

用 Delphi 开发 C/S 数据库应用程序的相关技术

Tips for Using Delphi to Develop a Database Application System Based on C/S

宋 玲 李陶深 严 毅 吴景莉
Song Ling Li Taoshen Yan Yi Wu Jingli

(广西大学计算机与信息工程学院 南宁 530004)

(Computer & Information Engineering College, Guangxi University, Nanning, 530004)

摘要 介绍运用 Delphi 面向对象程序设计工具在 C/S 结构的数据库应用程序开发过程中的一些关键技术及经验。

关键词 Delphi5.0 SQL Server 7.0 数据库应用程序 客户/服务器 ODBC BDE Query 报表

中图法分类号 TP 311.11

Abstract Some tips from our experiences of using Delphi in the course of developing a database application system based on a client/server architecture were released.

Key words Delphi5.0, SQL Server 7.0, database application system, client/server, ODBC, BDE, Query, report

1 Delphi 数据库编程技术概述

Delphi 是目前开发 Client/Server 结构数据库应用程序强有力的工具, 它真正实现了面向对象, 其支持语言是 Object Pascal。

Delphi 利用数据库组件实现对各种数据库的访问, 常用的数据库组件有: Tdatabase、Ttable 和 Tquery。Tdatabase 用于处理应用程序与服务器的连接; Ttable 用于实现对任一数据库表的访问; Tquery 是 Delphi 最强大的数据库访问组件, 它实现了 Delphi 对 SQL 语言的支持, SQL 语句通过 Tquery 控件传递到要访问的数据库系统的数据库引擎中, 由数据库引擎具体执行 SQL 语句, 从而实现对数据的操作。

Delphi 通过数据控制元件实现与用户的交互, 常用的数据浏览控件有: TDBGrid、TDBEdit、TDBText、TDBListBox、TDBComboBox 等, 它们可以提供显示、修改数据库中数据的界面; QuickReport 控件可建立、浏览和打印数据库表中的数据。

我们使用 Delphi5.0 作为前台开发工具, 以 SQL Server7.0 为后台数据库系统, 开发了一

个基于 C/S 结构的管理信息系统，以下介绍其中的一些主要技术。

2 用 Delphi 开发数据库应用程序的相关技术

2.1 实现 Delphi 与 SQL Server 的前后台连接

要让 C/S 系统工作，首先实现前后台的连接，将 Delphi 与后台数据库系统 SQL Server 相连可使用 ODBC 和 BDE/IDAPI 两种方式。

(1) 利用 ODBC 连接

打开控制面板，双击 ODBC Data Sources，在 ODBC Data Source Administrator 窗口中选择 User DSN，点击 Add 键，选择数据驱动程序 SQL Server，单击“完成”，进入 Microsoft SQL Server DSN Configuration 窗口，依次填入数据库的别名、数据库的描述和所连接的服务器名，单击“下一步”，选择 With SQL Server authentication using a login ID and password entered by the user 单选钮，填入登录名和密码，单击“下一步”，选择 Change the default database to 复选框，选择所要连接的数据库名，单击下一步，完成后，进入 ODBC Microsoft Server Setup 窗口，单击“OK”，至此便完成了用 ODBC 实现与后台 SQL Server 的连接。

(2) 利用 BDE/IDAPI 连接

BDE 是 Inprise 公司支持 Delphi 的核心数据库引擎和互连软件，它提供了支持多种数据库以及任何 ODBC 数据源的统一和一致的应用程序编程接口。利用 BDE 建立与后台数据库连接的方法是：

首先是参数设置。最重要的是别名设置，设置 SQL Server 所使用的驱动程序、通信协议、语言驱动程序等参数。Delphi 提供了强大的 SQL Server 本机驱动程序，可以直接根据别名连接到后台数据库。

启动 BDE Administrator 应用程序，单击 Configuration 页依次选择 Configuration/Drivers/Native/MSSQL，进入 MSSQL 的 Definition 窗口，在 Definition 窗口中出现 MSSQL 的许多参数，这些参数的设置，可以在 SQL Server 的 Enterprise Manager 中找到，在 Definition 页中填好各项设置后，便完成了 MSSQL 的数据库驱动程序的设计。

必须建立一个 SQL Server 的别名让 Delphi 使用。在 Databases/Configuration 窗口中，选择 Database 页，单击鼠标右键，选择“New”项，出现数据库别名对话框，选择 MSSQL，然后在 Definition 窗口中，必须设置以下参数：DATABASE NAME、SERVER NAME、SQL-PASSTHRU MODE、USER NAME，这几个参数的设置应与 Configuration 页中的设置一致。完成设置后，关闭 BDE Administrator 应用程序，并保存设置。此时，在 Windows NT 的 system 目录下应有一个名为“ntwdblib.dll”的文件，它是 SQL Server 的一个动态链接库的接口文件，这个文件对以后能否在前台使用 SQL Server 至关重要。

在实践中，我们对以上两种连接后台数据库的方式进行了比较，结论是 ODBC 在健壮性和稳定性方面比 BDE/IDAPI 好。

2.2 设置缓存机制

当用户修改数据库时，常会发生输入错误，若不使用缓存机制，错误信息将直接被写回数据库中。如果为 TDataSet 组件设置了缓冲机制，它会将用户对数据表的更新先放在 1 个缓冲区中，使用户方便地执行撤消 (Undo) 操作。以下介绍一些与缓冲机制有关的属性和方法。

CachedUpdates (属性) 设置数据集缓冲机制有效 (True) 或无效。

ApplyUpdates (方法) 将缓存中更新的记录写回数据库。

RevertRecord (方法) 只对光标所在的一条记录执行一个“Undo”操作。

CommitUpdates 清除缓存操作。

我们在程序中多次使用了缓冲机制, 例如: 在销售子系统的提货处理中, 使用缓冲机制的程序段如下:

```
{进入发票输入窗口时}
.....
table1.CachedUpdates:=true; //打开数据集缓冲机制
.....
{按发票输入窗口的“取消”按钮时}
table1.append;
table1.revertrecord; //执行一个“Undo”操作
.....
{按发票输入窗口的“存盘”按钮时}
table1.ApplyUpdates; //将缓存中更新的记录写回数据库
.....
{按发票输入窗口的“Close”按钮时}
table1.CommitUpdates; //清除缓存
table1.cacheupdates=false; //关闭数据集缓冲机制
.....
```

在编程过程中, 我们发现, Revertrecord 操作每次撤消连条记录, 因此我们在每次执行 Revertrecord 操作之前执行一次 append 操作。另外, 在使用缓冲机制时, post 语句并不能将数据集的变动写回到数据库中去, 只能用 ApplyUpdates 语句实现这一功能。

2.3 关于 Query 组件

2.3.1 Query 控件的动态 SQL 查询

“动态查询”是指在操作过程中动态改变查询条件、动态生成 SQL 语句, 使得窗体查询功能更强大。在本系统中, 我们大量使用了动态 SQL 查询语句, 也就是在程序中使用嵌入式 SQL 语句。

在 Tquery 控件中使用嵌入式 SQL 语句时主要使用的方法和属性有:

Tquery.Open (方法)、Tquery.execSQL (方法)、Tquery.prepare (方法)、Tquery.Params [index] (属性)、Tquery.ParamsByname (方法)、Tquery.SQL (方法)、Tquery.Close (方法) 等。

例如:

```
.....
query1.close; //关闭 query1
query1.sql.clear; //清空 query1 的 SQL 语句
query1.sql.add ('select * from dbo. 销售合同目录库'); //在 query1 的 SQL 书信中加
语句
query1.sql.add ('where 合同编号=: s1');
query1.paramsbyname (s1').asstring:=edit1.text; //给参数赋值
query1.prepare; //优化 SQL 查询语句
query1.open; //执行 SQL 查询语句
.....
```

```

query1.close;
query1.sql.clear;
query1.sql.add ('update dbo. 统计 ');
//在 query1 的 SQL 属性中加语句
query1.sql.add ('set 岩 2 上 = (select sum (数量) from dbo. 出库凭证目录库 ');
query1.sql.add ('where (年 = : var1) and (月 <= : var2) and (产品名称 = " 2# 岩
石" ) ');
query1.sql.add ('and (地区 = : var3) ');
//给参数赋值
query1.params [0].asinteger := spinedit1.value - 1;
query1.params [1].asinteger := spinedit2.value;
query1.params [2].asstring := table1.fieldbyname ('地区名').asstring;
//执行 SQL 查询语句
query1.ExecSQL;
.....

```

2.3.2 RequestLive

RequestLive 是 Tquery 控件的一个属性,它影响着 Tquery 控件的查询特性。RequestLive 属性设定应用程序是否可以通过 BDE 返回一个实际结果集。如果返回的是一个实际结果集,用户可以通过数据感应控件直接修改数据;返回只读集则不能。

要使 query 组件返回的数据表格是可以修改的,它必须同时满足 2 个条件:一是 RequestLive 属性要设置为 true,二是 query 内的查询指令是简单的 select 语句,这时所返回的数据便为可修改的。简单的 select 语句,要满足 3 点要求:a)只含一个数据表;b)不能有 ORDER BY 子句;c)不能使用像 SUM 或 AVG 之类的聚集函数。

另外,我们在编程中发现,当动态生成 query 组件的 SQL 语句时,要使返回的数据表格可以修改,除了满足以上要求外,还必须两次写入 SQL 语句。

例如,以下是查询 dbo. 采购合同目录库程序段:

```

.....
form15.query1.close;
form15.query1.sql.clear;
form15.query1.sql.add ('select * from dbo. 采购合同目录库 ');
form15.query1.prepare;
form15.query1.sql.clear; // * *
form15.query1.sql.add ('select from dbo. 采购合同目录库 '); // * *
if i1 and (not i2) and (not i3) and (not i4) then
begin
form15.query1.sql.add ('where 合同编号 = : s1');
form15.query1.parambyname ('s1').asstring := edit1.text;
end;
form15.query1.open;
.....

```

按照 Delphi 有关书籍介绍,上述带 * * 的两个语句是不需要的,但经过实践,我们发现如果不写这连个语句,在执行时显示错误信息“multirecord found,, but only one was expected”。因此,在动态生成 Query 组件的 SQL 语句时,先在 SQL 中写入一些语句,然后

删除, 再写入正式的语句即可。

2.4 报表的制作

报表的制作是开发一个管理信息系统的重要环节。Delphi5.0 的 Qreport 标签页中的组件是制作报表的元素。在介绍有关 Delphi 的书籍中, 介绍报表的制作及打印的内容很少, 我们通过实践, 总结出以下一些解决问题的方法和技巧。

(1) 在 QuickReport 中, 当纸张设置为 custom 时, 预览无问题, 但打印机却不打印, 而使用 A4 或其它非定义纸张可以打印。原因是与 Windows 系统自定义的纸张有关。解决方法: 退出 Delphi 环境, 设置打印机纸张为自定义大小后, 再进入 Delphi 定义报表的纸张大小。

(2) 制作繁琐的中式报表

如果报表的总体格式固定, 可以在 FORM 中直接画报表, 用程序控制横/竖线和数值的输出, 即利用 canvas.textout() 函数将数值输出到固定位置。此方法比较繁琐, 要将字段中的数值先赋给数组, 但对制作中式报表很有效。

(3) 在使用 QRGroup 时, 我们发现不能把 Summary 加在 GroupFooter 之后, 它总是出现在 GroupFooter 之前。解决方法: 把 GroupHeader 的 FooterBand 属性设置为 GroupFooter 所在的 Band, 即将 Summary 加在 GroupFooter 之后。

(4) DBGrid 对定义为 Text 型字段的记录显示不出来, 只显示 [Memo], 但在制作报表时, Text 型用 DBText 显示正常, 并且用 DBRichText 可以换行显示。解决方法: 在 Delphi 中 Text 要用 Memo 显示, 如果一定要在 DBGrid 网格中显示, 应将其改为 Char 型, 但改后就无法用 DBRichText 换行显示。

(5) 直接调用报表文件 (*.pas) 预览和打印时, 程序出错。解决方法: 假设有一个报表名为 rp—jjgb, 在写命令 rp—jjgb.preview 之前加上语句: Application.CreateForm (Trp—jjgb, rp—jjgb); 即可。

(6) 在一些报表中要用两个 Bank: Summary 算出“每页小计”和“累计”, 但实际预览时 Qreport 只运行一个 Bank 的结果, 即只能实现一个功能, 用 Page Footer 也不行。解决方法:

a) 在 Bank (Detail) 下加一个 Bank (Page Footer), 在此 Bank 上加上相应的 Label 和 QR-Expr 组件 (注意: 这里的 QRExpr 中 [ResetAfterPrint] 属性必须设置为 True, 用作实现“每页小计”功能。

b) 将上面的 Bank (Page Footer) 中 [HasChild] 属性设置为 True, 在弹出相应的 Child 上加上相应的 Label 和 QRExpr 组件 (注意: 这里的 QRExpr 中 [ResetAfterPrint] 属性必须设置为 False, 用作实现“累计”功能。

c) 要是该表上还有页脚则必须把 b) 步骤中的 Bank (Child) 中 [HasChild] 属性一项再设为 True, 将页脚组件加在弹出相应的 Child 上。

3 结语

Delphi 是一个真正面向对象的强大数据库应用开发工具, 以上是在开发一个 MIS 系统过程中所采用和探讨的一些关键技术, 具有一定的实际意义。