

第15章 Client/Server应用程序开发

- Delphi的Client/Server版支持应用客户/服务器（Client/Server）技术开发数据库应用程序。Client/Server技术在目前程序开发中得到了广泛的应用。这种技术的优点在于它将处理工作按照一定的比例分配到客户端和服务器的上去执行，这样减少了网络传输的工作量，从而合理地利用了资源，提高了应用程序开发的效率。因此近年来，Client/Server技术得到了充分的发展。
 - 15.1 Client/Server的基本概念和体系结构
 - 15.2 构造Client/Server的应用开发环境
 - 15.3 Client/Server的应用开发技术

15.1 Client/Server的基本概念 和体系结构

- 15.1.1 Client/Server的基本概念
- C/S（客户/服务器）结构包括连接在一个网络中的多台计算机。按照各自实现功能的不同可以分为客户机（Client）和服务器（Server）。
- 1. 客户机
 - 客户机是用来处理应用程序、请求服务的计算机。
 - 客户机主要是运行使用户产生其服务请求的程序，将数据库请求发送到服务器。客户机主要实现的功能包括：管理用户接口、从用户处接受数据、处理应用逻辑、产生数据库请求、向服务器发送数据库请求、从服务器接收结果并格式化接收到的结果。客户机执行的这些计算称为前端处理（Front-end processing），前端处理提供了对数据进行操作和与显示数据有关的所有功能。

15.1 Client/Server的基本概念 和体系结构

- 客户机的软件包括网络接口软件，支持用户需求的应用程序以及实现网络能力的实用程序。其中，网络接口软件是用来提供各种数据传输服务的；支持用户需求的应用程序主要用于执行具体的任务，例如生成电子表格和数据库查询等；实现网络能力的实用程序主要用于执行网络用户要求的标准任务。

15.1.1 Client/Server的基本概念

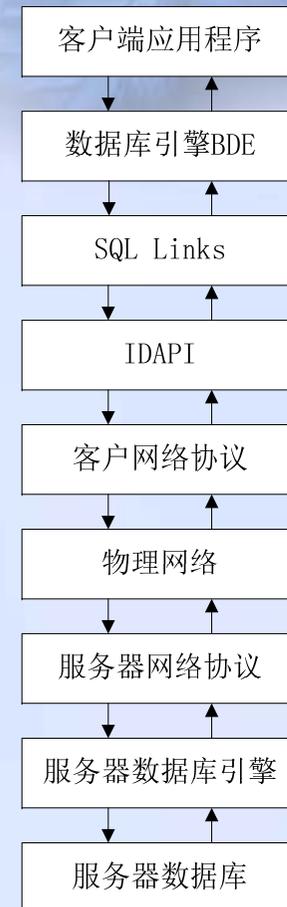
- 2. 服务器
 - 服务器是用来处理数据库的计算机。
 - 服务器主要是用来接受客户机的请求和传送结果的。服务器实现的功能主要包括：从客户机接受数据库请求、处理数据库请求、格式化结果并传送给客户机、执行完整性检查、提供访问控制、执行恢复以及优化查询和更新处理。在服务器上执行的这些计算称为后端处理（**Back-end processing**），后端处理主要提供管理数据资源和存储数据、维护数据以及对数据进行操作等数据库引擎功能。
 - 服务器的软件既包括OSI或其他网络结构的网络软件，也包括提供给客户机的应用程序和服务软件。

15.1.1 Client/Server的基本概念

- 3. 中间件
 - 一般来说，客户机通常是比较便宜的PC机，而服务器是功能强大，运行可靠的大型机或小型机。在数据库环境下，二者通过中间件进行交互。
 - 中间件（**Middleware**）是一种程序设计公共接口，在网络数据库编程中它们也是依靠服务器的扩展程序。它们的作用是提供应用程序和数据库中间的连通性，它们可以使开发人员在应用程序开发过程中避免受各种通讯协议、操作系统以及数据库管理系统的影响。
 - 最常见的中间件类型有应用程序设计接口、远程过程调用、网络通信、数据库访问以及计算机辅助软件工程工具。

15.1.2 Client/Server的体系结构

- 1. Client/Server的体系结构
 - Client/Server的体系结构如图所示，客户端的请求经过BDE和SQL Links后传递给IDAPI，经过IDAPI应用程序解释后经网络传输给服务器。服务器响应请求的过程与以上的顺序正好相反。



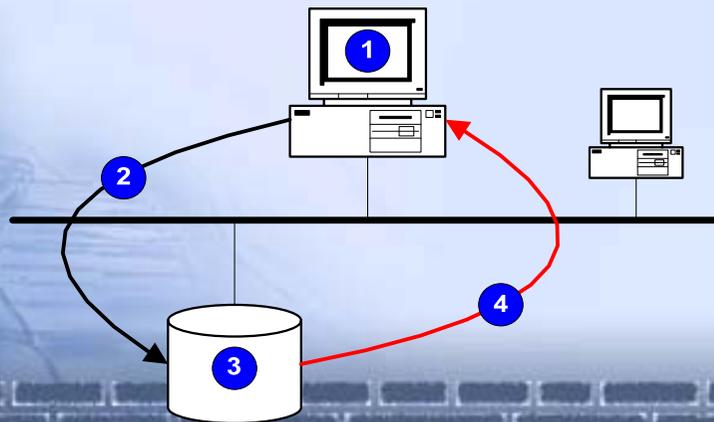
15.1.2 Client/Server的体系结构

■ 2. SQL Links

- 从Client/Server的体系结构图中可以看出，SQL Links在Client/Server数据库应用程序中起着非常重要的作用。
- Borland提供的SQL Links其实是一系列驱动程序的集合，通过它，Borland的应用程序用户可以访问SQL数据库的数据而不需要掌握SQL的知识。SQL Links的工作原理如图15-2所示。SQL Links完成与SQL服务器之间的联系，将查询译为相应的SQL语言并将它们提交给SQL数据库，当此过程结束后，SQL数据库将查询结果返回到客户端。

15.1.2 Client/Server的体系结构

- SQL Links的工作的具体步骤如下：
 - ① BDE的用户发出对SQL数据库的查询要求。
 - ② SQL Links驱动程序完成与SQL数据库服务器的连接，并将用户的查询转换为与服务器兼容的SQL语言后传输给SQL数据库服务器。
 - ③ SQL数据库服务器执行对SQL语言的检查后，执行查询过程。并将查询的结果返回给BDE用户。
 - ④ SQL Links驱动程序将SQL数据库服务器返回的结果转换为客户端的句法，客户端格式化结果并将结果显示给用户。



15.1.2 Client/Server的体系结构

- Borland提供的SQL Links可以应用于所有支持Borland Database Engine的应用程序，或者是支持IDAPI（Integrated Database Application Programming Interface），包括Windows的Paradox、dBASE、Windows Database Desktop的Quattro Pro以及应用Borland Database Engine的用户程序。
- SQL Links支持的数据库有InterBase、Informix、ORACLE，SYBASE以及Microsoft SQL Server数据库。

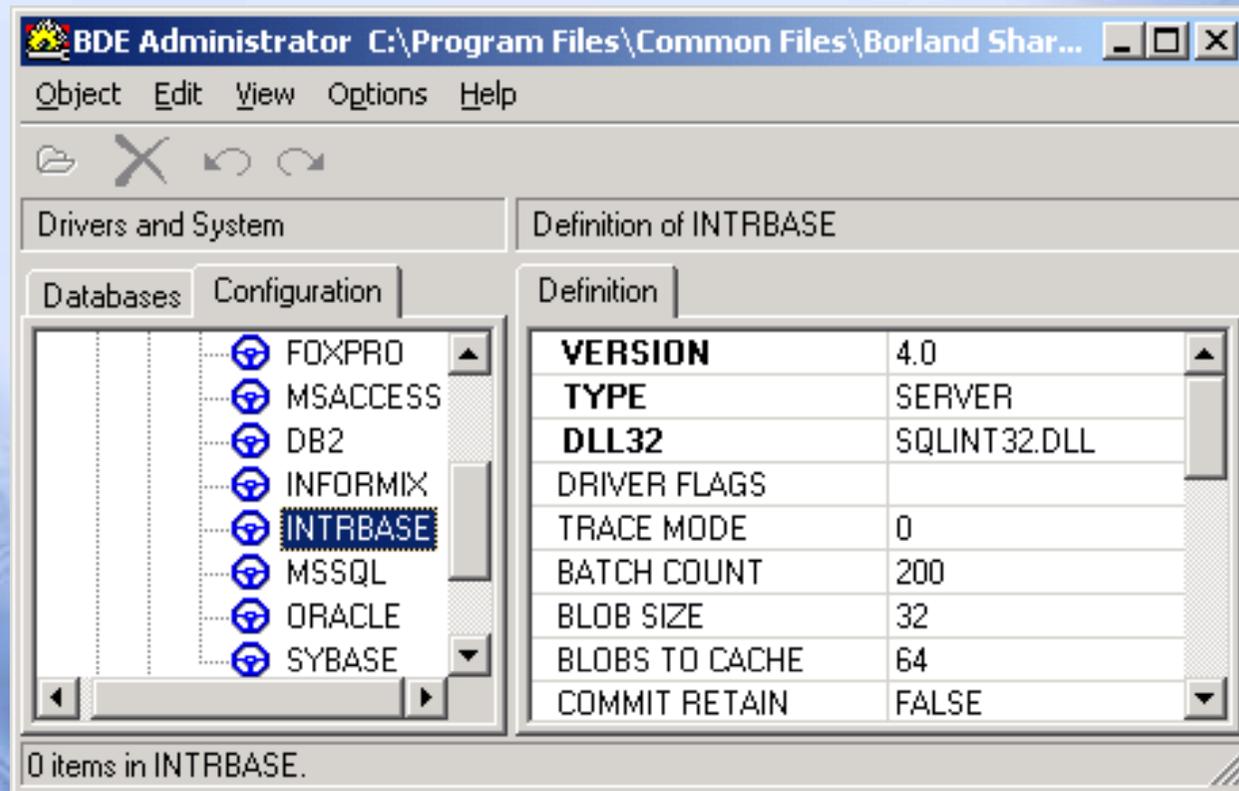
15.1.2 Client/Server的体系结构

- 使用SQL Links的BDE应用程序实际上起到了扩展应用程序的功能，使它继承了使用SQL数据库的优点：直接向SQL服务器查询，支持SQL形式的传输，改善了记录存储，实现数据锁定。
- 通过SQL Links驱动程序访问SQL服务器还使应用程序继承了workstation数据库的优点，它允许用户：打开一个数据表，实现数据集表和结果集表之间的双向传输，利用允许的索引给数据排序，在数据库中，为当前的session建立书签，以便于在以后的session中使用，通过即时地访问数据源动态地处理SQL数据。

15.2 构造Client/Server 的应用开发环境

- Client/Server的数据库应用程序应该完成客户端和服务端两端的环境配置。在Delphi的Client/Server的应用开发系统中，对于客户端的环境配置主要是对BDE的配置，靠设置SQL Links完成。
- SQL驱动程序的安装决定了Borland Database Engine与目标SQL服务器的连接方式和SQL数据库的打开方式。在安装前必须正确地安装Borland SQL Links驱动程序。
- 在Delphi中，对SQL Links驱动程序的安装是通过BDE Administrator来完成的。在安装Delphi的Client/Server版时会自动地安装BDE的配置工具BDE管理器（BDE Administrator）。
- 这个工具帮助用户修改客户端应用程序的环境配置。配置参数被保存在二进制文件IDAPI.CFG中，在运行应用程序时，将自动地读取该文件。

15.2 构造Client/Server 的应用开发环境



15.2.1 BDE Administrator

数据库页的操作

- 1. 查看BDE数据库别名
- 2. 增添BDE数据库别名
- 3. 修改BDE数据库别名
- 4. 删除BDE数据库别名

15.2.2 BDE Administrator配置页的操作

- 1. 系统的配置
- (1) System INIT的配置
 - System INIT的配置是BDE用来设置一个应用程序的开始的。这些配置信息被存储在Windows Registry中。
- (2) System Formats的设置
 - ① System Formats Date的配置。
 - 指明如何将字符串值转换为日期值。
 - ② System Formats Time的配置。
 - 指明如何将字符串值转换为时间值。
 - ③ System Formats Number的配置。
 - 指明如何将字符串值转换为数字值。
- 2. 驱动程序的配置

15.3 Client/Server的应用开发技术

- Client/Server的应用开发技术与本地应用程序的开发差别不大，主要体现在与数据库服务器的连接以及在服务器上实现存储过程中。
 - 15.3.1 连接数据库服务器
 - 15.3.2 实现服务器上的存储过程

15.3.1 连接数据库服务器

- 在应用程序中建立与数据库之间的联系并通过它对数据库进行控制主要依靠**TDatabase**组件来实现。当应用程序连接到一个远程的**SQL**数据库服务器上并要求控制与**BDE**相关事件的处理时，**TDatabase**组件显得非常重要。
- 当应用程序不需要显式地控制与数据库的联系时，可以不需要在应用程序中显式地创建**TDatabase**对象。在这种情况下，应用程序在运行时会自动生成一个具有默认属性的临时性**Tdatabase**对象。但在应用程序需要控制与数据库的联系时，用户就需要在应用程序中显式地为每个需要控制的联系创建相应的**TDatabase**对象。

15.3.2 实现服务器上的存储过程

- 使用存储过程可以充分利用服务器的处理功能，减少网络传输的工作量。服务器上的存储过程主要靠TStoredProc组件实现。