通过双击, 过大约 30 秒钟如果可以弹出 iReport 的主窗体即表明你的系统已经可以运行 iReport 了, 如果不能弹出主窗体, 一般是第一步骤错误, 或是没有完成。

### 1.2.3 准备数据库

iReport 支持绝大部分数据库, 只要该数据库能提供 JDBC 驱动器。本文提供 MySQL 数据库作为例子, 但是文章最后会提到如何配置 Oracle 的支持。关于数据库的安装和建立表不属于本文的范围, 请参考其他资料。本文假设已经安装了 MySQL 和在 MySQL 已经有一些表, 并且确定表中已经有数据了。

\*【特别提示】MySQL 的版本要求与 iReport 文件夹下的 Lib 目录的使用 MySQL 驱动程序兼容, 笔者建议到 <http://dev.mysql.com/downloads/> 下载最新版本 mysql 的驱动器, 这样就不会应为 JDBC 驱动器的问题而当心数据库的支持问题。

#### 1.2.4 启动 MySQL 服务

确定 MySQL 使用的字符集是重要的问题，特别是对需要中文报表的朋友，应该特别注意这个问题。

#### 1.2.5 确定商务逻辑

也就是希望完成什么样的报表任务。需要实现的报表的详细描述，这是实现报表的业务条件，否则所有的任务绝大部分没有意义。本文使用一个 Bug 记录表为例，本文的例子是制作一个根据项目和项目中的模块分组的 Bug 量统计。

### 1.3 开始配置基本信息

#### 1.3.1 配置界面使用的语言和报表输出路径

第一次进入系统是英文环境，可以通过【Tools】—【Options】开启配置 iReport 系统的基本信息对话框。在“Language”选项里面选择你需要的界面语言，比如“中文—中国”。点击【Apply】按钮，系统既把整个界面中文化。

在配置 iReport 系统的基本信息对话框中选择【编译】Label，之后决定你的报表输出路径，可以把“编译在报表数据文件夹”选择打勾，这样报表的 jrxml 文件和 jasper 文件就放置同一文件夹。（在新建报表时会要求你把 jrxml 文件保存到指定的文件夹）

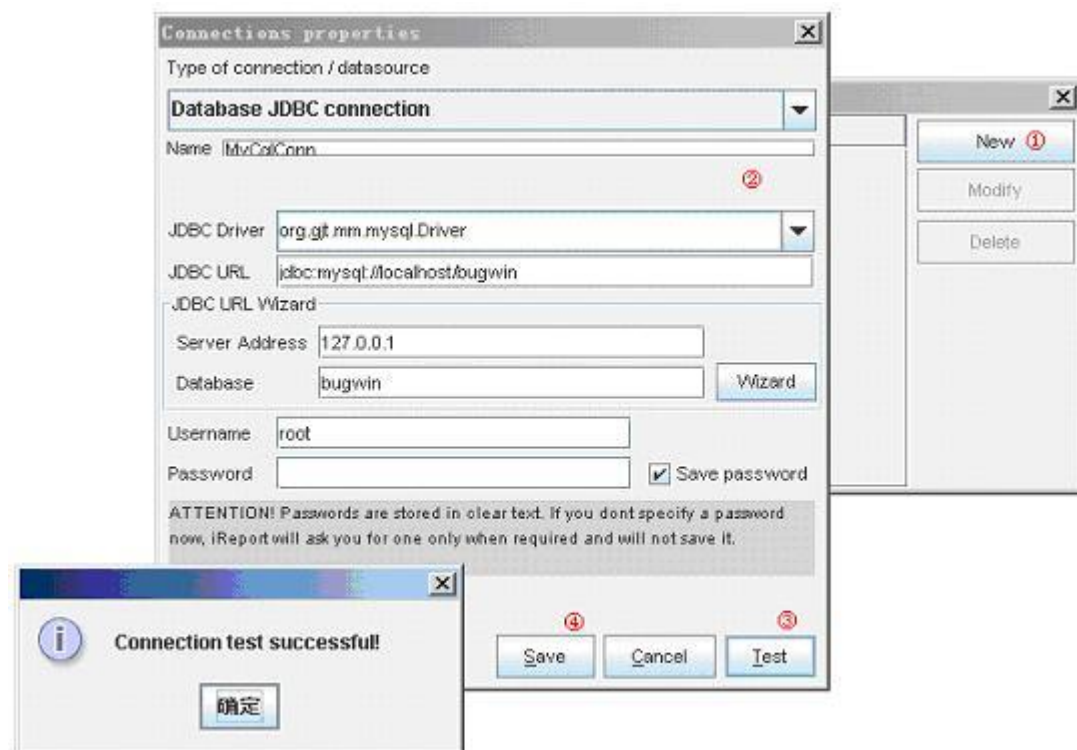
【Options】选项中的一部分参数修改不能通过【Apply】按钮直接起作用，比如“Look&Feel”，必须重新启动 iReport 才能起作用。不知道是不是 iReport 的 Bug？！最后【存档】。

#### 1.3.2 配置 MySQL 的数据库连接

这就是报表与数据库的接口。可以通过【资料来源】—【连接/资料来源】开启配置列表对话框，iReport 会记录以前使用的所有连接，除非你手工删除这些

连接，否则所有的连接都会存在连接/资料来源配置列表对话框中，不管是否确实可用。

点击【New】进入配置新连接界面，如图：



填写 JDBC 连接需要的信息，iReport 支持多种数据源连接，如图：



本文只是介绍 DataBase JDBC Connection 连接方式，这也是最常用的方式，特别是在嵌入式报表应用。所有的信息填写并测试通过之后，最后就是保存信息。回到配置列表对话框，关闭对话框，完成 MySql 数据库 JDBC 连接配置。

提示：如果你需要报表提供中文内容显示可以在 JDBC URL 下工夫，比如输入：

```
jdbc:mysql://localhost/SUBRDB?user=****&password=****&useUnicode=true&characterEncoding=GB2312
```

其中的\*\*\*\*号替换成数据库的用户名和密码。

### 1.3.3 新建一个空报表的基本配置

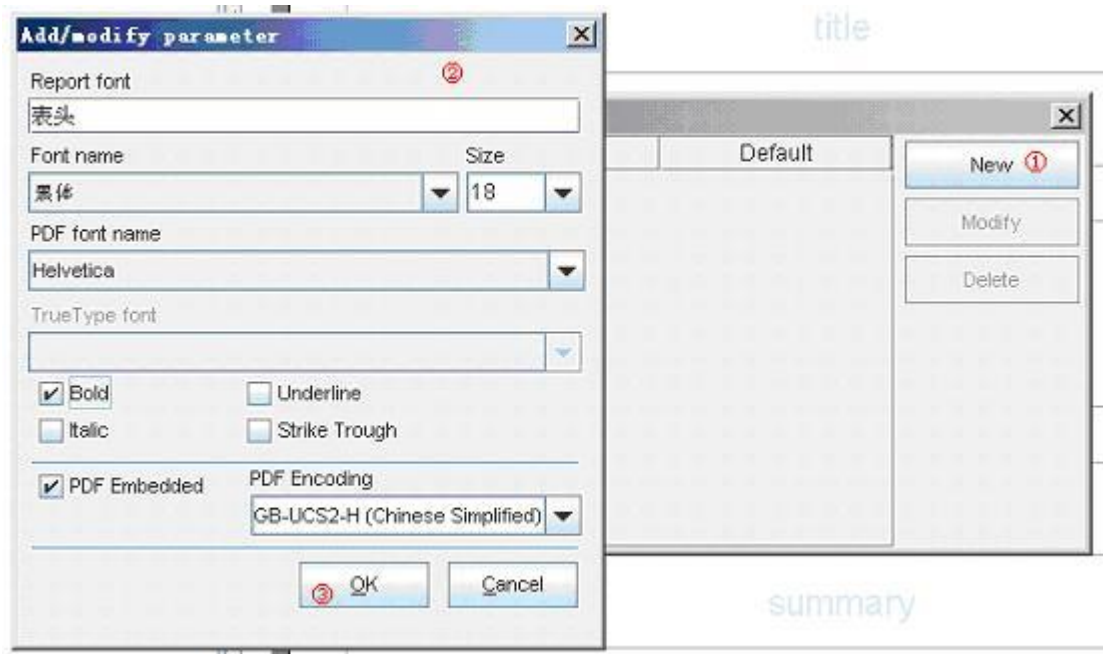
单击工具栏的第一个工具“New Report”，新建一个报表，输入报表名称和定义报表的一些参数，比如名称输入 BugsRpt（例子是做一个项目的 Bug 量统计报表），单击【More....】选择标签，填写或是选择 XML 编码，这是关系到你的 XML 支持的字符集的选择，请根据需要选择，比如需要你的 XML 文件支持中文，那么可以输入 GB2312 或是 GBK，之后单击【OK】按钮，进入报表的设计界面。

\*【特别提示】请在开始任何工作之前保存报表，这时 iReport 提示保存报表的位置，选择合适的位置之后输入 BugsRpt 作为名称。

### 1.3.4 定义报表可能需要的字体类型及其属性

一个报表的内容五花八门，有表头、栏位名、数据、其他变量信息等等，如果这些信息都是一致的字体和属性（比如颜色），那么整个报表就死气沉沉，显得很粗糙了。我们可以在为报表添加每个元素时定义元素的属性，但是那是一个多么费时的工作，如果能预先定义一些属性的组合，之后在创建每个元素时只需选择这些组合的其中一个即可，省事又快速。

单击【预览】—【报表字体】开启自定义组合对话框。单击【New】进入定义详细对话框，如图：按照图中的顺序填写信息和步骤，依次定义“表头”、“组”、“列”、“列内容”、“统计计算”、“其他”等 6 中字体组合。注意 PDF 内嵌字体的选择，如果你需要报表时以 PDF 文件格式提供，那么对此需要作出选择。



## 1.4 理解几个重要的概念

### 1.4.1 iReport 的输出格式

iReport 的预览输出格式可以支持以下几种：

PDF、HTML、CSV、JAVA2D、Excel、纯文字、JRViewer，其中最常用的是 PDF、JRViewer。本文以 JRViewer 为例子。JRViewer 是直接以 C/S 方式作为报表的输出格式，在 JFrame 框架下输出。Jasperreport 提供默认的 JRViewer 输出类。

### 1.4.2 报表的动态对象变量、参数、字段

在使用 iReport 的过程中会碰到很多与变量（Variables）、参数（Parameters）、字段（Fields）这些有关的内容，我们要介绍这些对象的使用和意义：

- 字段（Fields）：是数据库抽取出来的，希望在报表中出现的数据库内容。比如一个 ID 的所有值。`$F{ fieldsName }`

- 参数 (Parameters): 这是你的应用需要提供给报表的入口, 比如你希望在报表被解释的时候提供 Where 语句的条件值, 那么就可以使用参数(Parameters)。

`$P{ parameterName }`

- 变量 (Variables): 这是报表中一些逻辑运算的表现, 比如统计值。

`$V{ variablesName }`

每种对象的定义格式如每个对象的后面说明, 比如定义一个变量(Variables), 那么表达式就写成`$V{ variablesName }`, 报表中出现的就是这个变量的名称。后文会详细的介绍使用方法。

### 1.4.3 编译、静态运行、动态运行

Jasperreport 运行时就需要的就是一个 jasper 后缀的文件, 编译过程其实就是把 jrxml 后缀的文件生成 jasper 后缀的文件。(可以参考 Jasperreport 的运行原理)

静态运行和动态运行是相对的, 后者带数据源运行, 比如带数据库运行。前者就是静态文本运行, 和数据源无关, 如果报表中出现和数据源有关的对象, 则以 null 显示。

### 1.4.4 报表结构

一个报表的结构大致是几个部分: title、pageHeader、columnHeader、detail、columnFooter、pageFooter、summary、groupHeader、groupFooter。

- Title: 每个报表一般会有一个名字, 比如×××销售报表, title 就是搁置这个名称的最好地方了, 当然你也可以根据需要搁置在合适的地方。

- pageHeader: 报表的一些公共要素, 比如页码、创建时间、创建人等信息放置在这里是比较好的选择。

- columnHeader: 无可非议的这里是放置列的名称, 记住不是列数据。

- Detail: 放置需要循环的数据, 比如销售记录数据。

- columnFooter: 放置列级别的统计计算值或是列的说明。

- pageFooter: 放置页级别的统计值或是页的说明。

- **Summary:** 可能需要对几页（你的报表可能有几个页组成）的统计值。比如 50 个销售记录共占用了 3 页，那么放置这些统计记录的统计值最好的地方就是 summary。
- **groupHeader:** 每个表的内容可能需要根据某个属性进行划分显示内容和计算内容，比如希望以月份为单位每组分开显示销售记录，那么就可以定义一个组（组的定义参考后文），groupHeader 就是放置组说明或是组标志最好的地方。
- **Groupfooter:** 放置组的统计或是说明

## 1.5 向表添加对象

### 1.5.1 添加静态对象

可以通过工具栏的工具添加静态对象，比如文本，点击【**T**】，之后在报表的空白处单击，如此即可把静态对象添加到报表，然后拖动对象的边框，使它的大小合适，双击对象弹出对象的属性配置对话框，切换到【**Font**】Label，在“Report font”的 ComBox 选择“表头”字体，（表头字体是前文提供的自定义属性组合），再切换到【**Static Text**】Label，修改表头的名字，比如“Bug 统计报表”或是“销售记录统计表”等等与业务有关的内容。添加图片，请点击【**Image Tool**】，之后的操作与 Text 类似。其他静态对象操作步骤类似。

### 1.5.2 使用连接

还记得前文提供的（3.2 节）配置 MySQL 数据库连接吗？这里我们将要使用前文配置的连接了。选择菜单【**建立**】—【**使用动态连接**】开启可选的动态连接，选择任何一个你需要的连接最后【**OK**】，保存报表，这样你的报表就使用了这个连接了。

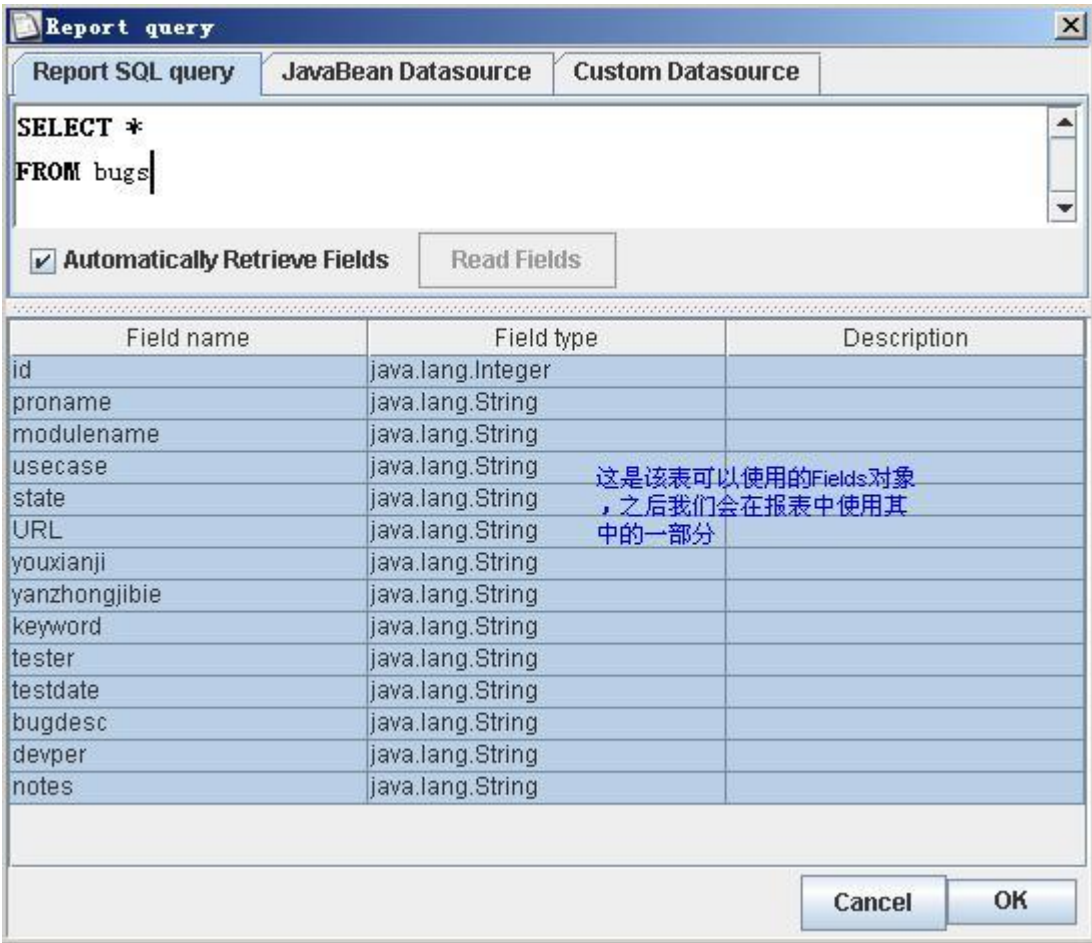
\*【**特别提示**】此连接必须与以后应用程序使用的连接一致。



### 1.5.3 创建 SQL 查询语句

SQL 语句是对任何 RDBMS 起作用的语言，外部用户需要使用这些语言管理维护数据库中的数据，同样的，iReport 也是需要这么做，我们需要提供查询数据库的语言—SQL 语句，这样，iReport 即可通过此 SQL 语句获取数据，之后组织到报表中并显示出来，以满足用户的需要为目的。

通过菜单【资料来源】—【报表查询】开启 SQL 输入对话框，并在【Report SQL Query】Label 中输入 SQL 语句如图：图中的“Automatically Retrieve Fields”checkbox 和“ReadFields”Button 是确认自动获取还是手动获取数据库表的可用 Fields。单击【OK】，保存报表。



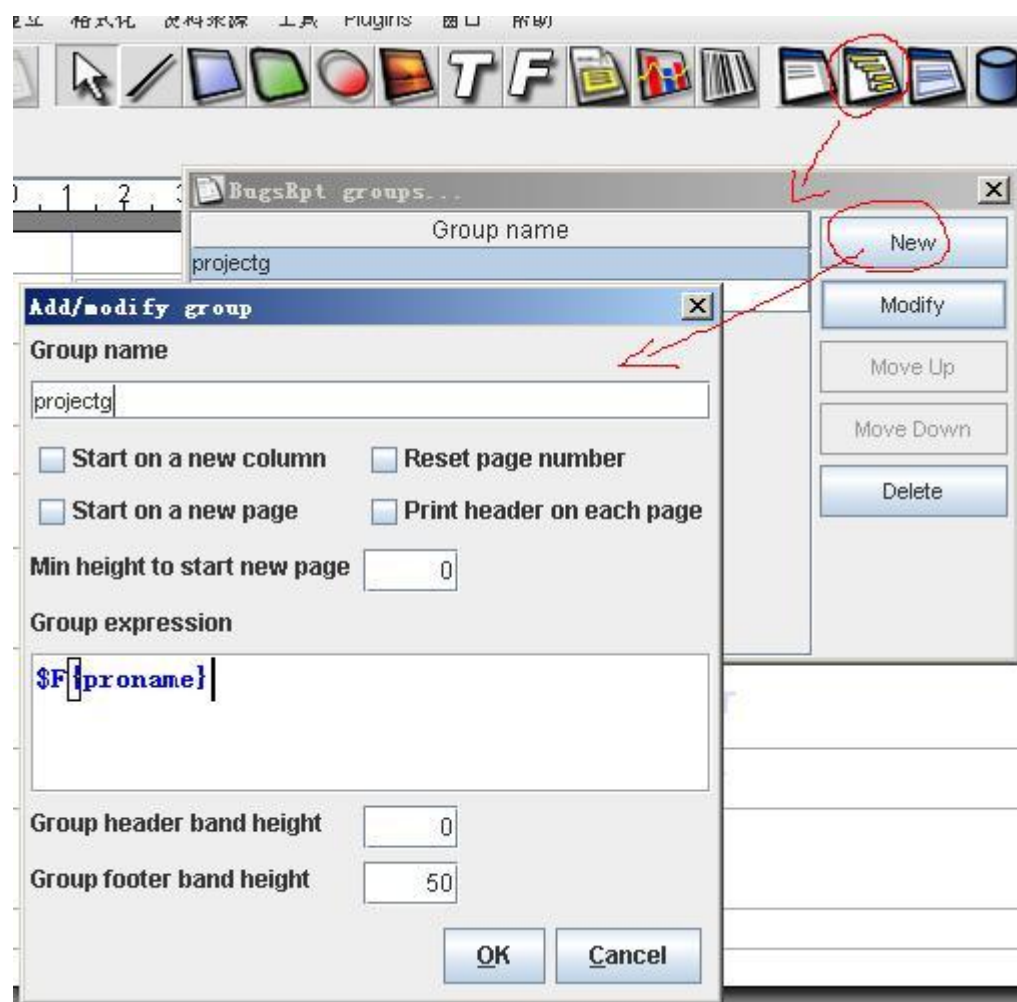
### 1.5.4 创建字段动态对象

报表的动态对象有变量、参数、字段，前文提及了他们的概念，这里将要一一讲解如何使用。

字段也就是数据库中的字段，通过菜单【预览】—【报表字段】开启字段的列表（工具条上可以找到相应的工具），可以拖放任意字段到报表的任何位置，比如拖动一部分 Bug 的内容字段到 detail 段（内容无关紧要，只要知道原理）。

### 1.5.5 创建组

组是一个很重要的概念，一个报表可以多个组，每个组以一个关键字为标记，比如希望 Bug 统计是根据项目（或是产品）进行统计的。那么可以设立一个项目标记的组。如图：



组的参数设定可以看界面即可理解部分，其中最主要的是“Group expression”，这是必须输入格式正确的并且存在的字段名称，本文的“praname”是【字段】中的一个元素。依此类推，建立其他的组对象。每建立一个组，在报表的界面上都会出现该组对应的段，如图：至

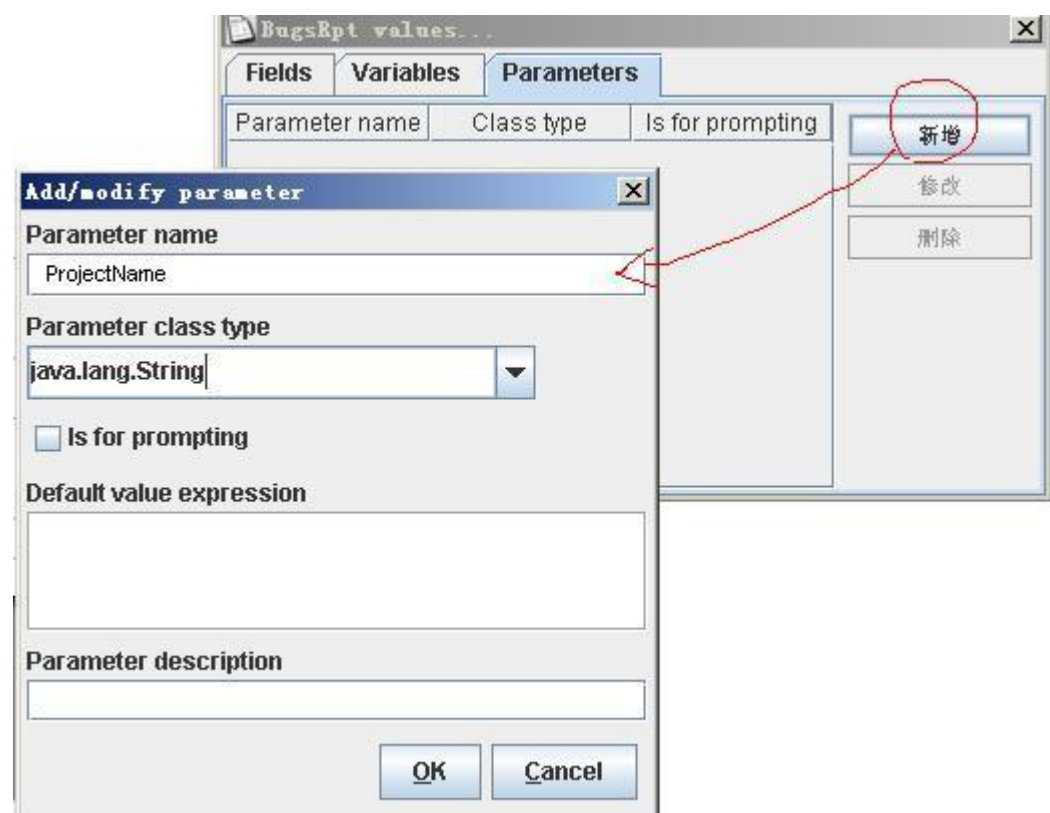
于他们的意义和容纳的内容参考“4.4 报表结构”，他们是首尾对应出现的。(Header 和 Footer)

[B U G 统 计 报 表]				第	\$V	页
优先级别	严重等级	测试人员	测试日期			
✓projectgHeader						
✓odulegHeader						
\$F{youxianji}	detail		\$F{tester}	\$F{testdate}		
✓odulegFooter						
✓objectgFooter						
columnFooter						
pageFooter						
summary						

1.5.6 添加参数和使用参数

我们重申参数作用，一般是需要外界提供参数给报表的入口，比如 SQL 语句的 where 条件的表达式。通过【预览】—【报表参数】开启报表参数列表对话框（工具条上可以找到

相应的工具)。如图：输入名称及其他参数。【ok】,保存报表。



那么如何使用呢？打开 SQL 语句对话框，参考“5.3 创建 SQL 查询语句”，这时候的 SQL 语句应该是：

SELECT \*

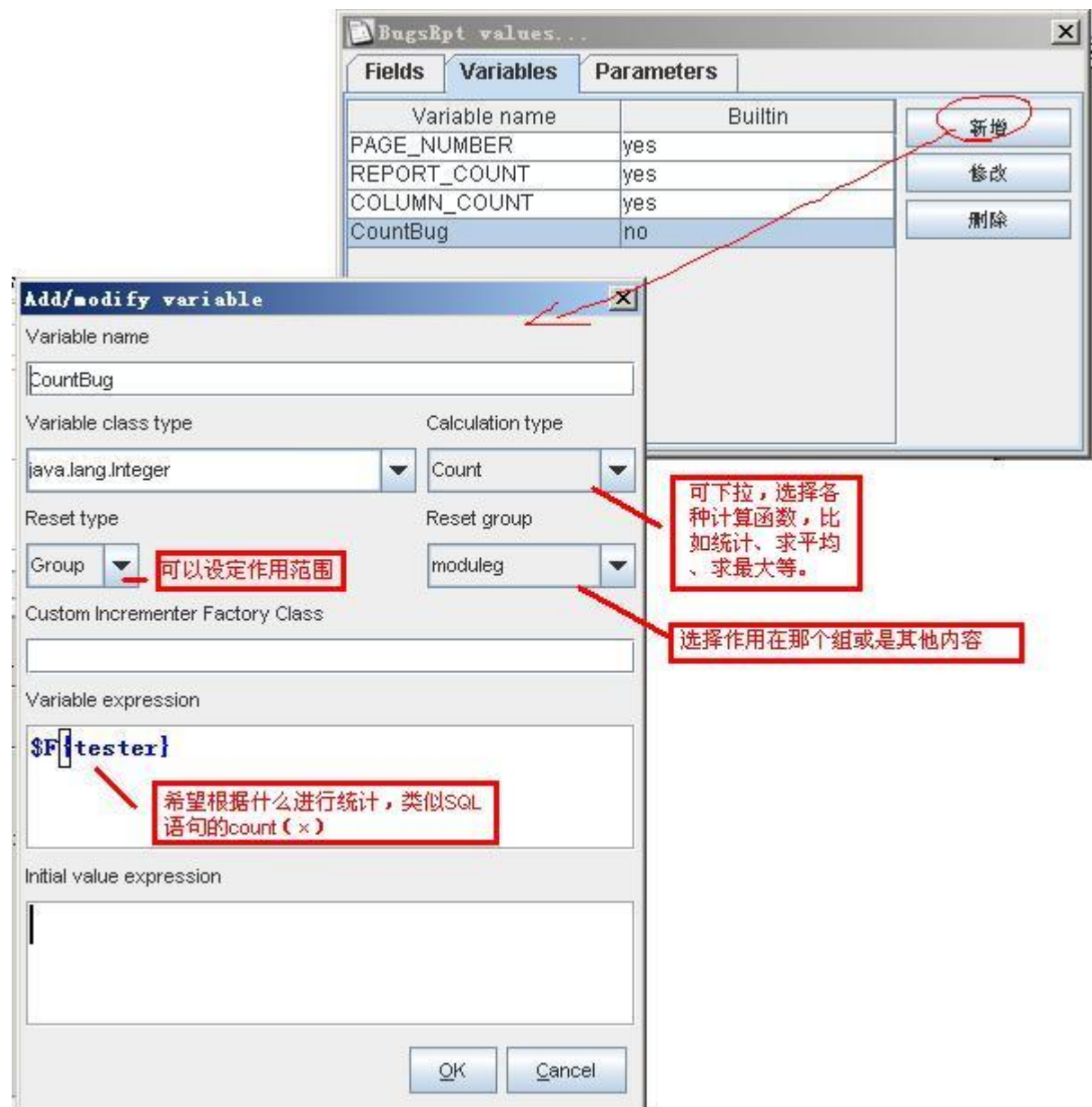
FROM bugs where pronaame=\$P{ProjectName} order by pronaame,modulename

注意其中的红体字部分，就是把刚才定义的变量运用到 SQL 语句了。这样当应用提供参数时，只要指定提供给这个参数，那么报表解释引擎即可替换这些变量然后再执行 SQL 语句，在第二部分提到编程时，会提供参数设定代码。

### 1.5.7 添加变量和使用变量

变量的定义类似参数，通过【预览】—【报表变量】开启报表变量列表对话框（工具条上可以找到相应的工具），如图，图中定义的变量的作用是：定义一个 Bug 的计数器，数据类型是 java.lang.Integer,使用 Count 函数进行统计字段 tester，作用范围是模块组，也就是统计模块的 Bug 量。其中的 tester 可以改成其他非组对象，比如 pronaame 是组对象，就不要用作这里的统计参数。以上提供的是自定义变量，其实 iReport 系统还有提供一些内嵌（Buildin）的变量，比如页

码，行记录数等，视需要而使用。



1.6 最后的报表

1.6.1 完成后的报表

[BUG 统计 报表]				第	\$V	页
静态文本内容		内嵌变量：页码				
优先级别	严重等级	测试人员	测试日期			
\$F {proname} — \$F {modulename}		组定义内容				
\$F {youxianji}	\$F {yanzhongjibie}	\$F {tester}	\$F {testdate}			
模块BUG统计：	\$V {CountBug}	详细信息 ( Detial )				
组结尾	变量	线				
pageFooter						
columnFooter						
pageFooter						
summary						

6.2、预览报表

点击动态运行报表，出现如图内容：





## 1.7 总结

第一部分只是介绍了如何制作一个动态数据报表，其实 iReport 还有提供很多的特性供开发人员使用，比如柱状图、饼图、及各种形状的图形等，满足企业绝大部分应用的需求。希望你能继续研究并充分利用。

关于使用 Oracle 数据库作为数据源的内容：提供与使用的 Oracle 版本对应的 JDBC 驱动，把驱动放置在 iReport 的 lib 目录，配置数据库的 JDBC 连接时如图：其它操作基本没有区别。

**Connections properties** [X]

Type of connection / datasource

**Database JDBC connection** ▼

Name

JDBC Driver  ▼

JDBC URL

JDBC URL Wizard

Server Address

Database

Username

Password  ☒ Save password

ATTENTION! Passwords are stored in clear text. If you dont specify a password now, iReport will ask you for one only when required and will not save it.

## 1.8 补充内容



1.8.1 实现表格

可以在 Detial 中加入必要线条实现表格,配合 columnHeader、columnHeader、Detial 这 3 个位置实现, 您可以试试画线的位置!

[BUG统计报表]

第 2 页

优先级别

严重等级

测试人员

测试日期

正常排队	其他：属参数设置方面的问题或建议	齐琳	2004-01-08
正常排队	处理结果正确，显示内容不明确	齐琳	2004-01-08
正常排队	严重错误，导致测试无法继续	李克喜	2004-01-08

模块BUG统计： 3

危化监管系统 — 会员公共信息模

正常排队	其他：属参数设置方面的问题或建议	李克喜	2004-01-07
正常排队	处理结果正确，显示内容不明确	李克喜	2004-01-08
正常排队	处理结果正确，显示内容不明确	李克喜	2004-01-08

模块BUG统计： 3

危化监管系统 — 会员管理模块

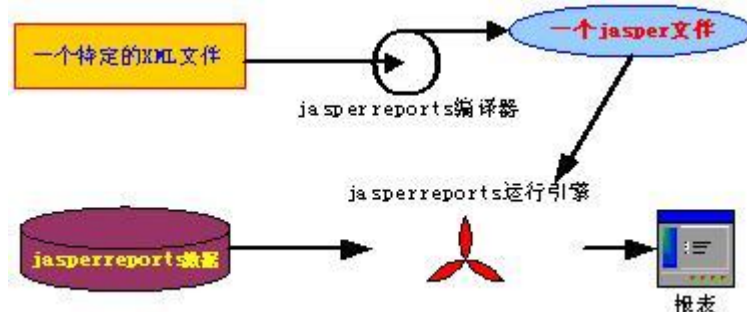
正常排队	一般错误，不影响系统运行	齐琳	2004-01-07
正常排队	一般错误，不影响系统运行	齐琳	2004-01-07
正常排队	一般错误，不影响系统运行	齐琳	2004-01-08
正常排队	一般错误，不影响系统运行	齐琳	2004-01-15
正常排队	一般错误，不影响系统运行	齐琳	2004-01-15
立即解决	一般错误，不影响系统运行	齐琳	2004-01-08

## 2 使用 Jasperreport 作为报表控件开发胖客户端报表应用

### 2.1 概述

我们对第一部分的内容做个简要的回忆，第一部分主要是介绍使用 iReport 如何制作一个数据报表，我认为文章比较详细的介绍“如何从零到满足大部分需求报表出现”。但是文中没有涉及 Jasperreport 的任何内容，目的是让你纯粹的理解如何做报表，因为做报表和把报表内嵌到应用程序编程是可以分工的，便于整合也便于分解。

我们知道 iReport 是一个 Jasperreport 的前端开发工具，iReport 用来制作和预览报表，为应用使用报表提供足够的前端支持。现在简要的介绍 Jasperreport 是如何工作的，这样你可以更好的理解 iReport 是如何帮助 Jasperreport 实现前端的工作，其实这些工作在我们看来就是“脏活”，为什么呢？看看下面的资料就知道了。



通过上图你大概已经明白 Jasperreport 的工作原理了。首先是要有一个 XML 文件（一般是以 jrxml 后缀），那么这个 XML 文件从那里来呢？做什么用呢？这个 XML 文件就是报表的定义文件，整个报表的每一个细节都在这个 XML 文件之中定义，一个简单报表的 xml 文件就有几百行，你可以手工编辑这个 XML 文件（一行一行，一段一段的编辑吧——这就是所谓的“脏活”）。如果是手工制作这个 XML 文件，单从效率上考虑就不允许，特别是现在很多应用系统的开发时间变得越来越紧张，总是会在时间上出问题。节省时间最好的办法就是充分的利用自动化工具，詹姆斯.马丁的软件工程思想也是反复的强调过程的自动化，如何做到自动化呢？就是充分的利用自动化工具集成到开发流程，说了一大堆不就是为了说要用 iReport 来做报表吗！这是我们第一部分已经完成的工作，但是这

里重要的是理解 Jasperreport 的工作原理。

## 2.2 建立 Application

参考 Jbuilder 有关资料。只要是普通 Window 窗口应用即可。

### 2.2.1 引入 JasperReports 需要的库文件

通过 jbuilder9 【Tool】 — 【configure Libraries】 — 按最左边的【New】 — 输入一个名称，比如 Report — 【add】按钮，导航到 iReport 的 lib 目录，把 lib 目录里面的所有文件引入。如图：



按两次 OK 回到 Jbuilder 的开发界面。

### 2.2.2 向工程添加 Report 系列库文件

通过右击工程，选择【properties】 — 【paths】 Label — 【Required Libraries】

Label—【Add】按钮—选择“Report”—按两次 OK 回到 Jbuilder 的开发界面。这样就添加完成了需要的库文件了。这么做是比较理想的，但是报表运行时可能不需要 iReport 的 lib 下所有的 jar 文件。

### 2.2.3 在主界面上添加需要的组件

在主 Frame 添加一个 Button 和一个 Label 以及一个 TextBox，当然你也可以通过创建菜单连接，如图：



这个窗体时用来做测试用的，实际的界面可能很复杂，其中的 TextBox 就是为报表的变量做准备的。这里填写的值就是要传递给报表的变量，也就是第一部分定义的带\$P{}符号的变量，程序会把他们对应起来。Button 的单击事件代码如下：

```
/**
 * RptDialog 对话框是用来承载报表的显示。
 * @param e
 */
void jButton1_actionPerformed(ActionEvent e) {
    //请创建一个对话框类 RptDialog
    RptDialog dlg = new RptDialog(this.jGroupID.getText());
```

```

        Dimension dlgSize = dlg.getPreferredSize();
        Dimension frmSize = getSize();
        Point loc = getLocation();
        dlg.setLocation( (frmSize.width - dlgSize.width) / 2 + loc.x,
                        (frmSize.height - dlgSize.height) / 2 + loc.y);

        dlg.setModal(true);
        dlg.pack();
        dlg.show();
    }

```

RptDialog 是接下来制作的一个 Dialog 对象，通过 Jbuilder 新建一个 Dialog 对象并命名 RptDialog。RptDialog 的代码如下：

```

import java.awt.*;
import javax.swing.*;
import dori.jasper.engine.*;
import dori.jasper.view.*;
import dori.jasper.engine.util.JRLoader;
import java.sql.Connection;
import java.io.File;
import java.util.Map;
import java.util.HashMap;
import mytest.trac.CommectionDB;

/**
 * <p>Title: RptDialog </p>
 * <p>Description: 报表承载对话框</p>
 * <p>Copyright: Copyright (c) 2004</p>
 * <p>Company: *****</p>
 * @author 李克喜
 * @version 1.0
 */
public class RptDialog extends JDialog {

```

```

JPanel ReportPan = new JPanel();
BorderLayout borderLayout1 = new BorderLayout();
//Bug 项目名称
String pronaame = "";
//显示报表需要的控件
JRViewer jrview;
public RptDialog(String GroupID) {
    pronaame = GroupID;
    try {
        jbInit();
        pack();
    }
    catch (Exception ex) {
        ex.printStackTrace();
    }
}
private void jbInit() throws Exception {
    ReportPan.setLayout(borderLayout1);
    getContentPane().add(ReportPan);
    //装载报表,在 Jbuilder 工程的目录创建一个 Reports 文件夹,并把报表的
jasper 文件搁置在该文件夹。
    String reportPath = System.getProperty("user.dir") + "\\Reports\\
BugsRpt.jasper";
    JasperReport jasperReport =
(JasperReport) JRLoader.loadObjectFromLocation(reportPath);
    //创建数据库的连接, 参考 java 的 JDBC 编程资料创建连接方式
    ConnectionDB conndb = new ConnectionDB();
    //注意: 这个连接要求与制作报表时使用的连接一致
    Connection conn = conndb.getDbConnection();

```

//报表配置参数，前文提过的 SQL 语句的 Where 条件参数就是与这里对应。

//条件的值可能是通过多种方式得到的，比如上一个页面传递过来的

```
Map parameters = new HashMap();
```

//**ProjectName** 就是 iReport 的变量\${ProjectName}的名称，参考第一部分的 5.6 添加参数和使用参

proname 就是从界面上获取的值。

```
parameters.put("ProjectName", proname);
```

```
JasperPrint jasperPrint =
```

```
    JasperFillManager.fillReport(
```

```
        jasperReport,
```

```
        parameters,
```

```
        conn
```

```
    );
```

//装载过程，注意其中的红体字部分

```
jrview = new dori.jasper.view.JRViewer(jasperPrint);
```

```
ReportPan.setLayout(borderLayout1);
```

```
ReportPan.setPreferredSize(new Dimension(800, 600));
```

```
getContentPane().add(ReportPan, BorderLayout.CENTER);
```

```
ReportPan.add(jrview,null);
```

```
}
```

```
}
```

到这里，所有需要设置和编程的工作基本完成了。

#### 2.2.4 运行

确定数据库已经启动了，在 JBuilder 运行应用程序。在 TextBox 输入有意义的组值，比如“BugWin 系统”，这是我的测试值，实际情况是根据你的需要来确定的。单击 Button，系统会弹出对话框，运行结果和第一部分 6.2、预览报表结果一致。

## 2.3 总结

随着技术的进步，应用环境的不断变化，胖客户端的应用可能会逐渐退出主流，但是它的存在是必要的。所以我要写胖客户端的应用方面的报表使用技术。文中简要明了的介绍了如何使用 Jasperreport 编程报表程序。希望对你有一点帮助。

## 3 第三部分：使用 Jasperreport 作为报表控件开发 Web 报表应用

### 3.1 概述

如何实现发 Web 报表有很多的选择，自定义 CSS+HTML 或是 XSLT+XML 或是其他控件，特别是支持图表的控件，比如：jfreechart。本文作为 Web 报表的一种实现方式，建议你使用 Jasperreport 作为报表控件，第一、二部分已经对制作报表和开发胖客户端报表应用做了介绍，其实有很多的文章可以参考实现第三部分的内容，不止是我写的这篇文章。

### 3.2 数据连接建立

参考 <http://blog.csdn.net/jemlee2002/archive/2004/09/28/JJEM.aspx>，这里有详细的介绍数据库的连接过程。

### 3.3 拷贝必要的 jar 文件到 Web 应用的 WEB-INF\lib 目录

每个 Web 应用都会有 WEB-INF 目录，但是 lib 是不一定有的，如果没有就创建它，本文需要的 jar 库文件有 3 个：

jasperreports-0.5.3.jar : jasperreports 执行时需要的 API

iTextAsian.jar : 亚洲字符集支持

itext-1.02b.jar : 其他字符集支持



如果你的报表全英文或是不需要支持亚洲字符集，那么 `iTextAsian.jar`、可以不要。

### 3.4 创建 `repotrs` 目录并导入 `.jasper` 文件

在 Web 应用中根目录下创建 `repotrs` 目录，其实这是一种建议，没有必要完全按照这样做，你可以根据你的业务需要创建 N 个目录或是层次目录。

把 `.jasper` 文件拷贝到 `repotrs` 目录下，比如例子中的 `BusinessRpt.jasper` 文件。

### 3.5 例子程序

`Test.jsp` 文件的内容：

```
<%@ page session="false" %>
<%@ page import="dori.jasper.engine.*" %>
<%@ page import="javax.naming.*"%>
<%@ page import="java.sql.*"%>
<%@ page import="javax.sql.*"%>
<%@ page import="java.util.*" %>
<%@ page import="java.io.*" %>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title></title>
<%
    DataSource ds = null;
    try{
        InitialContext ctx=new InitialContext();
        ds=(DataSource)ctx.lookup("java:comp/env/jdbc/mysql");
        Connection conn = ds.getConnection();
        //装载 jasper 文件
```

```

        File business_rpt = new
File(application.getRealPath("/reports/BusinessRpt.jasper"));

        //配置参数，可以参考《第二部分：使用 Jasperreport 作为报表控件开发胖客户端报表
应用》

        // http://blog.csdn.net/jemlee2002/archive/2004/10/08/JJem3.aspx
        //ProjectName 就是 iReport 的变量${ProjectName}的名称，
        //参考第一部分的 5.6 添加参数和使用参数
        //praname 就是从界面上获取的值。

        Map parameters = new HashMap();
        parameters.put("ProjectName ", praname);

        // JasperRunManager 是一个输出控制管理类，下文会提及部分内容

JasperRunManager.runReportToHtmlFile(business_rpt.getPath(),parameters,conn);

        //如果创建报表成功，则转向该报表，其实可以把报表套在框架内，这样实现比较有意
义的报表格式。

        response.sendRedirect("/reports/BusinessRpt.html");
    }catch(Exception ex){
        out.print("出现例外，信息是："+ex.getMessage());
        ex.printStackTrace();
    }
%>
</head>
<body>
</body>
</html>

```

### 3.6 关于 JasperRunManager

JasperRunManager 有很多的静态方法，控制输出的格式，比如输出格式是 pdf 或是 html 等，建议浏览 JasperRunManager 的一些方法，这样对开发报表输出有帮助。

### 3.7 输出内容

例子中输出格式是以 HTML 文件格式，所以 web 服务器可以直接解释并显示，效果不错。

### 3.8 总结

终于把 3 个部分的内容全部写完，到这里我可以休息一小会儿了，就像一休大师说得：“休息,休息……”。