

# 利用 Kylix 开发 Linux 平台上的数据库应用程序

潘 栋 章思耀 (北京清华大学精密仪器系 100084)



摘要: 本文系统地介绍了 Kylix 数据库应用程序的体系结构, 并对 Kylix 数据库应用程序的各组成部分作了简要的说明。

关键词: Kylix Linux 数据库

## 1 前言

Kylix 是 Borland 公司于今年 3 月初推出的 Linux 平台上第一个可视化快速开发工具 (RAD), 被人们称为“Delphi for Linux”。众所周知, Linux 平台上一直缺乏好的可视化开发工具, 这是限制 Linux 应用的一个很大的障碍。Kylix 将 Borland 优秀的编译器技术与高度可视化设计环境结合在一起, 进行 GUI、Web 程序和数据库程序开发, 极大地提高了 Linux 应用程序的开发效率, 是目前 Linux 平台上唯一拥有上述优点的开发工具。Kylix 将广大从事 Linux 图形界面编程的人员从烦琐的代码中解脱出来, 其意义可以和 Windows 平台上 Visual Basic 和 Delphi 的出现相当, 是 Linux 平台上编程的一场革命。

我们都知道, 开发数据库应用程序是 Delphi 的强项, Kylix 也秉承了这一特色。但是, 由于 Kylix 使用的组件库是 Borland 公司推出的新一代组件库 CLX (Common Library for Cross Platform), 而不是 VCL (Visual Component Library), 在数据库应用程序的体系结构上有较大的改动。下面本文将就在 Kylix 中如何进行数据库编程做一个简要而系统的介绍。另外, 由于 Borland 公司宣布在 Delphi 6 中将同时支持 CLX 和 VCL, 因此本文所述内容亦适用于 Delphi 6。

## 2 Kylix 数据库应用程序的组成

数据库应用程序是由用户界面 (UI) 元素、数据集 (Dataset) 组件、数据源 (DataSource) 组件, 以及连接数据库的组件组成的。这些组件的组织结构和相互关系

决定了数据库应用程序的体系结构。图 1 是一个数据库应用程序的基本框架。

### 2.1 连接数据库的组件

Kylix 数据库应用程序使用 dbExpress 数据库引擎来连接数据库。dbExpress 是由一系列轻量数据库引擎组成的, 它向编程者提供了一个独立于特定数据库的统一的编程接口, 提供了对 SQL 数据库的快速访问能力。与 Delphi 和 C++ Builder 的 BDE 数据库引擎比起来, dbExpress 是一个轻量、快速、易发布的数据库引擎。



图 1

目前, dbExpress 支持的数据库有 Oracle、DB2、InterBase 和 MySQL, 其他数据库如 SQL Server 等可通过 dbExpress-ODBC 桥访问。

TSQLConnection 组件封装了 dbExpress 的数据库连接, 它总是和单向数据集 (Unidirectional Dataset) 联合使用。多个单向数据集可以共用一个 SQLConnection 对象, 但值得注意的是, 对于某些数据库 (如 Oracle 和 InterBase), 一个 SQLConnection 对象可以支持多个事务 (Transaction), 而对其他一些数据库 (如 DB2), 一个

SQLConnection 对象只支持一个事务。

## 2.2 数据集组件 (Dataset)

数据集组件是数据库应用程序的核心。一个数据集对象代表了一个来源于数据库的记录集，这个记录集可能是一个查询或存储过程的结果集，也可能来自于另一个数据集。

### 2.2.1 数据集组件的继承关系

所有的数据集组件都是 TDataset 这个虚基类的子类，在 TDataset 中声明了所有数据集组件所共有的属性和方法。

数据集组件分为两大类：客户数据集 (Client Dataset) 和单向数据集 (Unidirectional Dataset)。TDataset 有两个子类 TCustomClientDataset 和 TCustomSQLDataset，分别是这两类组件的基类。数据集组件的继承关系图如图 2 所示。

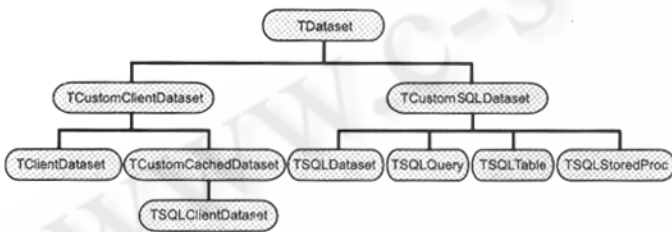


图 2

### 2.2.2 客户数据集 (Client Dataset)

客户数据集实现了 TDataset 中声明的所有数据访问、编辑、浏览、数据约束、过滤等功能，它们的特点是，将所有数据都放在内存缓冲区中，并在内存缓冲区中对数据进行操作。

Kylix 提供了两个 Client Dataset 组件：

(1) TClientDataset 使用时不一定需要数据库连接，是一个独立于数据库的数据集组件。

(2) TSQLClientDataset 与 TClientDataset 不同的是内置了一个 TDatasetProvider 组件，可以使用 dbExpress 数据库引擎来获取数据。

这两个客户数据集组件都能够直接读写文件。TClientDataset 组件还能够通过外部的 TDatasetProvider 组件从其他数据集中获取数据。TDatasetProvider 组件能够把源数据集中的数据传递到客户数据集，也能够把客户数据集对数据的修改传回源数据集。

TSQLClientDataset 组件内置了一个 TDatasetProvider 组件，它将单向数据集的快速数据访问能力和客户数据

集的编辑、浏览等功能结合了起来，十分方便。但是，由于其内置了 TDatasetProvider 组件，就不能再使用外部的 Provider 组件了。

### 2.2.3 单向数据集 (Unidirectional Dataset)

单向数据集提供了从 SQL 数据库读取数据的能力。它们是轻量的数据库访问组件，其工作机制为：单向数据集向数据库发送一条 SQL 命令，如果该命令返回记录集，那么就产生一个单向光标。它们不在内存中缓冲记录集，因此它们比客户数据集速度快，而且资源占用较少。但是，在浏览记录集时，只支持单向移动，而且不能对数据进行编辑、过滤，也不支持查询字段 (lookup fields)。

Kylix 提供了四个单向数据集组件：TSQLDataset、TSQLQuery、TSQLTable 和 TSQLStoredProc。其中，TSQLDataset 是一个功能全面的组件，支持查询、存储过程等各种操作，Borland 公司也推荐使用。而其他三个组件主要是用来移植 Windows 下的 Delphi 程序的。

## 2.3 数据源组件 (Datasource)

数据源组件的作用是作为用户界面 (或者说数据感知控件 Data-aware controls) 和数据集之间的一个管道。

## 3 Kylix 数据库应用程序的体系结构

### 3.1 单层结构 (One-tier Architecture)

这是最简单的数据库应用程序，其结构如图 3 所示。其实这是一个基于文件的结构，并不使用数据库。数据文件存储于本地文件系统中，可以是二进制文件也可以是 XML 文件。在这种结构中，数据的读取等操作实际上都是文件操作。读文件时使用的是客户数据集的 LoadFromFile 方法，写入数据时使用的是 MergeChangeLog 和 SaveToFile 方法。

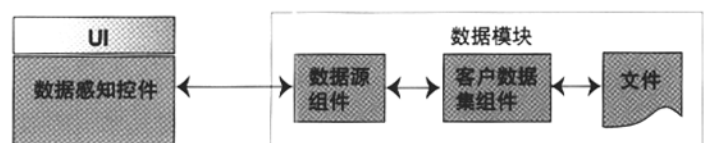


图 3

由于在这种结构中并没有数据库服务器的存在，Borland 公司又将这种应用程序称为 MyBase 应用程序。

### 3.2 客户-服务器结构 (Client-Server Architecture)

这是一种双层结构 (当然，在下列结构中，当连接

的是 Local InterBase 这类基于文件的数据库时，这些结构实际上是一种单层结构)。Kylix 要求使用单向数据集和 dbExpress 来连接数据库服务器，因此有下列两种典型结构。

### 3.2.1 直接使用单向数据集

该模型的结构图如图 4 所示。这种结构提供了快速的数据访问能力，而且可以处理巨大的结果集，因为不需要缓冲数据。但是，使用这个模型时，一次只能显示一条记录，而且只能单向移动，数据也都是只读的。

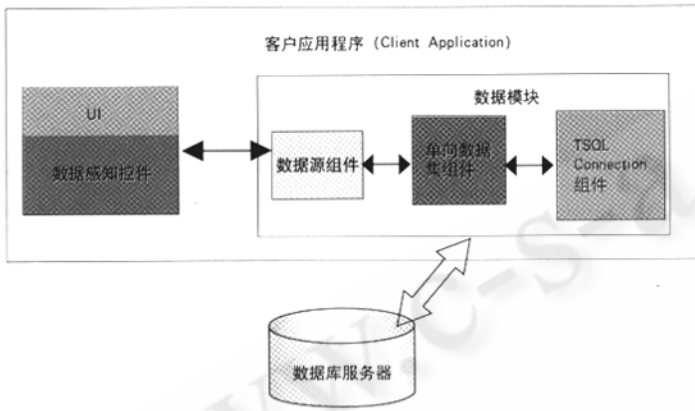


图 4

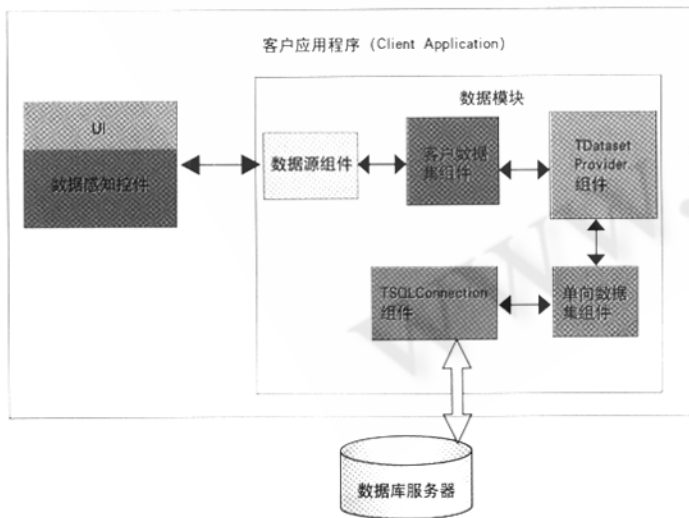


图 5

这个模型最适用于那些读取数据并进行统计处理或生成报表的程序。

### 3.2.2 使用客户数据集来缓冲数据

要想同时显示多条记录并对记录进行编辑、浏览，就必须使用客户数据集。该模型结构如图 5 所示。

为了简化这种结构，Kylix 提供了内置 TDataSetProvider 的 TSQLClientDataset 组件。该模型的结构图如图 6 所示。

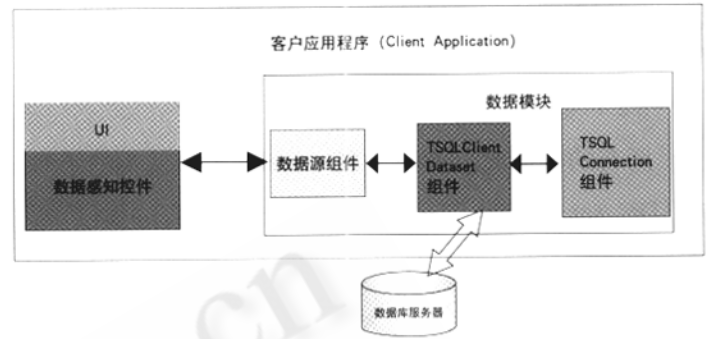


图 6

与单层结构不同的是，虽然这里也使用了客户数据集，但是由于是对数据库进行操作，在修改数据时，使用的是 ApplyUpdates 方法。

### 3.3 多层结构 (Multi-tier Architecture)

与双层应用比较，三层应用的中间层负责数据库服务器和客户端的连接工作，大幅度减轻数据库服务器的负担，提高了数据库服务器的响应速度和数据的安全性。同时，对数据库中的数据进行操作的规则都移到了中间层 - 应用服务器 (Application Server) 上。这种将数据操作的规则集中到中间层的结构，使得一个业务规则层可以重用多个客户应用程序而不破坏数据的一致性。而且，数据操作的规则移到中间层后，实现了瘦客户应用程序，更易于安装、配置和维护。另外，这种分布式处理的结构，减小了网络流量，系统整体性能也得到了提高。

## 4 结束语

综上所述，利用 Kylix 可以在 Linux 平台上方便地开发出各种规模和层次的数据库应用程序，这在 Linux 平台上是一个革命性的飞跃。可以预见，Kylix 的推广使用必将极大地丰富 Linux 平台上应用程序的数量，进而扩大 Linux 的使用范围。■

#### 参考文献

- 李文生、叶宁，采用 Delphi 实现三层客户服务器数据库应用程序，计算机工程，2000，26 (7): 170-172。