**MesReport开发手册**

**北明智通（北京）科技有限公司**

**2014-04-03**

**目 录**

[1 Birt介绍 5](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341370)

**[1.1](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341371)****[报表介绍](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341371)** [5](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341371)

**[1.2](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341372)****[基本概念](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341372)** [5](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341372)

**[1.3](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341373)****[丰富的报表](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341373)** [5](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341373)

**[1.4](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341374)****[BIRT报表由四部分组成](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341374)** [6](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341374)

**[1.5](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341375)****[报表设计器](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341375)** [6](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341375)

[2 Birt插件安装 8](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341376)

**[2.1](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341377)****[插件版本说明](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341377)** [8](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341377)

**[2.2](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341378)****[插件安装](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341378)** [9](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341378)

[2.2.1 完全安装 9](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341379)

[2.2.2 更新安装： 9](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341380)

[2.2.3 分步安装 9](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341381)

[3 Birt报表开发实战 13](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341382)

**[3.1](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341383)****[创建报表](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341383)** [13](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341383)

[3.1.1 创建报表工程 13](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341384)

[3.1.2 创建报表 14](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341385)

**[3.2](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341386)****[数据源配置](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341386)** [16](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341386)

**[3.3](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341387)****[创建数据集](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341387)** [19](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341387)

**[3.4](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341388)****[网格报表](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341388)** [20](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341388)

[3.4.1 创建数据源 21](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341389)

[3.4.2 创建数据集 21](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341390)

[3.4.3 对报表进行布局 21](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341391)

[3.4.4 格式化与预览 23](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341392)

**[3.5](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341393)****[分组报表](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341393)** [24](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341393)

[3.5.1 创建数据源 25](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341394)

[3.5.2 构建数据集 25](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341395)

[3.5.3 对报表进行布局 25](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341396)

[3.5.4 设定分组数据 25](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341397)

[3.5.5 格式化报表 27](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341398)

**[3.6](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341399)****[交叉报表](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341399)** [28](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341399)

[3.6.1 构建数据源 28](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341400)

[3.6.2 构建数据集 28](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341401)

[3.6.3 在报表中加入交叉报表 28](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341402)

[3.6.4 数据准备 29](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341403)

[3.6.5 向交叉表插入数据 32](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341404)

[3.6.6 加入汇总 33](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341405)

[3.6.7 格式化报表 34](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341406)

**[3.7](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341407)****[嵌套报表](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341407)** [35](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341407)

[3.7.1 构建数据源 36](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341408)

[3.7.2 构建数据集 36](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341409)

[3.7.3 布局嵌套报表 38](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341410)

[3.7.4 将Table绑定数据集 39](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341411)

[3.7.5 设定主表数据 39](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341412)

[3.7.6 设定子表数据 40](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341413)

**[3.8](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341414)****[交互报表](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341414)** [42](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341414)

[3.8.1 创建报表、数据源 43](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341415)

[3.8.2 创建数据集 43](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341416)

[3.8.3 加入响应参数 45](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341417)

[3.8.4 布局报表 47](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341418)

[3.8.5 插入报表数据 48](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341419)

[3.8.6 格式化与预览 49](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341420)

**[3.9](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341421)****[报表钻取](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341421)** [51](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341421)

[3.9.1 设定响应参数 51](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341422)

[3.9.2 预览结果 54](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341423)

**[3.10](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341424)****[图表报表](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341424)** [55](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341424)

[3.10.1 为报表插入图表 55](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341425)

[3.10.2 设定图表属性 55](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341426)

[3.10.3 图表预览 60](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341427)

**[3.11](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341428)****[多层交叉报表](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341428)** [61](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341428)

[3.11.1 构建数据源 61](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341429)

[3.11.2 构建数据集 61](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341430)

[3.11.3 在报表中加入交叉报表 62](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341431)

[3.11.4 数据准备 63](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341432)

[3.11.5 向交叉表插入数据 63](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341433)

[3.11.6 格式化与预览 64](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341434)

**[3.12](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341435)****[多层交叉汇总报表](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341435)** [65](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341435)

[3.12.1 加入汇总 65](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341436)

[3.12.2 格式化与预览 66](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341437)

**[3.13](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341438)****[其他](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341438)** [67](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341438)

[3.13.1 格式化报表 67](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341439)

[3.13.1.1 表头与标题 67](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341440)

[3.13.1.2 设定单元格格式 68](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341441)

[3.13.1.3 样式套用 68](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341442)

[3.13.1.4 数据元素格式化 69](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341443)

[3.13.2 分页、页眉与页脚 70](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341444)

[3.13.2.1 设置分页显示 70](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341445)

[3.13.2.2 设置分页打印 70](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341446)

[3.13.2.3 设置页眉页脚 71](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341447)

[3.13.3 隔行高亮度显示 73](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341448)

[3.13.4 对报告进行布局 76](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341449)

[3.13.4.1 添加|删除行或列 76](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341450)

[3.13.4.2 指定行高或列宽 76](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341451)

[3.13.4.3 添加图像 77](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341452)

[3.13.5 数据绑定 79](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341453)

[4 Birt报表script 82](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341454)

**[4.1](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341455)****[avascript脚本编写](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341455)** [82](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341455)

[4.1.1 隔行高亮度显示 82](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341456)

[4.1.2 多参数查询 84](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341457)

[4.1.3 脚本创建数据源 85](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341458)

**[4.2](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341459)****[java脚本编写](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341459)** [88](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341459)

[4.2.1 创建java类 88](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341460)

[4.2.2 java类与报表交互 92](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341461)

**[4.3](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341462)****[BIRT脚本参考](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341462)** [98](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341462)

[5 Birt报表部署 99](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341463)

**[5.1](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341464)****[直接使用运行birt-runtime-2\_2\_0包中的WebViewerExample](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341464)** [99](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341464)

**[5.2](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341465)****[集成到已有web应用](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341465)** [100](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341465)

**[5.3](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341466)****[数据库驱动位置](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341466)** [107](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341466)

**[5.4](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341467)****[报表url说明](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341467)** [107](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341467)

**[5.5](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341468)****[web.xml配置](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341468)** [113](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341468)

**[5.6](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341469)****[viewer.properties说明](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341469)** [115](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341469)

**[5.7](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341470)****[BIRT在jsp中tag使用](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341470)** [116](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341470)

**[5.8](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341471)****[关于BIRT编码](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341471)** [123](file:///C:\\Users\\Administrator\\Desktop\\MesReport开发手册.doc" \l "_Toc176341471)

# MesReport介绍

## **报表介绍**

MesReport采用开源的BIRT作为引擎构建。

BIRT――商业智能和报表工具。

BIRT是一个Eclipse-based开放源代码的报表系统，它主要是用在基于Java和J2ee的web应用程序上。BIRT主要由两部分组成：一个是基于Eclipse的报表设计器和一个可以添加到应用服务器的运行组建。BIRT同时提供一个图形报表制作引擎。

BIRT拥有和Dreamweaver一般的操作界面，可以像画table一样画报表，也可以生成图片、导出Excel、html、pdf、ppt等分页，样式比script设置简单，另外BIRT还有OLAP导航功能。

## **基本概念**

* 数据源：数据的来源，或提供者。如xml数据源、jdbc数据源等。
* 数据集：数据集合，它必须与数据源关联，可以理解为查询的结果。
* 报表以及报表项，报表可视为是针对一组数据集的表现形式，而报表项这是这个表现形式的某个具体的单元。它们之间的关系，与窗体和控件的关系非常类似。报表、数据集、数据源三者间的关系：数据源 --- 数据集 --- 报表。
* 报表参数：查询参数的表现形式，使用它可以构建更灵活的报表。
* 模板和库：主要用于复用报表设计，提高报表开发的效率

## **丰富的报表**

列表 － 列表是最简单的报表。当列表变长时，你可以把相关数据增加到同一分组（基于客户的订单分组，基于供应商的产品分组）。如果数据是数字类型的，你可以轻松的添加到“总数”、“平均”、或其他汇总中。

图表 － 当需要图表表现时，数字型数据比较好理解。BIRT 也提供饼状、线状以及柱状图标等。

交叉表 － 交叉表（也叫做十字表格或矩阵）用两种维度展示数据：sales per quarter or hits per web page。（交叉表在 1.0.1 中没有提供，但计划在将来提供。）

信函和文档 － 通知、信件、以及其他文本文档都很容易通过 BIRT 方便建立。文档包括正文、格式、列表、图表等。

混合报表 － 很多报表需要联合以上所有的报表构成单一文档。例如，一份客户声明会列出客户所需要的信息，为当前促进(promotions)提供文本，以及提供并行的出帐和入帐列表。一份财政报表将包括声明、图表、表格，所有这些都将进行全方位的格式化，来匹配共有的配色方案。

## **报表由四部分组成**

**数据** － 数据库、Web 服务、Java 对象，这些都可以作为 BIRT 报表源。1.0.1 版本提供 JDBC 支持，也支持利用编码来获取其他地方的数据。BIRT 的 ODA(Open Data Access) 框架允许任何人构建新的 UI 以及运行时支持任何类型的表格式数据。未来，单一报表可包含从任意多个数据源获取数据。

**数据转换** － 报表通过对数据的分类、统计、过滤以及分组来适应用户需求。当然，数据库能实现这些功能，当遇到普通文件和 Java 对象时 BIRT 必须以 "simple" 数据源方式处理。BIRT 允许复杂的操作，比如总合分组、整体共计的百分比，等等。

**业务逻辑** － 真实世界的数据很少提供你理想的结构良好的报表。许多报表要求用具体逻辑把原始数据转换成用户的有用信息。如果该逻辑仅仅用于该报表，你可以用 BIRT 的 JavaScript 脚本支持。如果你的程序中已包含这些逻辑，你可以调用已有的 Java 代码。

**表现** － 一旦数据准备好了，你可以在很大的范围内选择表现形式。表格、图表、文字等等都可以。单一数据集可以有多种方式表现，而单一报表可以表现多个数据集。

## **报表设计器**

**数据浏览器** － 把你的数据源（连接）以及数据集（查询）组织起来。数据集编辑器允许你测试数据集，以确保报表接收数据的正确性。

**布局视图（Layout）** － 所见即所得编辑器为你的报表提供以拽方式来创建表现内容。包含一个标准报表条目调色板。

**属性编辑器（Property Editor）** － 以便利的格局表现大多数通用的用户属性使编辑更快速和容易。BIRT 也集成了标准 Eclipse 属性视图，为每个条目提供详细的属性列表。

**报表预览（Preview）** － 你可以在任何时间采用真实数据测试你的报表。预览窗口直接内嵌在 Eclipse 中。

**代码编辑器(Script)** － 在访问数据以及报表生成或浏览时，脚本把业务逻辑添加给报表。在编辑脚本时代码编辑器提供标准的 Eclipse 特性：语法加色、自动完成等等。BIRT 用很简单的脚本来表达，expression builder 能更容易的创建这些表达。

**略图（OutLine）** － BIRT 报表被组织为一个树型结构作为整体报表的根，并且为样式、报表内容、数据源、数据集、报表参数等分类。略图视图提供你整个报表结构紧凑的预览。

**Cheat Sheets** － 学习新工具永远是种挑战，但是 Eclipse 提供一种创新方案：cheat sheets。它们是一些帮助你完成新任务的文档。

# Birt插件安装

## **插件版本说明**

EMF, GEF 和 BIRT 都是需要通过 Eclipse download page下载的。请注意以下的配置要求：

* **BIRT**
* Eclipse
* GEF
* EMF
* JRE/JDK

GEF：Graphical Editing Framework，图形编辑框架，是一个通过 BIRT UI 使用的Eclipse 插件。

EMF：Eclipse Modeling Framework，用来制作图表的框架。

DTP: DataTools Platform，管理数据源驱动程序，以及访问特殊数据源实例的配置。

WTP: Web Tools Platform，

另外，本手册中介绍实例使用的derby数据库。

## **插件安装**

### 完全安装

到Eclipse的官方网站<http://download.eclipse.org/birt/downloads>去下载(注意现在使用的是4.5版本) 带有BIRT 插件的Eclipse平台（birt-report-designer-all-in-one-5\_5\_0），解压后，它就是一个拥有 BIRT插件的Eclipse，您可以直接使用(完整版只有32位的windows版本)。

下载所在网页：  
<http://www.eclipse.org/downloads/download.php?file=/birt/downloads/drops/R-R1-4_5_0-201506092134/birt-report-designer-all-in-one-4.5.0-20150619-win32-win32-x86.zip>



### 更新安装：

BIRT 2.1开始，可以在 Eclipse 里直接下载。在Eclipse里按照：[Help]—>[Software Updates]—>[Find And Install]的路径来进行即可。

### 分步安装

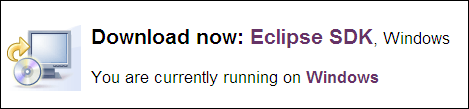
首先下载安装JRE 1.7；

1. **下载安装Eclipse 4.5**

下载地址：

打开网页：

点击下图中的“Eclipse SDK”链接即可下载。

[](http://download.eclipse.org/eclipse/downloads/drops/R-3.3-200706251500/winPlatform.php#EclipseSDK)

下载到文件eclipse-SDK-3.3-win32.zip，解压至安装目录，比如我们把它安装到C:\Program Files中；

1. **下载安装插件WTP 2.0，**

下载地址：

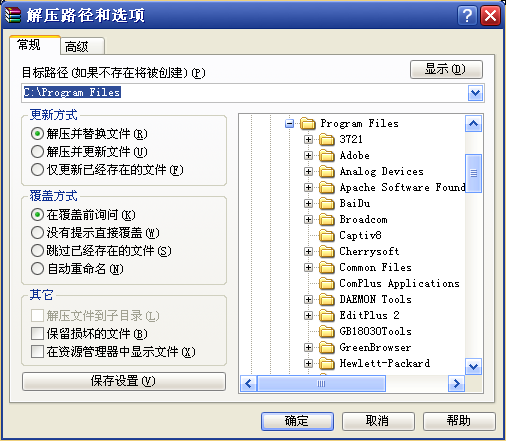
打开下面的网页点击网页中Download From后的链接即可进行下载。

下载到文件wtp-R-2.0.zip，将其解压至eclipse安装目录C:\Program Files即可；

方法如下：右键点击压缩文件，选择解压文件，



弹出解压路径和选项：



将目标路径改为我们安装eclipse的安装目录：C:\Program Files

点击确定。在解压时，如果出现“文件已存在”，选择“全部选是”进行覆盖就可以了。

1. **下载安装GEF 3.3 Release, GEF Runtime，**

下载地址：

打开网页：

点击：Runtime中Download下方的链接即可。



下载到文件GEF-runtime-3.3.zip，解压至安装目录，方法同上。

1. **下载安装EMF & SDO RT version 2.3，**

下载地址：

打开下面的网页点击网页中Download From后的链接即可进行下载。

下载到文件emf-sdo-SDK.zip，解压至安装目录，方法同上。

1. **下载安装DTP version 1.5，**

下载地址：

打开下面的网页点击网页中Download From后的链接即可进行下载。

下载到文件dtp-sdk\_1.5.1.zip，解压至安装目录，方法同上。

1. **下载安装BIRT Release 2.2，**

下载地址：

打开下面的网页点击网页中Download From后的链接即可进行下载。

下载到文件birt-report-framework-2\_2\_0.zip，解压至安装目录，方法同上。

安装完毕，即可进入使用。

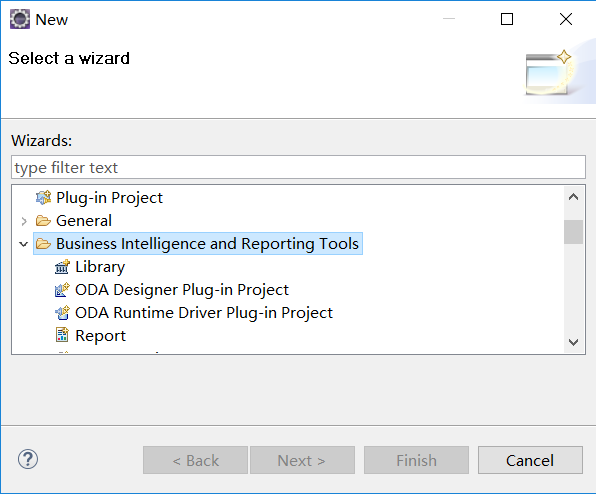
如果您只是缺少其中的某几个插件，下载后按照上述方法安装即可，但要注意版本的匹配。

# Birt报表开发实战

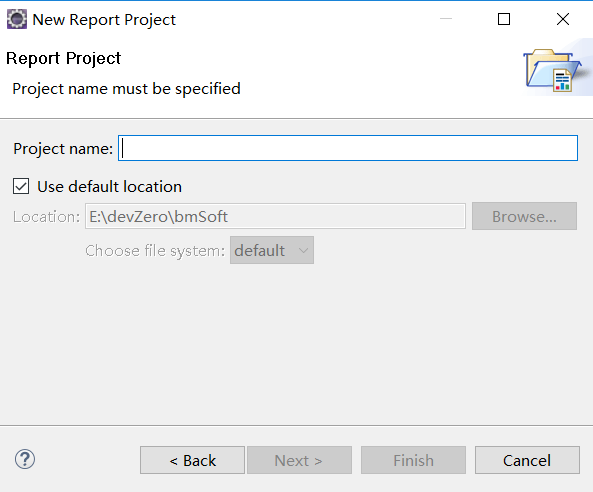
## **创建报表**

### 创建报表工程

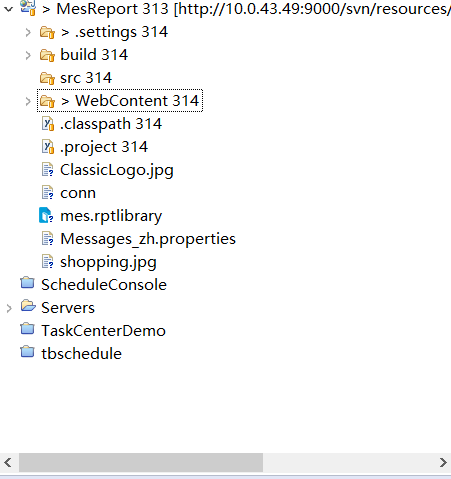
1. 选择文件->新建->项目。如下图中的新建项目将显示可以创建的项目的类型。



1. 展开商业智能和报表工具（Business Intelligence and Reporting Tools），选择报表项目（Report Project），然后选择下一步。
2. 在新建报表项目的工程名称中输入下列文本，如图所示：

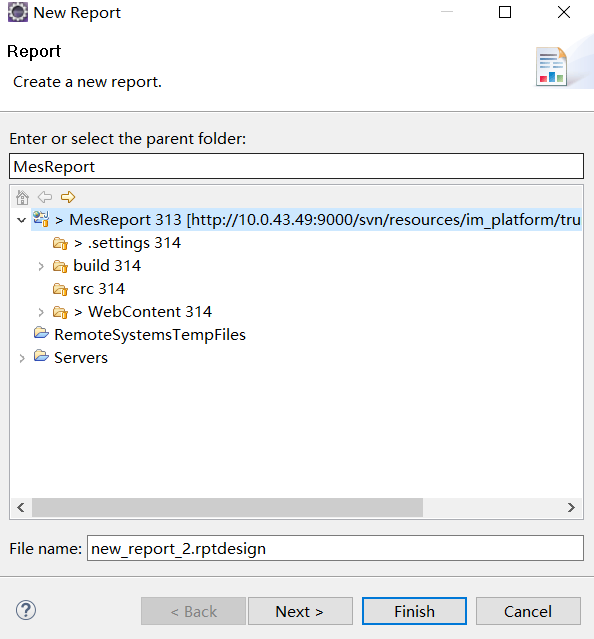


1. 要添加项目，请选择完成。现在可在导航器视图中看到该项目，如图所示。

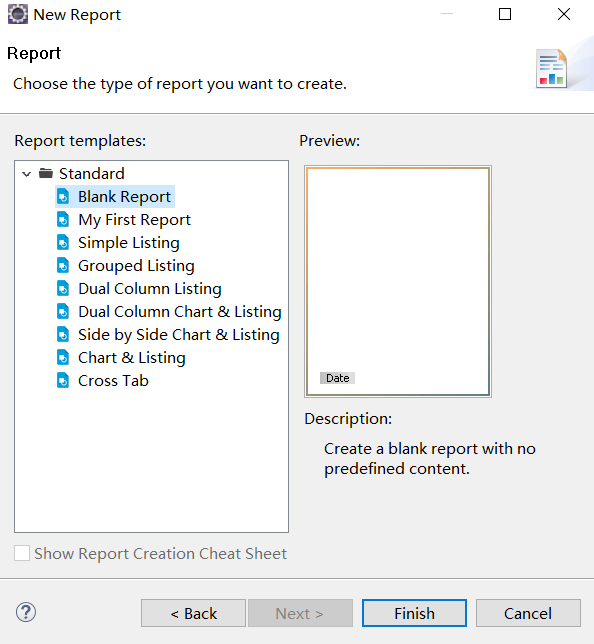


### 创建报表

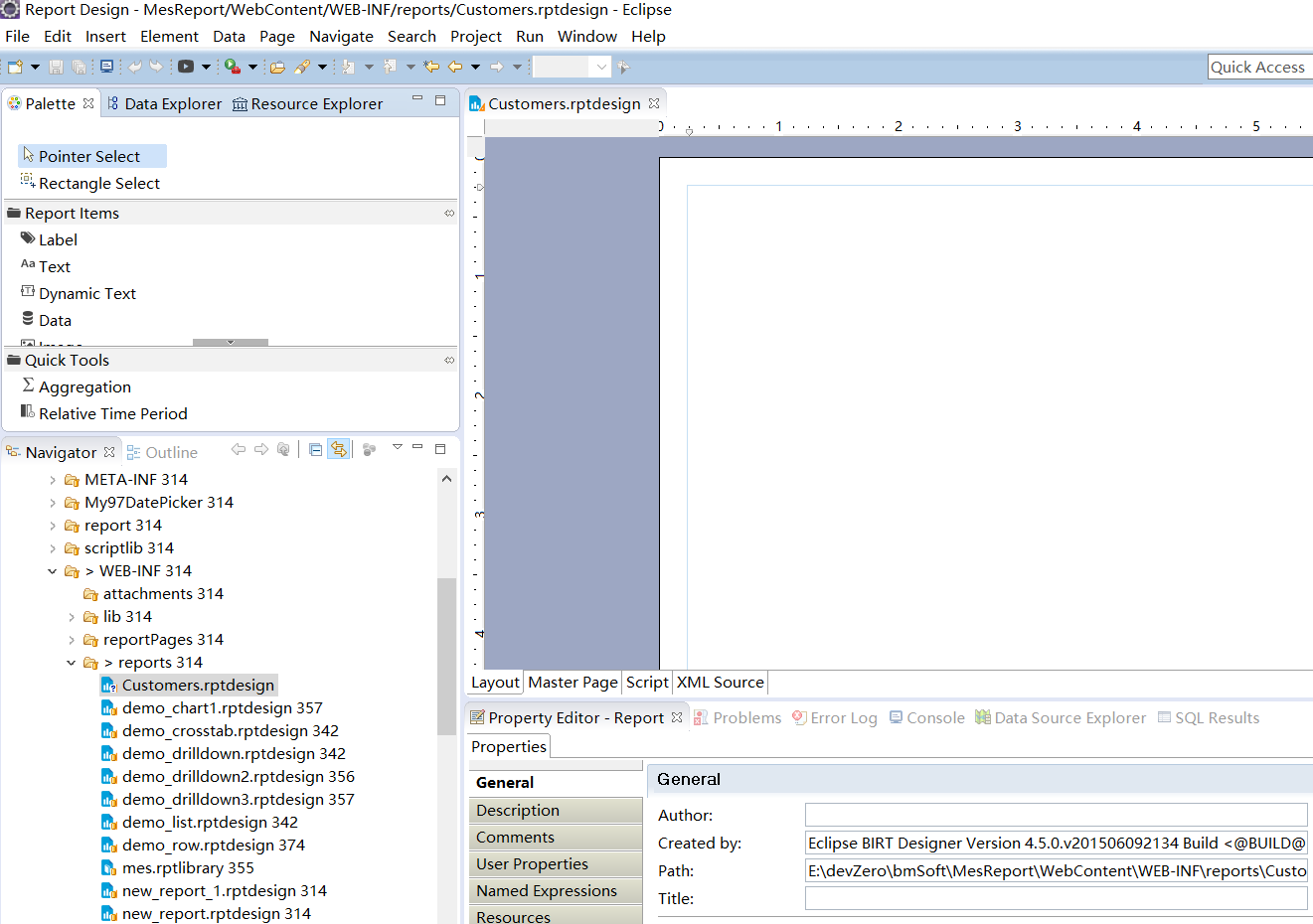
1. 选择文件->新建->报表。如下图：



1. 选择MesReport，输入报表文件名称，如：Customers.rptdesign
2. 选择Next。
3. 可以通过一下方式创建报表：

* 从空白报表开始设计
* 使用预定义的报表模版

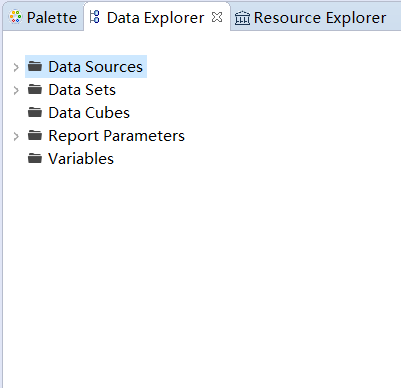
1. 选择Finish。新报表将出现在主窗口中，布局编辑器将显示空的报表页。如下图所示：



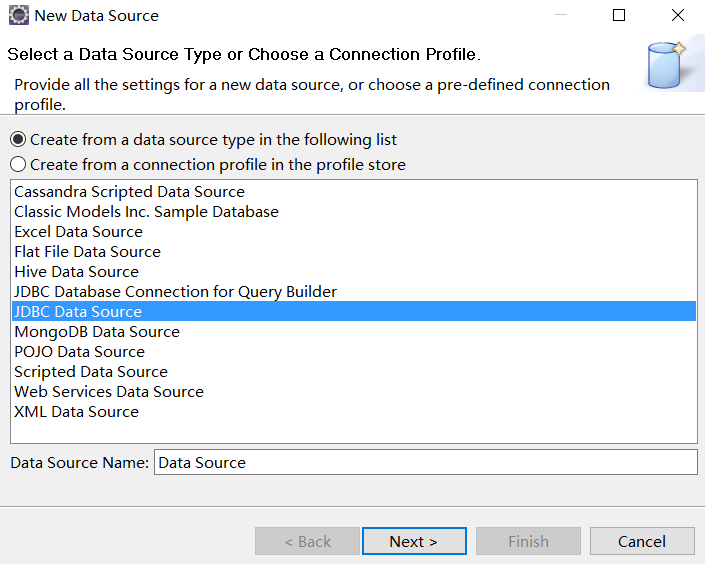
## **数据源配置**

在布局编辑器中开始设计报表之前，构建 BIRT 数据源以将报表连接至数据库或其他类型的数据源。构建数据源时，要指定驱动程序类、数据源名称和其他连接信息（例如，用户名和密码）。从mysql为演示

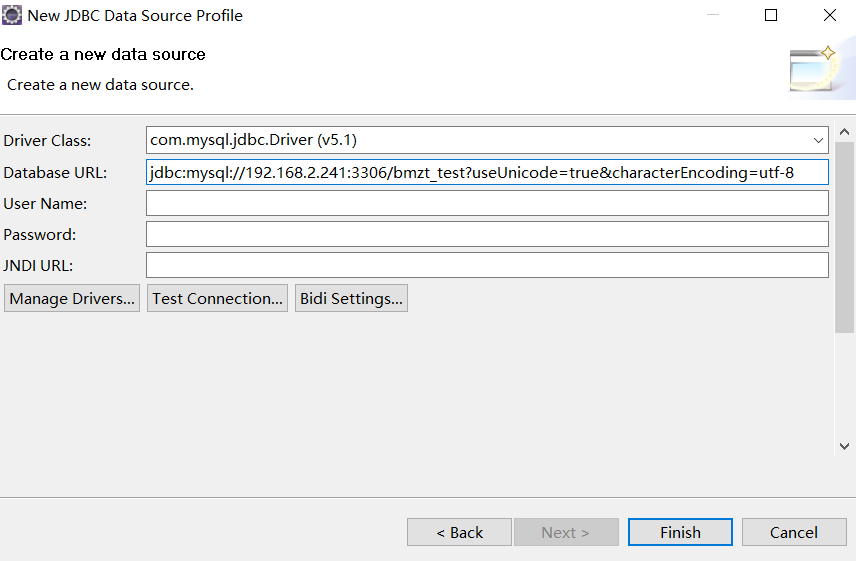
* + - 1. 选择数据资源管理器(Data Explorer)。如果使用缺省报表设计透视图，则数据资源管理器位于布局编辑器的左边，在选用板（Palette）的旁边，如图所示。如果它尚未打开，则选择窗口->显示视图->数据资源管理器。

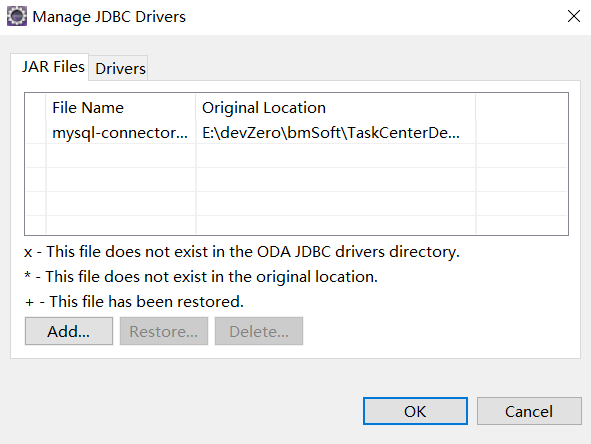


* + - 1. 右键单击Data Sources然后从上下文菜单中选择新建数据源。新建数据源显示可以创建的数据源的类型，如图所示。



* + - * + Classic Models Inc.Sample DataBase ———上面说过是BIRT样本数据库。
        + Flat File Data Source ———从CSV、SSV、TSV、PSV四种格式的文件获取数据源。
        + JDBC Data Source ——— 通过配置jdbc连接数据库。
        + Script Data Source ——— 通过编写脚本获取数据源。
        + Web Services Data Source ——— 通过web service方式获取数据源。
        + XML Data Source ——— 从xml文件获取数据源。

* + - 1. 这里我们介绍通过JDBC Data Source 配置数据源。选择JDBC Data Source ，输入数据源名称，点击Next,配置Jdbc信息。如下图：
* 首先，单击Manage Drivers添加驱动包。如下图：



单击add，选择derby数据库的derbyclient.jar驱动包。点击ok，驱动完成添加。

* 配置信息：

Driver Class: org.apache.derby.jdbc.ClientDriver (v10.1)(通过下拉菜单可以找到)

Database URL: jdbc:derby://【host】:1527/【database】

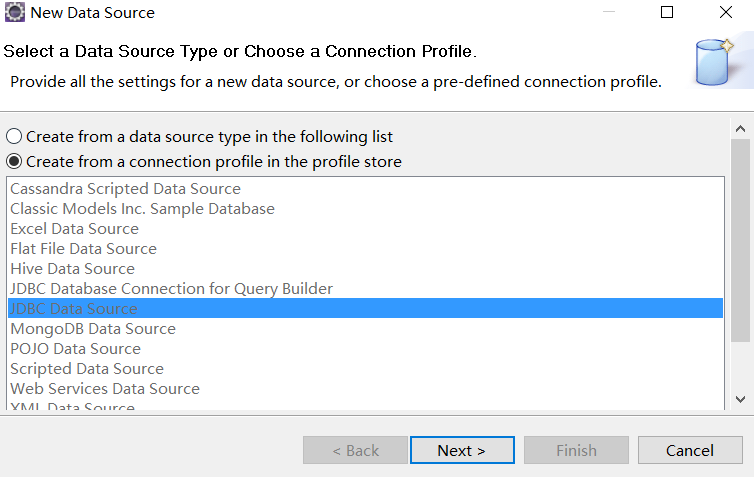
User Name：用户名

Password：密码

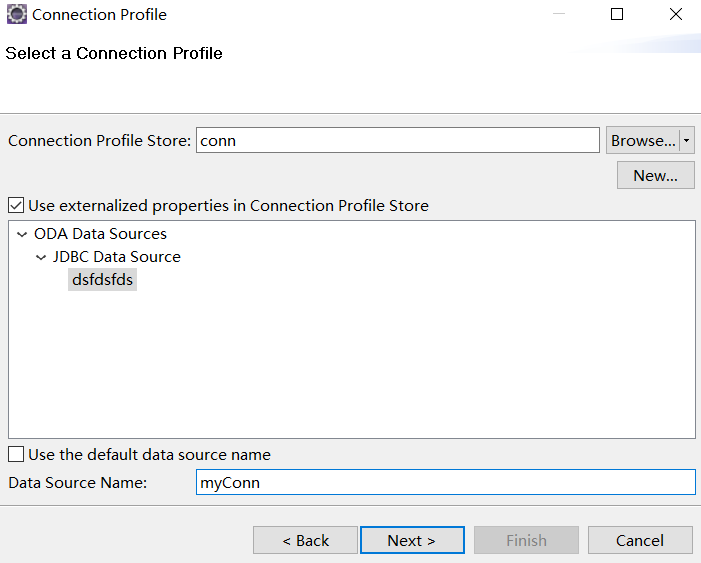
* + - 1. 测试连接，测试前查看derby数据库是否启动；点击Test Connection。出现Connection successful.对话框表示数据源创建成功。点击Finish完成。

## **使用配置文件创建数据源**

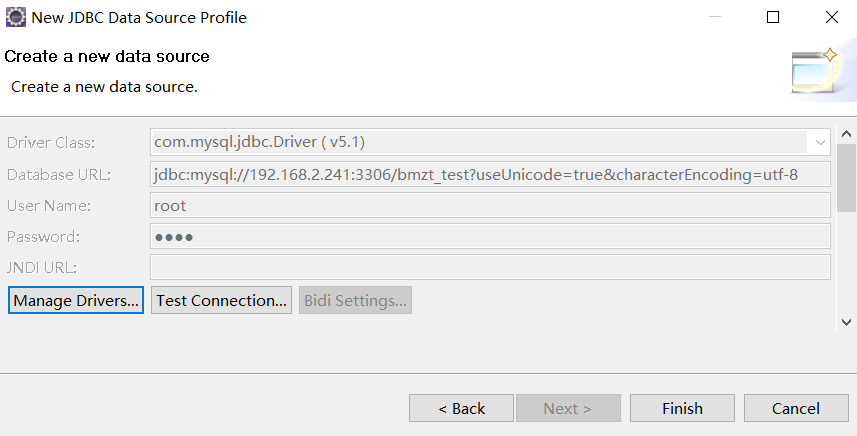
每个报表若单独设置数据源，不利于项目实现情况，遇到用户名密码时，所有文件都要修改，现在介绍通过连接配置文件生成数据源，首先新建数据源



然后下一步，点击浏览，选择已有的连接文件，若没有，点击新建



然后下一步

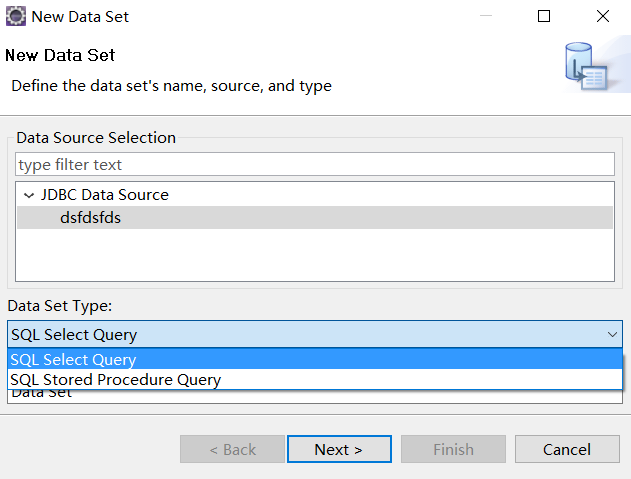


后面操作方法和3.2一致

## **创建数据集**

数据集标识要从数据源检索的数据。我们是通过连接至 JDBC 数据源，则使用 SQL SELECT 语句来标识要检索的数据。

1. 在数据资源管理器（Data Explorer）中，右键单击数据集（Data Set）并从上下文菜单中选择新建数据集。
2. 在新建数据集的数据集名称中，输入下列文本，如图所示：



1. 对于其他字段选择缺省值。

Data Set Types 数据集类型一般使用jdabc数据源连接分为两种：一种是SQL Select Query为select查询，一种是SQL Stored Procedure Query存储过程查询（使用存储过程查询，必须数据库中已创建存储过程）。

Data Source显示先前创建的数据源的名称。

Data Set Types 指示该数据集使用 SQL Select Query查询。

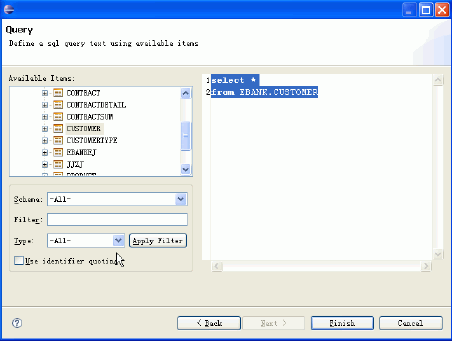
1. 选择Next。

查询显示可帮助您创建 SQL 查询的信息。可用项列示数据库中的所有表。可以单击表旁边的加号（+）标志来显示该表的各个列。编辑数据集右边的文本区域显示 SQL SELECT 语句的必需关键字：

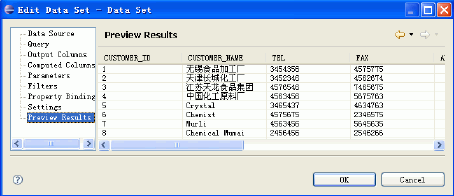
1. 在文本区域中，输入下列 SQL SELECT 语句以指定要检索的数据：

如：**select \*** **from** EBANK.CUSTOMER

虽然数据集编辑器以大写字母显示表和列名称，但可用您喜欢的方式输入这些名称，因为 SQL 不区分大小写。如果您不想输入查询，则可将列和表从可用项拖动到文本区域。



1. 选择Finish以保存数据集。编辑数据集将显示在查询中指定的列，并提供编辑数据集的选项。
2. 选择预览结果（Preview Results）以确保查询有效并且它返回正确的数据。如果正确输入了 SELECT 语句，您应该看到图中显示的结果。这些是查询返回的数据行。



1. 选择ok，数据集创建成功。

## **网格报表**

网格报表是报表最通用的模式，在大多数报表中，都需要或部分需要将一组（多条）数据结构完全相同的数据以表格形式顺序展现出来。如：



通过此例我们先来看一看如何制作一张简单的报表。我们需要建立一张显示所有银行客户及其相关信息的的报表。

### 创建数据源

方法同上例3.2.

### 创建数据集

方法同上例3.3. 数据集名称为：Customer。

客户的信息存在CUSTOMER表中，我们需要输入如下语句：

**select** EBANK.CUSTOMERTYPE.TYPENAME,EBANK.CUSTOMER.\*

**from** EBANK.CUSTOMER,EBANK.CUSTOMERTYPE

**where** EBANK.CUSTOMER.CUSTOMERTYPE **=** EBANK.CUSTOMERTYPE.TYPEID

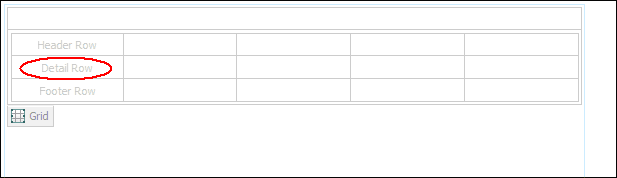
### 对报表进行布局

**1．**在本过程中，您在报告页中插入元素以显示先前创建的数据集的数据。先插入一个表元素，然后在该表中插入数据元素。

**2．**选择选用板“Palette”。选用板显示所有可布置在报告中的元素。

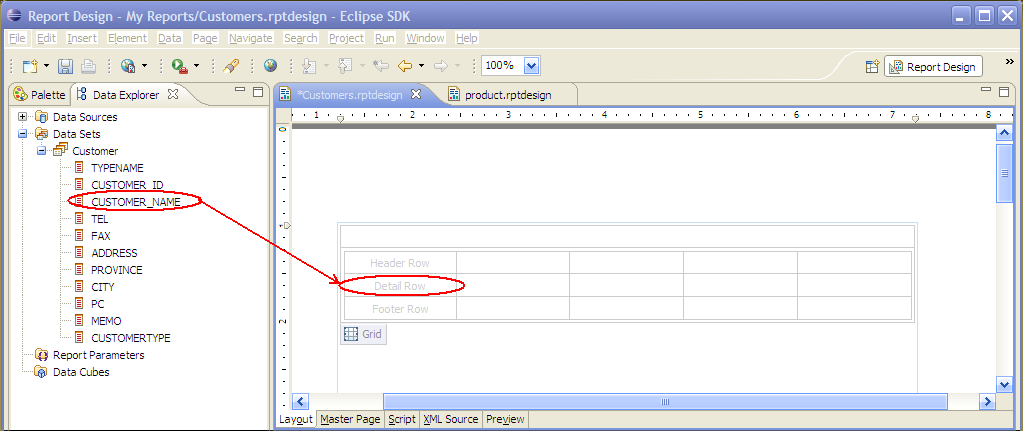
**3．**为了以后的表布局美观，我们先从选用板中的“Gird”元素拖拽放入布局编辑器内（即文件的“Layout”视图）的报告中，插入时提示您要制定创建的行列数的详细信息：输入1列2行，

**4．**然后，将选用板中的表元素“Table”拖放到布局编辑器内的报告中。插入表提示您指定要对此表创建的列数和详细信息行数。指定5列和1个详细信息行，然后选择“确定”。具有5列和1个详细信息行的表将出现在布局编辑器中。现在，您就可以在该表中插入数据了。

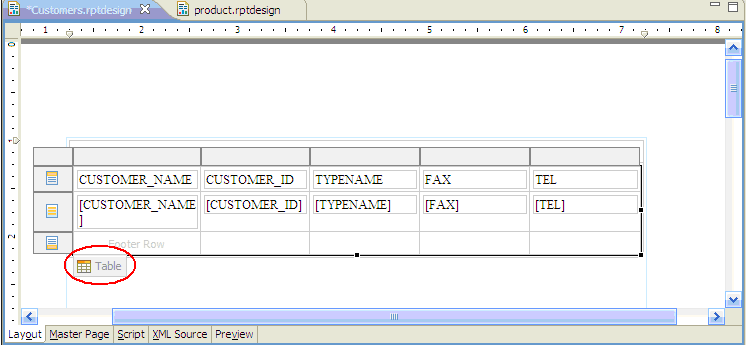


**5．**选择数据资源管理器Data Explorer视图。在Data Explorer视图中，展开“Data Sets”，然后展开Customers，在查询中指定的列将出现在Customers下面。

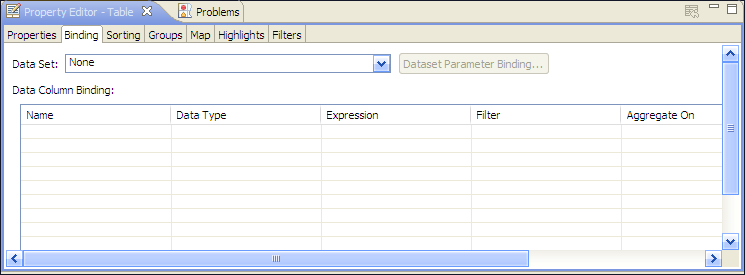
将CUSTOMER\_NAME从Data Explorer中拖放到表的详细信息行“Detail Row”中的第一个单元格内（如下图）。详细信息行显示报告中的主要数据。在完成的报告中，详细信息行重复显示数据集中的所有数据行。



**6．**然后采用同样的方法将CUSTOMER\_ID，TYPENAME，FAX，TEL这四个数据项从Data Explorer中拖放到表的详细信息行中，完成的报告中，详细信息行重复显示数据集中的所有数据行。如下图：



**7．**选中报表格左下角的“”（如上图），在下方的视图中会显示本Table的信息：



选择第二项“binding”，将Data Set的下拉框选为“Customer”，下方的Data Colum Binding中就会显示我们数据集中的数据项，通过此步我们将本报表与数据集绑定起来，否则将不能显示（一般将数据集中的数据项拖拽如Table中后，会自动将次数据集的数据项与本报表绑定，我们可以不用进行这步操作。但我们必须得清楚这个操作）。

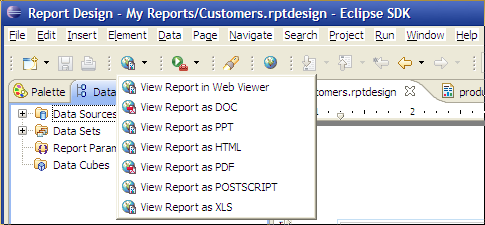
完成后，表格的基本设置就算做完了。

### 格式化与预览

通过布局编辑器中选择“Preview”可对我们设计的报表进行预览。

我们通过一系列格式化操作，将报表进行美化。

完成以后则可，通过工具栏中提供的各种不同的视图模式来进行预览，如下：



使用默认的View Report In Web Viewer预览结果如下：



## **分组报表**

在网格报表的基础上，我们可能还需要对数据进行进一步的处理。比如，将同类数据显示在一个组中，并对这个组进行统计运算。如下图所示，对数据按省份进行分组，并将同一省份数据放在一个组内进行汇总统计。



下面我们将上例的网格报表进行一下小的改进，建立一个按用户类型进行分组的报表。

### 创建数据源

方法同上例。

### 构建数据集

本例所用的数据集与上例完全相同，输入与上例3.4.2相同的sql语句建立数据集即可。

### 对报表进行布局

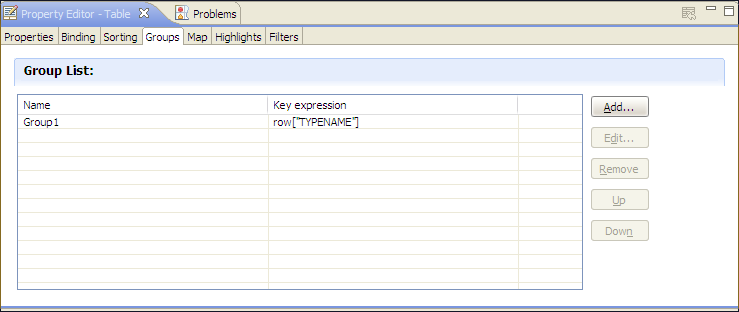
**1．**做好上面的工作之后，先在布局编辑器中建立一个1列2行的Grid，

**2．**同上例一样，从选用板“Palette”中拖拽一个5列1行的Table至第二行的网格中，

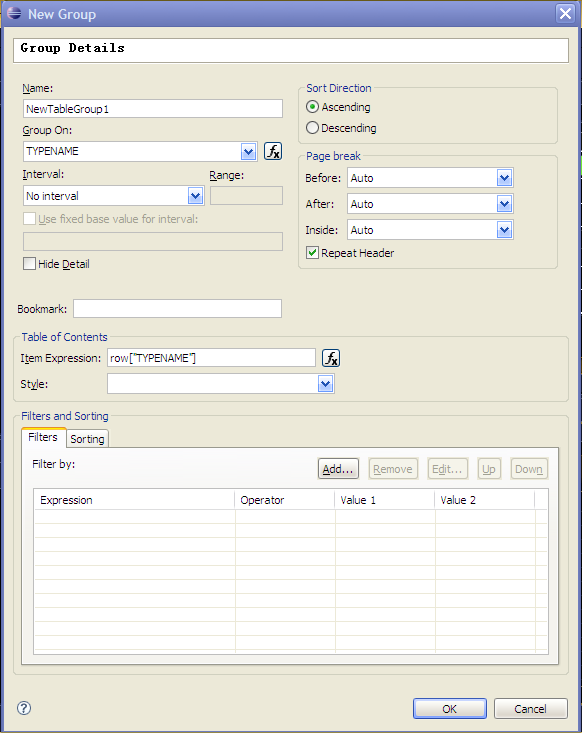
**3．**从Data Explorer视图中的“Data Sets”项中将TYPENAME拖拽至此Table的第一个详细信息行中，然后依次将，CUSTOMER\_ID，CUSTOMER\_NAME，FAX，TEL拖拽入详细信息行内。这里不再赘述。

### 设定分组数据

**1．**选中Table，在右下方的Table的属性编辑器“Property Editor - Table”中，选择Group，进入如下分组设定界面：

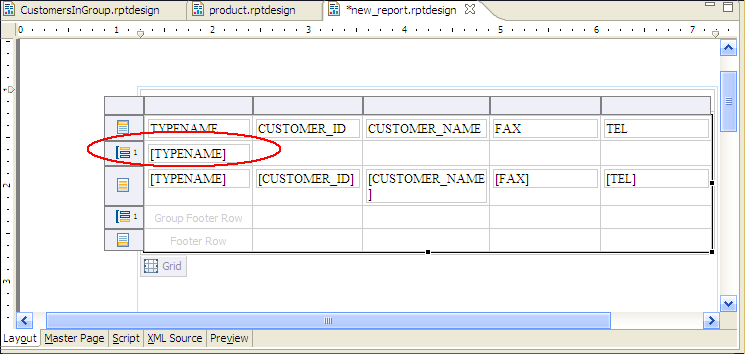


**2．**点击“Add”按钮进行添加分组：



从“Group on”的下拉框选择TYPENAME，其他默认，（可根据需要对“Name”项进行命名），然后选择“OK”。完成分组设定。

**3．**这时Table中多出一行Group行，如下图：



将详细信息栏中的TYPENAME（即Table中第三行第一列数据项，如上图标记下方的数据项）删除。

这样，一个按照用户类型建立的分组报表就建立起来了。

### 格式化报表

进行报表格式化处理后，预览如下效果：



## **交叉报表**

### 构建数据源

方法同上。

### 构建数据集

**1．**在定义好数据集的基本信息之后，进入SQL SELECT 语句查询界面，

输入如下SQL语句，进行数据集的创建：

select EBANK.CUSTOMER.CUSTOMER\_NAME,

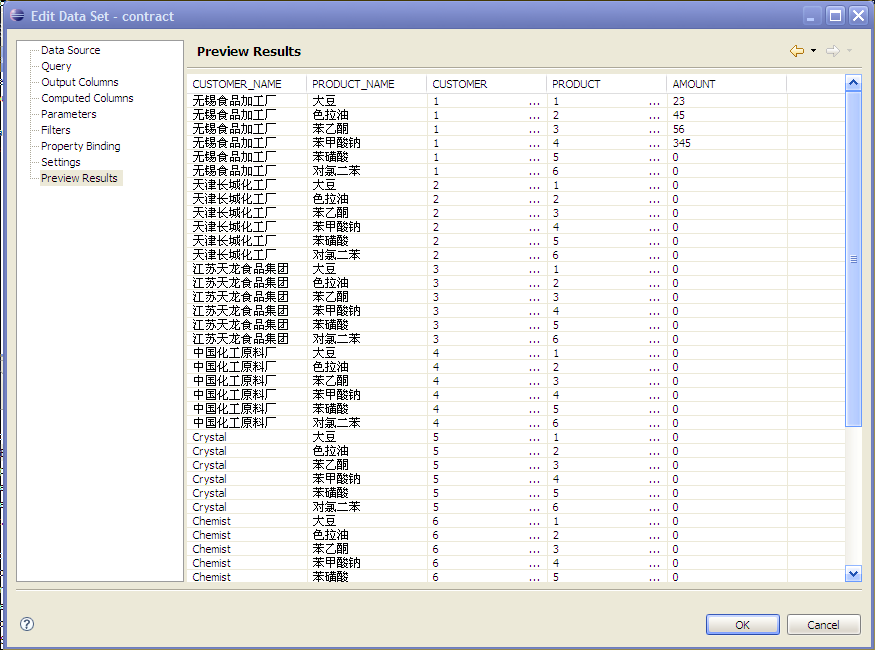
EBANK.PRODUCT.PRODUCT\_NAME,EBANK.CONTRACTSUM.\*

from EBANK.CUSTOMER,EBANK.PRODUCT,EBANK.CONTRACTSUM

where EBANK.CONTRACTSUM.CUSTOMER = EBANK.CUSTOMER.CUSTOMER\_ID

and EBANK.CONTRACTSUM.PRODUCT = EBANK.PRODUCT.PRODUCT\_ID

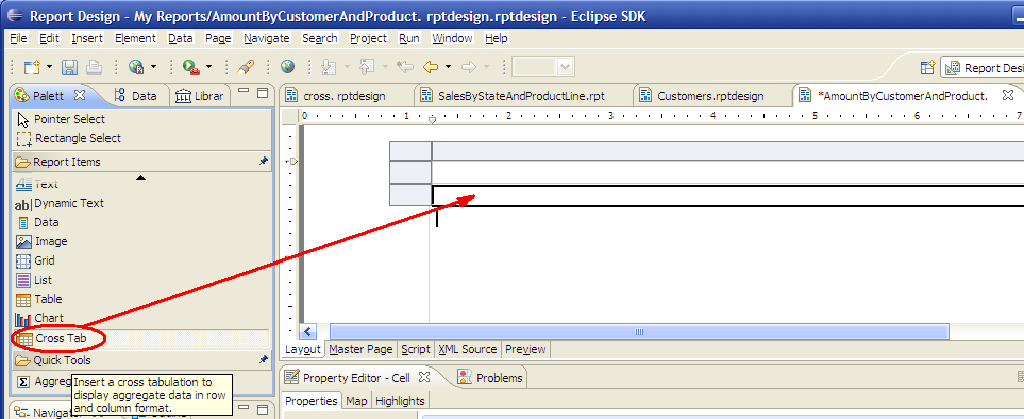
**2．**如果创建正确，在结果预览窗口“Preview Relusts”中将看到一下信息：



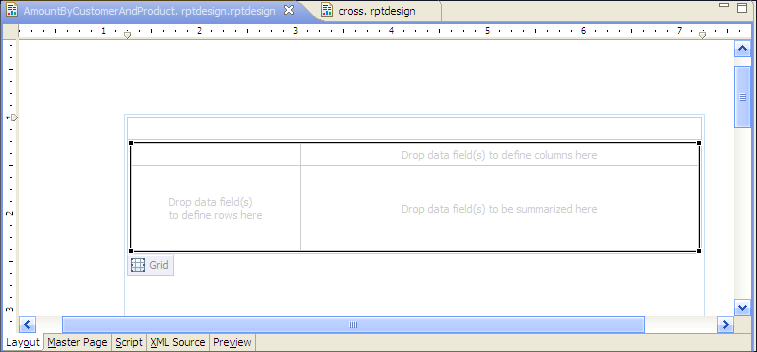
点击“Ok”完成数据集的创建。

### 在报表中加入交叉报表

1. 同上例，先在布局编辑器中建立一个1列2行的Grid。
2. 从选用板“Palette”视图中拖拽一个“Cross Tab”至第二行的网格中。

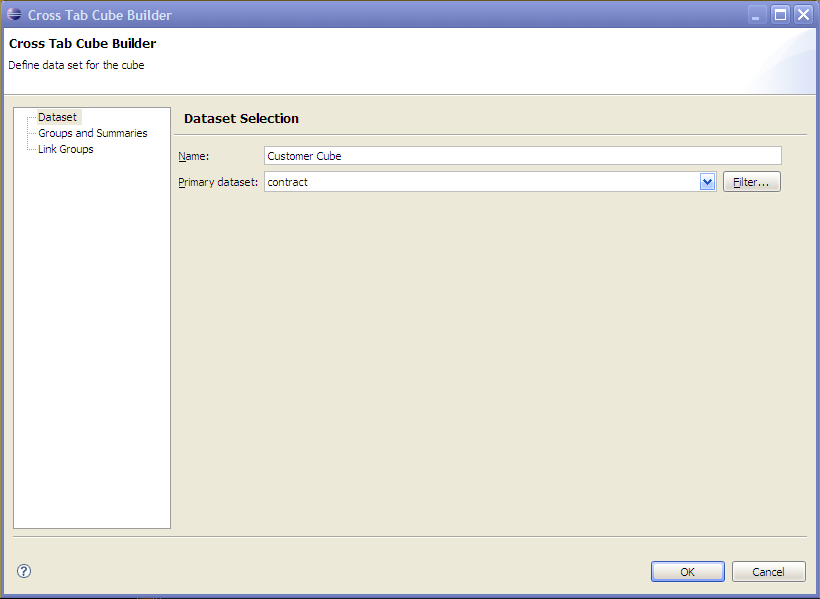


在布局编辑器中可以看到我们创建的Cross Tab的样式。

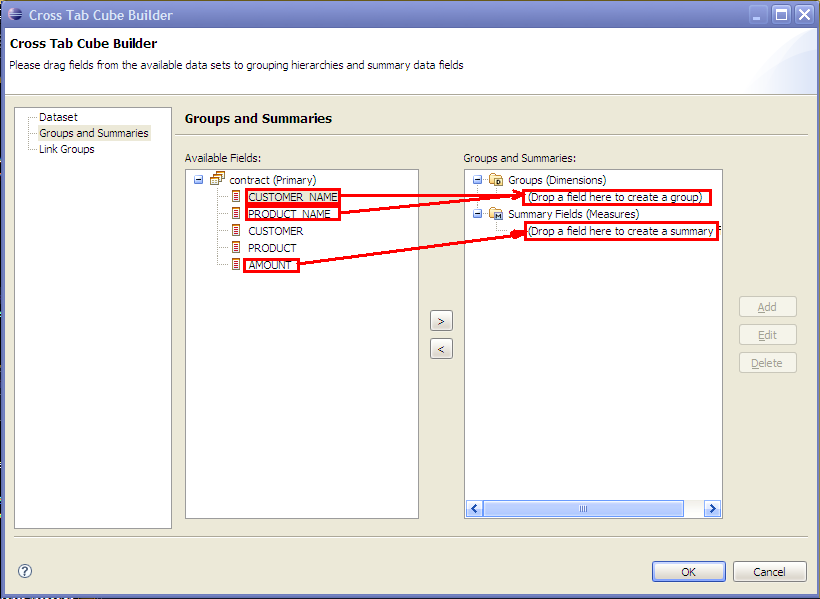


### 数据准备

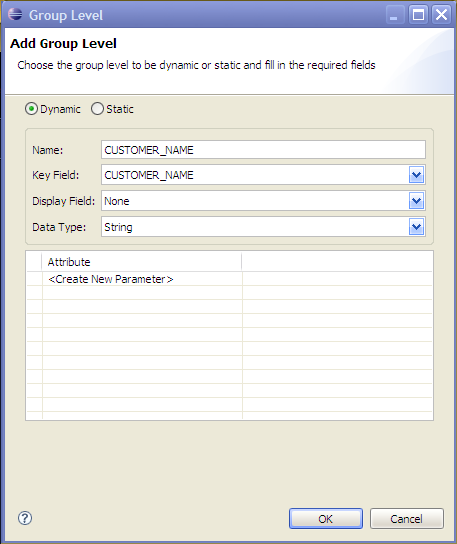
1. 选择数据资源管理器“Data Explorer”。右键单击“Data Cubes”，选择“New Data Cube”，进入如下交叉表数据编辑界面：
2. 默认选择的是“Dayaset”，  
   Name：为Cube命名,Customer Cube  
   Primery dataset：选择相关的数据集。打开下拉菜单选择之前建立的数据集contract



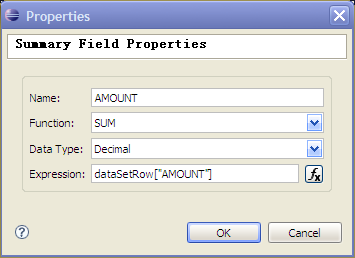
1. 在左边的主菜单中选择第二项“Groups and Summeries”，出现如下窗口（），  
   在将Available Fields中的CUSTOMER\_NAME拖拽到右边Groups (Dimensions)中标有“(Drop a field here to create a group)”的地方，如下图。



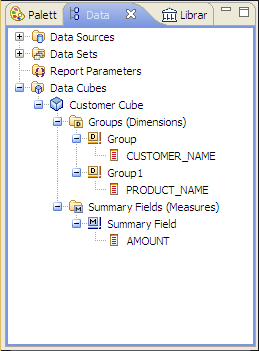
拖拽完毕之后弹出如下窗口：所有选项都选默认数据。点击“OK”进入下一步



1. 按照地4步的方法，将Available Fields中的PRODUCT\_NAME也拖拽到右边的对应地方。
2. 将Available Fields中的AMOUNT拖拽到右边Summary Fields (Measures)中标有“(Drop a field here to create a summary field)”的地方。弹出下面窗口，同样选择默认数据，点击“OK”进入下一步。

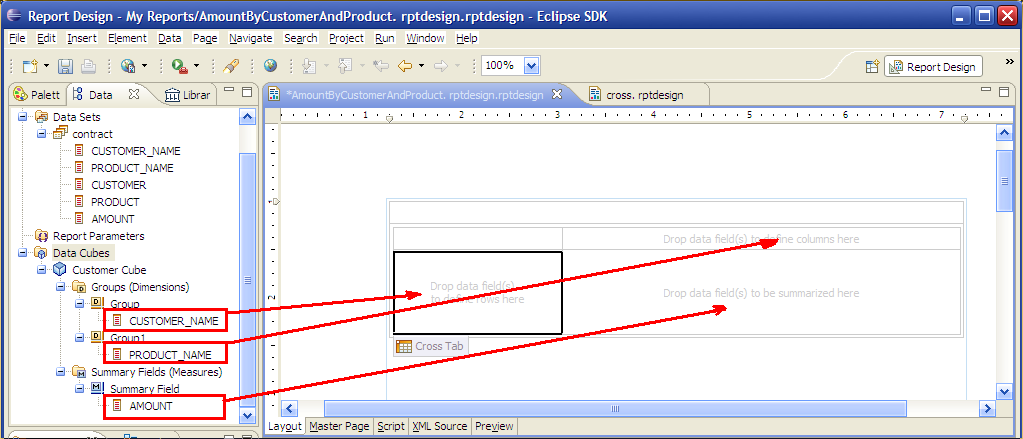


数据设定好以后，我们可以在“Data Explorer”看到Data Cubes的信息如下图：



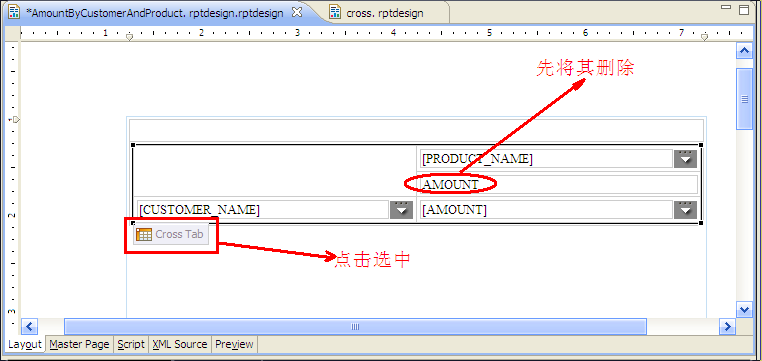
### 向交叉表插入数据

1. 在 Data Explorer视图中，将Customer Cube中的CUSTOMER\_NAME拖拽并放置于布局编辑器中Cross Tab的标有“Drop data field(s) to define rows here”的区域中。如下图：



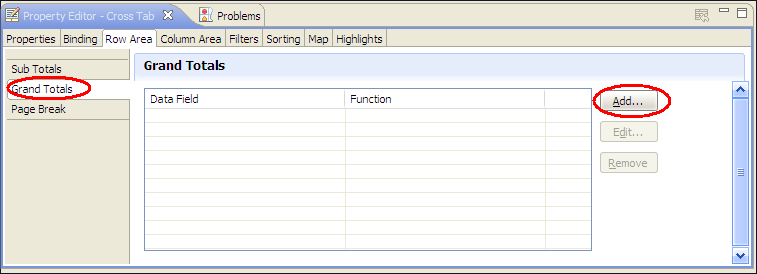
同第一步，将Customer Cube中的PRODUCT\_NAME拖拽并放置于布局编辑器中的Cross Tab中标有“Drop data field(s) to define rows here”的区域中；然后将Customer Cube中的AMOUNT拖拽并放置于布局编辑器中的Cross Tab中标有“Drop data field(s) to be summarized here”的区域中。上图所示：

1. 完成上述步骤之后，布局编辑器中的Cross Tab出现如下图中的样式：我们将Cross Tab中第二列的AMOUNT的标题菜单删掉。如果不去掉，就会在报表的产品标题菜单下方出现一行“AMOUNT”的标题，，不仅多于，而且影响美观。

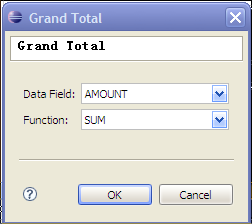


### 加入汇总

1. 点击Cross Tab左下角的“Cross Tab”图标选中本交叉表。在Property Editor属性编辑器中选择“Row Area”，如下图：  
   在左边的选项栏中选择第二个选项“Grand Totals”

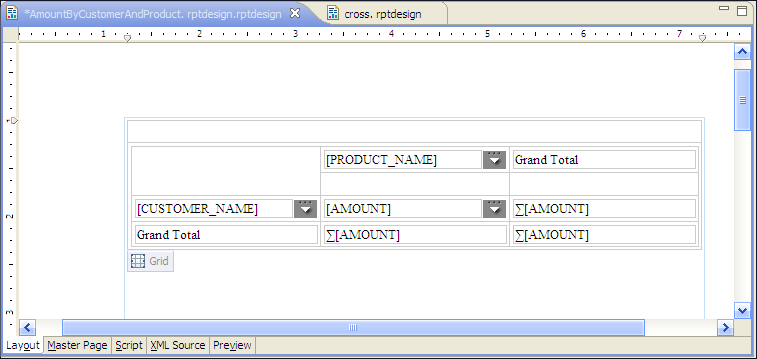


1. 点击“Add”，添加行汇总元素。弹出如下窗口：



选用默认的数据，点击“OK”进入下一步。

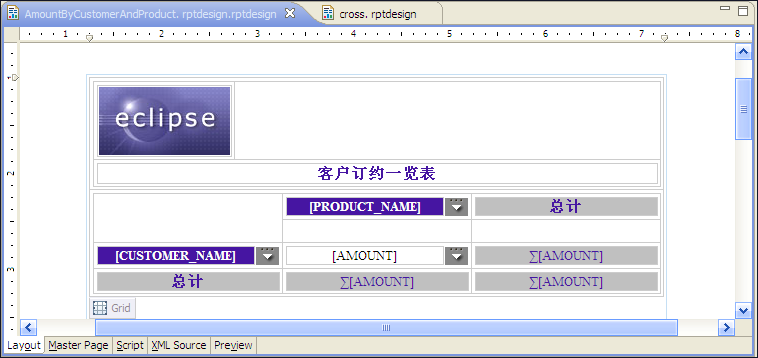
1. 在Property Editor属性编辑器中选择“Colum Area”，同第一步加入列汇总。添加完毕，布局编辑器出现如下视图：



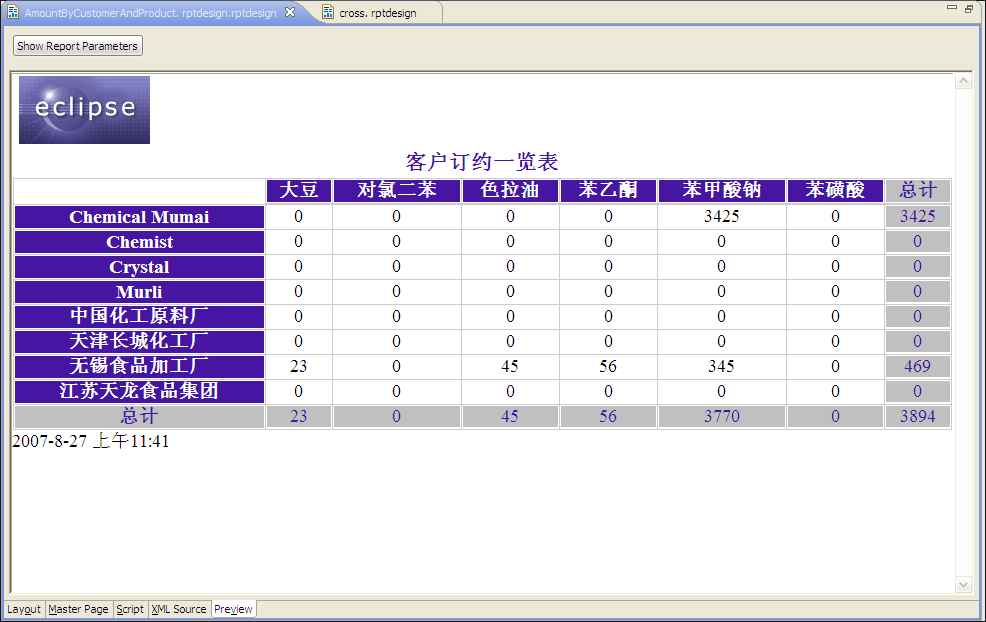
这样，一张交叉汇总表基本上完成了。

### 格式化报表

格式化报表：



预览如下图：



## **嵌套报表**

　嵌套报表其实是一种设计技巧的应用，类似编码过程中的模块化设计。当报表的复杂程度让设计者开始觉得逻辑混乱的时候，可以使用嵌套报表设计方法，将报表拆分为多个不同的设计页面，然后使用一个总的页面将这些设计页面整合起来展现。  
　　此外，嵌套报表有时候也会用于主从结构的实现，通过逐层嵌套，可以实现无限层次的主从结构数据展现。

本例，我们将每个客户的信息包括客户表中基本的信息以及他们的订单信息，在同一张表中按客户一一列出来，所使用的就是嵌套报表。

报表名我们定为：nestingTab.rptdesign

### 构建数据源

本例我们使用BIRT自带的数据源，在新建数据源选择数据源类型为默认的第一项：Classic Model Inc，Sample Database

数据库驱动类别；org.eclipse.birt.report.data.oda.sampledb.Driver (Classic Models Inc. SampleDB Driver)；URL：jdbc:classicmodels:sampledb

### 构建数据集

本例我们需要建立2个数据集：

1. 建立数据集payments，所用SQL语句：（这里我们加入条件选择订单大于150000）

**select** CUSTOMERNUMBER,sum(AMOUNT) amount

**from** CLASSICMODELS.PAYMENTS

**group by** CUSTOMERNUMBER

**having** sum(AMOUNT)>150000

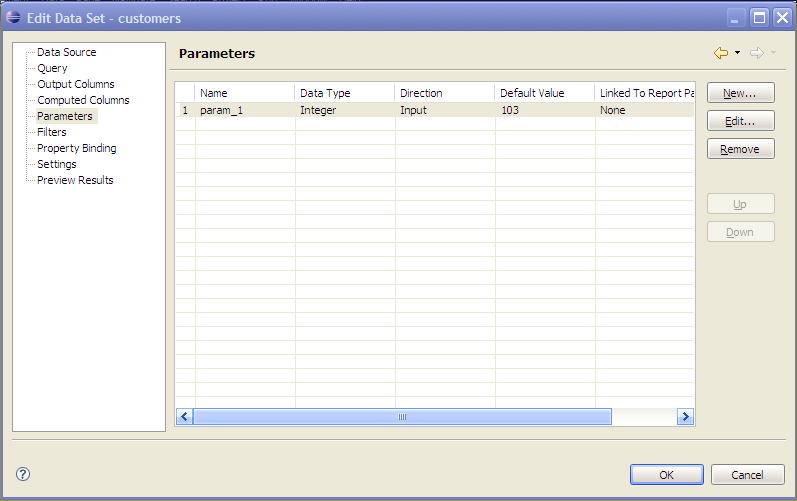
**order by** amount **desc**

1. 建立数据集customers：所用SQL语句：

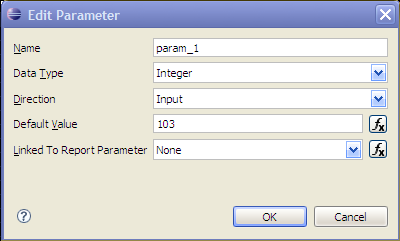
**select \***

**from** CLASSICMODELS.CUSTOMERS **where**

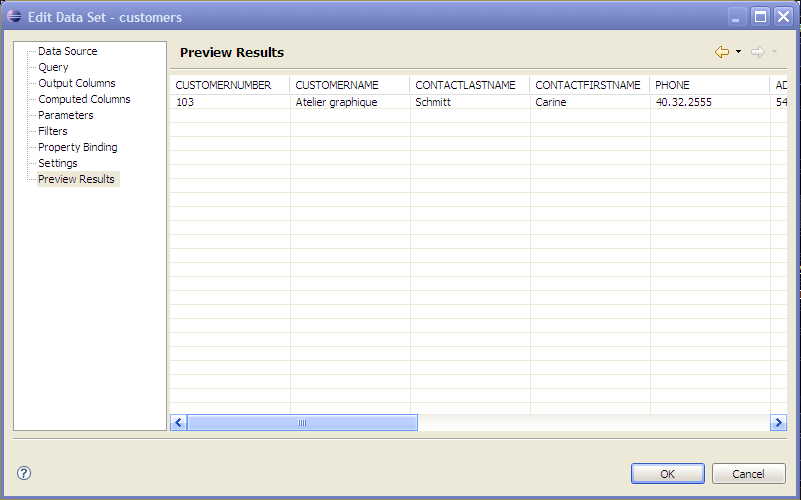
CLASSICMODELS.CUSTOMERS.CUSTOMERNUMBER = ?

输入SQL语句后，还要在数据集编辑窗口选择Parameters选项。  


双击默认参数Pram1或者选择“Edit”，弹出如下窗口：



输入：  
Name：CustID；  
Default Value：103；  
其它默认  
完成后，在Preview Result选项中就可看见结果预览：

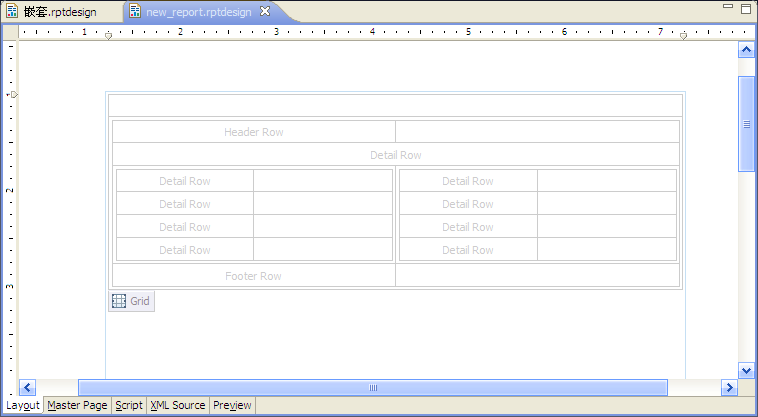


### 布局嵌套报表

1. 同上例，先在布局编辑器中建立一个1列2行的Grid。
2. 从选用板“Palette”视图中拖拽一个2行2列“Table”至第二行的网格中；
3. 选中Table的第一个“Detail Row”行，右键点击后选择“Merge Cells”将本行两列合并为一列。本行用来显示主表的信息
4. 从选用板“Palette”视图中拖拽一个“Table”的至第二个“Detail Row”行中的第一格中，设定其为2列4行。完成后将这个刚插入的子表的“Header Row”和“Footer Row”删除掉。
5. 向主表的第二个“Detail Row”行的第二格进行与上一步同样的操作，插入一个相同的子表。

第4、5两步所加的两个子表其实是用来加入客户信息的，所用到到的数据都是customers数据集里的数据，这里用两个表来进行展现完全是为了整体布局的美观。

布局配置完毕，布局编辑器中会呈现如下图：



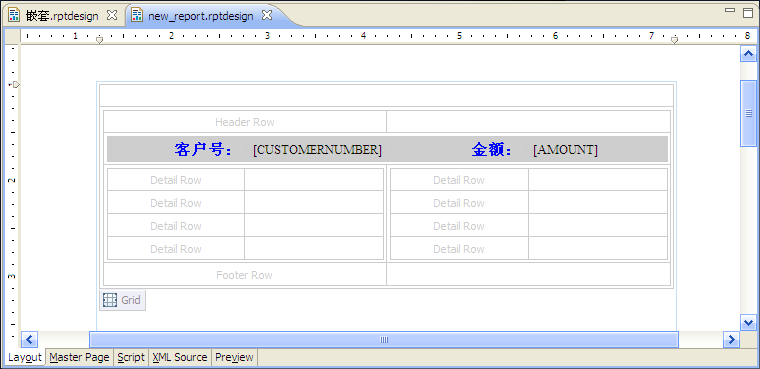
### 将Table绑定数据集

1. **将Table绑定数据集：**在布局编辑器中选中主表Table，下方出现Table的属性编辑器Property Edit，选中标题栏的Binding栏，从Dat Set中选择payments数据集。弹出窗口选择“Yes”即可。点击“Ok”完成。
2. **设定排序：**  
   接着在Property Edit编辑器中选中“Sorting”栏，点击“Add”添加排序变量，从弹出窗口的下拉菜单中选择“CUSTOMERNUMBER”，即按照编号排序，排序方法选择“Ascending”（升序）。点击“OK”完成。（读者根据实际情况的需要可以通过同样的方法选择按照不同的变量来进行排序）

### 设定主表数据

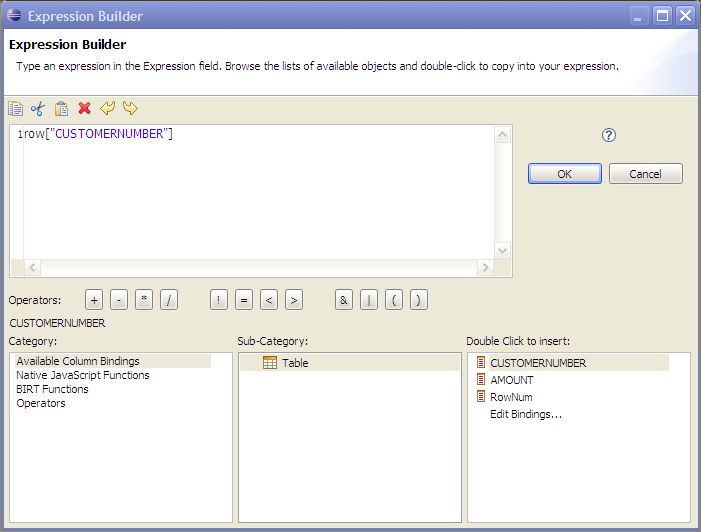
1. **加入网格：**从选用板“Palette”视图中拖拽一个Grid至主表Table的第一行“Detail Row”中，设定其为4列1行。
2. **插入数据:**展开选择数据资源管理器“Data Explorer”，将数据集payment中“CUSTOMERNUMBER”与“AMOUNT”两个数据项分别拖拽至我们刚插入的Grid的第2格与第4格中。并将其格式都设定为居左。
3. **加入数据标题Lable** 从选用板“Palette”视图中拖拽Lable至Grid的第1格与第3格中，分别输入“客户号：”与“金额：”，并将其格式都设定为居右

设定好之后，视图如下：



### 设定子表数据

1. **设定参数绑定：**在布局编辑器中选中子表Table，下方出现Table的特定编辑器Property Edit，选中标题栏的Binding栏，在“”点击按钮“Dataset Parameter Binding”，进入后双击默认的变量进入进行设置，点击“Value”标签后的函数符号“”，进入如下界面：

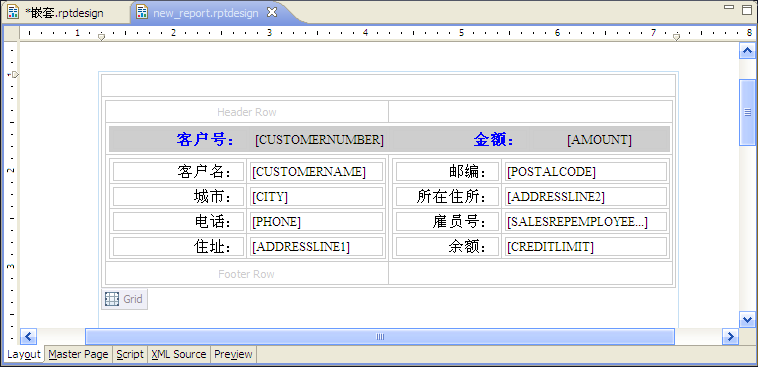


如图依次选择需要插入的数据，在第三个框中，双击“CUSTOMERNUMBER”进行插入。依次点击“OK”，完成

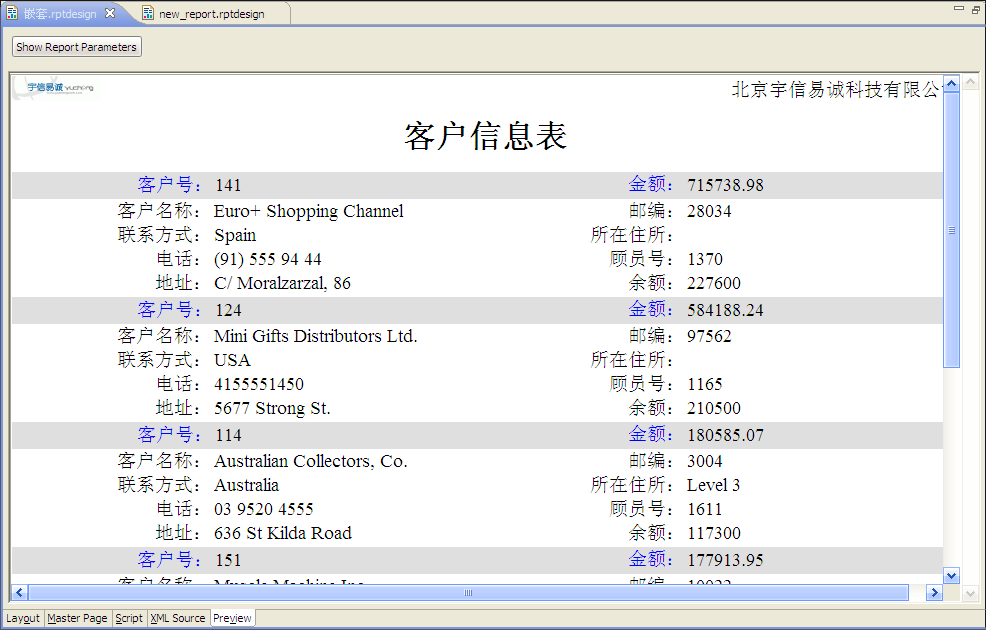
对另外一个子表也需进行如上操作。

1. **插入数据**  
   展开选择数据资源管理器“Data Explorer”，将数据集customers中的“CUSTOMERNAM”、“CITY”、“PHONE”、“ADDRESSLINE1”拖拽至第一个子表Table的第二列中，将第一列的各个单元格同样拖拽进去一个“Text”，分别输入子项标题“客户名”、“城市”、“电话”、“住址”。然后将数据集customers中的“POSTALCODE”、“ADDRESSLINE2”、“SALESREPEMPLOYEENUMBER”、“CREDITLIMIT”拖拽至第二个子表Table的第二列中，将第一列的各个单元格同样拖拽进去一个“Text”，分别输入子项标题“邮编”、“所在住所”、“雇员号”、“余额”

这样整个嵌套报表的数据配置已经完毕，如果配置正确，布局将会如下：



结果预览：



## **交互报表**

交互式报表就是存在与用户交互的通道，在查询交互式报表的结果时，需要用户输入一定的参数，输出结果就会显示出参数所对应的结果。

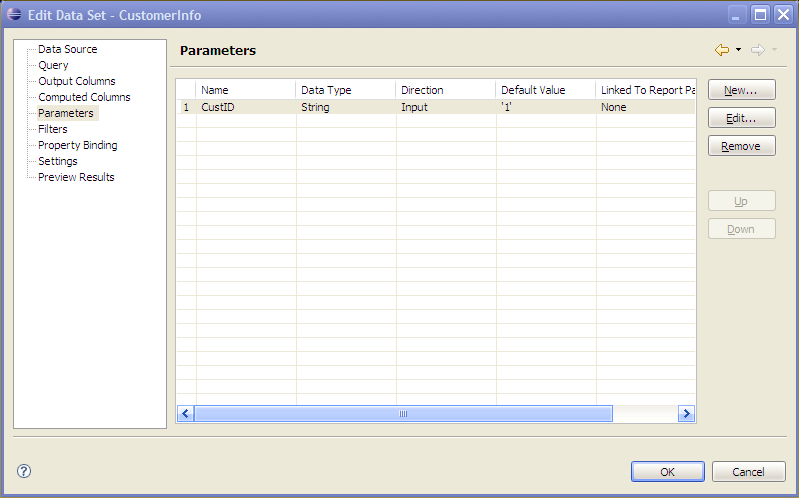
在本例中，我们将客户信息做成交互式报表，在用户输入客户的编号之后，我们就会看到对应客户的信息。

### 创建报表、数据源

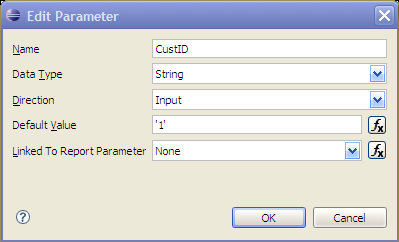
方法同上，报表名我们定位：PeportOfAlternant.rptdesign

### 创建数据集

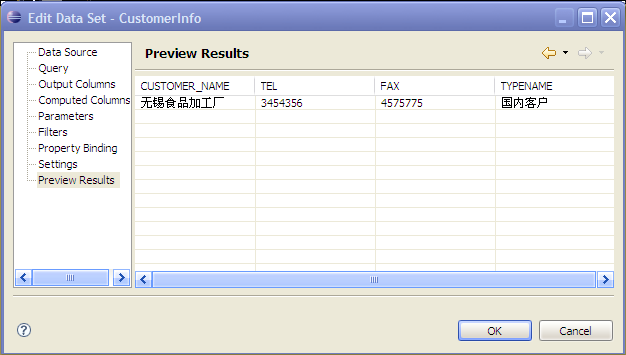
本例我们需要建立3个数据集：

1. 建立数据集CustomerId，所用SQL语句：  
   **select**EBANK.CUSTOMER.CUSTOMER\_ID,  
   EBANK.CUSTOMER.CUSTOMER\_NAME **from** EBANK.CUSTOMER  
   **where   
   EXISTS   
    (SELECT** EBANK.CONTRACTSUM.CUSTOMER **FROM** EBANK.CONTRACTSUM **WHERE** EBANK.CUSTOMER.CUSTOMER\_ID = EBANK.CONTRACTSUM.CUSTOMER)
2. 建立数据集CustomerInfo：所用SQL语句：  
   **select** EBANK.CUSTOMER.CUSTOMER\_NAME,  
   EBANK.CUSTOMER.TEL,  
   EBANK.CUSTOMER.FAX,  
   EBANK.CUSTOMERTYPE.TYPENAME  
   **from** EBANK.CUSTOMER,EBANK.CUSTOMERTYPE  
   **where** EBANK.CUSTOMER.CUSTOMERTYPE=EBANK.CUSTOMERTYPE.TYPEID  
   **and** EBANK.CUSTOMER.CUSTOMER\_ID **=** ?  
   输入SQL语句后，还要在数据集编辑窗口选择Parameters选项。

双击默认参数Pram1或者选择“Edit”，弹出如下窗口：



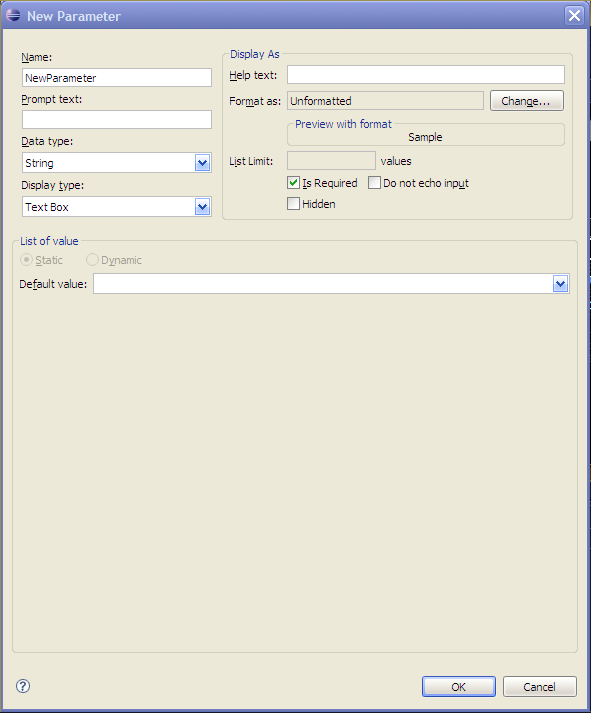
输入：  
Name：CustID；  
Default Value：’1’；  
其它默认  
完成后，在Preview Result选项中就可看见结果预览：

建立数据集Contract：所用SQL语句：  
**select** EBANK.CONTRACTSUM.AMOUNT,  
EBANK.PRODUCT.PRODUCT\_NAME  
**from** EBANK.CONTRACTSUM,EBANK.PRODUCT  
**where**   
EBANK.CONTRACTSUM.PRODUCT **=** EBANK.PRODUCT.PRODUCT\_ID  
**and** EBANK.CONTRACTSUM.CUSTOMER **=** ?

同上一步一样，还需对参数设置默认值，方法同上。

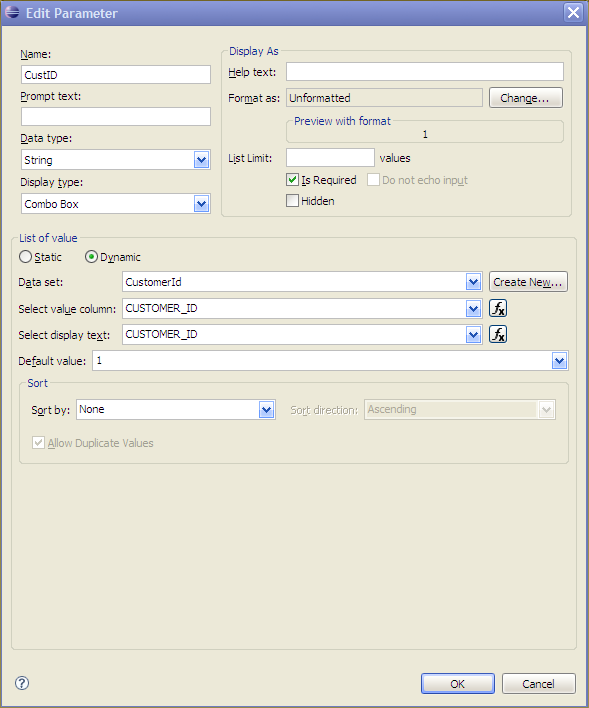
### 加入响应参数

**１.**建立好数据集之后，展开选择数据资源管理器“Data Explorer”，右键点击“Report Parameters”选择“New Parameter”，弹出如下窗口：



输入：  
Name：CustId  
Prompt：请输入用户编号  
Data Type：选择默认String  
Display Type：从下拉菜单中选择Combo Box

**２.** 将Display Type值选为Combo Box后，下方的“List of Value”的两个选择按钮变亮，选择“Dynamic”，窗口变为如下：



输入数据：

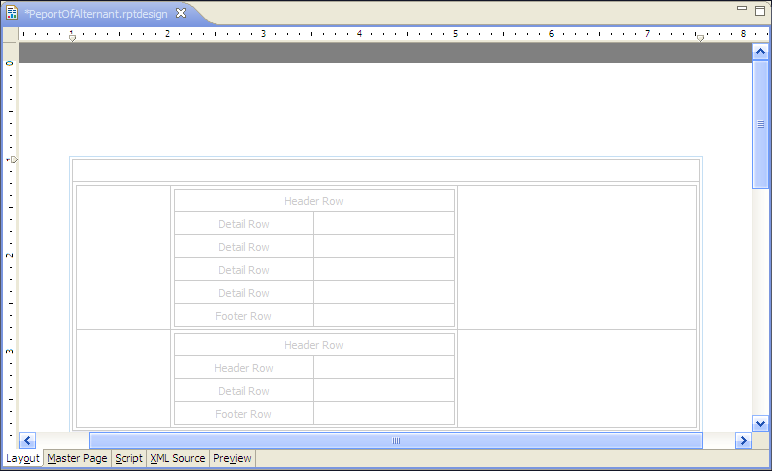
Data Set：从下拉框中选择CustomerId；  
Select Value Column：从下拉框中选择CUSTOMER\_ID；  
Select Display text：从下拉框中选择CUSTOMER\_ID；  
Default Value：选择<select value…>之后在弹出的窗口中选择任意一个即可，这里选1

点击“OK”完成

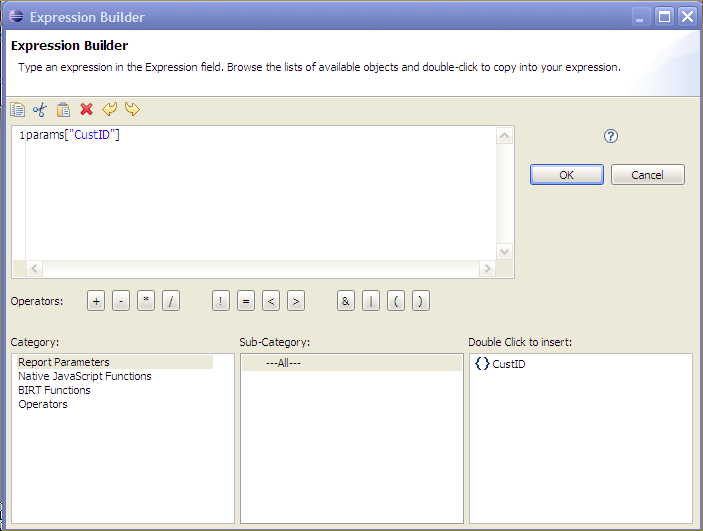
### 布局报表

1. 先在布局编辑器中建立一个1列2行的Grid；
2. 在所建立的Grid的第二格中再加入一个3列2行的Grid；
3. 选中3列2行的Grid的第一列，在属性编辑器Property Edit，将默认菜单下的General项中的 “Width”设定为20%，其中“%”符号从下拉框中选择；选中第二列，同样方法设定其宽为60%；
4. 从选用板中向选中3列2行的Grid的第二列第一行的空格拖拽一个2列4行的Table，选中这个Table的第一列，用第3步中的方法设定其宽为50%，选中这个Table的第一行“Header Row”，右键点击选择“Merge Cells”将此行合并，并从选用板中拖拽一个“Lable”，输入“客户基本信息”，使其居左；
5. 再在这个3列2行的Grid的第二列第二行的空格中加入一个2列1行的Table，对此Table作与第4步同样的处理，在合并“Header Row”之前先选中“Header Row”，右键点击选择“insert”，插入一行新的Header Row。然后将第一个“Header Row”合并，在合并后的“Header Row”中加入的“Lable”中输入“客户订单信息”使其居左

布局后的视图：



### 插入报表数据

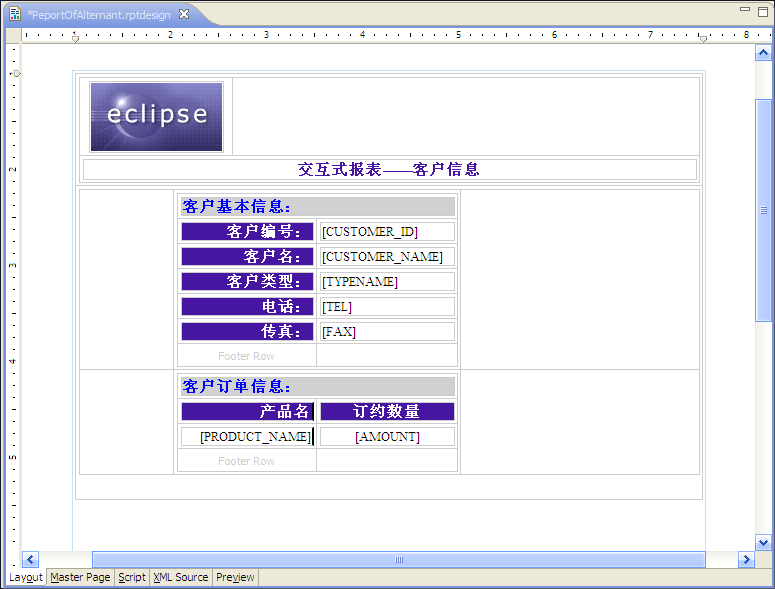
1. **参数绑定：**选中我们在3.5.4中第3步加入的第一个Table，下方出现Table的属性编辑器Property Edit，选中标题栏的Binding栏，在“Data Set”的下拉框中中选择数据集“CustomerID”；然后点击其后的按钮“Dataset Parameter Binding”，进入后双击默认的变量进入进行设置，点击“Value”标签后的函数符号“”，进入参数设定界面：  
   

从下方的设定窗口依次选择：Report Paramrters——All——{}CustID，双击{}CustID将其加入。依次点击各个窗口的“OK”，完成参数绑定。

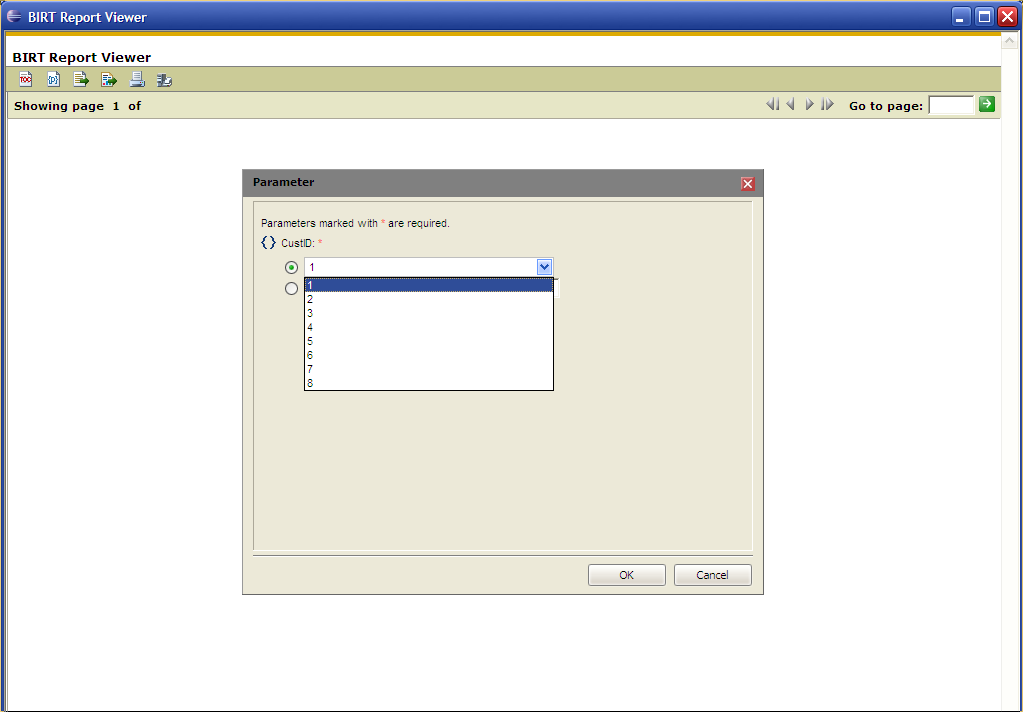
1. 将数据集CustomerInfo中的数据项一一拖拽至第一个Table的第二列的“Detail Row”中，在第一列中加入“Lable”并输入标题信息。
2. 对第二个Table也做与第一步的相同操作：注意在Table的属性编辑器中进行参数绑定之前在“Data Set”中选择的数据集应该是Contract。做好参数绑定之后，将数据集Contract中的数据项一一拖拽到Table的“Detail Row”的两个空格中，然后将“Header Row”中自动生成的标题Lable更改。这样，我们就完成了交互报表的基本工作。

### 格式化与预览

格式化报表：



点击View Report In Web Viewer结果预览图表，此时弹出对话框，需要我们选择输入客户编号：



选择“1”，结果如下：



通过点击左上方第二个图标“”可以重新进行参数选择。

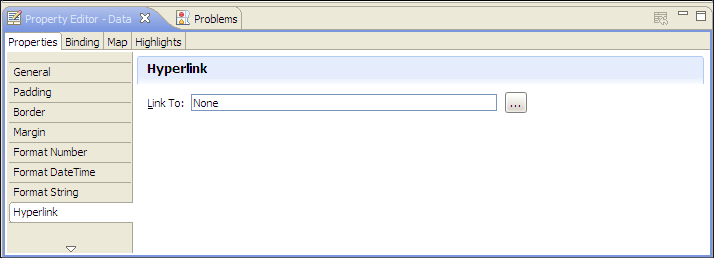
## **报表钻取**

报表钻取是指从一个表中获取另外一个表的数据。

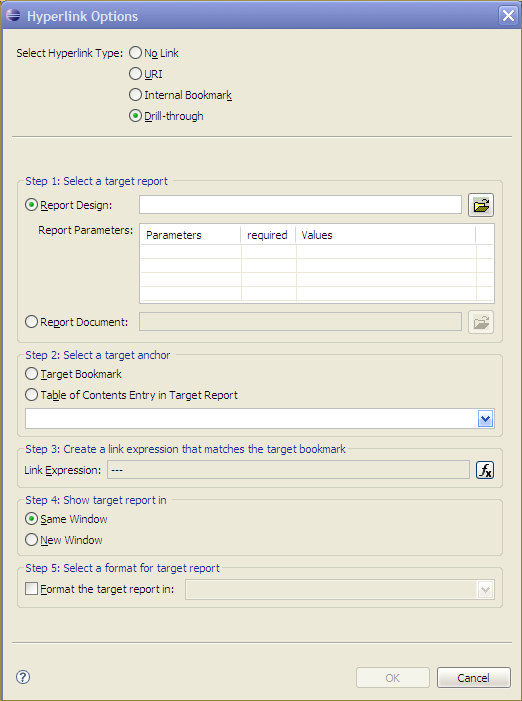
本例我们从3.1所建的客户信息表中获取上例3.5中的客户的详细信息。

### 设定响应参数

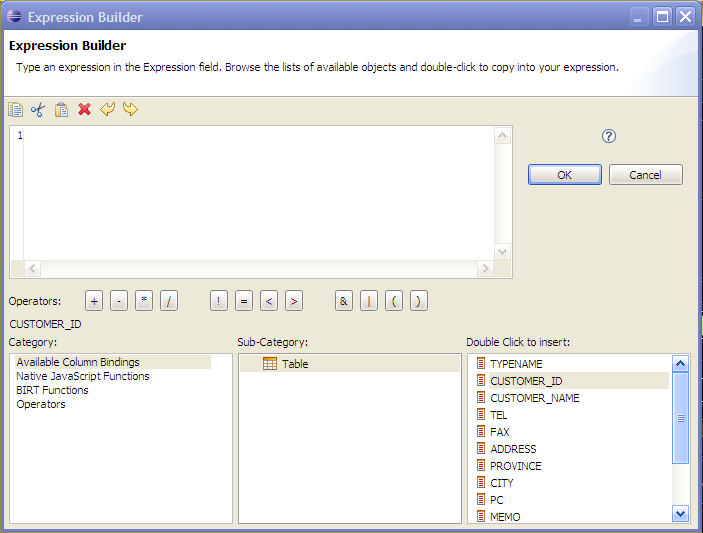
1、打开3.1中所建表CustomersInGroup.rptdesign，从布局编辑器中选择数据项CUSTOMER\_ID；从下方的特定编辑器Property Edit中选择默认的Properties项，选择树形菜单中的“Hyperlink”项，下图所示：



2、点击“…”框，进入超级链接选择界面，选择第四项：Drill\_through，出现如下视图：



3、在Step1中，打开Report Design后方的“”选择我们所要链接文件PeportOfAlternant.rptdesign；在Report Parameters表格中，点击Parameters列下方的空白框，会出现我们我们在PeportOfAlternant.rptdesign文件设定的参量CustID，选中它，然后点击“required”，会出现一个绿色的“√”，然后点击Value下方的空白框，，点击框中出现的省略号框，出现下图：



如图示依次选择到CUSTOMER\_ID，双击加入。点击“OK”完成。

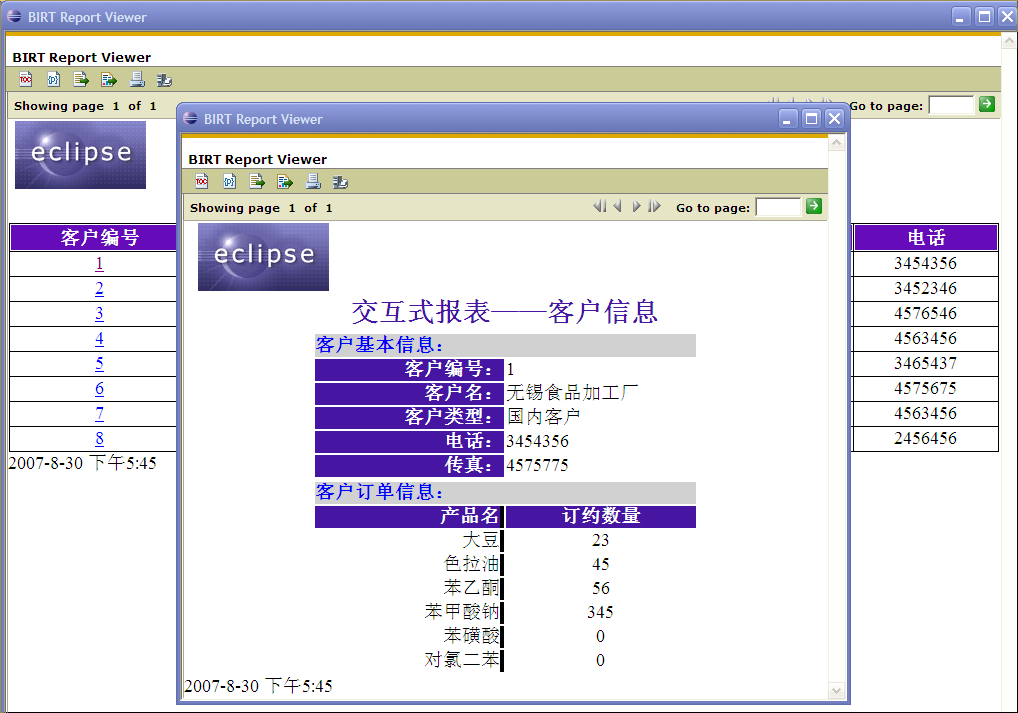
此步表示将CUSTOMER\_ID与另一表中的参量CustID进行映射。

4、在Step2，Step3中我们这里不用设定。在Step4中选择钻取数据是否在新窗口出现，我们选择第二项“New Window”；在Step5中选择窗口的视图模式，我们可以不用设定。

点击“OK”完成数据超级链接设定。之后我们就可以在预览中观看数据钻取的效果了。

### 预览结果

打开预览视图：



如图所示，点击客户编号“1”后直接弹出客户的详细信息。

## **图表报表**

图形报表是非常常见的一种报表样式，用于直观地表现数据规律。常用的类型包括：柱状图、饼图、折线图、甘特图、雷达图等。Eclipse的Birt插件为我们提供了多种图表的表现形式。

这里我们利用3.6的交叉报表，为它做一张简单直观的柱形图。

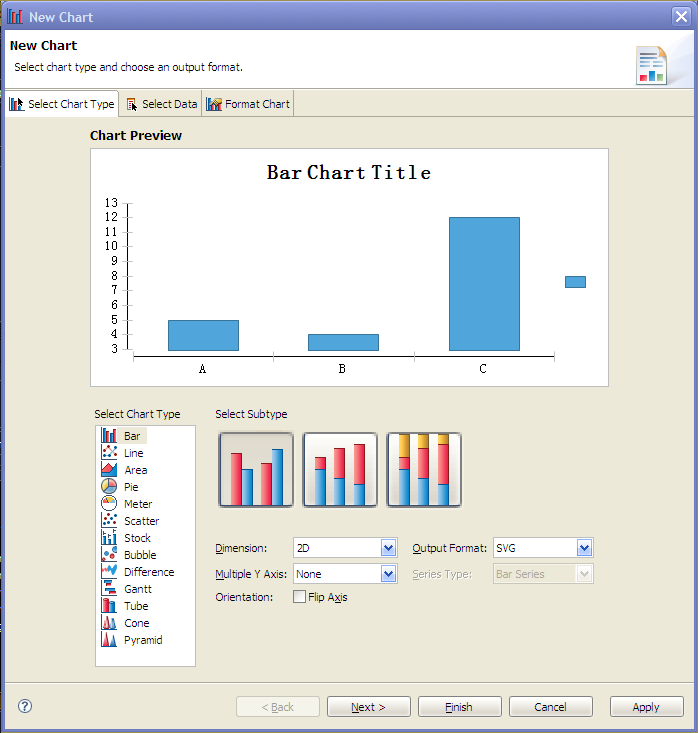
### 为报表插入图表

本例我们直接利用3.6交叉报表的数据集，直接在3.6所创建的报表上插入图表。  
在选用板“Palette”视图中拖拽一个“Chart”至布局编辑器中交叉报表“Cross Tab”的下方。

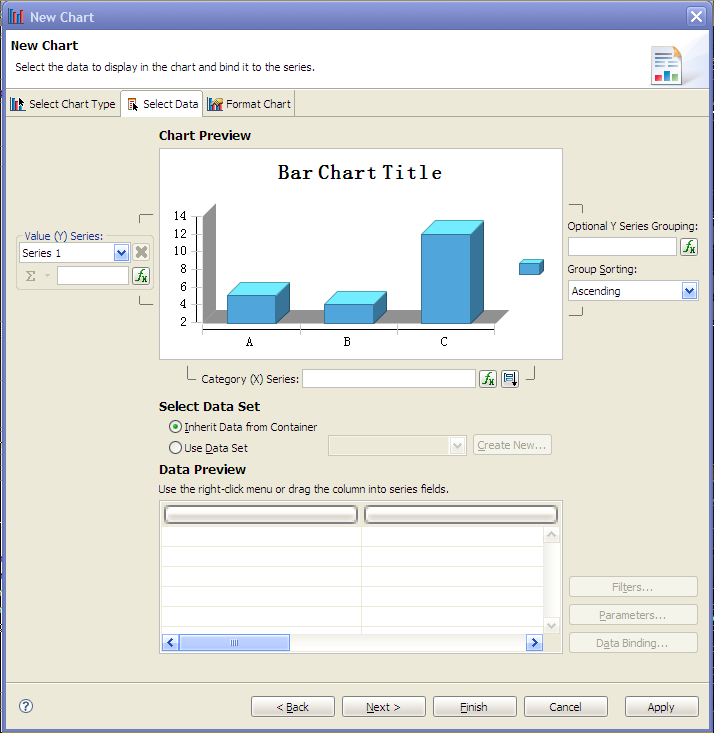
### 设定图表属性

1. 选择图表类型

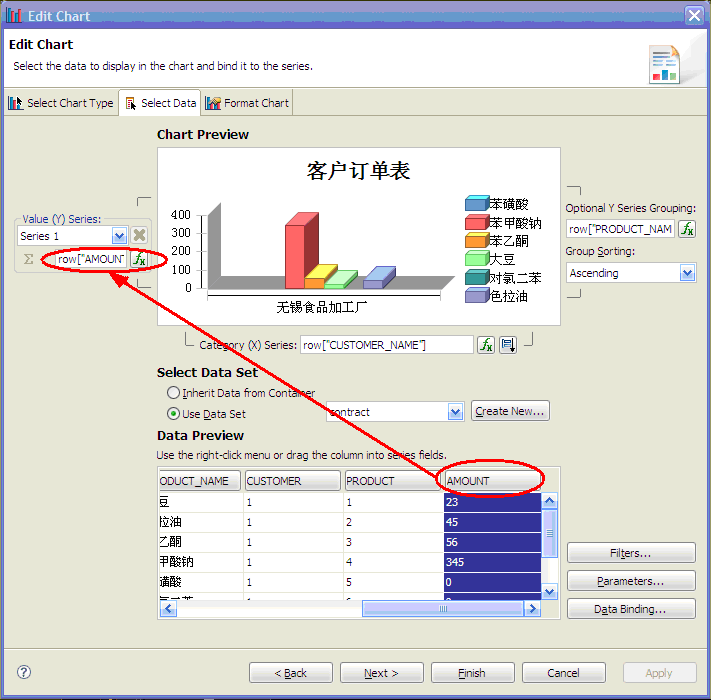
进行上一步拖拽之后，会弹出如下窗口：在此窗口我们首先要进行图表基本属性的设定。系统提供给我们有13种可供选择的图表，这里我们选择第一种“Bar”即柱形图。  
在窗口中部右方的“Select Subtype”部分有三种类型可供我们选择，这里我们选择第一个，其他的数据设定：  
Dimension：2D With Depth (设定图形维度)  
Output Format：SVG（设定输出图表的格式）  
Multiple Y Axis：none



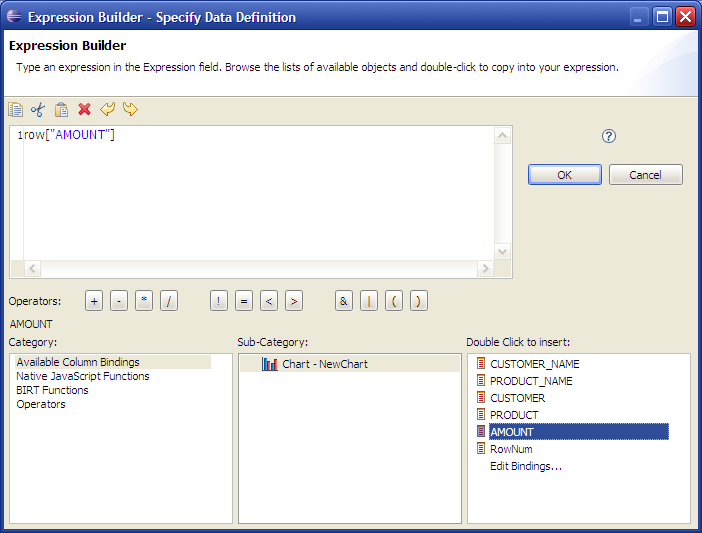
选择“Next”进入下一步：



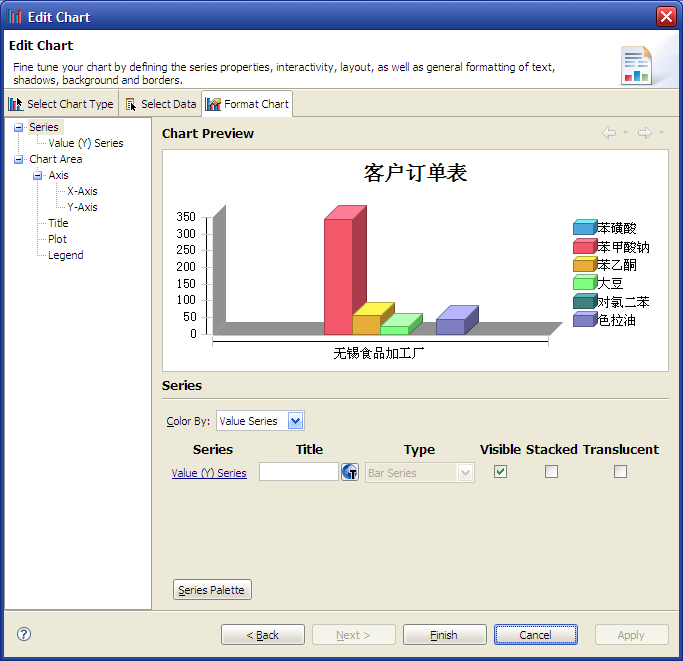
1. 设定好图表基本类型之后，进行图表的数据设定。窗口哦界面如上图。  
   在窗口中部的Select Data Set中，选择Use Data Set，然后从它的下拉菜单中选择我们所创建的唯一数据集contract，  
   此时，Data Preview窗口出现了数据集中的各个数据项：



1. 将数据项中的AMOUNT拖拽至左上方“Value（Y）Series”中的“∑”框内。表示在图表的Y轴的变量设定为amount变量。（也可以通过点击“∑”框左边的函数符号来进行设定。点击进入后，在下方三个小窗口依次选择“Available Column Bindings”→“Chart-New Chart”→“AMOUNT”，点击“AMOUNT”时进行双击。或者可以直接在编辑窗口输入“row["AMOUNT"]”即可。点击“Ok”完成）



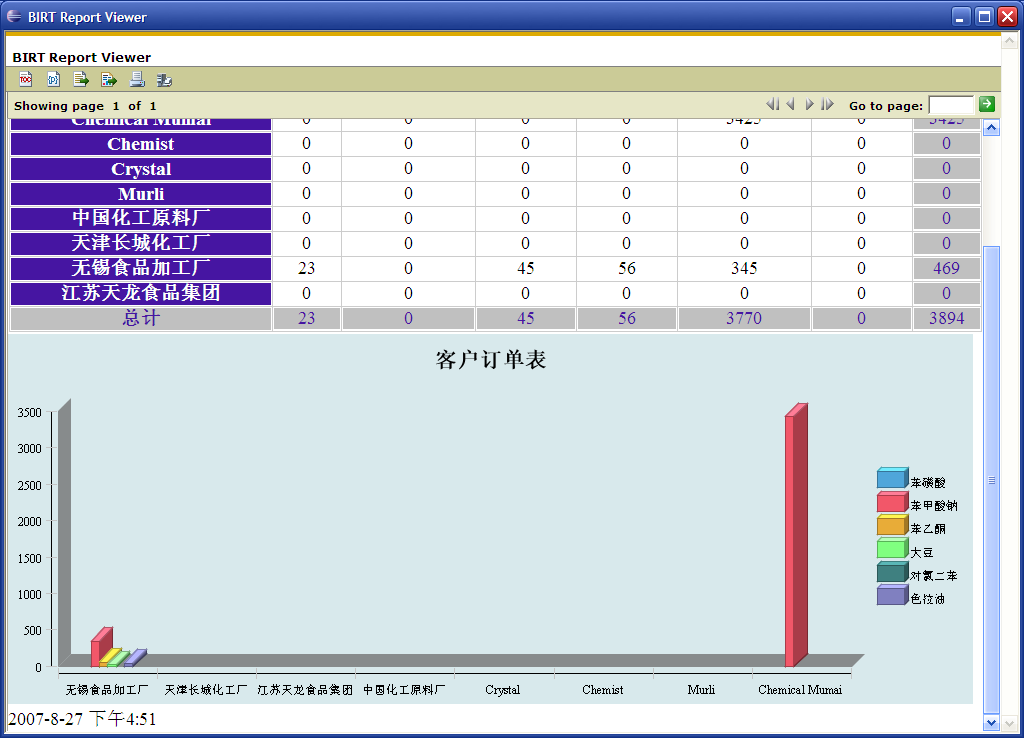
1. 用与第三步相同的方法，设定：  
   Cetogory（X）Series：row["CUSTOMER\_NAME"]（设定X轴变量）  
   Optional（Y）Series Grouping：row["PRODUCT\_NAME"]（设定分组变量）  
   Group Sorting：Ascending（选择默认，指排序方式为升序）

选择“Next”进入下一步。  


1. 本部分进行图表的格式化，对图表进行一些显示的处理。这里我们只修改一下图表的标题，点击Title框左边的“”图表，进入编辑器，输入标题：“客户订单表”。

### 图表预览

在布局编辑器中将图表，拉长拉大一些，以便于在显示时清楚。最后我们图表显示如下图：每个客户上方不同颜色的图条表示的是客户所订的不同产品。十分的清晰明了。



## **多层交叉报表**

多层交叉报表是指在交叉报表中对同一观察角度另外分层次观察，比如，我们想观察某一年各地区的天气情况，可以建立一个关于(时间\_地区)的交叉报表，在“地区”维度上，我们可以先分省，再分市、县，这样就实现了同一维的多层次化。

本例中我们将3.3所建立的交叉报表，在客户维度与产品维度上分别加入客户类型与产品类型两个新的维度，这两个新维度显然是原来两个维度的父维度，这样我们就实现了交叉报表的多层次化。

### 构建数据源

方法同3.2.

### 构建数据集

**1．**在定义好数据集的基本信息之后，进入SQL SELECT 语句查询界面。

输入如下SQL语句，进行数据集的创建：

select EBANK.CUSTOMER.CUSTOMER\_NAME,

EBANK.PRODUCT.PRODUCT\_NAME,

EBANK.CONTRACTSUM.AMOUNT,

EBANK.CUSTOMERTYPE.TYPENAME,

EBANK.PRODUCTTYPE.TYPENAME

from EBANK.CUSTOMER,EBANK.PRODUCT,

EBANK.CONTRACTSUM,

EBANK.CUSTOMERTYPE,

EBANK.PRODUCTTYPE

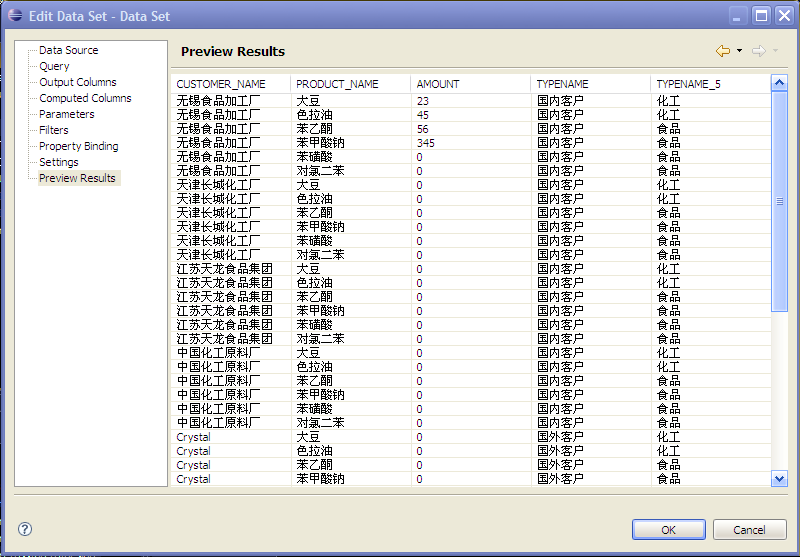
where EBANK.CONTRACTSUM.CUSTOMER = EBANK.CUSTOMER.CUSTOMER\_ID

and EBANK.CONTRACTSUM.PRODUCT = EBANK.PRODUCT.PRODUCT\_ID

and EBANK.CUSTOMER.CUSTOMERTYPE = EBANK.CUSTOMERTYPE.TYPEID

and EBANK.PRODUCT.TYPE = EBANK.PRODUCTTYPE.TYPEID

**2．**如果创建正确，在结果预览窗口“Preview Relusts”中将看到一下信息：



点击“Ok”完成数据集的创建。

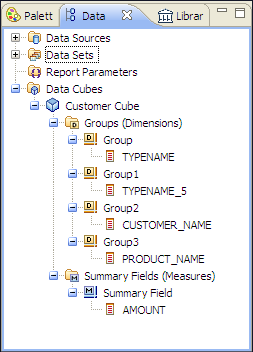
### 在报表中加入交叉报表

方法同3.6.3

### 数据准备

1. 方法同3.3.5，这里需要在维度数据中加入四个分组。

数据设定好以后，我们可以在“Data Explorer”看到Data Cubes的信息如下图：



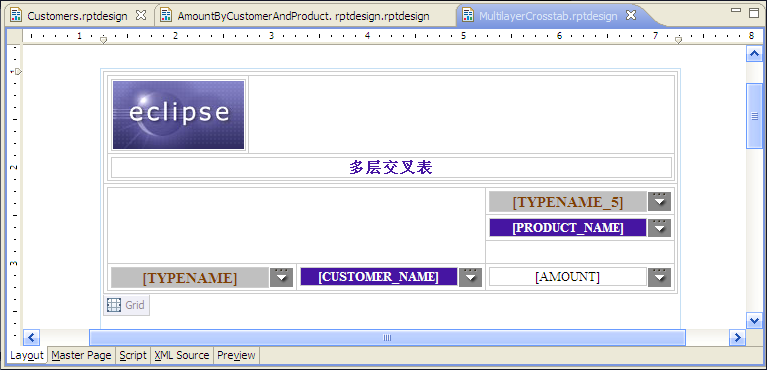
### 向交叉表插入数据

1. 在 Data Explorer视图中，将Customer Cube中的TYPENAME拖拽并放置于布局编辑器中Cross Tab的标有“Drop data field(s) to define rows here”的区域中。（参见3.3.6），然后将Customer Cube中的CUSTOMER\_NAME拖拽并放置于同样的区域中，这时，CUSTOMER\_NAME会出现在TYPENAME后面。
2. 将Customer Cube中的TYPENAME\_5拖拽并放置于布局编辑器中的Cross Tab中标有“Drop data field(s) to define rows here”的区域中；然后将Customer Cube中的PRODUCT\_NAME拖拽并放置于同样的区域中这时，PRODUCT\_NAME会出现在TYPENAME\_5的下面。
3. 然后将Customer Cube中的AMOUNT拖拽并放置于布局编辑器中的Cross Tab中标有“Drop data field(s) to be summarized here”的区域中，将Cross Tab中第二列的AMOUNT的标题菜单删掉。

此时，我们即完成了多层交叉报表的设计工作。

### 格式化与预览

格式化报表：



预览如下图：



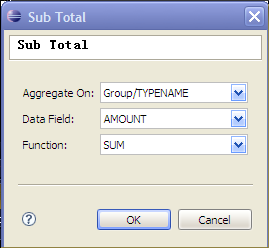
本图可以清晰的看见多层交叉表的层次

## **多层交叉汇总报表**

### 加入汇总

**1．**在3.8所建立的报表MultilayerCrosstab.rptdesign基础上，点击布局编辑器中Cross Tab左下角的“Cross Tab”图标选中本交叉表。在Property Editor属性编辑器中选择“Row Area”，（可参考图3-11）。

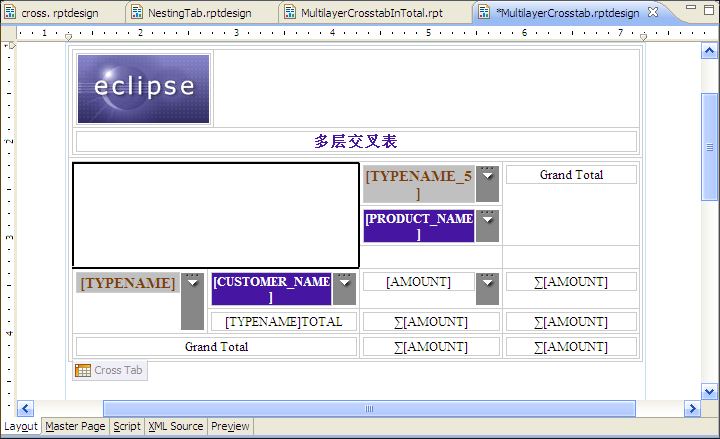
**2．**在左边的选项栏中选择第一个选项“Sub Totals”，点击编辑器右边的“Add”按钮，弹出加入次级汇总（本例即根据客户类型进行汇总）的窗口：



全部选择默认数据：点击“OK”完成创建。

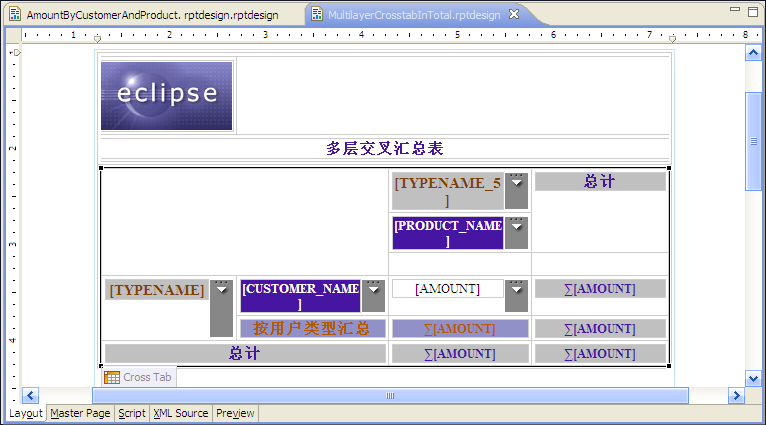
**3．**在左边的选项栏中选择第二个选项“Grand Totals”，点击编辑器右边的“Add”按钮，弹出窗口加入行汇总，同样,数据全选默认。点击“OK”完成创建。

**4．**在属性编辑器Property Editor视图中选择“Column Area”，同样在左边的选项栏中选择第二个选项“Grand Totals”，点击编辑器右边的“Add”按钮，弹出窗口加入列汇总，同样,数据全选默认。点击“OK”完成创建。

创建完毕后，布局编辑器如下图：  


### 格式化与预览

格式化后如下图：



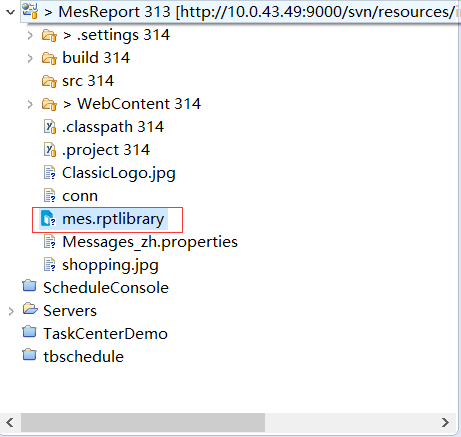
预览结果如下：

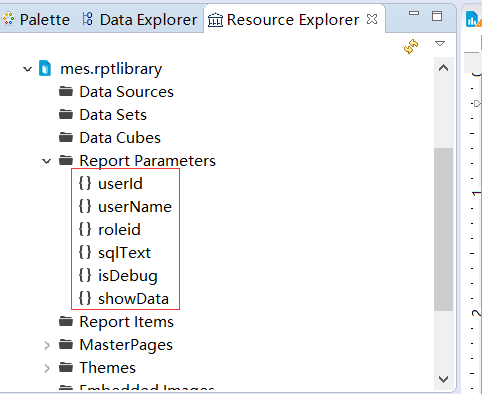


## **其他**

### 报表参数

根据项目实际需要，为了报表复用，已经定制了报表资源文件mes.rptlibrary，【mes.rptlibrary】含有报表基本样式，和额外添加的6个参数，由于mes.rptlibrary版本可能会更新，请联系项目组得到最新版的mes.rptlibrary





### 格式化报表

格式化报表是指对报表的显示形式进行设计，

我们以3.1的网格报表为例，来示例一下对报表的简单格式化处理：

#### 表头与标题

为报表加入标题并设置报表的表头。

我们在进行报表布局时，第一步曾要求大家在空白的报表中先加入一个1列2行的Grid，我们把表格都放入了地二格中，而此Grid的第一格则是我们准备用来加入表头和标题的。下面我们以3.1为例来加如前面试图所看见的标题与图片。

1. 从选用板“Palette”视图中再拖拽一个2行2列的Grid到第一格中（根据您的设计要求可以选用其他3\*3等样式的网格，）。
2. 在这个2\*2的Grid中，把第二行合并，方法如下：选中第二行，右键点击后从菜单中选择“Merge Cells”即可进行合并。
3. 从选用板“Palette”视图拖拽一个“Lable”到合并后的空格中（或者用右键点击后选择“Insert”→“Lable”插入即可），然后输入标题内容：客户信息一览表。
4. 在左上格中，从选用板“Palette”视图拖拽一个“Image”（或者右键点击后进行插入，同上），然后选择图片来源即可进行图片的插入。我们这里选择Embedded image，通过“Add”加入所用到的图片，然后选中以后确定即可。

#### 设定单元格格式

选中所要设置的Grid，在属性编辑器Property Editor中

选择“General”进行网格的基本样式设置，（注意，不仅是对于Grid，对于其他的报表元素，如“Lable”、“Table”、“Data”、“List”、“CrossTab”等这个选项都存在，进行的设置选项也基本一样）。比较常用的是：Text Alignment：网格中的文本对齐方式，我们这里选择居中。其他选项根据需要进行设置，这里不做解释，请大家在实际应用之中进行了解。“Background Color”设置背景颜色，我们这里不做设置；

对于其他选项，我们也不做设置，因为每个元素所要选用的样式不一样，而在Grid里设置的所有它自身网格里的样式，在这里我们就不用同意设置，而对不同元素我们进行不同的设置。

我们选中标题多对应的“Lable”，在属性编辑器Property Editor中对它的格式进行一些简单设置：

General：Color：RGB(70,21,162)； Size：Large； 字体加粗：

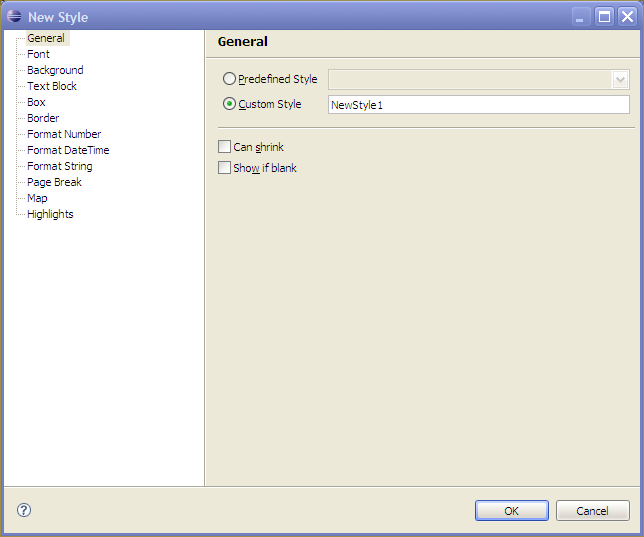
在对其他元素进行格式设置时，同样先选中它，然后通过Property Editor属性编辑器进行编辑即可，所进行的设置都大同小异，我们这里不再举例。

#### 样式套用

在同一张表中，如果多个相同元素（如本例中的几个标题Lable以及数据项Data）需要设定相同的样式，我们可以设定一个样式，然后所有需要用到的元素都套用此样式即可。

例如，我们在3.1中网格报表的列名所选用的都是同样的样式，方法如下：

**1、**右键点击所要加入样式的Lable，选择“Style”→“New Style”，进入新样式编辑页面：



**2、**在此页面，设置样式的字体、背景、边框等。

在例3.1中我们设置的有：

General：Custom Style：Lable Style（对样式命名）

Font：Color：White；Size：Large；Weight：Bold

Backgrand：Backgrand Color：RGB(70,21,162)

**3、**选中其他要需选用这个样式的Lable，右键点击后依次选择“Style”→“Apple Style” →“Lable Style”来套用这个样式就可以了。

**4、**样式编辑，若要对所建立的样式进行修改，可以通过右键点击选用此样式的元素以后，通过选择“Style”→“Edit Style” →“Lable Style” 对我们建立的样式进行修改。

#### 数据元素格式化

对某些特定格式的数据，如日期、时间、货币、电话号码等，我们希望在显示的时候能够以恰当的格式来进行显示，在BIRT也可以对此进行设置：

1. 选定表中数据项，在Property Editor属性编辑器视图中的默认项Properties中选择Format Number来设定数字的格式；选择Formt Date Time来设定时间日期的格式；选择Format String来设定字符型数据的格式。
2. 例如，我们要设置货币格式，选择Format Number，在“Format As”中选择“Currency”，  
   Use 1000s separator：选中表示使用千分符号  
   Symbol：选择货币标志，如美元、人民币等  
   Symbol position：货币标志的位置，一般选择Before
3. 如果要设置电话号码，选择Format String，在“Format As”中选择“Custom”，在“Custom Setting”中选择“Phone Number”，然后在Fomat Code中更改电话号码的格式就可以了。

### 分页、页眉与页脚

大多数报告分多页显示和打印。在设计报告时，您应该考虑多页报告的显示和打印方式。例如，可以将报告设计成按逻辑位置换页。对于包含一系列子报告的报告来说，可以指定让每个子报告都另起一页。

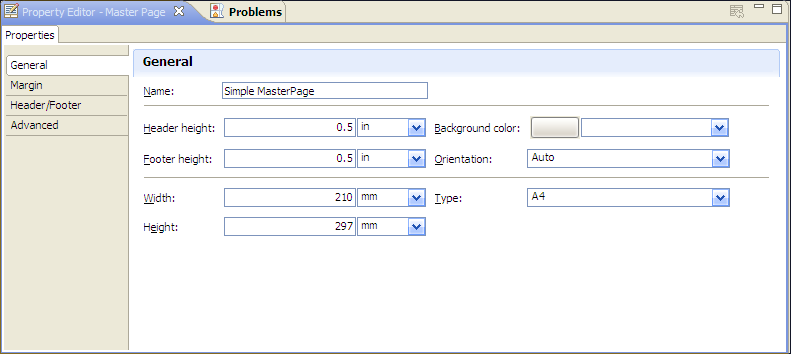
还可以设计能够增强报告的外观和可用性的页面布局。页面布局指定用于显示报告数据的页面的样式。例如，可以指定所有页面都在左上角显示报告标题、在右上角显示公司徽标、在右下角显示页号并在左下角显示报告生成日期。

#### 设置分页显示

选择要分页显示的表格Table，在属性编辑器Property Editor中的默认项Properties中选择“Page Break”，对Page Break Interval进行设置，是指在显示时每页显示的记录条数，如不设置就是默认的200条每页。

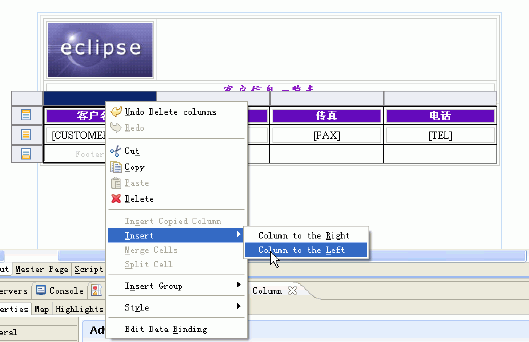
#### 设置分页打印

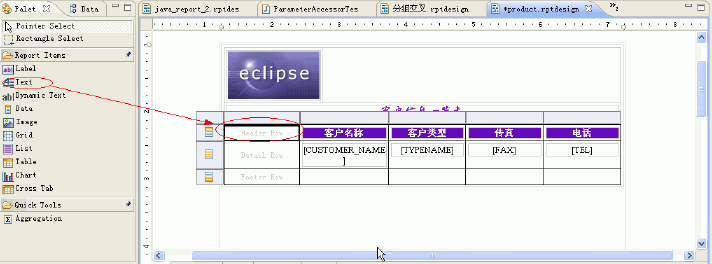
在布局编辑器中，选择“Master Page”选项卡，布局编辑器将显示主页。点击报表的空白处，下方试图出现主页的属性编辑器Property Editor- Master Page，选择General，在最后的Type中设置打印时的纸张格式，比如我们可以选择“A4”，表示在分页打印时采用A4纸大小。我们可以通过Doc视图或者PDF视图预览时查看纸张类型进行查看。



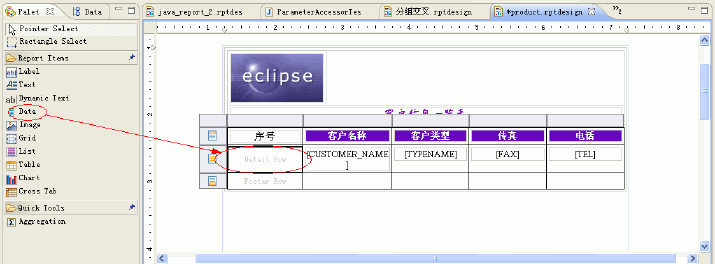
#### 设置页眉页脚

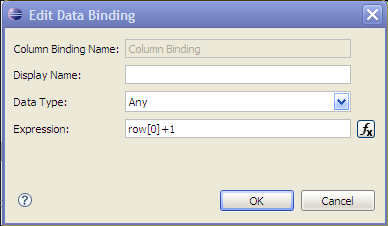
在布局编辑器中，选择“Master Page”选项卡。布局编辑器将显示主页。选用板将在“Auto Text”下显示专门用于主页的附加元素。

1. 要在页眉或页脚显示单个文本元素：将期望的元素从选用板中拖放到页眉或页脚。例如，要显示页码，从选用板中拖放Page 元素或 Page n of m 元素插入。
2. 要在页眉或页脚显示多个文本元素：在页眉或页脚插入网格元素“Grid”。请先从默认的页脚中删除日期元素。在插入网格“Grid”，指定网格的列数和行数。例如，如果想要在同一行显示两个元素，则指定 2 列和 1 行，然后将每个文本元素插入到一个网格单元格中。
3. 点击报表的空白处，下方试图出现主页的属性编辑器Property Editor- Master Page，选择General，可以设置页眉页脚的高度。报表中table添加行号
4. 在表中插入一列，选中表的第一列，点击右键，“insert”→“Column to the Left”加入新的一列；。如下图： 
5. 在第一列第一个单元格插入一个“文本”项，输入“序号”。



1. 在第一列第二个单元格中插入一个“Data”项。



1. 从选用板中拖拽一个“Data”元素至新列的“Detail Row中”，弹出对话框如下：

在表达式“Expression”中加入表达式：

row[0]+1

选择“OK”完成。

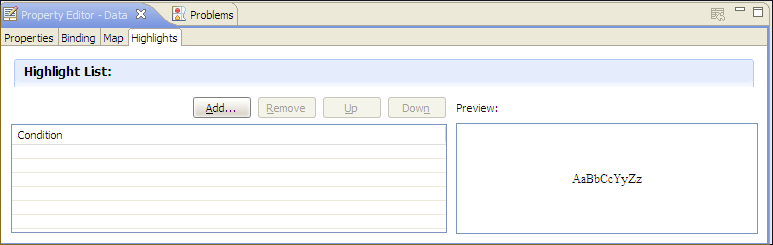
1. 预览结果：



### 隔行高亮度显示

**1、**在布局编辑器中，选择想要有条件格式化的报告元素。

在“属性编辑器”中选择“突出显示”选项卡。将出现“突出显示”页，如图所示。如果尚未对所选元素指定任何突出显示规则，则此页面是空的。



选择“Add”以添加新的突出显示规则。

**2、**在“新建突出显示规则”上，通过完成下列步骤来创建规则以对报告元素应用特定格式：

首先考虑纯英语格式的规则。例如：

If the order total is greater than $50,000.00, then set

the font color to blue and the font style to bold.

规则分为两个部分：If 和 Then。

“新建突出显示”对话框会帮助您指定规则的“If”部分和“Then”部分，方法是将它们分为更具体的部分。

**3、**通过完成下列步骤来指定规则的“If”部分：

在第一个字段中，指定“If”表达式的第一个部分。在使用示例规则时，此部分是订单总量：

如果订单总量值直接来自所选元素，则从下拉列表中选择“此数据项的值”。

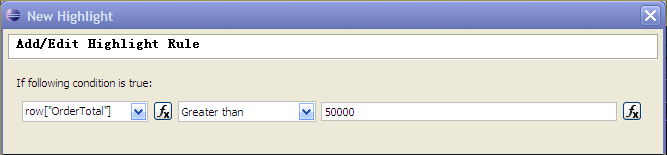
如果订单总量值来自另一个数据元素，则指定引用相关列绑定的表达式，该列绑定引用您需要的数据集值或计算出的值。以下表达式是一个示例：

row["OrderTotal"]

如果想要帮助构造表达式，则通过选择“”按钮来使用“表达式构建器”。

在第二个字段中，通过从列表中选择选项来指定“If”表达式的第二个部分。在使用示例规则时，此部分是“大于”。

在第三个字段中，指定“If”表达式的第三个部分。在使用示例规则时，此部分是值 50000，如图所示。



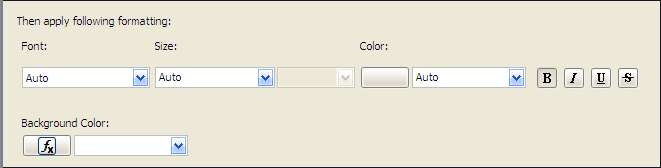
现在您已经完成了规则的 If 部分，它指定：

If row["OrderTotal"] is greater than 50000

**4、**通过完成下列步骤来指定规则的“Then”部分，即“then set the font color to blue and the font style to bold”：

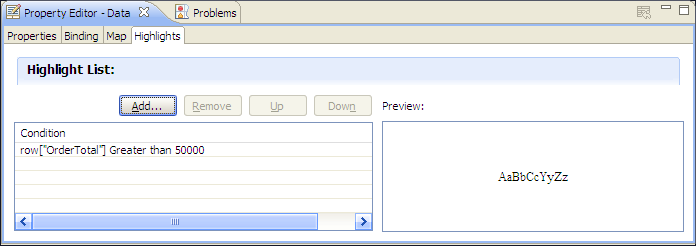
选择“Color”，然后从颜色拾取器中选择一种颜色。

选择 B 以设置粗体格式，如图所示。



选择“OK”。

创建的格式化规则将出现在“突出显示规则列表”中，如图所示。该规则将在下一次运行报告时生效。



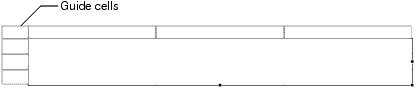
预览报告以测试格式化规则。

### 对报告进行布局

#### 添加|删除行或列

1. 选中行或列

在布局编辑器中，将鼠标指针悬浮在网格“Grid”左下角，直到显示“Grid”的选项卡为止，然后选择该选项卡。指导单元格出现在网格左上角，如图所示。



1. 插入与删除  
   右键单击想要添加行或列的位置旁的指导单元格，从上下文菜单中选择下列其中一项以在期望的位置添加行或列。

插入（Insert） → 行（Row） → 在上面（Above）

插入（Insert） → 行（Row） → 在下面（Below）

插入（Insert） → 列到右边（Column to the Right）

插入（Insert） → 列到左边（Column to the Left）

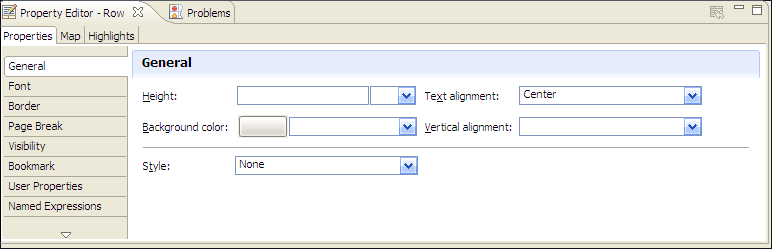
1. 合并与拆分

选中两个单元格（单击后长按住鼠标，然后拖动鼠标进行选择），右键点击，从上下文菜单中选择“Merge Cells”进行合并，合并后如要拆分，选中后右键点击，从上下文菜单中选择“Spilt Cell”进行拆分。

#### 指定行高或列宽

1、制定行高和列宽

选择网格或表左下角的选项卡。指导单元格出现在网格或表的左上角。选中一行或者一列。选择想要调整大小的行或列列的指导单元格。“属性编辑器”将显示行属性，如图所示。

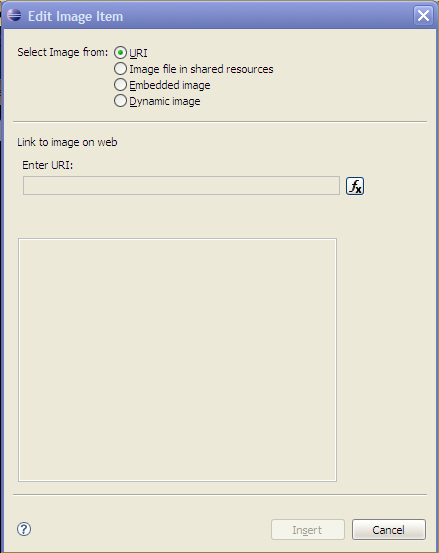


在“General”类别中，指定对应高度的值。此值将设置行的最小高度。可指定不同的度量单位，包括英寸、厘米、毫米和磅值。

#### 添加图像

我们此处介绍两种比较常用的方法：

1. 插入链接的图像：  
   将选用板中的图像元素“Image”拖放到页面上的期望位置。将出现“新建图像项”，如图所示。



要链接至存储在任何位置的图像：

在“Select Image from”中，选择 URI。

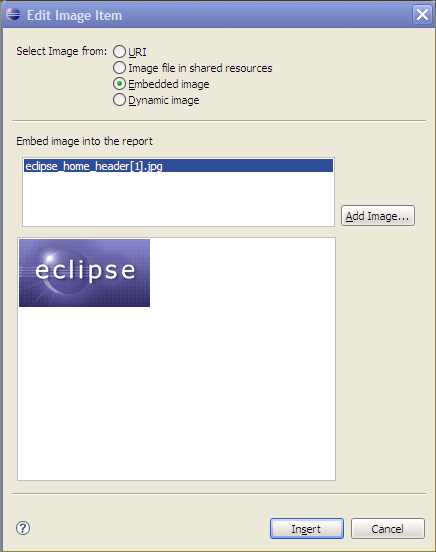
在“Link to Image on Web”中，指定图像文件的位置。用双引号（" "）将 URI 引起来。以下表达式是一个远程位置中的文件 URL 的示例。

"http://mysite.com/images/companylogo.jpg"

以下表达式是一个本地文件系统中的文件 URI 的示例：

file:///c:/myprojects/images/companylogo.jpg

1. 插入嵌入式图像：  
   在新建图像时在“Select Image from”中，选择“Embedded image” 如果先前插入了图像，则“新建图像项”会显示这些图像的名称，如图所示

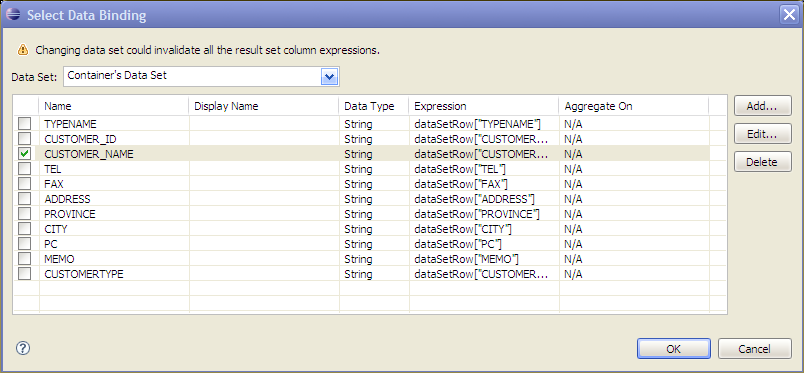


要嵌入新图像，请选择“Add Image…”，查找并选择要嵌入的图像，然后选择打开。此时该图像被显示在下面的预览框中。选择“Insert”。该图像出现在页面上。

### 数据绑定

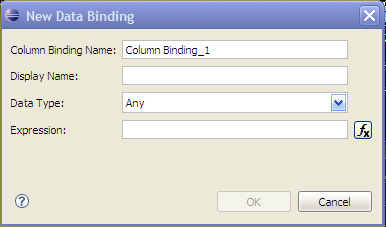
我们所创建的数据集将返回要在报告中使用的数据，在使用或显示报告中的数据集数据之前，必须先创建必需的数据绑定。正如第一个教程所演示的那样，要显示报告中的数据，只需将数据集字段从数据资源管理器“Data Explorer”中拖至布局编辑器中的表。BIRT 报告会根据你拖入的数据项自动创建数据绑定。

要查看数据绑定，通过右键点击数据项→选择“Change Data Column”可以查看绑定的数据项，也可以更改数据绑定。图中的这个对话框显示每次插入数据集字段时 BIRT 报告设计器创建的数据绑定。此数据绑定（称为列绑定）定义用于指定要显示的数据的表达式。



* + 创建新的列绑定：

从选用板中拖拽一个“Data”元素至一个空格中，或者选择这个空格右键点击→“Insert”→“Data”，点击“Add”，弹出如下对话框：



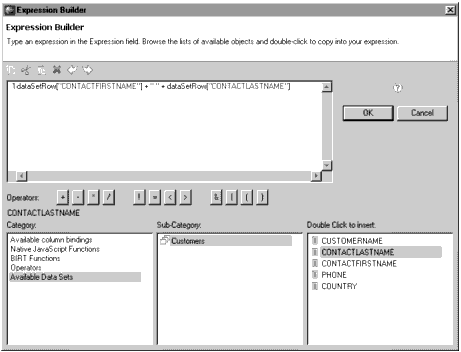
在“Column Binding Name”中制定列绑定名；

在“Data Type”中，选择适合于接下来指定的表达式所返回数据的数据类型。如果不确定数据类型，请使用缺省类型“任何”；

在表达式“Expression”中，使用下一步中的一个方法指定用于指示要返回的数据的表达式

* + 创建列绑定表达式：  
     可直接在“表达式”字段中输入表达式。

如果需要帮助构造表达式，选择“”启动表达式构建器。显示了表达式构建器中的一个表达式，该表达式用于将从Customers数据集中选择的两个数据集字段的值加在一起。完成构造表达式后，请选择“确定”。



编辑完毕，点击OK就可以创建列绑定。然后就可以将这个列绑定制定给“Data”元素了。

# Birt报表script

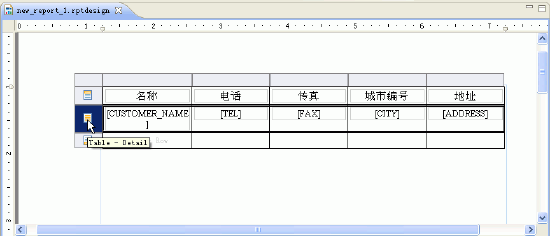
在使用 Eclipse 工作台开发 BIRT 报表时，您可以使用 Java 或 JavaScript 编写定制事件处理程序。在使用 Eclipse ECP（Eclipse CORBA Plugin (ECP)提供一个可针对CORBA IDL文件语法进行着色的编辑器）开发 BIRT 报表时，只能编写 JavaScript 事件处理程序。无论是使用 Java 还是 JavaScript，可以编写的事件处理程序集合是相同的。

可以使用 Java 来编写一些事件处理程序并使用 JavaScript 来编写另一些事件处理程序。如果同一个事件同时有 Java 和 JavaScript 事件处理程序，则 BIRT 将使用 JavaScript 处理程序。

## **avascript脚本编写**

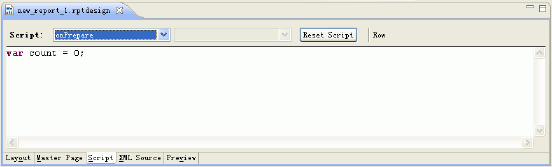
### 隔行高亮度显示

1. 在布局（Layout）中,选择Table－Detail的行。如下图：



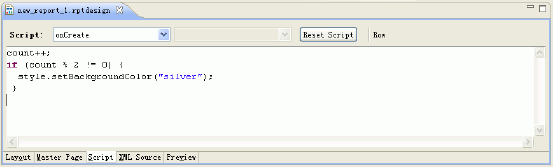
1. 然后选择Script，打开脚本窗口顶部的方法列表并选择onPrepare，如图所示。

var count=0; 错误更正:去掉var

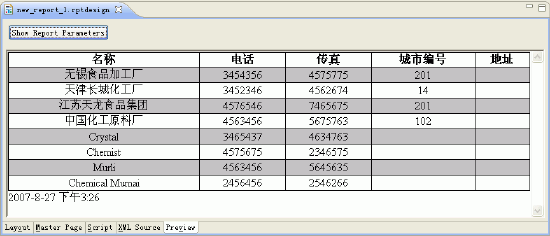


1. 最后打开脚本窗口顶部的方法列表并选择 onCreate，如图所示：

count++;  
if (count % 2 != 0) {  
  style.setBackgroundColor(silver);  
 }



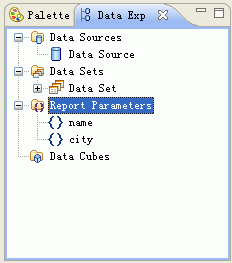
1. 保存，点击Preview预览结果如下图。



### 多参数查询

在多参数查询时，根据输入的参数不同，查询出不同的结果集，这就需要对查询sql语言进行拼凑。在birt环境下这种拼凑可以在data set的script编辑框中完成。前端包含在标签中的查询参数提交上来后，通过params[parameterName]即可引用。

1. 首先，按照3.3创建数据集的步骤创建数据集。Sql
2. 创建报表参数，sql需要的条件参数。如下图：



1. 点击Data Set（创建的数据集），然后点击编辑区的Script出现脚本编辑器。打开脚本窗口顶部的方法列表并选择brferOpern如下图：

**if**(!params[name].equals() || !params[city].equals())

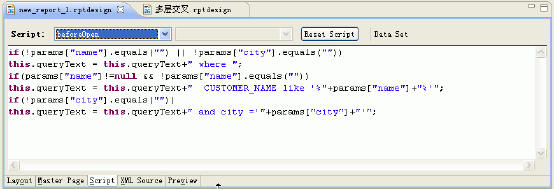
**this**.queryText = **this**.queryText+ where ;

**if**(params[name]!=**null** && !params[name].equals())

**this**.queryText = **this**.queryText+ CUSTOMER\_NAME like '%+params[name]+%';

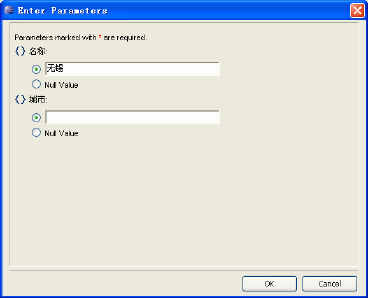
**if**(!params[city].equals())

**this**.queryText = **this**.queryText+ and city ='+params[city]+';

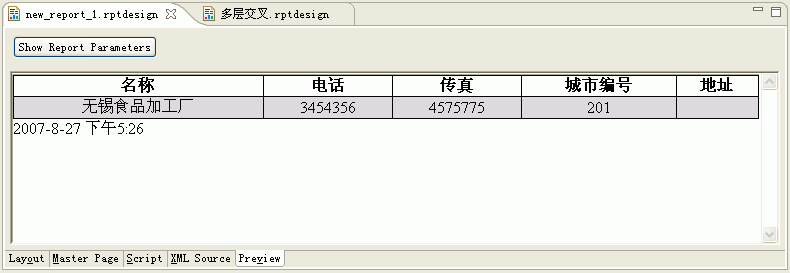


说明：**this**.queryText获取之前创建的没有参数的sql语句。

1. 保存，点击编辑区的Preview预览。会弹出输入参数的对话框。如下图：



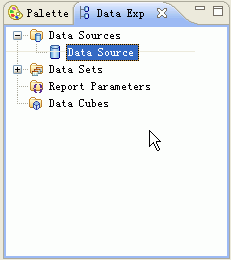
1. 点击ok，就在预览窗口看到根据条件查询结果。如下图：



### 脚本创建数据源

通过脚本从properties配置文件取得jdbc数据源信息。

1. 在已创建的jdbc数据源的基础编写脚本。
2. 选择数据资源管理器(Data Explorer)－>Data Source（创建的数据源），如下图：



1. 点击编辑区的Script脚本编辑器，打开脚本窗口顶部的方法列表并选择brferOpern如下图：

importPackage( Packages.java.io );

importPackage( Packages.java.util );

fin = **new** java.io.FileInputStream( **new** String("c:/temp/report.properties") );

**if**(fin!=**null**){

props = **new** java.util.Properties( );

props.load(fin);

}

extensionProperties.odaURL = **new** String(props.getProperty("dbURL"));

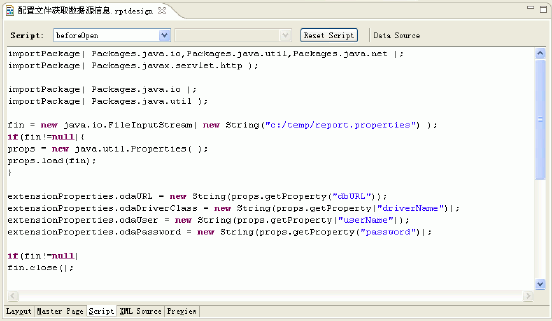
extensionProperties.odaDriverClass = **new** String(props.getProperty("driverName"));

extensionProperties.odaUser = **new** String(props.getProperty("userName"));

extensionProperties.odaPassword = **new** String(props.getProperty("password"));

**if**(fin!=**null**)

fin.close();



由于在这里的代码只支持绝对路径，在服务器部署的情况下可能不太适用，可以修改一下支持相对路径，相对于你发布的应用/WEB-INF/目录。如下：

importPackage( Packages.java.io,Packages.java.util,Packages.java.net );

importPackage( Packages.javax.servlet.http );

req = reportContext.getHttpServletRequest( );

propPath = **new** String(params["propFile"]);

**if** ( propPath.charAt(0) != "/" )

propPath = "/WEB-INF/" + propPath;

url= req.getSession().getServletContext().getResource( propPath );

props = **new** java.util.Properties( );

props.load( url.openStream( ) );

extensionProperties.odaURL = **new** String(props.getProperty("dbURL"));

extensionProperties.odaDriverClass = **new** String(props.getProperty("driverName"));

extensionProperties.odaUser = **new** String(props.getProperty("userName"));

extensionProperties.odaPassword = **new** String(props.getProperty("password"));

## **java脚本编写**

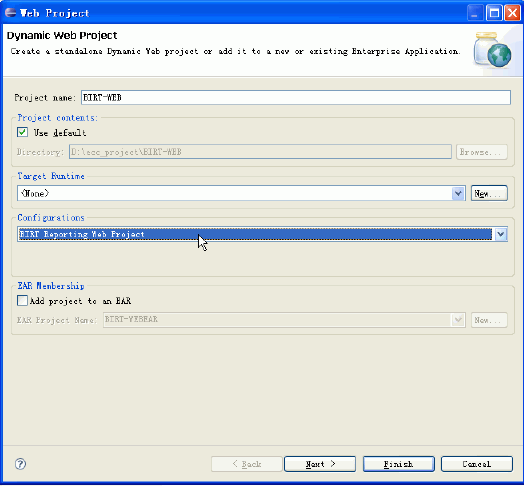
由于不是简单地在 BIRT 报表设计器中直接输入 Java 代码，所以与创建 JavaScript 事件处理程序相比，Java 事件处理程序的创建工作略为复杂。要创建 Java 事件处理程序类，必须编译 Java 类的源代码并确保该类对 BIRT 可视。但是，由于 Eclipse 是健壮的 Java 开发环境并支持将 Java 项目与 BIRT 项目集成，所以，为 BIRT 创建 Java 事件处理程序的过程已大为简化。

### 创建java类

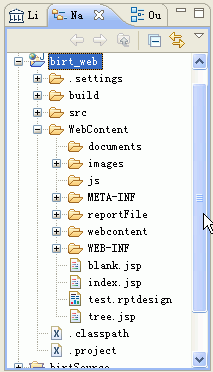
1. 创建java工程或者报表web工程。

如果创建报表web工程，可以将报表创建在web目录下，方便测试。

* + 选择File->New－>Web Project,在窗口Configuretions下拉菜单选择BIRT Reporting Web Project。如图：

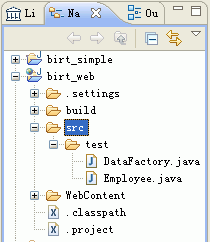


* + 点击Finish，完成报表工程创建。如图：



1. 编写java类

在工程src目录下创建类。如图：



* + 创建Employee类，职工信息类。

**package** test;

**public** **class** Employee {

**private** String id = "";

**private** String name="";

**private** String sex="";

**private** String address ="";

**public** **void** setId(String id){

**this**.id = id;

}

**public** String getId(){

**return** **this**.id;

}

**public** **void** setName(String name){

**this**.name = name;

}

**public** String getName(){

**return** **this**.name;

}

**public** **void** setSex(String sex){

**this**.sex = sex;

}

**public** String getSex(){

**return** **this**.sex;

}

**public** **void** setAddress(String address){

**this**.address = address;

}

**public** String getAddress(){

**return** **this**.address;

}

}

* + 创建DataFactory类，数据集合类。

**package** test;

**import** java.util.ArrayList;

**import** java.util.List;

**public** **class** DataFactory {

/\*\*

\* 定制数据集合，这里也可以从数据库获取

\* **@return**

\*/

**public** **static** List getData(){

Employee emp1 = **new** Employee();

emp1.setId("00501");

emp1.setName("张三");

emp1.setSex("男");

emp1.setAddress("西安");

Employee emp2 = **new** Employee();

emp2.setId("00502");

emp2.setName("李四");

emp2.setSex("男");

emp2.setAddress("北京");

Employee emp3 = **new** Employee();

emp3.setId("00503");

emp3.setName("王五");

emp3.setSex("男");

emp3.setAddress("深圳");

List list = **new** ArrayList();

list.add(emp1);

list.add(emp2);

list.add(emp3);

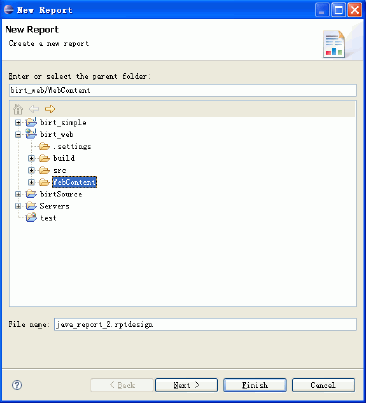
**return** list;

}

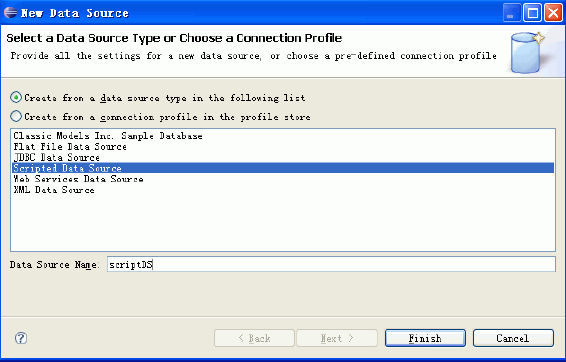
}

### java类与报表交互

1. 创建报表。在前面已经讲过怎么创建报表，在这里只要选择报表存放目录为工程的WebContent。如图：

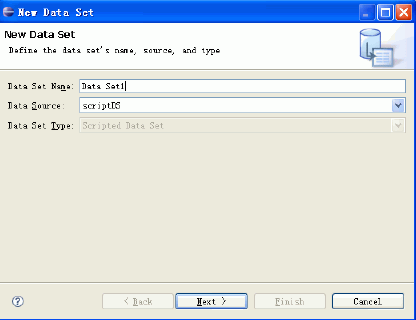


1. 在报表文件中新建一个脚本数据源。如下图：

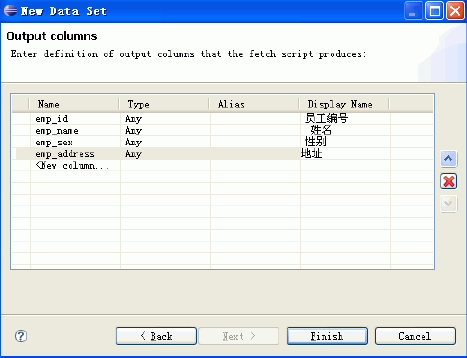


点击Finish，完成脚本数据源的创建。

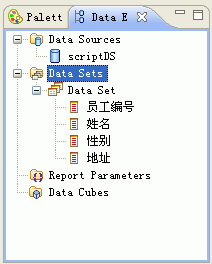
1. 创建一个数据集。
   * + 1. 数据类型，如下图：



* + - 1. 点击Next，为数据集定制输出列。如下图：



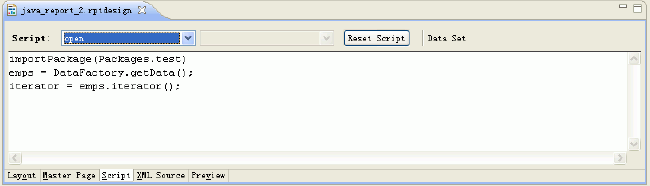
点击Finish，完成数据的创建。如下图：



1. java类与报表交互

选择Data Set（创建的数据集），在报表设计视图点击Script选项卡进入脚本编辑界面。

* 打开Script脚本窗口顶部的方法列表并选择open方法，输入下列代码。



importPackage(Packages.test)

emps = DataFactory.getData();

iterator = emps.iterator();

* 打开Script脚本窗口顶部的方法列表并选择fetch方法，输入下列代码。

**if**(iterator.hasNext() == **false** ){

**return** **false**;

}

**var** emp = iterator.next();

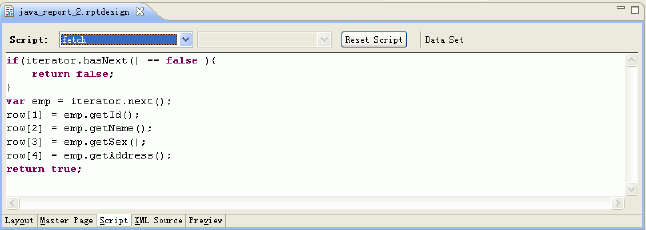
row[1] = emp.getId();

row[2] = emp.getName();

row[3] = emp.getSex();

row[4] = emp.getAddress();

**return** **true**;

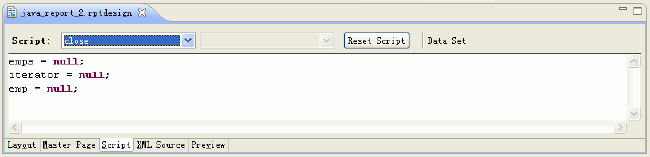


* 打开Script脚本窗口顶部的方法列表并选择close方法，输入下列代码。如下图：

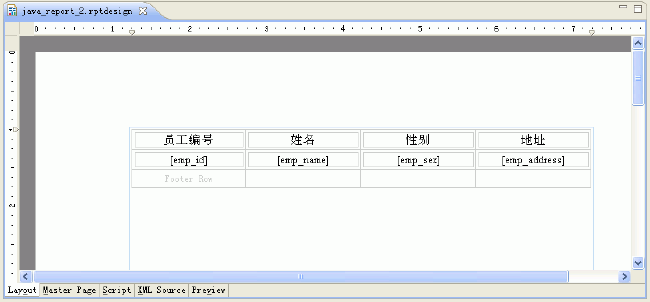
emps = **null**;

iterator = **null**;

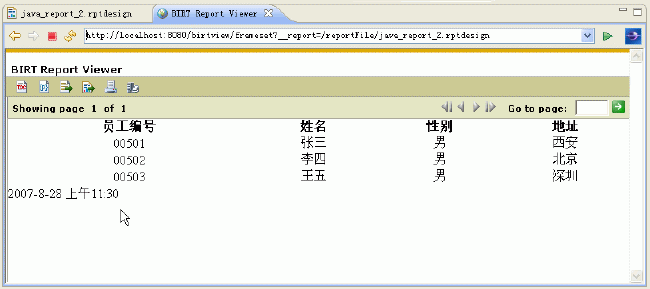
emp = **null**;



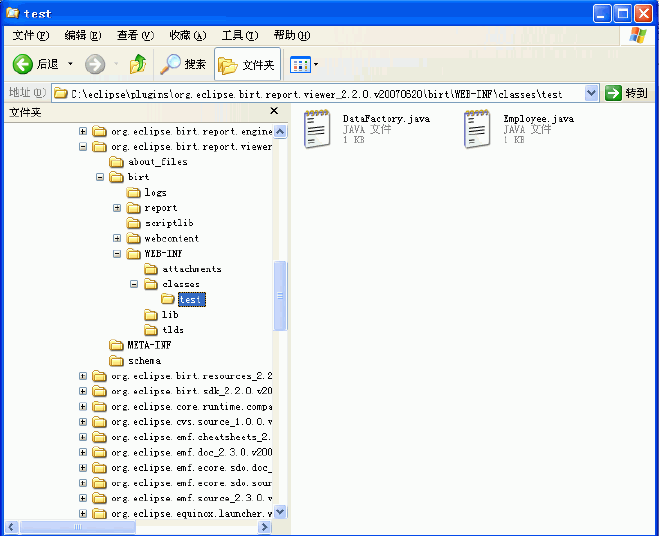
1. 在报表设计视图点击Layout（布局），选项卡进入报表布局编辑界面。如下图：



1. 启动服务器测试报表。结果如下图：

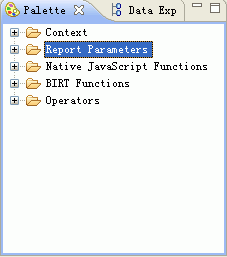


注意：如果在报表设计视图的Preview预览，需要将java拷贝到如下目录：C:\eclipse\plugins\org.eclipse.birt.report.viewer\_2.2.0.v20070620\birt\WEB-INF\classes。



## **BIRT脚本参考**

在设计视图点击Script，然后点击Palette，可以看BIRT脚本导航目录。如下图：



* Context――报表的上下文信息。
* Report Parameters ――报表参数。
* Native JavaScript Functions ――本地javascript函数。
* BIRT Functions ―― 报表函数。
* Operators ―― 操作。

BIRT Script API参考：

<http://help.eclipse.org/help31/index.jsp?topic=/org.eclipse.birt.doc/enginescript/api/index.html>

# Birt报表部署

运行资源：

Server：Tomcat5.5

Runtime：birt-runtime-2\_2\_0

Jdk：jdk1.5.0\_02

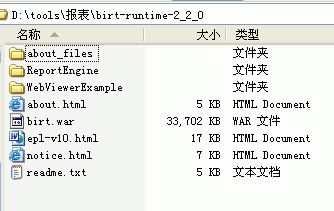
## **直接使用运行birt-runtime-2\_2\_0包中的WebViewerExample**

1. 下载birt-runtime-x\_x\_0运行包。如下：

[](http://www.eclipse.org/downloads/download.php?file=/birt/downloads/drops/R-R1-2_2_0-200706261010/birt-runtime-2_2_0.zip%26url=http://download.actuatechina.com/eclipse/birt/downloads/drops/R-R1-2_2_0-200706261010/birt-runtime-2_2_0.zip%26mirror_id=385)

**Download from:**

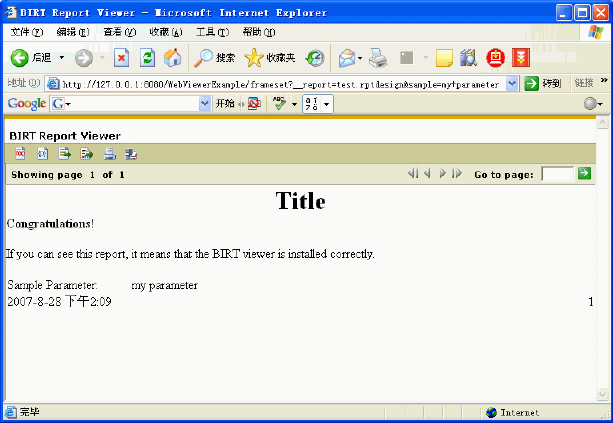
1. 下载解压包完整目录如下：



1. 将WebViewerExample，复制到％tomcat\_home%\webapps\目录下。
2. 将报表文件\*.rptdesign 和\*.rptconfig文件放于% tomcat-Home%\webapps\ WebViewerExample目录下。
3. 启动tomcat服务，浏览器地址栏输入：

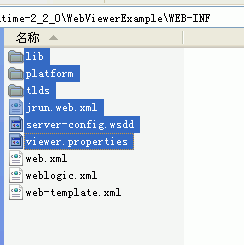
<http://127.0.0.1:8080/WebViewerExample/frameset?__report=test.rptdesign>

可以通过地址<http://127.0.0.1:8080/WebViewerExample/>测试，点击页面的 [View Example](http://127.0.0.1:8080/WebViewerExample/frameset?__report=test.rptdesign&sample=my+parameter) 。看到如下图所示表示部署成功：



## **集成到已有web应用**

1. 与5.1的1、2步骤相同。
2. 将WebViewerExample\ WEB-INF\目录下面的lib、platform、tlds、jrun.web.xml、server-config.wsdd、viewer.properties拷贝到要集成的目标WEB应用的\ WEB-INF\目录下。如下图：



1. 将WebViewerExample\ webcontent文件夹拷贝到要集成的目标WEB应用的根目录下。
2. 在web.xml文件中添加如下信息。

<display-name>Eclipse BIRT Report Viewer</display-name>

<!-- Default locale setting -->

<context-param>

<param-name>BIRT\_VIEWER\_LOCALE</param-name>

<param-value>en-US</param-value>

</context-param>

<!-- Report resources directory for preview. Default to ${birt home} -->

<context-param>

<param-name>BIRT\_VIEWER\_WORKING\_FOLDER</param-name>

<param-value></param-value>

</context-param>

<!-- The generated document files directory. Default to ${birt home}/documents -->

<context-param>

<param-name>BIRT\_VIEWER\_DOCUMENT\_FOLDER</param-name>

<param-value></param-value>

</context-param>

<!-- If usr can only access the reprot resources under working folder. Default is false -->

<context-param>

<param-name>WORKING\_FOLDER\_ACCESS\_ONLY</param-name>

<param-value>false</param-value>

</context-param>

<!-- Output image/chart directory. Default to ${birt home}/report/images -->

<context-param>

<param-name>BIRT\_VIEWER\_IMAGE\_DIR</param-name>

<param-value></param-value>

</context-param>

<!-- Engine log directory. Default to ${birt home}/logs -->

<context-param>

<param-name>BIRT\_VIEWER\_LOG\_DIR</param-name>

<param-value></param-value>

</context-param>

<!-- Report engine log level -->

<context-param>

<param-name>BIRT\_VIEWER\_LOG\_LEVEL</param-name>

<param-value>WARNING</param-value>

</context-param>

<!-- Directory to store all birt report script libraries (JARs). Default to ${birt home}/scriptlib -->

<context-param>

<param-name>BIRT\_VIEWER\_SCRIPTLIB\_DIR</param-name>

<param-value></param-value>

</context-param>

<!-- Resource location directory. Default to ${birt home} -->

<context-param>

<param-name>BIRT\_RESOURCE\_PATH</param-name>

<param-value></param-value>

</context-param>

<!-- Preview report max rows limited. -->

<context-param>

<param-name>BIRT\_VIEWER\_MAX\_ROWS</param-name>

<param-value></param-value>

</context-param>

<!-- If always overwrite generated document file. For runtime,efalult to true -->

<context-param>

<param-name>BIRT\_OVERWRITE\_DOCUMENT</param-name>

<param-value>true</param-value>

</context-param>

<!-- Define BIRT viewer configuration file -->

<context-param>

<param-name>BIRT\_VIEWER\_CONFIG\_FILE</param-name>

<param-value>WEB-INF/viewer.properties</param-value>

</context-param>

<!-- If turn on the function that supports print on the server side. Default to on. -->

<context-param>

<param-name>BIRT\_VIEWER\_PRINT\_SERVERSIDE</param-name>

<param-value>ON</param-value>

</context-param>

<!-- If force optimized HTML output. Default to true -->

<context-param>

<param-name>HTML\_ENABLE\_AGENTSTYLE\_ENGINE</param-name>

<param-value>true</param-value>

</context-param>

<!-- Viewer Servlet Context Listener -->

<listener>

<listener-class>org.eclipse.birt.report.listener.ViewerServletContextListener</listener-class>

</listener>

<!-- Viewer HttpSession Listener -->

<listener>

<listener-class>org.eclipse.birt.report.listener.ViewerHttpSessionListener</listener-class>

</listener>

<!-- Viewer Servlet, Support SOAP -->

<servlet>

<servlet-name>ViewerServlet</servlet-name>

<servlet-class>org.eclipse.birt.report.servlet.ViewerServlet</servlet-class>

</servlet>

<!-- Engine Serlvet -->

<servlet>

<servlet-name>EngineServlet</servlet-name>

<servlet-class>org.eclipse.birt.report.servlet.BirtEngineServlet</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>ViewerServlet</servlet-name>

<url-pattern>/frameset</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>ViewerServlet</servlet-name>

<url-pattern>/run</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>EngineServlet</servlet-name>

<url-pattern>/preview</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>EngineServlet</servlet-name>

<url-pattern>/download</url-pattern>

</servlet-mapping>

<servlet-mapping>

<servlet-name>EngineServlet</servlet-name>

<url-pattern>/parameter</url-pattern>

</servlet-mapping>

<taglib>

<taglib-uri>/birt.tld</taglib-uri>

<taglib-location>/WEB-INF/tlds/birt.tld</taglib-location>

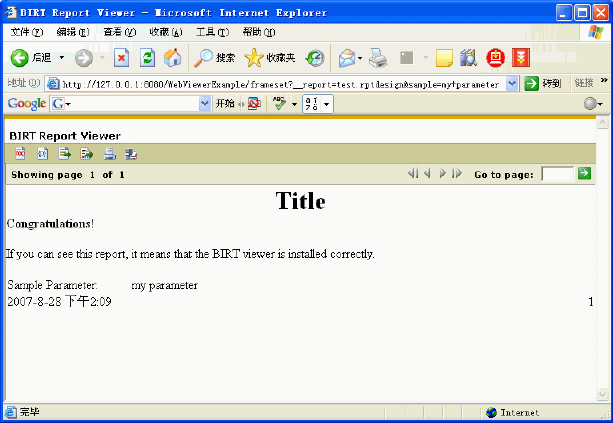
</taglib>

1. 将报表文件拷贝到目标应用的根目录下或者/report目录下面。

我们可以通过将5.1根目录下的test.rptdesign拷贝到我们的目标应用程序的根目录下，在地址栏输入：

[http://127.0.0.1:8080/【my web】/frameset?\_\_report=test.rptdesign](http://127.0.0.1:8080/【my%20web】/frameset?__report=test.rptdesign)

看到如下图所示表示部署成功：



## **数据库驱动位置**

把所使用的数据库的JDBC驱动jar包复制到/webapps/WebViewerExample/WEB-INF/platform/plugins/org.eclipse.birt.report.data.oda.jdbc\_2.2.0.v20070615/drivers目录下。

## **报表url说明**

<http://127.0.0.1:8080/WebViewerExample/frameset?__report=report/reportName.rptdesign>说明如下：

1. **Servlet模式说明：**

**Frameset** ―― 采用Ajax框架，可以显示工具条，导航条和TOC面板，实现复杂的操作，如分页处理，导出数据，导出报表，打印等等。该模式下会自动生成report document文件(预览report design文件)到特定的目录(用户可以用参数指定，也可以定义在web.xml里)。采用Ajax，速度较慢。

**run** ――也采用Ajax框架，但不实现frameset的复杂功能，不会生成临时的report document文件(预览report design文件)，也不支持分页，这个主要是应用在BIRT Designer里的preview tab里，可以支持cancel操作，其它不怎么常用。采用Ajax，速度较慢。

**Preview** ――没有用到Ajax框架，直接调用底层Engine API对报表进行render，把生成的报表内容直接输出到浏览器。这种模式和run模式调用的是相同的Engine API，唯一区别在于run采用Ajax获取报表内容，而preview直接输出到浏览器。如果要支持分页，用户需要在URL上定义\_\_page和\_\_pagerange参数，这两个参数也会在后面详细说明。需要特别说明的是，在这几种预览模式中，preview的速度是最快的。

**Document** ――该模式主要是为了从report design文件生成report document文件。用户可以在URL上提定document文件生成存放的路径(存放在server端)，如果未指定，会直接生成rptdocument发送到客户端浏览器，用户可以下载到客户端。

**output** ――该模式类似于frameset，会自动生成report document文件(预览report design文件)，区别在于output不采用Ajax，而是将生成的报表内容直接输出到浏览器。

**parameter** ――该模式主要用于生成一个参数对话框，一般用户不常用，用户可以直接通过提供的JSP Tag--parameterPage去实现参数对话框，不需要直接调用。

**download** ――用于导出报表数据为CSV格式，当你使用frameset工具条里的导出数据功能时，会用到这个模式。

1. **url参数说明：**

**\_\_report**

定义要预览的rptdesign文件路径，支持相对路径和绝对路径，相对路径是相对于web.xml中定义的工作目录。

**\_\_document**

定义要预览的rptdocument文件路径，同样支持相对和绝对路径。相对路径是相对于web.xml中定义的工作目录。在同时定义了\_\_report和\_\_document参数时，以\_\_document为优先，如未找到相应的document文件，才会从design文件生成document文件(frameset/output)或是直接去render这个design文件(preview/run)。

**\_\_title**

定义报表显示的标题。

**\_\_showtitle**

是否显示frameset模式下上方的标题部分。true | false

**\_\_toolbar**

是否显示frameset模式下的工具条。true | false

**\_\_navigationbar**

是否显示frameset模式下的导航条。true | false

**\_\_parameterpage**

是否强制弹出或不弹出报表参数对话框。true | false

**\_\_format**

输出报表的格式，默认为html。现在支持：pdf | doc | xls | postscript | ppt

**\_\_locale**

设置Locale信息，如 \_\_locale=zh\_CN， 注意必须是国家加语言。

**\_\_svg**

设置chart输出是否以SVG格式输出。true | false，frameset和run模式下，会采用javascript判断客户端浏览器是否支持svg，但并非对所有浏览器有效。

**\_\_bookmark**

设置页面要定位的书签名字。

**\_\_istoc**

指定定位的书签是不是一个TOC名字。如为true,就会根据\_\_bookmark参数值去获取一个真正的书签名，从而实现正常的跳转。这个主要用于定位到一个TOC上。

**\_\_rtl**

指定HTML页面输出是否需要right to left。支持不同国家的阅读习惯，如阿拉伯国家是从右到左的。

**\_\_page**

指定要输出的报表页数，这个依赖于报表的分页设计(page break)。

**\_\_pagerange**

指定要输出的报表页数范围。如1,3,5-9。

**\_\_resourceFolder**

定义资源目录路径。同web.xml中的BIRT\_RESOURCE\_PATH设置。

**\_\_asattachment**

是否以附件方式下载报表，如生成PDF或是其它格式里。默认是inline。

**\_\_masterpage**

是否要显示master page。true | false

**\_\_designer**

该参数主要是应用在BIRT Designer环境下，如会读取cache的报表参数等等，一般不用。true | false

**\_\_overwrite**

该参数同web.xml定义的参数，不过web.xml里是全局设置，在URL上通过参数可以定义本次操作的设置。

**\_\_imageID**

内部参数，用于image的引用，一般不用。

**\_\_maxrows**

设置Dataset查询的最大记录数，要注意这个设置是全局的，会影响后面所有的请求。主要用于BIRT Designer下，提高报表设计效率。同web.xml中的BIRT\_VIEWER\_MAX\_ROWS设置。

**\_\_maxlevels**

设置查询获取Cube的最大级数。同上面的\_\_maxrows，也主要用于BIRT Designer设计环境。

同web.xml中的BIRT\_VIEWER\_MAX\_CUBE\_LEVELS设置。

**\_\_cubememsize**

同web.xml中的BIRT\_VIEWER\_CUBE\_MEMORY\_SIZE参数设置。

**\_\_instanceid**

如果查看BIRT输出的HTML代码，你就可以看到一些HTML Element会有一个iid的属性(如table)，这个就是instanceid。这个是Engine动态生成的，不可提前预知。所以你需要从HTML代码中得到这个值。该参数主要是为了获取reportlet(报表片断，如只输出报表中的一个Table或是一个Chart)。需要配合\_\_isreportlet参数。

**\_\_isreportlet**

指定当前输出是不是一个reportlet。true | false

特别说明：为了输出一个reportlet，BIRT现在提供两种方式。

* 为要输出的对象(表格或是Chart)定义一个bookmark，然后可以用下面的URL输出reportlet.

http://localhost:8080/birt/frameset?\_\_report=test.rptdesign&\_\_bookmark=bk&\_\_isreportlet=true

* 采用instanceid，但这个值事先是无法预知的，需要预览一次后从HTML代码中得到。然后用下面的URL输出reportlet.

http://localhost:8080/birt/output?\_\_report=test.rptdesign&\_\_instanceid=iid&\_\_isreportlet=true

注意，reportlet只支持document文档。如果是预览design文档去输出reportlet，就必须要使用frameset/output(自动生成document文档)。

**\_\_clean**

BIRT里临时生成的一些文件都是和session相关的，比如临时document文件，还有image文件。这些文件也可以通过session进行管理，这个参数就是指定是否需要在session timeout的时候清除这些临时文件。默认值是true。

true | false

**\_\_dpi**

可以设置输出Chart的dpi数值。

**\_\_fittopage**

暂时这个参数只对PDF和postscript格式报表有效，指定是否调整至适合页面。

**\_\_pagebreakonly**

暂时这个参数只对PDF和postscript格式报表有效，指定是否只采用BIRT报表内定的分页设置。这个参数一般需要和\_\_fittopage联合使用。

**\_\_agentstyle**

同web.xml中的HTML\_ENABLE\_AGENTSTYLE\_ENGINE参数设置。

**后台Server端打印相关参数**

**\_\_action**

定义执行的指令名称。当前只支持print指令，用于后台服务器打印。

**\_\_printer**

后台打印机名称。

**\_\_printer\_copies**

对应打印机的打印份数参数。

**\_\_printer\_collate**

对应打印机的双面打印参数。

**\_\_printer\_duplex**

对应打印机的duplex参数。

**\_\_printer\_mode**

对应打印机的模式参数。是单色还是彩色。

**\_\_printer\_pagesize**

对应打印机的纸型参数。比如A4。

**JSP Tag相关参数**

**\_\_id**

viewer的ID号，这个参数一般不常用，主要用于JSP Tag中，如在一个页面插入两个BIRT Viewer，而且预览同一个报表文件，这时候因为在一个session下面，所以需要用不同的ID去生成单独的document文件。不至于都生成同一个document文件上，从而引发冲突。

**\_\_pattern**

在JSP Tag中用于指定要提交的Servlet Pattern名字，如frameset/output/run/preview等。主要用于采用parameter模式生成parameter dialog对话框时。

**\_\_target**

可以指定提交到的窗口名称。如\_blank,\_self等。

**\_\_nocache**

指定是否会用到cache的报表参数值，这些cache的值一般保存在rptconfig文件里。在设计报表并预览的时候，可以保存输入的报表参数值。这个在runtime的时候不常用。

**报表参数相关**

**\_\_isnull**

指定当前的报表参数为null值，后面是报表的参数名。

**\_\_islocale**

指定当前的报表参数值是和Locale/Format相关的，必须用特定的Locale/Format转化参数值(从String转化为Object)。格式为\_\_islocale=paramName。

**\_\_isdisplay\_\_**

指定报表参数的displayText值,格式为\_\_isdisplay\_\_paramName=displayText。可以在报表中引用displayText值，如params[p1].displayText。在URL上传displayText时如下(报表参数名为p1)：&\_\_isdisplay\_\_p1=hello

**\_\_islocale\_\_**

指定该报表参数值是Locale/Format相关的，同时给定了参数值。格式为\_\_islocale\_\_paramName=paramValue。

**Export Data参数**

**\_\_exportEncoding**

该参数应用于导出数据为CSV中，可以指定导出的文件编码，如GBK或是GB2312等。

**\_\_sep**

该参数应用于导出数据为CSV中，可以指定数据分隔符，如逗号，冒号等。

**\_\_exportdatatype**

该参数应用于导出数据为CSV中，可以指定是否输出数据类型。true | false

**ResultSetName**

要导出数据的记录集名字。

**SelectedColumnNumber**

要导出的栏位数。

**SelectedColumn**

要导出的数据栏位名称。

## **web.xml配置**

1. **[BIRT\_VIEWER\_LOCALE]**

设置默认的Locale信息，暂时没有太大意义。因为Locale的信息，首先以URL上定义的\_\_locale为准，如果没有定义，会找到当前浏览器的Locale信息，最后才会用到这里定义的信息。

1. **[BIRT\_VIEWER\_WORKING\_FOLDER]**

设置BIRT Viewer的工作目录。用户可以把report design或是report document文件存放在这个目录下，这样就可以在URL上采用相对路径去预览这些报表文件了。默认是当前根目录。

当前支持三种形式：

相对路径 --- 这个相对当前的WEB应用的context root.

绝对路径

JAVA系统变量 --- 可以在启动服务器时，定义JVM的系统变量，如java –Dmyworkingfolder=D:/reports。这样就可以在web.xml中用${myworkingfolder}进行引用了。

1. **[BIRT\_VIEWER\_DOCUMENT\_FOLDER]**

设置生成的document文件的存放路径。默认是documents目录。路径设置同上。

1. **[WORKING\_FOLDER\_ACCESS\_ONLY]**

简单的报表访问限制控制实现，如果设为true,哪就只能预览存放在工作目录下的报表文件。默认值是false。

1. **[BIRT\_VIEWER\_IMAGE\_DIR]**

设置生成的临时图片的存放路径。默认是report/images目录。路径设置同工作目录设置。

1. **[BIRT\_VIEWER\_LOG\_DIR]**

设置生成的日志文件存放路径。默认是logs目录。路径设置同工作目录设置。

1. **[BIRT\_VIEWER\_LOG\_LEVEL]**

设置日志的level，可选的值有：ALL|SEVERE|WARNING|INFO|CONFIG|FINE|FINER|FINEST|OFF。级别由高到低。

1. **[BIRT\_VIEWER\_SCRIPTLIB\_DIR]**

设置用户script lib文件的存放目录( 在报表中用到的Java Event Handler Class )。默认值是scriptlib。路径设置同工作目录设置。

1. **[BIRT\_RESOURCE\_PATH]**

设置用户资源存放路径，这些资源包括library文件，image文件等。默认是当前根目录。路径设置同工作目录设置。

1. **[BIRT\_VIEWER\_MAX\_ROWS]**

设置获取dataset的最大记录数。主要应用于设计报表的时候，预览报表如果记录数太多，会花费很多的时间，也可能会引起out of memory问题。默认是不限制。

1. **[BIRT\_VIEWER\_MAX\_CUBE\_LEVELS]**

设置CUBE查询的最大级数。和前面的参数作用类似。默认是不限制。

1. **[BIRT\_VIEWER\_CUBE\_MEMORY\_SIZE]**

设置在生成CUBE时，可以写在memory中的最大值，单位是MB。可以提高效率，写在内存会比直接写在硬盘快很多。但同时也要注意内存占用的问题。

1. **[BIRT\_OVERWRITE\_DOCUMENT]**

该参数主要用于frameset/output模式，它们会生成临时的document文件上。如果设为true，则每次刷新页面时，都会重新去生成document文件，如果为false，则不会重新生成，只会用原来的document文件去生成报表内容。

1. **[BIRT\_VIEWER\_CONFIG\_FILE]**

定义properties文件的路径，不可以修改。

1. **[BIRT\_VIEWER\_PRINT\_SERVERSIDE]**

在frameset工具条上，提供有后台服务器打印的功能，该参数可以设置是打开还是关闭后台打印的功能。默认是打开。可选值为: ON 和 OFF。

1. **[HTML\_ENABLE\_AGENTSTYLE\_ENGINE]**

这个参数是会传递给Engine的，主要用于一些CSS的兼容性方面的问题。默认值是true。

## **viewer.properties说明**

viewer.properties文件主要是定义一些扩展的参数。

# configurable variable for JSP base href. Please uncomment the below line.

#base\_url=http://127.0.0.1:8080

该设置主要应用于代理服务器的情况下，在使用代理服务器后，从request里获取的URI并非真正的URI，需要在这里定义。

# [EXTENSION SETTING]

viewer.extension.html=html

viewer.extension.pdf=pdf

viewer.extension.postscript=ps

viewer.extension.doc=doc

viewer.extension.xls=xls

viewer.extension.ppt=ppt

定义输出的报表文件的后缀名，和format相关联。

# [OUTPUT FORMAT LABEL NAME]

viewer.label.html=HTML

viewer.label.pdf=PDF

viewer.label.postscript=PostScript

viewer.label.doc=Word

viewer.label.xls=Excel

viewer.label.ppt=PowerPoint

定义导出报表对话框里的报表格式列表，和format相关联，这样名字会更有意义。

# [CSV SEPARATOR]

viewer.sep.0=,

viewer.sep.1=;

viewer.sep.2=:

viewer.sep.3=|

viewer.sep.4=\t

支持多种CSV分隔符，用户也可以增加新的分隔符(只支持char，而不是string)。但同时需要修改JSP文件和Messages.properties文件。

# [LOGGERS]

# "logger."+class=level

# if no level is specified or the text "DEFAULT",

# then the default level from the web.xml will be used

logger.org.eclipse.datatools.connectivity.oda=DEFAULT

logger.org.eclipse.datatools.enablement.oda=DEFAULT

当前的日志都是通过Engine输出的，如果需要其它组件的日志输出，可以在这里定义。注意格式必须为logger.org……。而且该组件必须实现了java的logger。

可以单独为该组件设置日志级别，如果设为DEFAULT，就会使用web.xml里的设置。

## **BIRT在jsp中tag使用**

1. Jsp中view标签的使用

对应 frameset/run 模式，基于Ajax。

<%@ page language="java" contentType="text/html; charset=utf-8"

pageEncoding="utf-8"%>

<%@ taglib uri="/birt.tld" prefix="birt" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>Viewer tag example-frameset</title>

</head>

<body>

<birt:viewer id="1"

reportDesign="test.rptdesign"

position="absolute"

width="500"

height="300"

left="50"

top="50"

format="html">

</birt:viewer>

</body>

</html>

参数说明：

|  |  |  |  |
| --- | --- | --- | --- |
| 参数名 | 参数说明 | 参数值 | 默认值 |
| id | 唯一标识 | 数字类型 |  |
| pattern | 用什么模式的servlet预览报表 | frameset，run | frameset |
| title | 报表标题 |  |  |
| showToolBar | 是否显示工具条 | false、true | true |
| showTitle | 是否显示标题 | false、true | true |
| isHostPage | 根据页面大小显示 | false、true | false |
| scrolling | 根据框架IFrame设置显示，但isHostPage值为true时，起作用。 | Auto、 yes 、no |  |
| position | 设置报表容器的样式位置，  isHostPage值为true时，起作用。 |  |  |
| style | 报表容器的样式，isHostPage值为true时，起作用。 |  |  |
| height | 报表高度，单位象素isHostPage值为true时，起作用。 |  |  |
| width | 报表宽度，单位象素isHostPage值为true时，起作用。 |  |  |
| top | 报表顶部的距离，单位为象素，isHostPage值为true时，起作用。 |  |  |
| left | 报表左边的距离，单位为象素，isHostPage值为true时，起作用。 |  |  |
| frameborder | 是否显示ifrem框架的边框，isHostPage值为true时，起作用。 | Yes、 No | no |
| reportDesign | 报表文件的路径或者url |  |  |
| reportDocument | 报表document的名字，可以是绝对路径或者url |  |  |
| showParameterPage | 是否显示参数页面。 | false、ture | true |
| showNavigationBar | 是否显示分页导航。 | false、ture | true |
| forceOverwriteDocument |  |  |  |
| resourceFolder |  |  |  |
| pageRange |  |  |  |
| reportletId |  |  |  |
| locale |  |  |  |
| bookmark |  |  |  |
| svg | 是否以矢量图格式输出 |  |  |
| format | 指定报表输出的格式 |  |  |
| rtl | 报表显示顺序，false为从左到右、true从右到左 | False、true | false |
| pageNum |  |  |  |

1. Jsp中Report标签使用

对应 preview模式。

1. Jsp中Parameter标签使用

是 View、Report标签的sub tag。

1. Jsp中ParameterPage标签使用

可以自定义参数对话框，提供用户自定义选项。

<%@ page language="java" contentType="text/html; charset=GBK"

pageEncoding="GBK"%>

<%@ taglib uri="/birt.tld" prefix="birt" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=GBK">

<title>ParameterPage tag example</title>

</head>

<body>

<birt:parameterPage

id="report1"

name="page1"

reportDesign="param\_group.rptdesign"

isCustom="true"

showNavigationBar="true"

showToolBar="true"

pattern="frameset">

国家: <birt:paramDef id="1" name="country" />

<br><br>

省: <birt:paramDef id="2" name="state" />

<br><br>

市: <birt:paramDef id="3" name="city" />

<br><br>

<input type="submit" name="buttom" value="提交" />

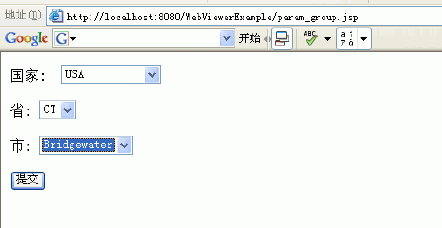
<br><br>

</birt:parameterPage>

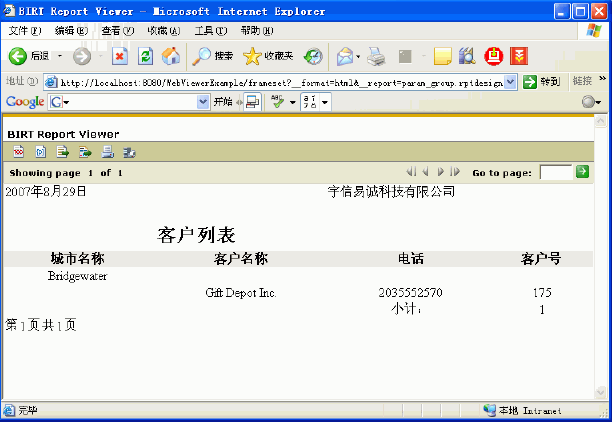
</body>

</html>

预览如下图：



提交如下图：



1. Jsp中Parameter Definition标签使用

parameterPage 标签的sub tag, 支持用户自定义参数。

<%@ page language="java" contentType="text/html; charset=utf-8"

pageEncoding="utf-8"%>

<%@ taglib uri="/birt.tld" prefix="birt" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

<title>ParameterPage tag example - user</title>

</head>

<body>

<birt:parameterPage id="parameterPage"

name="form1"

isCustom="true"

pattern="run"

reportDesign="test.rptdesign"

target="\_blank"

width="600"

height="400"

format="html">

<p align="center">This is a user-defined parameter page.</p>

<br/>

<p align="center">

Sample Parameter: <input type="text" name="sample">

<br/><br/>

<input type="submit" value="submit form">

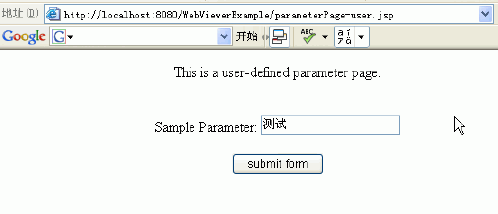
</p>

</birt:parameterPage>

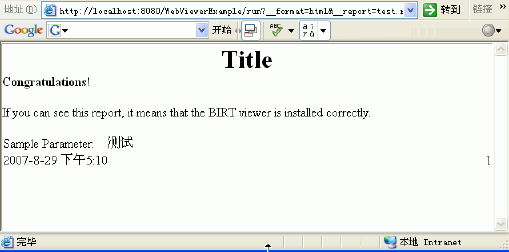
</body>

</html>

预览如下：



点击“submit form”，如下图：



## **关于BIRT编码**

BIRT报表使用”UTF-8”编码，在通过URL传递参数时，都需要对参数编码转换。如：

<%

String param1=java.net.URLEncoder.encode("公用信息表","UTF-8");

%>

<a href="<%=request.getContextPath()%>/frameset?\_\_report=test.rptdesign&sample=<%= param1%>">View Example</a>