

基于批量复制的高效 Excel 套打方法^①

王 俊, 张以维

(北京复兴路 22 号自动化站, 北京 100842)

摘 要: 针对应用系统中广泛存在的数据报表输出问题, 提出了一种基于 Excel 套打的方法, 该方法通过批量复制和模板的动态修改, 可以快速完成大数据量的报表输出。该方法的实现简单、输出的报表规范, 通过性能对比实验, 该方法在上万条数据记录的情况下具备良好的性能。

关键词: 数据报表; Excel 套打; 批量复制; Delphi

Efficient Excel Template-Print Based on Batch Copying

WANG Jun, ZHANG Yi-Wei

(Automation Station of FuXing Road N0.22, Beijing 100842, China)

Abstract: Many application systems face the problem of data reporting. This paper proposes one method based on Excel template-print. By batch copying and dynamic modification of Excel template, the approach can make a large amount of data reporting quick, simple and criterion. Compared with the other way by experiments, our method has good performance for processing more than ten thousand data records.

Keywords: data reporting; excel template-print; batch copying; Delphi

1 引言

数据的打印输出是信息系统中最常用的功能之一, 它的具体实现包括两种常用方法: 一是采用专门的报表控件, 这种方法专业性较强, 对开发人员有较高的要求, 而且存在额外的使用费用; 另一种方法是利用 Excel 进行套打¹, 这种方法开发简易, 而且允许用户在打印之前, 进行文档预览、修改等处理, 适合用户的工作习惯。在实际应用中, 我们发现在大数据量的情况下, 两种方法都存在效率低的缺点, 为了解决这个问题, 使报表打印规范、简捷、高效, 本文提出了一种基于批量复制的高效 Excel 套打方法, 该方法在项目实施中具有非常好的效果。

2 应用系统操作Excel的基本原理

Excel 电子表格软件是实际工作中最常用的软件之一, 使用 Excel 作为应用系统中报表打印的媒介非常适合用户的工作习惯, 实用性非常强, 因此 Excel 套打作为解决数据打印的一种方法, 在系统开发中被

广泛应用。这种方法首先根据用户需要制作 Excel 模板文件(普通的 Excel 文件亦可), 在该模板文件中设定好纸张大小、打印格局、页码格式, 以及已经确定的报表标题、行列描述等; 然后使用程序往模板文件中输出变化的数据。以 Delphi 开发为例, 对 Excel 的操作非常直观, 核心代码可以参考如下:

```
//创建 Excel 工作簿  
XL := CreateOleObject('Excel.Application');  
//打开指定的模板文件  
XL.Workbooks.add(excel_path + excel_name);  
//在指定位置输出数据  
XL.activesheet.cells[row, col].value:=data_output;  
//进行报表预览  
XL.visible := true;  
XL.activesheet.printpreview;
```

从中可以看出, 在进行数据输出时, 编程人员相当于根据行、列两个坐标往二维表里写数据, 这种方法在输出具有简单信息的报表时非常好用。

^① 收稿时间:2010-05-04;收到修改稿时间:2010-06-06

3 基于批量复制的高效套打方法

在很多系统中,需要处理的数据量比较大,比如项目中遇到的批量数据集的列表输出,要输出的数据项和记录数都比较多,在实际工作中,最初采用了逐个单元格填写的办法,产生了两个需要解决的难题。一是因为数据量巨大,报表生成特别慢,用户根本无法接受;二是无法确定输出数据的行数,在采用固定模板文件时,十分影响美观效果。

针对第一个问题,可以采用批量复制的方法,从数据源中将数据一次性快速输出到 Excel 文件中,这样就解决了输出慢的问题;从数据源到 Excel 中的批量复制,从查询到的有关资料来看,有多种不同的方法,包括逐行复制等,有的方法还做了中间处理。经过实际的多次试验,本方法中,我们最终确定直接调用微软提供的原始方法:

```
//打开数据集
adoqry1.open;
//批量复制
```

```
XL.cells[row,col].copyfromrecordset(adoqry1.Recordset);
```

该方法可以直接将数据集直接复制到 Excel 表指定区域中(上面代码中的 row, col 两个变量指定了该区域左上角第一个单元格的位置)。这样做,去掉了中间处理过程,在数据量多的情况下可以节省大量的时间。

针对第二个问题,本方法实现了一种动态修改模板文件的方法。该方法的原理是在 Excel 模板文件中制作好一定数量的样本行,设置好它们的格式,然后根据数据量的多少进行样本的批量复制,核心代码如下:

```
//这两行代码选定 20 行样本行
tempstr:= inttostr(row_begin + 1) + ':' +
inttostr(row_begin + 20);
XL.activesheet.ROWS[tempstr].SELECT;
// 计算复制次数
total_page:=lack_rows div 20;
mod_page:=lack_rows mod 20;
进行循环复制
for i := 1 to total_page do
XL.SELECTION.INSERT[-4121];
//补足零散行
tempstr:= inttostr(row_begin + 1);
XL.activesheet.ROWS[tempstr].SELECT;
for i := 1 to mod_page do
XL.SELECTION.INSERT[-4121];
```

从上述示例代码中,每次复制在 20 行样本行的基础上进行,实际的复制速度可以满足数据记录数少于 1000 行的情况。对于再大的数据量需要修改每次复制的样本行数。例如,在数据量在 1000 至 10000 之间时,可

以改为每次复制 100 行。总的来说,动态修改模板文件关键在于根据数据量级确定每次复制的样本行,正确计算零散行数,保证格式完好的数据行不多不少。在此基础之后,再使用批量复制的方法,完成数据的输出。

4 性能分析

该方法在实际工作中很好的满足了应有系统的性能要求,为了更进一步明确其性能,我们在 R60 笔记本上做了对比实验:在 Access 数据库上选了一个有 19 个字段的数据表,该表的数据类型包括整数型和文本型,然后实测了数据库记录数在 10、100、1000、10000、60000 条时系统的性能,该实验中记录了不同情况下从数据库到 Excel 文件的数据转换时间(不是指实际打印时间)。

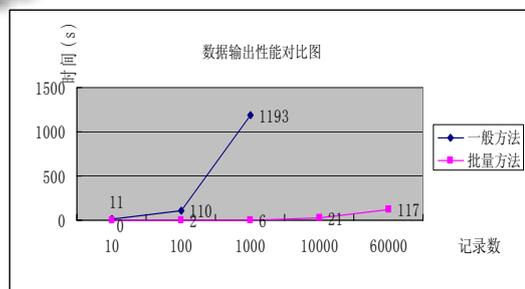


图 1 不同数据规模下,两种打印方法的性能对比

从对比中我们可以看出,一般方法是通过在 Excel 模板中逐格输出完成套打的,这种方法在数据字段比较多的情况下,随着数据量的增大,耗费的时间急剧上升,是用户根本无法接受的;相对于此,基于批量复制的高效 Excel 套打方法耗时平稳,在对比实验中 60000 条记录已经接近 Excel 2003 版本单表数据量的上限,117 秒的时间消耗在这样的数据量下是用户完全可以接受的。

5 结论

基于批量复制的高效 Excel 套打方法可以解决大数据量的报表输出问题,该方法的核心是批量复制和模板的动态修改,在实际应用中性能优异、报表规范、非常符合用户习惯,并且代码简洁,便于开发人员掌握,不失为一种大数据量的报表输出解决方案。

参考文献

- 伍远.Excel VBA 开发技术大全.北京:清华大学出版社,2009.21—55.
- 陈劲秋.Delphi 数据库编程.北京:机械工业出版社,2007.100—150.