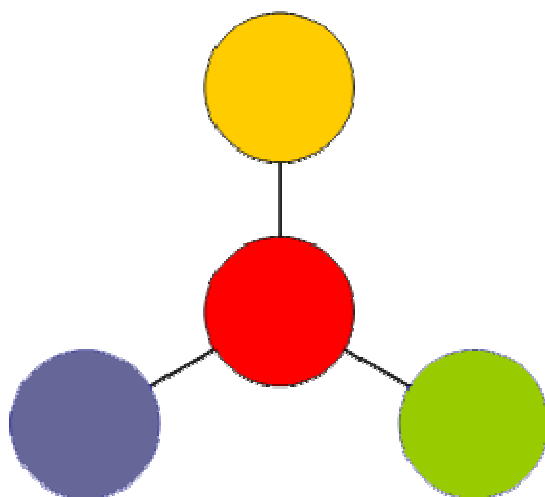


# **JasperReport + iReport**

## **报表工具详细开发手册**

**(V 1.1)**



**<http://www.damiya.cn>**

**2009-8-10**

# 目 录

1. 简介 .....	3
2. 安装 .....	3
2.1. IREPORT 安装.....	3
2.2. JASPERREPORT 安装.....	4
3. 入门篇.....	5
3.1. IREPORT 入门.....	5
3.1.1. 常用工具.....	5
3.1.2. 设置数据源.....	7
3.1.3. 新建空报表.....	7
3.1.4. 设置查询SQL .....	7
3.1.5. 设计报表.....	8
3.2. JASPERREPORT 入门.....	12
3.2.1. 在 WEB 中显示报表.....	12
4. 提高篇.....	13
4.1. 报表的结构 .....	13
4.2. 变量、参数及字段 .....	14
4.3. 图形报表 .....	16
4.3.1. 注意事项.....	16
4.3.2. 饼图(Pie) .....	16
4.3.3. 3D 饼图(3D Pie).....	17
4.3.4. 柱状图(Bar).....	17
4.3.5. 3D 柱状图(3D Bar).....	19
4.3.6. 堆栈柱状图(Stacked Bar).....	19
4.3.7. 曲线图(Line).....	19
4.3.8. 面积图(Area).....	20
4.3.9. 仪表盘(Meter) .....	20
4.3.10. 温度计(Thermometer).....	21
4.3.11. 气泡图(Bubble).....	22
4.4. 报表导出 .....	23
4.4.1. PDF 格式导出.....	23
4.4.2. EXCEL 格式导出.....	24
4.5. 调用 JAVA 代码.....	26
5. 技巧篇.....	28
5.1. 在 WEB 中显示数据不分页显示.....	28
5.2. 没有数据时仍然显示标题 .....	28
5.3. 数字型字段值到字符的转换 .....	28

## 1. 简介

JasperReport 是 Java 语言编写的报表引擎，它能解析.jasper 报表定义文件，并最终产生面向最终用户的报表界面。

iReport 是一个可视化的报表模板设计工具，使用此工具能方便地对报表进行可视化设计，设计结果保存成.jrxml 的 XML 文件，并能够把.jrxml 编译成.jasper 二进制文件供 JasperReport 报表引擎解析、显示。

本文档编写是基于以下版本：

jasperreports-3.5.2

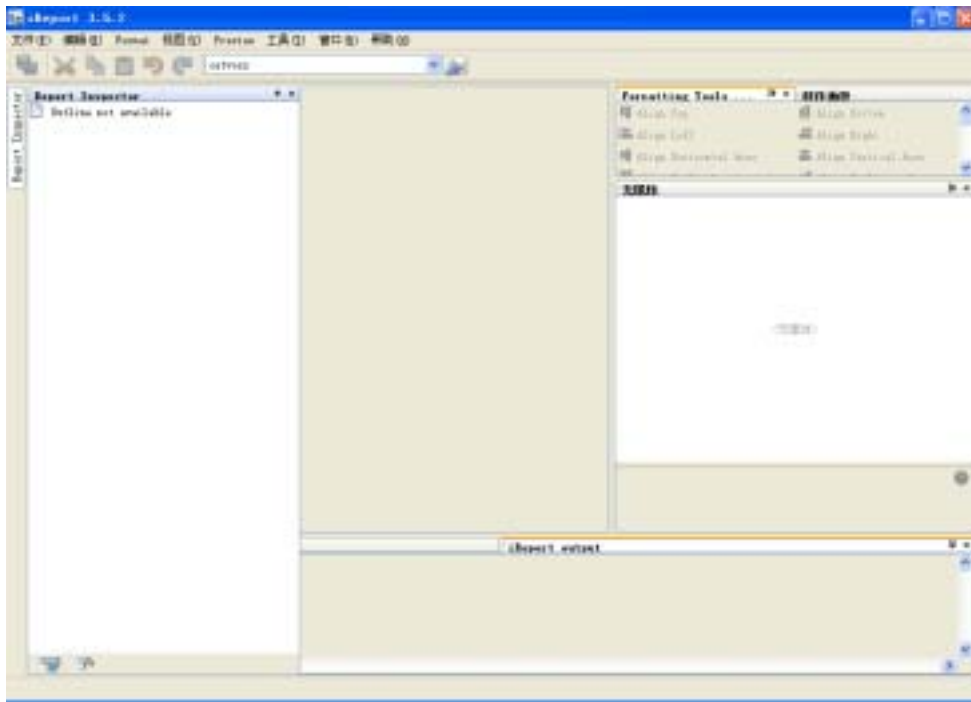
iReport-nb-3.5.2

## 2. 安装

从网站 <http://jasperforge.org> 上分别下载：iReport-nb-3.5.2.zip 和 jasperreports-3.5.2-project.zip 两个包。

### 2.1. iReport 安装

iReport 安装非常简单，只要解压缩 iReport-nb-3.5.2.zip 包后，点击运行\iReport-nb-3.5.2\bin\ireport.exe 程序即可。运行需要 Java 虚拟机，如果没有安装 Java 虚拟机，则必须先安装好 Java 虚拟机，并把 Java 的 bin 目录路径设置到 Path 环境变量中，然后才能运行 iReport。



## 2.2. JasperReport 安装

解压 jasperreports-3.5.2-project.zip 包 ,然后把需要的 jar 包拷贝到项目中即可。Jar 包有两个部分 ,一个是 JasperReport 本身的 jar 包 ,另一个部分是 JasperReport 依赖的第三方组件包。

\\jasperreports-3.5.2\\dist\\目录下是 JasperReport 本身的 jar 包 ,包括 :

jasperreports-3.5.2.jar

jasperreports-3.5.2-applet.jar

jasperreports-3.5.2-javafLOW.jar

\\jasperreports-3.5.2\\lib\\目录下的 jar 包是 JasperReport 依赖的第三方组件包。此目录下的 jar 包比较多。这些 jar 包中以 “ commons- ” 开头的都是必须要的 , 其他都是可选的 , 用到哪些功能 , 把对应的 jar 引入即可。

“ commons- ” 包有 :

commons-logging-api-1.0.2.jar

commons-logging-1.0.2.jar

commons-pool-1.3.jar

commons-vfs-1.0.jar

commons-beanutils-1.7.jar  
commons-codec-1.3.jar  
commons-collections-2.1.jar  
commons-dbcp-1.2.1.jar  
commons-digester-1.7.jar  
commons-httpclient-3.1.jar  
commons-javaflow-20060411.jar  
commons-math-1.0.jar

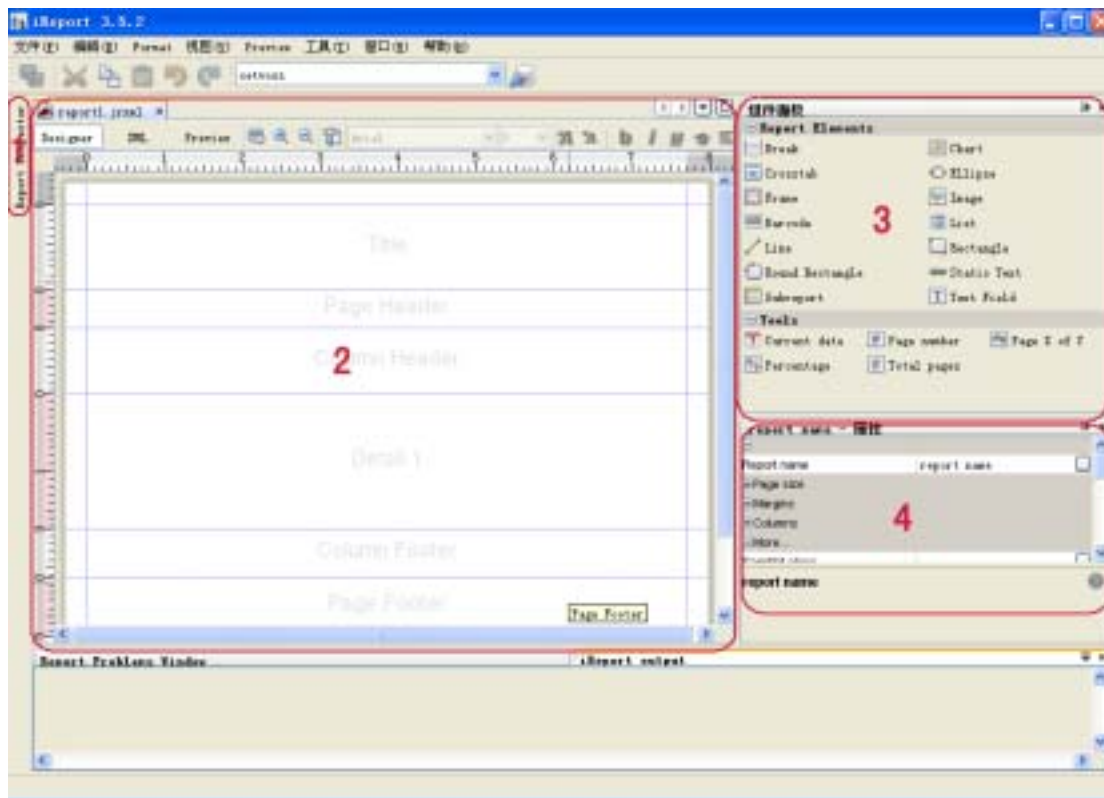
假如用到 JFreeChart 图形功能，则需要：

jfreechart-1.0.13-swt.jar  
jfreechart-1.0.13.jar  
jfreechart-1.0.13-experimental.jar  
jcommon-1.0.0.jar

### **3. 入门篇**

#### **3.1. iReport 入门**

##### **3.1.1. 常用工具**

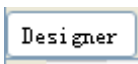



区域 1：Report Inspector 区域，报表用到的字段、参数、变量、ScriptLet 都在这里。如果没有此窗口，可在“窗口”菜单中打开。

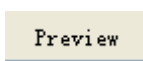
区域 2：报表设计区域，画报表的样式。

区域 3：组件面板，报表中用到的各种组件。如果没有此窗口，可在“窗口”菜单中打开。


区域 4：属性设置区域，报表、报表区域、报表组件等对象属性值的设置区域。如果没有此窗口，可在“窗口”菜单中打开。

设计与编辑模式按钮：

XML 编辑模式按钮：

预览(Preview)按钮：

设置数据查询 SQL：

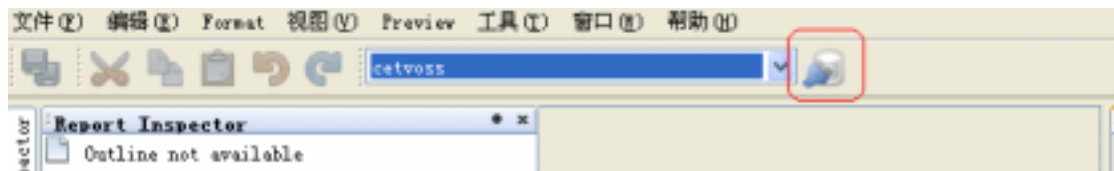
编译：

在组件面板中有两个常用控件：Static Text 和 Text Field。Static Text 输入字符串文本。Text Field 输入表达式，表达式有标签表达式和 Java 语句。Text

Field 控件中输入的必须是表达式，如果直接输入字符串则出错，如果需要输入字符串，则使用 Java 字符串语句：String.valueOf("字符串表达式")。

### 3.1.2. 设置数据源

点击图中红色框圈住的按钮设置数据源：



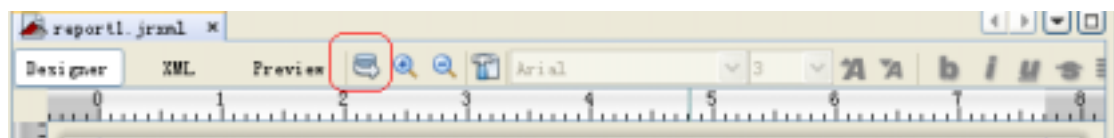
如果是连接数据库，一般使用“Database JDBC connection”，然后根据提示设置 JDBC 参数即可。保存之前可点击测试按钮测试一下是否连接成功。

### 3.1.3. 新建空报表

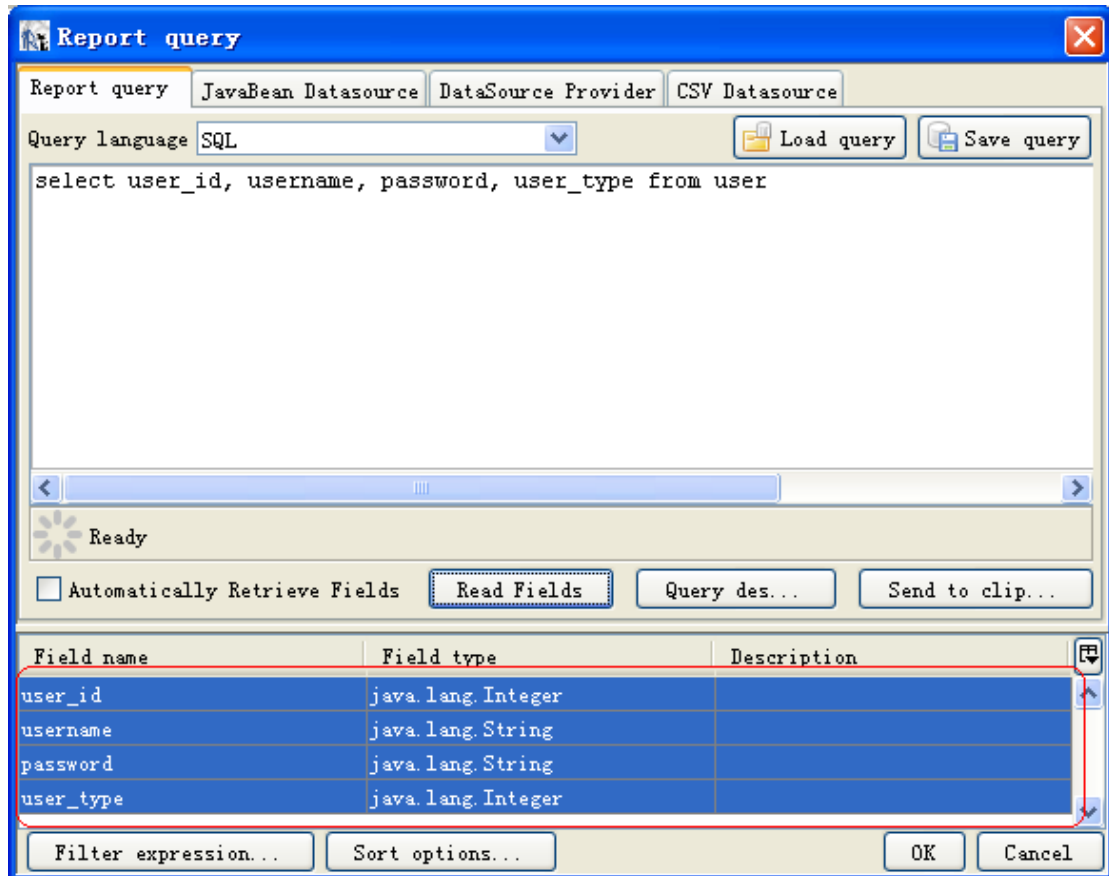
选择菜单：File -> New -> Empty report

### 3.1.4. 设置查询 SQL

点击报表中的红色框圈住的按钮，打开设置数据查询 SQL 语句的界面。



在 SQL 数据框中输入 SQL 查询语句：

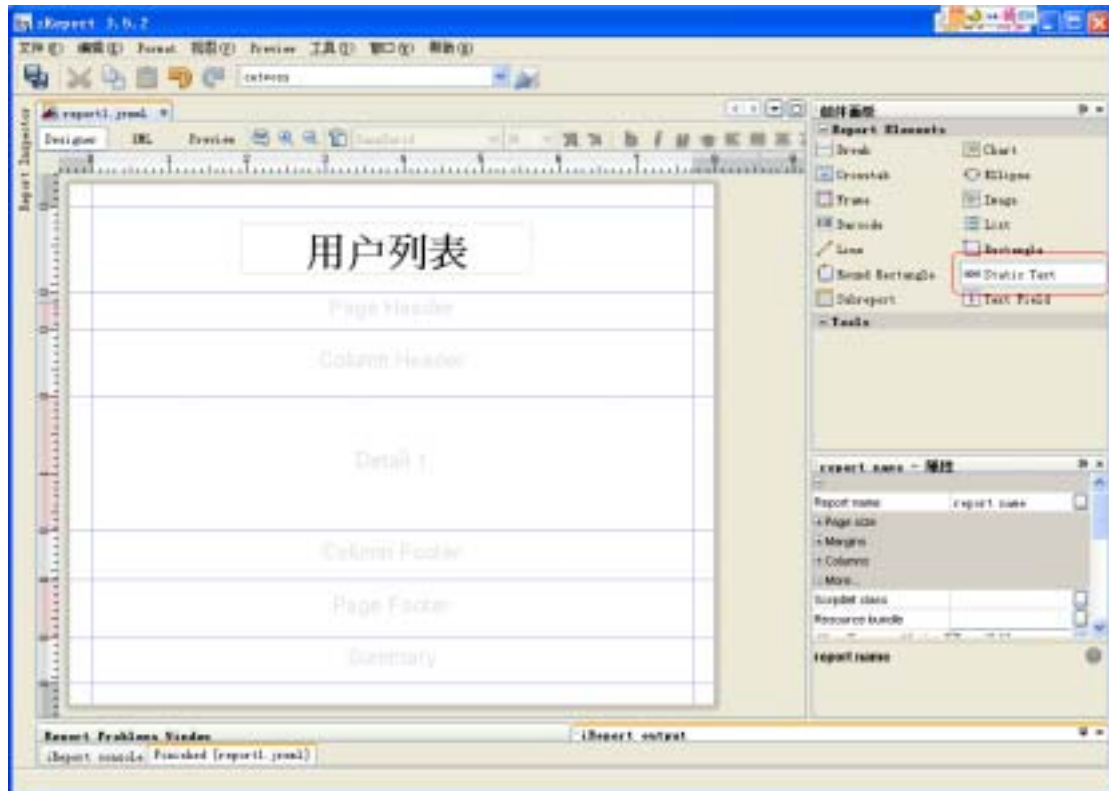


然后点击“ Read Fields ”按钮，系统把查询的字段列举在红色框位置处，然后点击“ OK ”按钮。

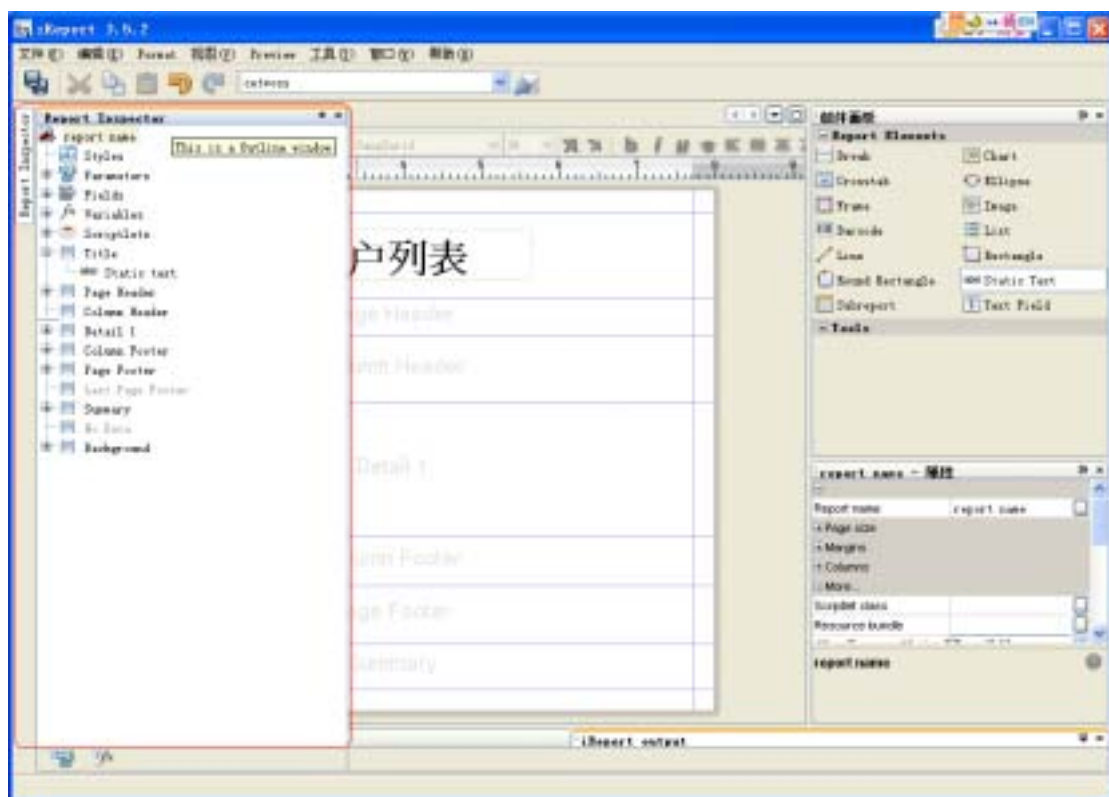
### 3.1.5. 设计报表

从右侧组件面板中拖拽一个“ Static text ”组件到报表中的“ Title ”处。并把“ Static text ”改成“ 用户列表 ”。



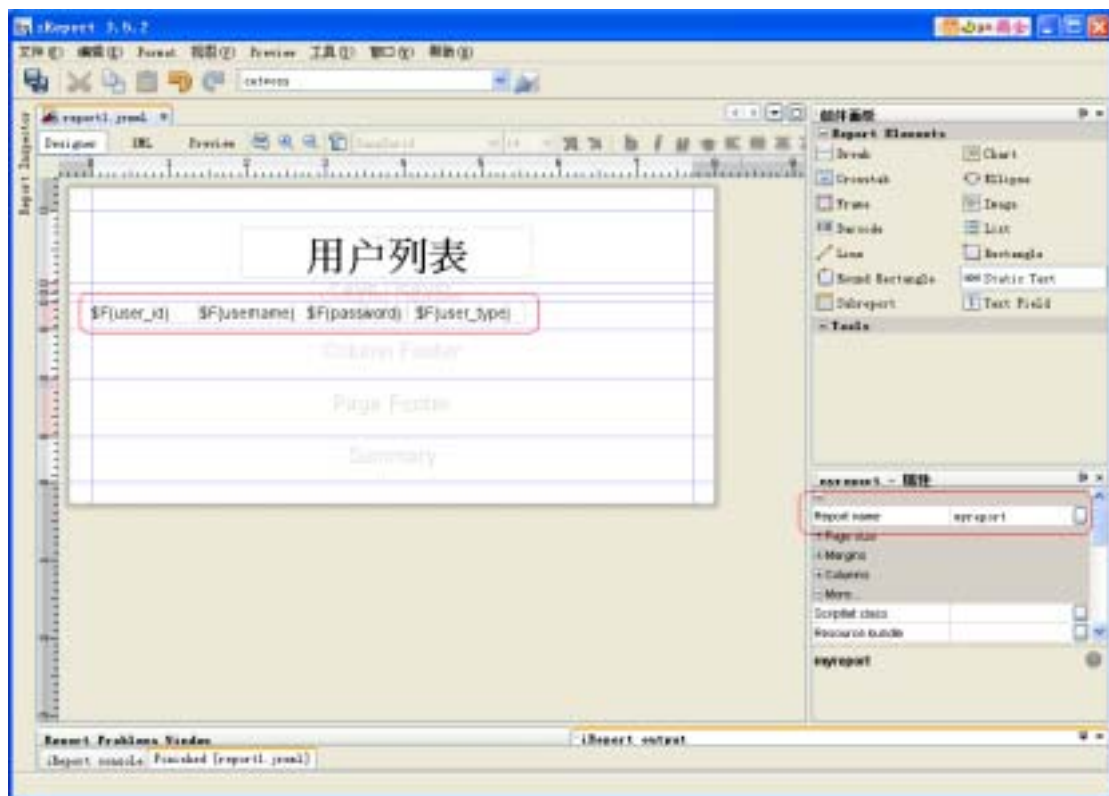


点击左侧的“ Report Inspector ”按钮打开 Report Inspector 面板：

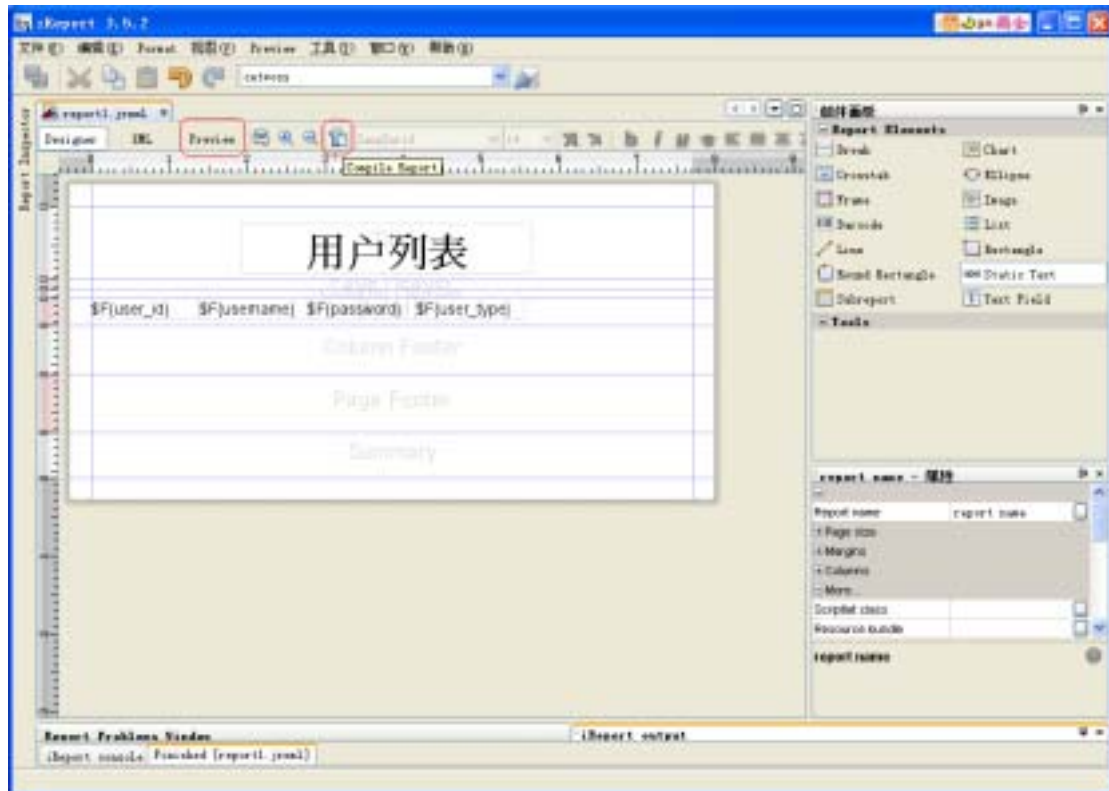


把 Report Inspector 面板中 Fields 中的字段都拖拽到报表中的“ Detail 1 ”区域中，排列好顺序，对齐，调低“ Detail 1 ”区域的高度。在属性设置区域中把报表名

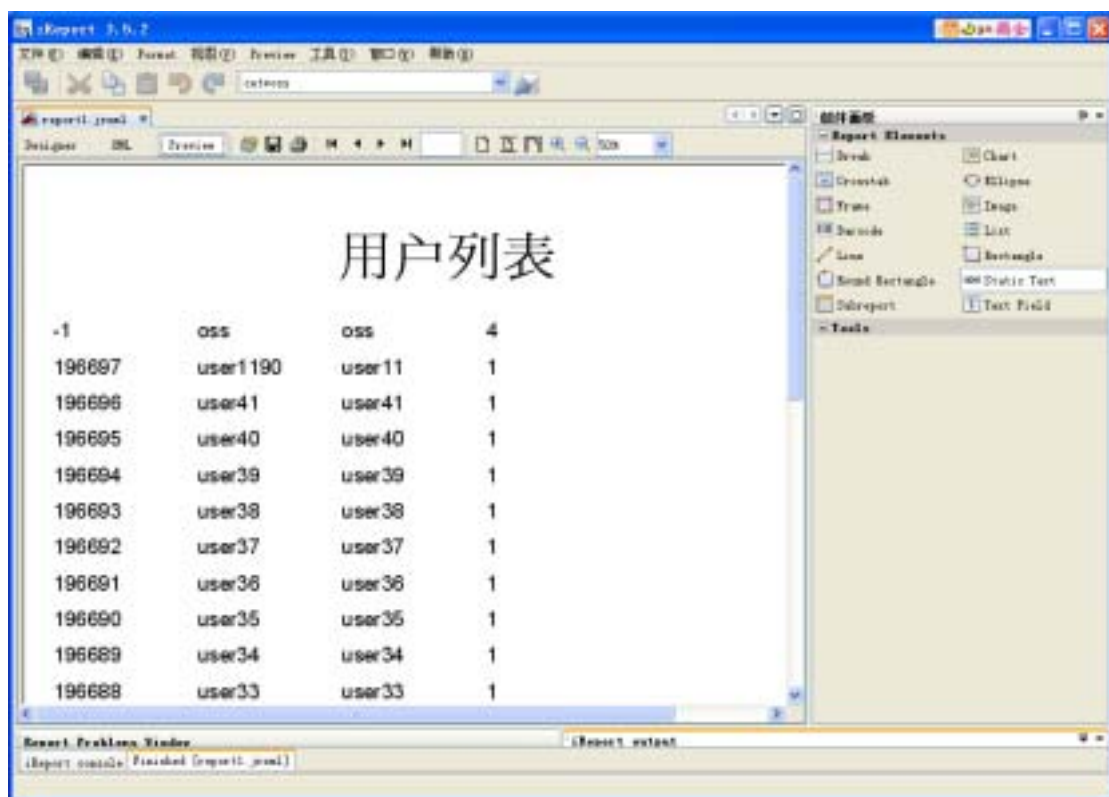
称由默认的“report name”修改成“myreport”，此名称在程序调用时需要用到。



点击编译按钮，编译后系统会生成一个后缀为“.jasper”的文件，后面我们将用到这个文件，编译完后在点击预览 Preview 按钮：



预览结果：



至此一个简单的报表便已经制作好。如果想知道如何在 WEB 中运行查看此报表请参看 JasperReport 入门一节。

## 3.2. JasperReport 入门

### 3.2.1. 在 WEB 中显示报表

首先需要搭建 WEB 环境，请参看 JasperReport 安装一节。在 WEB 服务根目录下建立 reportRun.jsp 程序。把报表编译时生成的后缀为 “.jasper” 的文件拷贝到 WEB 服务根目录下。reportRun.jsp 程序如下：

```
<%@ page contentType="text/html;charset=GB2312"%>
<%@ page import="net.sf.jasperreports.engine.*"%>
<%@ page import="java.util.*"%>
<%@ page import="java.io.*" %>
<%@ page import="java.sql.*" %>
<%
String ctxpath = request.getContextPath();
//创建数据库连接Connection对象
Class.forName("com.mysql.jdbc.Driver");
String url="jdbc:mysql://172.16.48.57:3306/cetvoss2";
String user="cetvoss2";
String password="cetvoss2";
Connection conn= DriverManager.getConnection(url,user,password);
//报表编译之后生成的.jasper文件的存放位置
File reportFile = new
File(this.getServletContext().getRealPath("/report1.jasper"));
//设置参数
Map parameters = new HashMap();
try {
    //执行报表程序
    JasperRunManager.runReportToHtmlFile(reportFile.getPath(),
parameters, conn);
    response.sendRedirect(ctxpath+"/myreport.html"); //此处的myreport是
报表的名称
}
catch (Exception e) {
    System.out.println( e.getMessage() );
}
finally {
    try {
        conn.close();
    }
    catch (Exception ex) {
        System.out.println( ex.getMessage() );
    }
}
```

```
}  
%>
```

执行 WEB 程序，WEB 中显示效果如下：



4. 提高篇

4.1. 报表的结构

一个报表的结构大致是几个部分：Title、PageHeader、ColumnHeader、Detail、ColumnFooter、PageFooter、Summary。

- 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。Title 只在第一页出现。
- 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：放置列的名称，不是列数据。
- Detail：放置需要循环输出的数据，比如销售记录数据。此处会按照数据的条数循环显示。

- ColumnFooter：放置列级别的统计值或是列的说明。
- PageFooter：放置页级别的统计值或是页的说明。
- Summary：需要对几页（报表可能有多个页）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地方就是 Summary。Summary 只在最后一页出现。

## 4.2. 变量、参数及字段

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，下面介绍这些对象的使用和意义：

- 变量（Variables）：由报表系统内部产生，不需要从外界传入的动态数据，如报表的打印时间等，可以通过定义一个变量来实现。调用方法：`$V{variableName}`。定义方法如下：从“窗口”菜单中打开“Report Inspector”窗口，鼠标右键点击“Variables”，选择“添加”项，然后在属性窗口中设置此变量的各个属性值。

以下是 JasperReport 的内置变量：

参数名	说明
<code>\$V{PAGE_NUMBER}</code>	代表当前页数（可以是页码也可以是页数，通过 TextField 的计算时间的不同值来设置）。
<code>\$V{PAGE_COUNT}</code>	当前页面中记录的数目。
<code>\$V{COLUMN_NUMBER}</code>	列号码。
<code>\$V{COLUMN_COUNT}</code>	当前列中记录的数目（是当前页里行的序号，在新页上会重置为 1）。
<code>\$V{REPORT_COUNT}</code>	当前文档中数据源记录数目。

- 参数（Parameters）：应用程序动态传给报表的参数，如希望报表动态执行某个 SQL 查询条件。定义方法如下：从“窗口”菜单中打开“Report Inspector”窗口，鼠标右键点击“Parameters”，选择“添加”项。例如动态定义 SQL 查询条件年份：

(1) 先定义变量：year

(2) 设置查询 SQL 如下(设置方法参看“设置查询 SQL”一节)：

```
select * from data where report_year = $P{year}
```

(3) 应用程序传参数给报表，方法参见“输出 PDF 的 JSP 程序”一节中给参数 parameters 设置值的方法。

参数的引用方式有两种，一种是\$P{}，另一种是\$P!{}，前者可以出现在任意的表达式位置，而后者则只能出现在 SQL 脚本里，用来替换查询语句，用以按照不同的需求替换查询语句。同时，在脚本里，也有些许区别：前者只能替换语句的部分字段，如 SELECT \* FROM bugs where name=\$P{Name} order by proname, modulename ,而对于\$P!{} 则可以直接在 SQL 脚本框里输入\$P!{SQL}。

以下是 JasperReport 的内置参数：

参数名	数据类型	说明
\$P{REPORT_PARAMETERS_MAP}	java.util.Map	这个参数总是指向一个 java.util.Map 对象，该对象保存了用户调用报表装填过程时传递给报表引擎的用户定的参数。
\$P{REPORT_CONNECTION}	java.sql.Connection	这个参数指向一个 java.sql.Connection 对象，报表引擎用来通过 JDBC 来执行 SQL 查询。
\$P{REPORT_DATA_SOURCE}	net.sf.jasperreports.engine.JRDataSource	可以由应用程序提供动态的数据源。
\$P{REPORT_SCRIPTLET}	net.sf.jasperreports.engine.JRAbstractScriptlet	这个内置的参数指向一个 net.sf.jasper.engine.JRAbstractScriptlet 实例，该实例默认是一个 net.sf.jasper.engine.JRDefaultScriptlet 对象。可以通过报表属性 Scriptlet class 项设置自己的 Scriptlet。

- 字段 (Fields) : 是数据源抽取出来的, 如果数据源是数据库, 则此处就是表字段名称。引用方法: `$F{ fileName }`。在“设置查询 SQL”一节中有获得字段的方法, 请参考。

### 4.3. 图形报表

#### 4.3.1. 注意事项

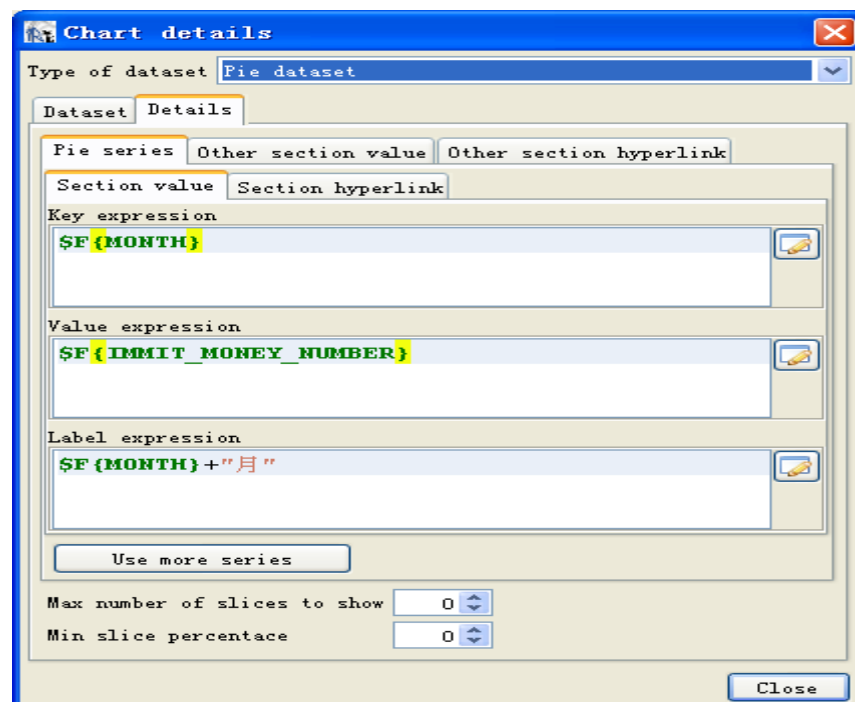
- 重复显示多个图形

在设计报表时, 如果发现显示图形时, 出现重复现象, 则是图形组件放置的位置不正确。

应该把图形组件放到 Column Footer 区域中, 如果放在 Detail 区域中则出现重复显示多个图形问题。

#### 4.3.2. 饼图(Pie)

把组件面板中的 Chart 组件拖拽到报表的 Column Footer 区域中, 并选择饼图。鼠标左键点击图形组件选中, 然后再点击鼠标右键, 选择弹出的菜单的“Chart Data”选项, 然后选择 Details 选项卡。



有 3 项必须填写表达式：

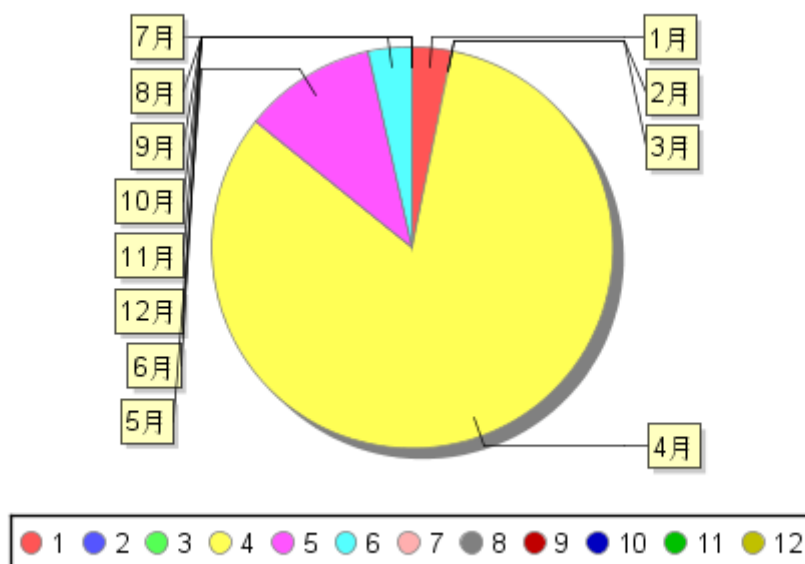


- Key expression：键表达式。如果是从数据库取，则可写成表字段，可通过右侧的按钮选择 Field 字段。
- Value expression：值表达式。可通过右侧的按钮选择 Field 字段。
- Label expression：标签表达式，图形区域的文本注释。

其它一些设置说明：

- Max number of slices to show：最多显示的切片数，如果切片数多于此数，则多于此数的其它切片值的合计放置在 Other 切片中。
- Min slice percentace(可能写错了，估计是 percentage)：最小切片所占的百分数值，如果切片的百分数小于此值，则不在显示。

饼图的预览效果：

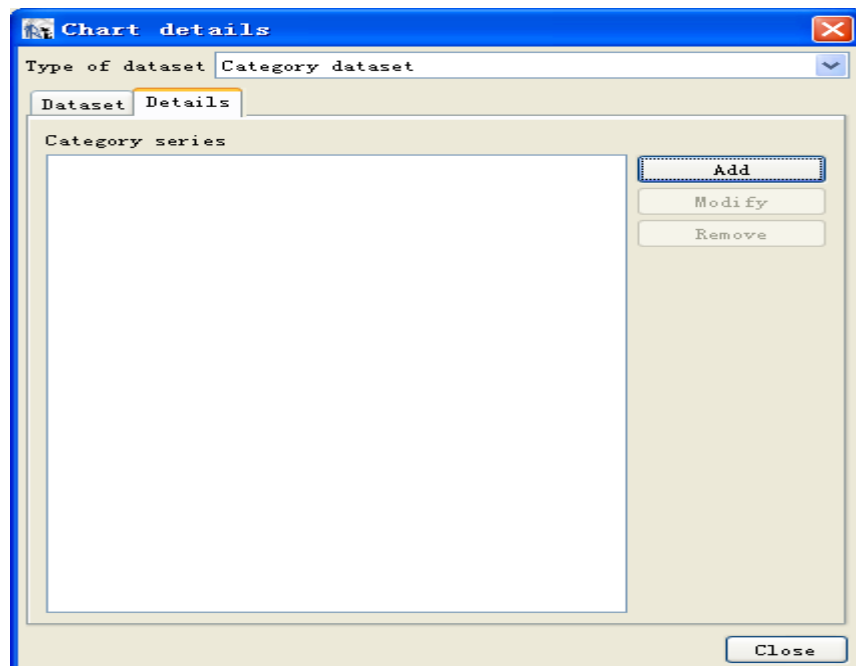


#### 4.3.3. 3D 饼图(3D Pie)

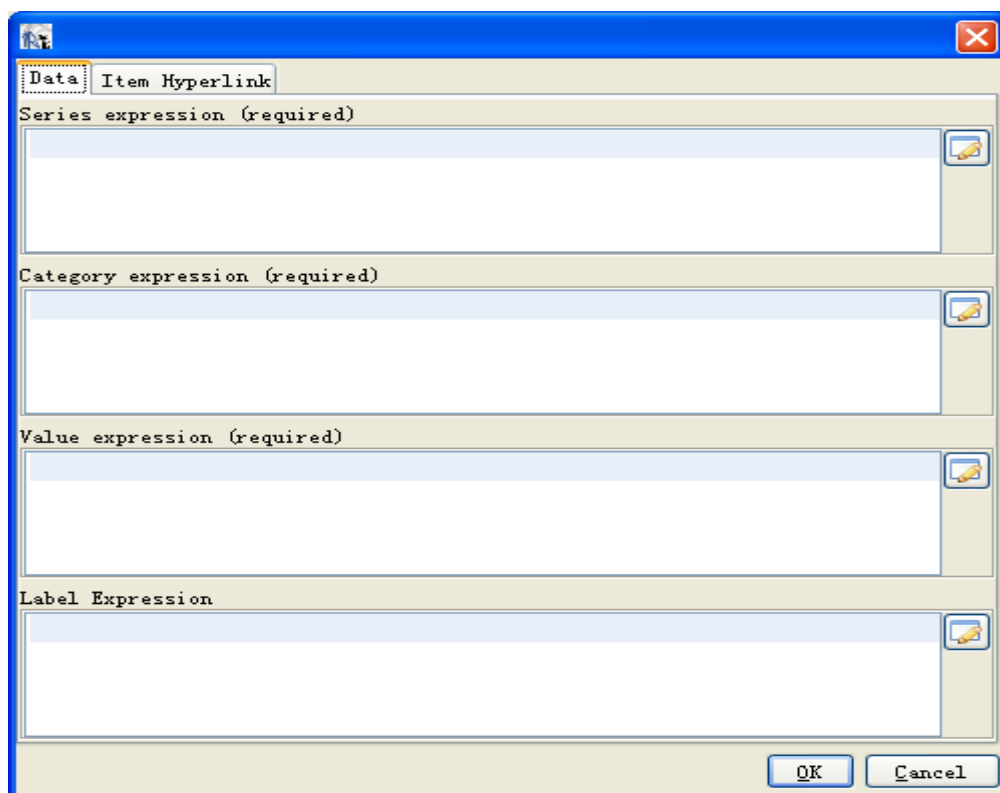
3D 饼图的用法和平面饼图的用法一样，只要在使用图形组件时，选择 3D 饼图即可。

#### 4.3.4. 柱状图(Bar)

把组件面板中的 Chart 组件拖拽到报表的 Column Footer 区域中，并选择柱状图。同样打开“Chart Data”界面的“Details”选项卡。



点击“Add”按钮，添加分类序列。



各输入项的说明：

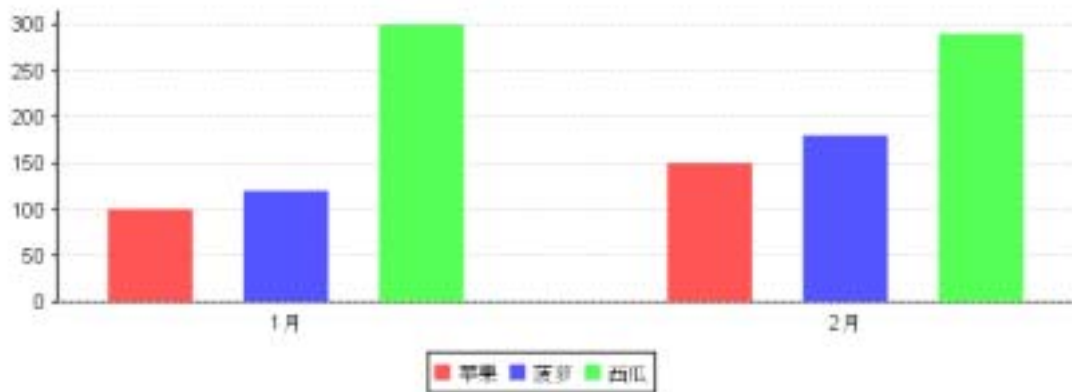
Series expression：序列名称表达式，也就是二级分类表达式，如果是从数据库取，则可写成表字段，可通过右侧的按钮选择 Field 字段。

Category expression：分类名称表达式，也就是一级分类表达式，可通过右侧的按钮选择 Field 字段。

Value expression：值表达式，可通过右侧的按钮选择 Field 字段。

Label expression：标签表达式，可通过右侧的按钮选择 Field 字段。

柱状图的预览效果：



#### 4.3.5. 3D 柱状图(3D Bar)

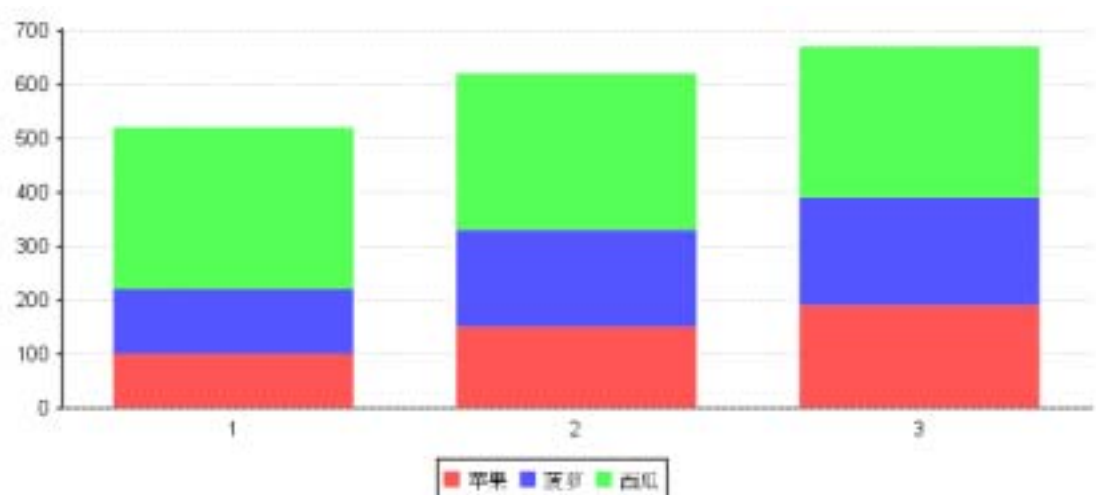
使用方法和平面柱状图一样。

#### 4.3.6. 堆栈柱状图(Stacked Bar)

堆栈柱状图的参数设置项和柱状图一样。只是选择图形组件时，选择曲线图形组件即可。

堆栈柱状图中 Series expression 分类(二级分类)，会以堆的形式现在一个柱上。

例如下图，Category(一级分类)是月份，Series 分类(二级分类)是水果种类。



#### 4.3.7. 曲线图(Line)

曲线图的参数设置项和柱状图一样。只是选择图形组件时，选择曲线图形组

件(Line)即可。

#### 4.3.8. 面积图(Area)

面积图的参数设置项和柱状图一样。只是选择图形组件时,选择面积图形组件(Area)即可。

#### 4.3.9. 仪表盘(Meter)

仪表盘的数据只需要一个值,而不是数组。

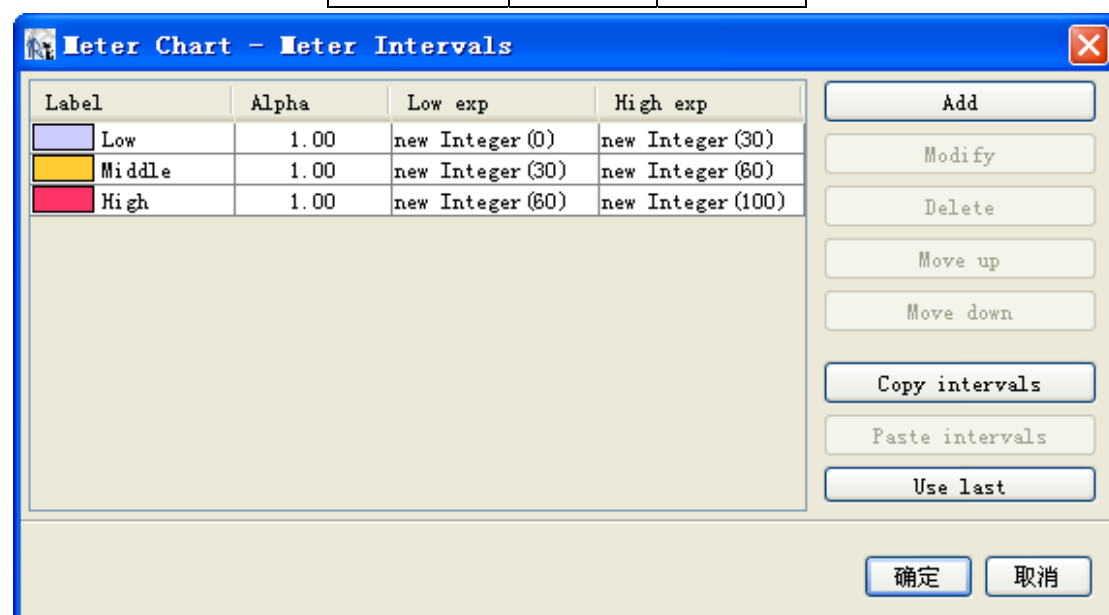
仪表的设置属性在属性设置面板中设置,常用到的属性如下:

Data Range Low Expression: 设置数值区间的最低值。例如: new Integer(0)

Data Range High Expression: 设置数值区间的最高值。例如: new Integer(100)

Meter Intervals: 设置切片区域。如下图设置了 3 个切片: Low、Middle、High。切片需要设置颜色,最低值,最高值。

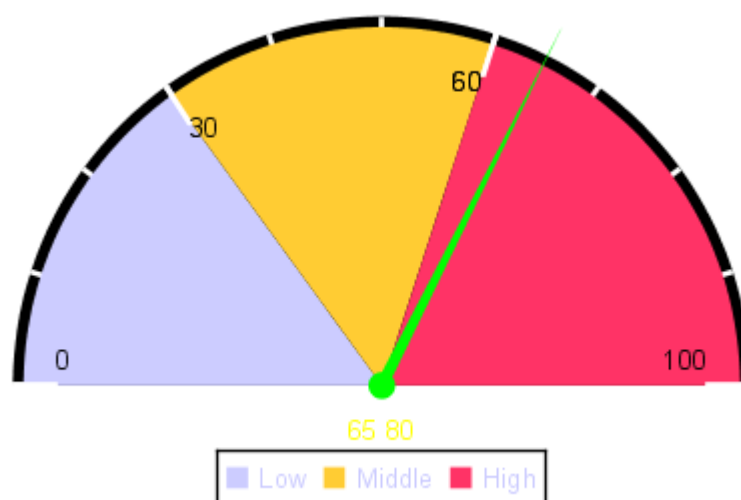
切片名称	最低值	最高值
Low	0	30
Middle	30	60
High	60	100



经过以上设置后,再设置仪表的数据值。选中图形,右键选择 Chart Data

项。此处设置的值为“new Integer(65)”。

仪表盘的显示效果如下：



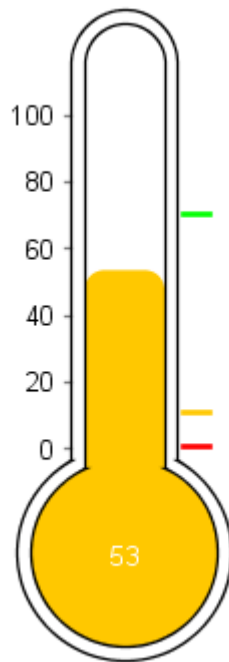
#### 4.3.10. 温度计(Thermometer)

温度计和仪表盘的设置类似，但不同的是仪表盘可以设置多个切片区域，而温度计只有固定的三个切片区域，分别是 Low、Medium、High。切片的设置如下图所示：

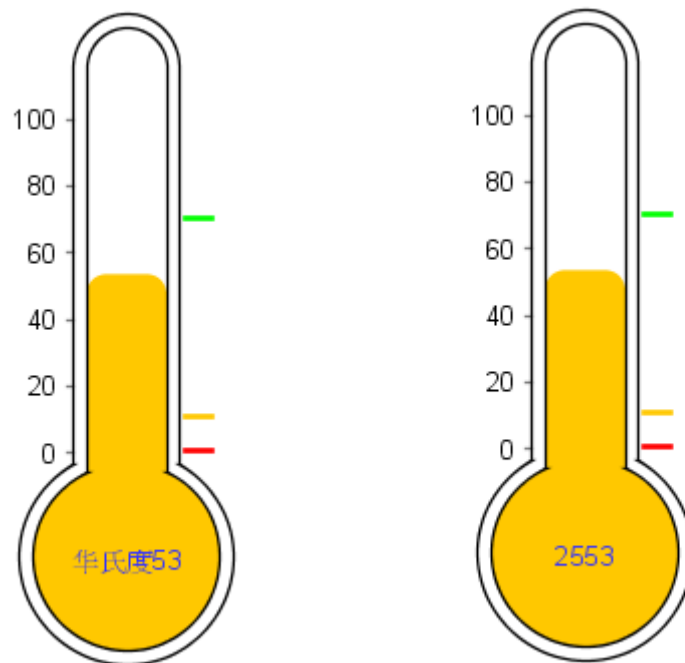
Thermometer Chart - 属性		
Mercury Color	null	...
Value Color	null	...
Mask		...
Value Font	sansserif 10	...
Data Range Low Expression	new Integer(0)	...
Data Range High Expression	new Integer(100)	...
Low Range Low Expression	new Integer(0)	...
Low Data Range High Expression	new Integer(10)	...
Medium Data Range Low Expression	new Integer(10)	...
Medium Data Range High Expression	new Integer(70)	...
High Data Range Low Expression	new Integer(70)	...
High Data Range High Expression	new Integer(100)	...

经过以上设置后，再设置温度计的数据值。选中图形，右键选择 Chart Data 项。此处设置的值为“new Integer(53)”。

温度计的显示效果如下：



Mask 属性 :在显示的数值前加的前缀 ,也就是上图中的圆形区域内显示的数值。  
Mask 属性值可以是数字 ,也可以是字符。如果是数字的话 ,后几位是 0 的话 ,  
0 都去掉。下图中 Mask 属性分别是 :华氏度 ,250。



#### 4.3.11. 气泡图(Bubble)

设置气泡图的 Chart Data ,设置的值有 4 项 :

Series expression : 分类名称，可是固定值，也可是表字段。

X value expression : X 轴值对应的表字段。

Y value expression : Y 轴值对应的表字段。

Z value expression。气泡大小值对应的表字段。

## 4.4. 报表导出

### 4.4.1. PDF 格式导出

#### ● PDF 中文问题

使用 PDF 格式导出数据时，需要用到 iText-2.1.0.jar 包，如果输出的 PDF 中有中文，则需要 iTextAsian.jar 包。

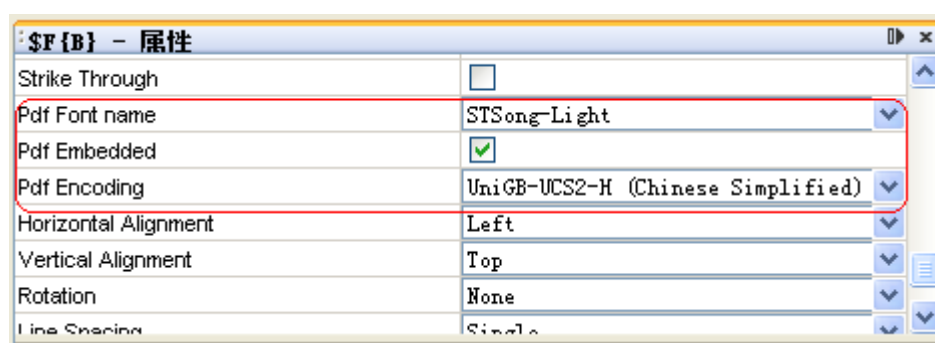
另外需要给输出中文的对象设置以下几个属性值：

Pdf Font name : STSong-Light

Pdf Embedded : 选中

Pdf Encoding : UniGB-UCS2-H (Chinese Simplified)

设置方法，鼠标左键点击文本显示对象，选中，然后在属性中找到 Pdf Encoding 等项，把其值改为上面说明的值。



#### ● 输出 PDF 的 JSP 程序

以下是一个完整的输出 PDF 的 JSP 程序：

```
<%@ page contentType="text/html; charset=GB2312"%>
<%@ page import="net.sf.jasperreports.engine.*"%>
<%@ page import="java.util.*"%>
<%@ page import="java.io.*" %>
<%@ page import="java.sql.*" %>
<%
    //报表编译之后生成的.jasper文件的存放位置
    File reportFile = new
```

```

File(this.getServletContext().getRealPath("test-pdf.jasper"));
    String url="jdbc:mysql://localhost:3306/cetvoss";
    Class.forName("com.mysql.jdbc.Driver");
    Map parameters = new HashMap();
    parameters.put("year", "2009");
    Connection conn = DriverManager.getConnection(url, "cetvoss",
"cetvoss");
    byte[]
bytes=JasperRunManager.runReportToPdf(reportFile.getPath(),parameters
,conn);
    response.setContentType("application/pdf");
    response.setContentLength(bytes.length);
    ServletOutputStream outStream = response.getOutputStream();
    outStream.write(bytes,0,bytes.length);
    outStream.flush();
    outStream.close();
    out.clear();
    out = pageContext.pushBody();
%>

```

#### 4.4.2. EXCEL 格式导出

使用 EXCEL 格式导出数据时 ,需要用到 poi-3.0.1-FINAL-20070705.jar 包。

##### ● 输出 EXCEL 的应用程序

TestExcel.java 完整程序如下：

```

package com.test;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.HashMap;
import java.util.Map;

import net.sf.jasperreports.engine.JRAbstractExporter;
import net.sf.jasperreports.engine.JRException;
import net.sf.jasperreports.engine.JRExporterParameter;
import net.sf.jasperreports.engine.JasperFillManager;

```



```

import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.export.JExcelApiExporter;

public class TestExcel {
    public static void main(String[] args) {
        File reportFile = new
File("D:\\proj\\JasperSample\\WebRoot\\test-excel.jasper");
        Map parameters = new HashMap();
        Connection conn = null;
        FileOutputStream output = null;
        try {
            String driver = "com.mysql.jdbc.Driver";
            String url="jdbc:mysql://localhost:3306/cetvoss";
            Class.forName(driver);
            conn = DriverManager.getConnection(url, "cetvoss",
"cetvoss");
            Map parameter = new HashMap();
            JasperPrint report = null;
            report =
JasperFillManager.fillReport(reportFile.getPath(), parameter,
conn);

            JAbstractExporter exporter = new JExcelApiExporter();
            output = new FileOutputStream("D:\\excel.xls");
            exporter.setParameter(JRExporterParameter.JASPER_PRINT,
report);

            exporter.setParameter(JRExporterParameter.OUTPUT_STREAM,
output);

            exporter.exportReport();
        }
        catch (Exception e) {
            e.printStackTrace();
        }
        finally {
            try {
                output.close();
            }
            catch (Exception e) {
                e.printStackTrace();
            }
            try {
                conn.close();
            }
            catch (Exception e) {

```

```
        e.printStackTrace();
    }
}
}
```

#### 4.5. 调用 JAVA 代码

报表中可调用我们自己编写的 JAVA 程序输出数据。步骤如下：

(1) 使用任意工具编写 MyScriptLet.java 程序，并编译：

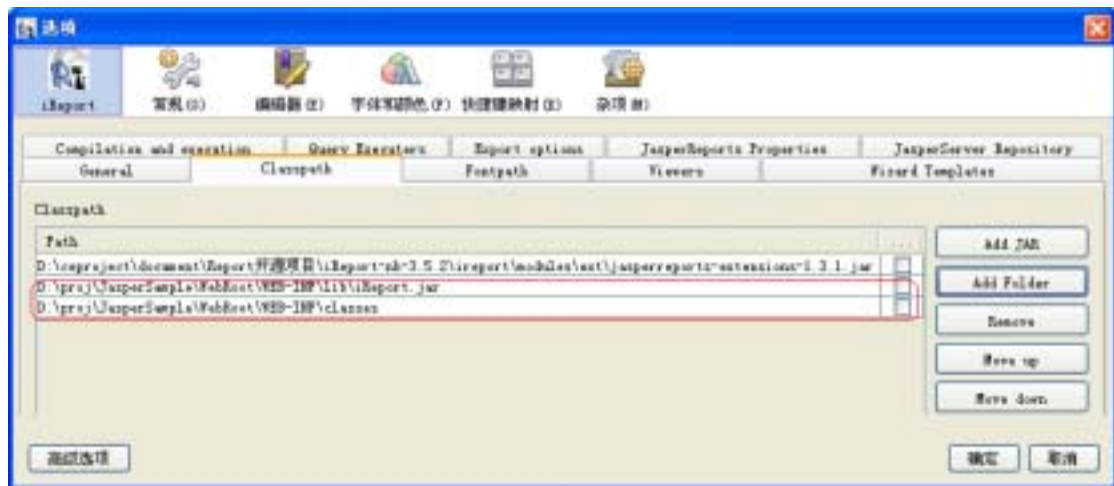
```
package com.test;
public class MyScriptLet extends
it.businesslogic.ireport.IReportScriptlet {

    public String getResult() {
        return "Hello my first ScriptLet example.";
    }
}
```

注意：IReportScriptlet 类在 iReport.jar 包中。编译时需要引用此类。

(2) 引用 iReport.jar 和 MyScriptLet.class 到 iReport 环境中

选择菜单“工具”->“选项”，再选择“Classpath”选项卡。然后使用“Add JAR”和“Add Folder”分别把 iReport.jar 和 MyScriptLet.class 添加到 iReport 环境中。本例子中的 MyScriptLet.class 完全目录：  
D:\proj\JasperSample\WebRoot\WEB-INF\classes\com\test\MyScriptLet.class



### (3) 设置报表属性 Scriptlet class 项

鼠标左键点击报表边缘的空白处，然后到属性面板中找到 Scriptlet class 项，并把值设置成：`com.test.MyScriptLet`，注意不能写成：`com.test.MyScriptLet.class`，或写成：`MyScriptLet`。

### (4) 使用 MyScriptLet 类

从组件面板中拖拽一个“Text Field”组件到报表中，并把值设置成：

```
((com.test.MyScriptLet)$P{REPORT_SCRIPTLET}).getResult()
```

注意把 MyScriptLet 类的包名写全，应写成“`com.test.MyScriptLet`”，如果写成“`MyScriptLet`”，则编译时报错。

### (5) 编译运行显示效果：

1	苹果	null	Hello my first ScriptLet example.
1	菠萝	null	Hello my first ScriptLet example.
1	西瓜	null	Hello my first ScriptLet example.
2	苹果	null	Hello my first ScriptLet example.
2	菠萝	null	Hello my first ScriptLet example.
2	西瓜	null	Hello my first ScriptLet example.
3	苹果	null	Hello my first ScriptLet example.
3	菠萝	null	Hello my first ScriptLet example.
3	西瓜	null	Hello my first ScriptLet example.

## 5. 技巧篇

### 5.1. 在 WEB 中显示数据不分页显示

置报表属性 Ignore pagination 项为选中状态。

### 5.2. 没有数据时仍然显示标题

默认的情况下，如果没有数据，系统不显示报表的标题、表头等信息。如果需要显示，则把报表的“ When No Data Type ”项属性值设置成 :All Sections, No Detail。

### 5.3. 数字型字段值到字符的转换

`String.valueOf($F{year})`