

PowerBuilder 中数据窗口的复制方法及性能分析

高延红 (济南山东师范大学物理系 250014)

董祥军 (济南山东轻工业学院计算机科学与技术系 250100)

摘要: 详细讨论了数据窗口复制的五种方法, 并对各自的性能进行了比较。这五种方法是使用 GetItem/SetItem 函数复制、使用剪贴板复制、使用结构数组复制、使用 RowsCopy 函数复制以及对数据窗口直接赋值进行复制。

关键词: PowerBuilder 数据窗口 复制

1 引言

在用 PowerBuilder 开发应用系统时, 会经常用到数据窗口的复制。PowerBuilder 提供了多种数据窗口的复制方法, 但不同的方法性能也不一样, 尤其是复制的数据量较大时差别比较明显。本文详细讨论了数据窗口复制的五种方法, 并对各自的性能进行了比较。

在以下的讨论中, dw-source 表示源数据窗口控件, dw-destination 表示目标数据窗口控件, 我们的任务是要把 dw-source 中的数据复制到 dw-destination 中。用于数据测试的数据窗口对象为 d_student, 共有四列: id (varchar), class (varchar), name (varchar), birthday (date), 分别表示学号、班级、姓名、出生日期; dw-source 中共有数据 3767 行, 约有 105KB, 测试过程中用到了一个自定义函数 of-calculate-time, 用于计算复制时间, 其脚本为:

```
Long ll-End
Double ldb-copy-time
ll-End = CPU()
```

// 将毫秒单位转换为秒

```
ldb-copy-time = (ll-End - il-Begin) / 1000
```

```
Return ldb-copy-time
```

测试界面如图 1 所示, 硬件为 C400、96MRAM。

2 数据窗口复制的五种方法

2.1 使用 GetItem/SetItem 函数

这种方法是先用 GetItemString(), GetItemNumber(), GetItemDate() 等函数从原数据窗口中读取每一行的数据, 然后用 InsertRow() 函数在目标数据窗口中插入一空行, 再用 SetItem() 函数来设置数据。

单击按钮“使用 GetItem/SetItem 函数”的单击事件脚本为:

```
long ll-Count, ll-RowCount
string S-Data
Date d-Data
SetPointer(HourGlass!)
// 清除目标数据窗口数据
```

```
dw-destination.Reset()
dw-destination.SetRedraw(False)
// 求出总行数
ll-RowCount = dw-source.RowCount()
// 开始计时
il-Begin = CPU()
For ll-Count = 1 To ll-RowCount
// 在目标数据窗口插入一空行
dw-destination.InsertRow(0)
// 用 GetItem... 函数取得数据, 用 SetItem 函数写入数据
S-Data = dw-source.GetItemstring(ll-Count, "d")
dw-destination.SetItem(ll-Count, "d", S-Data)
S-Data = dw-source.GetItemstring(ll-Count, "class")
dw-destination.SetItem(ll-Count, "class", S-Data)
S-Data = dw-source.GetItemstring(ll-Count, "name")
dw-destination.SetItem(ll-Count, "name", S-Data)
d-Data = dw-source.GetItemDate(ll-Count, "birthday")
dw-destination.SetItem(ll-Count, "birthday", d-Data)
Next
// 显示时间
st-getitem.Text = String(of-calculate-time())
dw-destination.SetRedraw(True)
```

2.2 使用剪贴板

这种方法是先用 SaveAs() 函数将源数据窗口中的数据复制到剪贴板上, 然后再用 Import String() 函数复制到目标数据窗口中。

Methods of Datawindow Copy and Their Performance Analysis in PowerBuilder

单击按钮“使用剪贴板”的单击事件脚本为:

```
SetPointer(HourGlass!)
dw-destination.Reset()
il-Begin = CPU() //
复制数据到剪贴板
dw-source.SaveAs("", Clipboard!, False)
// 将剪贴板中的数据复制到目标数据窗口中
dw-destination.ImportClipboard()
// 显示时间
st-clipboard.Text = String(of-calculate-time())
```

2.3 使用结构数组

这种方法是先将源数据窗口中的数据复制到一个结构数组中, 然后再将该数组中的数据复制到目标数据窗口中。这里用到一个结构类型 os-data, 定义时要与数据窗口各列的类型匹配。

单击按钮“使用结构数组”的单击事件脚本为:

```
// 定义数组
os-data lst-Data []
SetPointer(HourGlass!)
dw-destination.Reset()
il-Begin = CPU()
// 将数据从源始数据窗口复制到结构数组中
lst-Data = dw-source.Object.Data
```

```
// 将结构数组中的数据复制到目标数据窗口中 dw-
destination.Object.Data = lst-Data
st-array.Text = String(of-calculate-time())
```

2.4 使用直接赋值

使用这种方法时直接对数据窗口进行赋值。单击按钮“使用直接赋值”的单击事件脚本为:

```
SetPointer(HourGlass!)
dw-destination.Reset()
il-Begin = CPU()
// 直接将数据复制到目标数据窗口中
dw-destination.Object.Data = dw-source.Object.Data
st-direct.Text = String(of-calculate-time())
```

2.5 使用 RowsCopy 函数

这种方法是使用 RowsCopy() 函数, 该函数可以复制部分行到另一个数据窗口中。其语法结构为:

```
dwcontrol.RowsCopy (startrow, endrow, copybuffer,
targetdw, beforerow, targetbuffer) \
```

其中:

dwcontrol: 源数据窗口控件 (或 DataStore, 或 child DataWindow);

startrow: 复制内容的起始行号;

endrow: 复制内容的终止行号;

copybuffer: 被复制数据所在的缓冲区, 其值有:

Primary!, Delete!, Filter!

targetdw: 目标数据窗口控件 (或 DataStore, 或 child DataWindow), 可以与 dwcontrol 相同;

beforerow: 目标位置的行号, 被复制的数据将插入到该行号前面, 如果想复制到最后, 只需用一个大于总行数的值;

targetbuffer: 目标数据窗口缓冲区, 其值有:

Primary!, Delete!, Filter!

该函数执行成功时返回 1, 出错时返回 -1。

单击按钮“使用 RowsCopy 函数”的单击事件脚本为:

```
long ll-RowCount
SetPointer(HourGlass!)
dw-destination.Reset()
// 求出总行数
ll-RowCount = dw-source.RowCount()
il-Begin = CPU()
// 复制数据, 从第一行开始
dw-source.RowsCopy(1, ll-RowCount, Primary!, dw-
destination, 1, Primary!)
st-rowscopy.Text = String(of-calculate-time())
```

3 性能分析

从图 1 中的数据可以看出, 使用 RowsCopy() 函数时速度最快, 而使用 GetItem/SetItem 函数时速度最慢。但使用 GetItem/SetItem 函数也有自己的优点, 当要复制的行有一定的条件要求时, 可以在循环中加上适当的条件语句。另外运行时间不同, 这些数据也会略有差异。 ■

参考文献

- 1 温为民等, Power Builder 7.0 实例应用进阶 [M], 机械工业出版社, 2000.
- 2 樊金生等, Power Builder 6.5 使用教程 [M], 科学出版社, 2000.6.
- 3 PowerBuilder Code Examples.

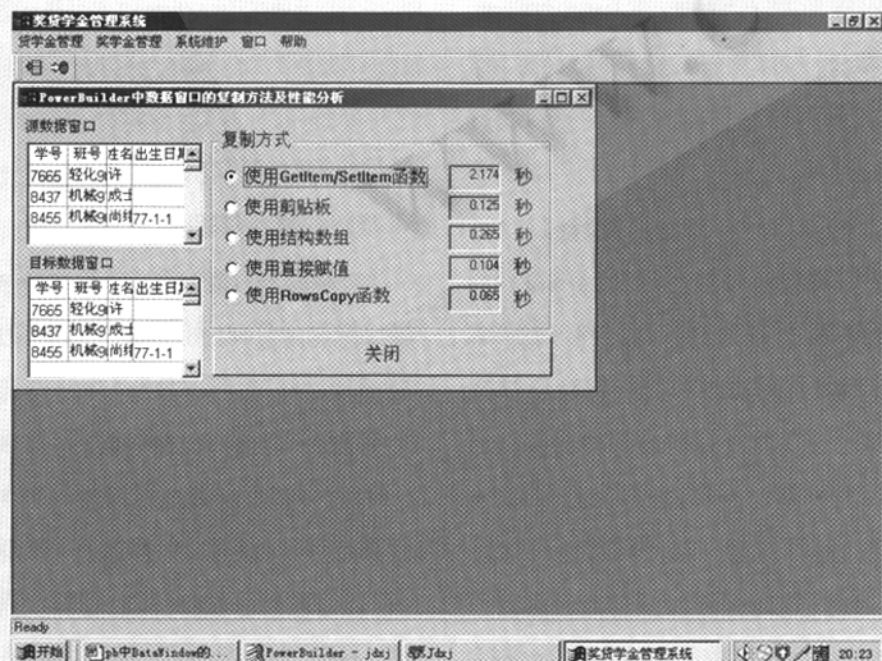


图 1 数据窗口复制及性能分析示意图