

VB6.0 新控件在数据库中的应用及编程技巧

资 刚 彭楚武 魏育成 (湖南大学电气与信息工程学院 410082)

邹明珠 (珠海华电股份有限公司 510951)

摘要:本文介绍了 VB 新控件在数据库管理系统中的应用技巧,并通过实例进行了详细说明。

关键词: VB6.0 数据库应用 形状语言 ADO DATA 控件 MSHFlexGrid 控件

一、引言

Visual Basic 6.0 是微软公司正式推出的 VB 最新版本,它不仅功能更强大,使用起来更方便,而且在许多方面有了实质性突破。综合来看,VB6.0 主要在两方面大大超越了 VB 以前的版本:

(1) Server 程序开发及 DHTML 的制作;

(2) 数据库访问及管理。

二者相比,后者特别引人注目:除了增强 Web 数据库功能外,新增的 ADO(ActiveX Data Objects)数据库访问方式及其相关控件更是令 VB 编程如虎添翼。笔者近期开发了一个医院 MIS 系统,采用 VB6.0 进行前端开发,SQLServer6.5 作为远程数据库。在开发过程中,笔者深切体会到使用 VB6.0 的新控件不仅可以大大减轻开发人员的工作量,而且能使开发出的界面更美观、更友好。本文将重点介绍这些新控件的使用技巧及编程时应特别注意之处。

二、新控件的使用方法及编程技巧

1. ADO 控件

ADO 是一种全新的数据对象,与 VB 传统的 DAO(Data Access Objects)和 RDO(Remote Data Objects)相比,它具有对象模型简单,易于使用,数据库访问速度快,内存及磁盘空间占用较少,功能更强的特点。在新产品的开发,特别是开发 Client/Server 和 Browser/Server 类应用软件时,利用 OLEDB 作为数据提供者,并使用 ADO 技术通过 OLEDB 访问网络数据库成为现今流行的首选方案。ADODC 即为使用 ADO 方式访问数据库的可视化控件,与使用代码实现的方式相比较,它具有编程简单(甚至不用写代码)、使用方便以及有一个图形界面的优点。该控件的基本使用方法在一些资料中已有介绍,此处不

再重复,下面仅介绍一些编程技巧。

(1) ConnectionString 与 CommandType 属性的配合使用。在 ADODC 中最重要的属性为 Data Source 与 ConnectionString,前者指明了 ADO 控件与哪个数据库以什么引擎(如 JET, ODBC, OLEDB 等)绑定,后者则指明了数据的命令文本(CommandText)。CommandType 是与 ConnectionString 配合使用的属性,它给出了命令文本的属性,即指明了命令文本是表格名称,SQL 语句,还是存储过程。CommandType 的缺省值是 adCmdUnknown),但是这会降低系统性能,因为在这一方式下,VB 必须调用数据提供者才能确定 ConnectionString 的属性,然后转向相关代码处理。因此编程者可配合使用这两个属性,直接指示 ADO 转向相关代码,从而提高访问数据库速度,二者最常用的属性可配合使用如下:

ConnectionString	CommandType	备 注
1 "Select...From...Where ..."(SQL 语句)	adCmdText	ADO 将其视为 SQL 命令文本来执行
2 "TableName" (数据库表名)	adCmdTable	ADO 将生成 SQL 查询以从表中 返回所有行
3 "ProcedureName" (存储过程名)	adCmdStoredProc	ADO 将 Source 视为存储过程
4 其 他 命 令 (非优化命令)	adCmdUnknown	ADO 将调用数 据提供者确定

(2) Supports 在 ADODC 中的应用。网络数据库的难题之一是异构平台数据库数据存取与对记录集的操作,对于各类关系或非关系数据库,有不同的数据提供者为相应的数据源提供访问接口,甚至对于同一数据源也可通过不同的数据提供者访问。但是各个数据提供者所支持的数据集操作功能不尽相同,为了提高程序对异构平

台的兼容性及鲁棒性,使用 Supports 方法返回当前数据提供者支持的功能就尤为重要。Supports 的一种使用方法如下:

```
Dim bln as Boolean
```

```
bln = ADODC1.Recordset.Supports(cursorOptions)
```

其中 cursorOptions 可填入各类记录集操作方法名称,常用的有:adApproxPosition,adBookmark,adDelete,adMovePrevious 等等。执行上述代码后,若数据提供者支持相应的方法则返回 True,否则返回 False。这样,对于数据提供者不予支持的方法编程者应想法另辟蹊径。

2. 利用 CoolBar 与 ToolBar 创建有自我特色的浏览器风格界面

浏览器风格界面是当今许多应用软件界面的首选风格,其主要特点是将各按钮图标分成三种状态:正常时各图标为平面、黑白的;禁止时为灰色的;当用户鼠标移至可用按钮图标上时,该图标会突出并现出彩色,同时给出该按钮功能提示。在以往的 Visual Basic 中,必须利用代码实现,如通过捕获鼠标移动事件以改变相应按钮的属性来实现。在 VB6.0 中,利用 CoolBar 和 ToolBar 即可轻松实现浏览器风格的界面,具体步骤如下:

首先在窗体中放置一 CoolBar 并将其“Align”属性设置为 vbAlignTop 使其位于窗体顶部或紧贴菜单条;其次在 CoolBar 上指定的分隔带间放置一 ToolBar,并在 CoolBar 属性页带区栏相应索引的“子”项栏目中填入该 ToolBar。注意:只有这样才能使 ToolBar 真正嵌入 CoolBar 中,否则 ToolBar 只能“浮”在 CoolBar 上。

接着在窗体上放置三个 ImageList 控件,一个命名为“imlColor”,用于存放按钮的彩色图片;第二个命名为“imlBW”,用于存放按钮的黑白图片;最后一个命名为“imlBrown”,用于存放按钮的灰色图片。在此步骤中应特别注意各控件图片像素尺寸必须一致,而且图片索引必须一一对应。然后在 ToolBar 的属性页“通用”一栏中,将“图象列表”设为 imlBW,将“热图象列表”设为 imlColor,将“禁用图象列表”设为 imlBrown。同时应将“外观”设为“0 - ccFlat”,将“样式”设为“1 - tbrFlat”,否则无法显示突出的按钮。最后在按钮一栏中将对应位置的按钮一一插入并按 ImageList 中的顺序设好索引。

上述步骤仅为基本步骤,读者还可改变其他属性或加入浮雕效果等方式建立有自我特色的浏览器风格界面。

3. 用 MSHFlexGrid 控件清楚展示记录间层次关系

在许多大型数据库中,不同类型的数据往往存储于几张表中,然后通过同一字段关联,对于用户指定查询的 A 表中的一个记录在 B 表中也许有多个记录相对应,而 B 表中的某些记录在 C 表中也有多个记录相对应。在查询并显示这类数据集时,MSHFlexGrid 控件提供了一种易于使用,而且能有效、清楚地揭示记录间层次关系的方法。使用这一控件最简单的方法是将 MSHFlexGrid 的 Data Source 属性赋给数据环境,而使用形状化查询语言则较难,但其功能更强,下面将重点介绍:

(1)形状(Shape)数据操作语言概念的引入。对于上述记录之间一对多的关系,我们可理解为父记录与子记录的关系,类似于树形层次结构,一个父记录下可接多个子记录。形状数据操作语言即为 ADO2.0 中推出的针对父-子关系记录集操作的语句,它可使查询结果表示为分级的记录集。与 SQL 语句中的 JOIN 子句相比(使用该子句也可关联记录集),使用分级的父-子方式更有效,因为前者将使创建的 Recordset 的每一行中产生源于所查询某个表的大量信息冗余,而后的子记录只有唯一一个父记录与之对应。

(2)形状语言的使用方法。使用形状语言前必须指定提供者,在 ADO 方式中,既可在 ConnectionString 中指定,也可在记录集打开命令的 ActiveConnection 中指定。而使用 Shape 语言有两种方法:

①将子记录追加到父记录;②对子记录计算后产生父记录。下面将分别分析其语法并举例说明:

·追加方式语法:

```
“SHAPE {parent - command} [[AS]table - alias]
```

```
APPEND {child - command}
```

```
RELATE (parent - column TO child - column)”
```

说明:

·parent - command, child - command:返回父、子记录对象的查询命令,二者必须使用结构化查询语言(SQL)且必选。

·parent - column 为由 parent - command 返回的 Recordset 中的列;child - column 为由 child - command 返回的 Recordset 中的列,二者父记录表与子记录表中同一字段。

·table - alias - 用于引用 parent - command 返回的 Recordset 的别名。

例如在数据库“Dbtest”中有两张表,分别为“A”表和“B”表,程序通过 ODBC 方式访问该数据库,数据源名称为“dbt”,为简便起见,口令、用户身份识别等均未设。

表的内容分别如下:

表 A:

姓名	性别	年龄	地址
李四	男	21	湖北省武汉市
王五	女	32	广东省广州市
张三	男	50	湖南省长沙市
赵六	女	18	广西省桂林市

表 B:

姓名	入院日期	治疗费用	住院时间(天)
张三	97-3-2	250	1
李四	97-5-8	124	0
王五	97-12-12	3089	21
张三	98-6-10	10	0
张三	98-10-25	17	0
王五	98-11-13	59	0
王五	99-5-17	479	3
赵六	99-6-1	138	1

查询要求:显示每个病人姓名,性别,年龄,地址,入院日期,住院时间以及治疗费用。

我们只要在查询窗体上放置一个 MSHFlexGrid 控件,然后在合适的位置写入如下程序即可:

```
'定义 ADO 记录集对象
Dim rt As New ADODB.Recordset
'指定形状语言提供者,ODBC 打开方式及数据源名称
rt.ActiveConnection = "provider=msdatashape.1;dsn=dbt"
'按指定查询项目写 SQL 语句
rt.Source = "shape {select * from a} append ({select 姓名,入院日期,住院时间(天),治疗费用 from b} relate 姓名 to 姓名)"
'打开数据库生成记录集
```

```
rt.Open
'显示记录集
Set MSHFlexGrid1.DataSource = rt
则显示结果如下:
```

姓名	性别	年龄	地址	入院日期	住院时间(天)	治疗费用
张三	男	50	湖南省长沙市	1997-10-02	1	250
				1998-06-10	0	10
				1998-10-25	0	17
李四	男	21	湖北省武汉市	1997-05-08	0	124
王五	女	32	广东省广州市	1997-12-12	3	479
				1998-11-13		
				1999-05-17		
赵六	女	18	广西省桂林市	1999-06-01	1	138

·计算方式语法。形状语言的功能十分强大,除了具有上面介绍的的分层显示特色外,它还可对指定的子记录进行计算,并将结果赋给父记录,从而直接完成数据库统计功能。

```
"SHAPE {child-command} [[AS] table-alias]
COMPUTE Aggregate-Command Field-List
[BY grp-field-list]"
```

说明:

- Aggregate-Command - 合计函数
- Field-List - 所操作的字段列表
- grp-field-list - 指定子记录集 Recordset 中行次序的列的列表。如果没有指定 BY 子句,将只返回合计函数的结果;如果指定了 BY 子句,则子记录集 Recordset 被追加到所生成的父记录集 Recordset。

其他字段的说明见上节。

合计函数有 SUM(计算指定字段中值的总和),AVG(计算指定字段中所有值的平均),MAX(计算指定字段中的最大值),MIN(计算指定字段中的最小值),COUNT(计算指定字段的行数),STDEV(计算指定字段的标准偏差),CALC(expression)(只对当前行计算专有表达式)等等。限于篇幅,此处不介绍具体例子。

(来稿时间:1999 年 6 月)