

# VB6.0 数据报表对象 技术及应用

王青 柳虹 杨东勇 (杭州浙江工业大学信息工程学院 310014)

**摘要:** 本文介绍了使用VB6.0数据报表(DataReport)对象定制报表的实现方法。讨论了实现过程中若干值得注意的重要技术环节的解决方法,给出了设计实例,提出了高效开发打印报表的新途径。

**关键词:** 数据报表 数据源 打印

## 1 引言

报表打印是大多数应用系统必须具备的功能。在 windows 系统中,所有打印的工作最终都被 windows 系统接管,所以 Visual Basic 6.0 的打印也是针对 windows 系统的打印。在 windows 的应用程序中打印是最复杂的任务之一。对 VB 程序员而言,创建报表是开发应用系统过程中最繁琐的工作。如果用 print 方法和图形方法创建自己的报表函数,会花很多时间和精力。如果使用第三方的内置报表功能,如 Crystal Report,虽然可以提供配置报表的简单方法,但是存在一系列包括额外成本和 DLL 配置等其他问题。

Visual Basic6 中 Microsoft 新增加了一种在应用程序中创建报表的方式: Microsoft 数据报表设计器 (Microsoft Data Report designer),它是一个多功能的报表生成器,提供了创建联合分层结构报表的能力。当同数据源,如数据环境设计器 (Data Environment designer) 一起使用,可以从几个相关的不同表创建报表。除创建可打印报表之外,也可以将报表导出到 HTML 或文本文件中。数据报表能很容易地从一个 ADO (ActiveX Data Objects) 数据源中显示一个带有打印和输出按钮的预览打印屏幕。用户所要做的是提供数据和报表格式。我们在开发“车辆称重系统”时,其中“车辆信息查询模块”、“车辆称重模块”、“车辆信息汇总表的打印模块”都有报表及打印功能。采用 VB6.0 的 DataReport 技术,设计开发了报表功能,取得了良好的效果。

## 2 报表打印的实现方法

### 2.1 建立数据源

由于数据报表(DataReport)只能显示 ADO (ActiveX Data Objects) 数据源中的数据,因此必须利用 ODBC 管

理器先建立一个 ODBC 数据源。然后,在窗体的 load 事件中加入代码以便打开数据源。在窗体的 unload 事件中加入代码关闭数据源,并且释放本身占用的内存资源。具体代码如下:

```
Private Sub Form_Load()
    Set cn = New ADODB.Connection
    cn.Open "dsn=chengzhong" ' 打开数据源
End Sub

Private Sub Form_Unload(Cancel As Integer)
    rs.Close
    cn.Close ' 关闭数据源
    set rs=nothing
    set cn=nothing ' 释放内存资源
End Sub
```

### 2.2 设置数据源 (DataSource)

为了在数据报表(DataReport)中显示数据必须指定数据报表的 DataSource。数据报表(DataReport)的 DataSource 可以在属性中直接指定,例如将数据报表(DataReport)和数据环境设计器 (Data Enviroment designer) 直接相连显示数据。这种实现方法只需在属性域中直接指定数据报表(DataReport)的 DataSource 为数据环境设计器(Data Enviroment)的名称。这一方法是不能按照用户的实际需求动态地输出所需的报表,特别是要满足我们所涉及的系统所要求的按照用户指定日期动态地输出报表。我们必须采用 VB 代码,用 Sql 查询所得的 ADO (ActiveX Data Objects) 纪录集动态地当做数据报表(DataReport)的数据源 (DataSource) 的方法。具体代码如下:

```
sl = "select kehu.客户名, zhongbiao.车号, Sum
```

```
(zhongbiao.净重) AS 总净重,count(zhongbiao.净重) as
总车数 FROM huo INNER JOIN (kehu INNER JOIN
zhongbiao ON kehu.客户号 = zhongbiao.客户号) ON
huo.货号 = zhongbiao.货号 where zhongbiao.日期
between #" & begin_date & "# and #" & end_date
& "# group by zhongbiao.车号
```

上述代码应用 Sql 查询语句实现用户要求的根据指定日期输出一张包含客户名、车号、总净重、总车数的报表,并且此报表按照车号排序。

```
Set rs = cn.Execute(s1)
```

```
Set DataReport1.DataSource = rs ' 用代码为
DataReport 对象设置 DataSource
```

### 2.3 数据报表设计

首先要在 Visual Basic 工程中加入一个新的 DataReport 对象。方法是选择 Project ,再选择 Add Data Report,把名为 DataReport1 的新 DataReport 对象加入到 Project Explorer 窗口中。通常情况下,程序中每个 DataReport 对象表示一个单独的报表。但这种报表是固定报表,不能根据自己需要或临时变化自由调整,有一定的局限性。如果在系统中有多张风格相同,但内容不同的报表,需要做大量重复的编程工作。例如,在我们的项目开发过程中,单就“车辆信息汇总表的打印模块”来说,如果按常规方法在报表相应的区域上画出报表控件的话,则必须设计不同内容的报表 10 多张。但实际上本质不同的报表只有三张。因此,采用对一个 DataReport 对象采用 VB 代码处理多张表的方法。这样既减少了编程量,也减少了程序执行过程中系统资源的利用,优化了应用系统的性能。

## 3 报表打印具体实现技术

### 3.1 标签控件的处理

首先在数据报表(DataReport)的相应位置画出不附带内容的标签控件(Rptlabel),该标签代表的具体内容在程序中用代码添加。对于数据报表(DataReport)的“报表标头(Report Header)”和“页标头(Page Header)”的标签控件(Rptlabel)的Caption 属性的设置,程序采用如下代码:

```
If Option3(0).Value = True Then
```

```
one_string = " 货名 "
```

```
End If
```

```
If Option3(1).Value = True Then
```

```
one_string = " 车号 "
```

```
End If
```

```
DataReport1.Sections("section4").Controls("label2").
```

```
Caption = one_string
```

### 3.2 文本控件的处理

数据报表(DataReport)的细节区域(Detail Header)是显示 ADO 数据源中每一条记录的那部分报表。要使细节区域与数据库的数据相连必须添加文本控件(RpttextBox),实现与数据字段的绑定。与标签控件(Rptlabel)的处理一样,首先在细节区域(Detail Header)相应位置画出不附带内容的文本控件(RpttextBox),当显示"Unbound"时,表明还没与数据字段相连。此时,如果不用代码实现,则直接在它的属性域的DataField 属性中输入要显示的数据库中的字段名。这种方法只能实现一个数据报表(DataReport)对象对应与一张输出报表,不能实现一个数据报表(DataReport)对象输出多张报表的功能。为了实现一个数据报表(DataReport)对象实现多张报表的输出功能,本文采用了程序代码的实现方法,具体代码如下:

```
If Option3(0).Value = True Then
```

```
one_string = " 货名 "
```

```
End If
```

```
If Option3(1).Value = True Then
```

```
one_string = " 车号 "
```

```
End If
```

```
DataReport1.Sections("section1").Controls("text1").
```

```
DataField = one_string
```

上述处理的优点用图 1 所示的车辆信息汇总表来说明:当用户选择“附加条件”的“货名”,那么就在报表中输出“货名”,若用户选择“附加条件”的“车号”,那么就在报表中输出“车号”,这样就可以利用一个数据报表(DataReport)对象实现多张报表的输出。

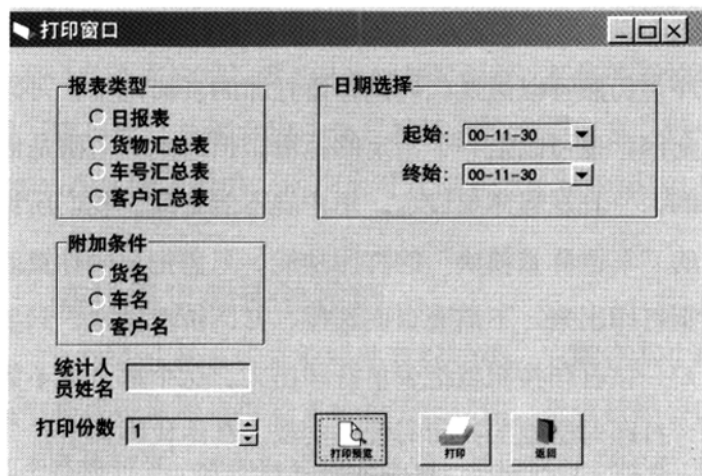


图 1 车辆信息汇总表的打印模块界面

### 3.3 打印预览实现

用 DataReport 实现报表打印预览非常容易,用如下代码实现。

```
Private Sub Comprint_show_Click()
    DataReport1.Show
End sub
```

### 3.4 打印功能实现

打印一个报表时通常有显示打印对话框的打印和不显示对话框的打印两种。打印对话框允许用户选择打印机型号、打印到文件、打印的页面范围并指定要打印的份数。

#### 3.4.1 显示“打印对话框”的打印

实现显示“打印对话框”的打印,我们采用两种方法。代码如下:

方法一:通过 CommonDialog 控件控制打印

On Error GoTo Cancel '捕捉错误

CommonDialog1.PrinterDefault = True '选择默认打印机

CommonDialog1.ShowPrinter '显示“打印”对话框

CommonDialog1.Copies = Text2.Text '允许用户设定打印数,在程序中加以控制

```
For intloopindex = 1 To CommonDialog1.Copies
```

```
DataReport1.PrintReport False
```

```
Next intloopindex
```

方法二:直接利用数据报表(DataReport)实现显示“打印对话框”

```
DataReport1.PrintReport True
```

#### 3.4.2 不显示“打印对话框”的打印

在有些情况下,不允许用户干预打印报表。PrintReport 提供不显示“打印对话框”打印的方法。当然,程序员仍然可以通过代码选择要打印的页面范围,可以是全部,也可以是一个指定的范围,但选定的范围是固定的,一旦在程序里设定,用户就不能更改。我们所设计的“车辆称重模块”的打印功能,只需把车辆称重的数据打印出来,不需要页面选择,打印份数设定,只要用户一按打印按钮就把表单打印出来。这个部分用不显示“打印对话框”打印的代码实现。具体代码如下:

```
DataReport1.PrintReport False '默认打印全部内容也可以用代码指定打印的页面范围:
```

```
DataReport1.PrintReport False, rptRangeFromTo, 1, 2
```

'表示打印1-2页之间的表单,但不显示打印对话框

## 4 数据报表设计重要技术细节的解决

### 4.1 报表宽度

在使用数据报表(DataReport)时,如果遇到一行要输出的数据项非常多,在打印预览时很可能出现这样的错误“报表宽度大于纸的宽度”。解决的办法是:在数据报表的属性中减少 LeftMargin 的值,把 RightMargin 的值设为0,这样报表的内容就可以尽可能地往两边扩展,当然,报表内容不能超过纸的宽度。

### 4.2 不涉及数据库的报表设计

有时,用户想要输出的报表可能根本不涉及数据库的内容,如在我所设计的“车辆称重模块”中的打印部分,该部分要实现如下功能:实时打印车辆过磅秤时通过传感器传到电脑的一系列数据,以及操作人员姓名、操作时间等信息。这些信息都可从界面上获得,无需调用数据库。那么就在页眉(Page Header)部分设计整张报表,且全部使用标签控件(Rptlabel)。但有一点必须注意:在程序中调用该张报表,依然必须为它设定数据源(DataSource),否则会出现提示为“无效数据源”的错误。尽管事实上这个数据报表根本不涉及任何数据库内容,根本不需要任何数据源。因此必须在程序中添加类似的代码。

```
Set rs = cn.Execute("select * from huo,zhongbiao,kehu where huo.货号=zhongbiao.货号 and kehu.客户号=zhongbiao.客户号")
```

```
Set DataReport4.DataSource = rs
```

### 4.3 行间距和控件高度的调整

在报表设计时,如果想要根据自己的需要调整报表行与行之间的间距,以及控件的高度,可以单击右键取消对“抓取到网格”的选择。这样就可自动调整标签控件(Rptlabel)、文本控件(RpttextBox)的高度,否则,只可调整这两个控件的宽度,高度必须与网格一致,将影响整个报表的美观。

## 5 结束语

目前,VB的有关资料中往往只介绍固定模式使用数据报表(DataReport),限制了数据报表对象的应用。本文介绍的用程序方式来使用数据报表对象的方法和技巧,能有助于充分挖掘数据报表对象的功能,简化报表设计的开发工作。■