

Visual Basic 的数据库访问技术漫谈

柴明建 (北京理工大学机电工程学院98研 100081)

摘要: 本文以VB为例,着力介绍了其在开发数据库应用程序时的数据访问接口问题。首先介绍了VB数据访问接口的底层技术,接着分析比较了几种通用的直接面向用户的访问接口,力图使读者在开发VB数据库应用程序时可以合理的选择数据访问接口。文中所介绍的同样适合于VC++。

关键词: DB-Library ODBC API OLE DB O/Jet RDO ADO

1 前言

利用高级语言开发数据库应用程序一般步骤是:

- (1) 打开对要访问数据库的连接;
- (2) 基于用户提供的条件执行查询,从数据库获得返回的结果集;
- (3) 将结果集发送到界面,并对结果集进行处理,从而获取相应的信息。

打开指向数据库的连接是第一步,也是最重要的一步。在利用VB开发数据库应用程序时,面对众多纷繁复杂的数据接口,广大的开发者、特别是一些新手们,该怎样合理地选择自己的数据接口方案呢?本文正是对这个问题做一些分析比较。

2 了解底层接口

2.1 DB-Library

DB-Library (DBLIB) 是最初的底层数据接口,它是基于API(应用程序编程接口)的,它是C语言的函数集,它使VB开发者通过VBSQL函数完成工作。它是第一个连接到SQL Server的本地接口。它已经过时,先是被ODBC API所替代,最近,又被OLE DB所替代。

在DBLIB底层接口上建立的数据访问接口是VBSQL,它实际上是C库的“VB化”接口,但它与作为ODBC API的纯粹的C接口相比,有很大的不同。

2.2 ODBC

ODBC (Open Database Connection) 即开放数据库互连。ODBC驱动程序最初是为连接异种数据库而设计的,特别是Microsoft的ODBC,后来逐渐成为数据库访问的工业标准。一个应用程序,通过附加不同的ODBC驱动程序,可以访问不同的关系数据库。在理论上,ODBC应用程序通过SQL服务器驱动程序连接到SQL Server,通过Oracle驱动程序连接到Oracle数据库,而不需要修改

应用程序可执行文件,这为应用程序访问不同的数据源提供了极大的方便。

在编写一般的ODBC接口时的困难同其他任何使用基于API的接口一样复杂。ODBC API函数有大约55个,每个ODBC调用函数都通常设计成能够接受多个参数,而且每个参数都有许多选项,开发人员使用起来往往会很困惑。

2.3 OLE DB

OLE DB被称为通用的接口集,可以让开发者建立象使用组件对象模型(COM—Component Object Model)的组件一样的数据访问工具,它是Microsoft最新的底层数据访问接口,随着时间的推移,终将代替ODBC而成为数据访问底层接口的工业标准。除了访问数据库外,访问电子邮件、电话服务或Internet等内容的需求越来越广泛,如图1左边的虚线框所示,在OLEDB出现以前,访问不同的数据需要不同的数据访问接口,这就意味着开发人员需要精通不同的数据访问方法。OLE DB的出现,只需要通过基于OLE DB底层接口的ADO对象,就可以轻松实现对各类数据信息的访问。如图1右边的虚线框所示,OLEDB被设计成为访问不同数据源的万能接口。

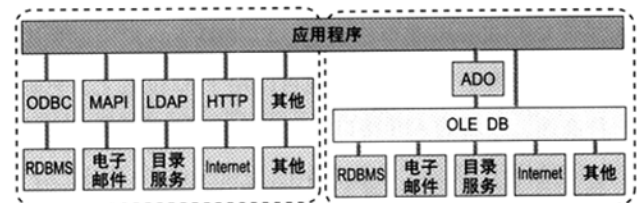


图 1

OLE DB是Microsoft的Universal Data Access平台的一部分。这个结构是为开发多层企业应用程序而设计的,它需要通过企业网或Internet访问不同的相关或不相关的数据源。Universal Data Access由包含显示数据的数

据提供者、使用数据的数据消费者、传输数据的服务组件组成。ADO数据访问接口正是建立在OLE DB之上的。无论是传统的客户机/服务器结构还是基于WEB的结构,这个新的数据访问接口都是适用的。

3 VB6 数据访问接口

VB6的数据访问接口技术最全面的体现在它对SQL Server的访问上,下面我们以VB6访问SQL Server数据库为例,介绍VB6的数据库访问接口技术,VB的数据访问接口共有VBSQL、ODBC API、DAO/Jet、DAO/ODBCDirect、RDO、ADO六种,它们都是建立在底层访问接口之上的。其对数据的访问以及与底层接口之间的关系可用图2表示。

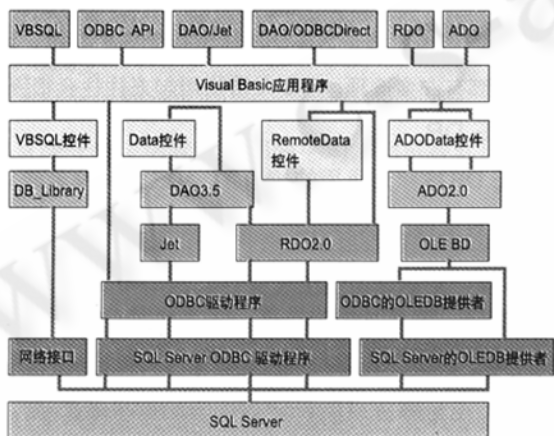


图 2

下面,将分别介绍上述各种数据访问接口,并对其进行分析比较。

3.1 VBSQL

VBSQL是第一个连接到SQL Server的本地接口,它是基于DB Library底层接口的,以VBX和OCX形式提供,既可以用于16位机上,也可以用于32位机上。它是一种高速、底层的API接口,使用自定义的Declare文件来访问适用于VB的C函数。起初,它部分的被ODBC API所取代,以现在的趋势,它连同ODBC API一起,都将最终被Activex数据对象(Activex Data Objects—ADO)所代替。

在SQL Server 7中,虽然仍然用Tabular Data Stream(TDS)协议与外部通信,然而,新的SQL Server 7有一些DB Library不支持的扩展,这种有限的支持表明,在利用VB开发新的应用程序时,你应该停止使用VBSQL。

3.2 ODBC API

ODBC API,即开放数据互连应用程序接口,是相关数据源的高速、底层数据库无关的接口,它使用Declare文件访问基于C的函数,它几乎可以访问SQL Server的所有特性。支持16位和32位系统。

事实上,多年以来,Microsoft ODBC API开发类型对VB的支持一直很少,也就是说,对于VB开发人员来说,直接采用ODBC API开发数据库应用程序并不是一个理想的选择。

3.3 DAO/Jet

Jet数据库引擎是用于执行一组特定功能的代码主体。一个引擎可以是数学处理器、商业规则处理器或其他可以智能控制的代码。Jet有多个版本,Jet1.1、Jet2.0、Jet2.5、Jet3.0、Jet3.5、Jet3.51分别存在于VB的各个版本中,每一个Jet引擎接口都由一个类库来实现,类库提供了一个在引擎接口中访问组件的方法,以便开发人员可以选择最合适的对象模型。其中,Jet3.51是最新的版本,它存在于VB6中。

因为VB6提供的Jet3.51版本不再支持16位的运行环境,因而开发人员对类库的选择是有限的。目前的VB6版本中,开发人员可以选择的类库是Microsoft DAO 3.51 Object Library,它只能用于32位的系统中。Microsoft DAO 2.5 Object Library是16位的,它基本上已经过时了,另外,Microsoft DAO 2.5/3.51 Compatibility Library为VB应用程序的16/32转换提供了最大的兼容性。

作为连接SQL Server的备用方法,在RDO和ADO出现以后,DAO/Jet就逐渐失去了市场,因为大多数开发者们发现DAO对访问ODBC数据源有太多限制。对SQL Server的支持—特别是存储过程和多结果集的支持有限。DAO/Jet访问数据有两种方式,一种是通过“编码”方式,另外一种是通过使用Data控件绑定到数据源。使用Data控件虽然需要更少的代码,但是其加载速度和执行速度都比使用“编码”方式慢,特别是在返回大量记录时,这种差别是非常明显的。

3.4 DAO/ODBCDirect

DAO/ODBCDirect接口是一个新的DAO程序设计模型,这个接口将DAO从Jet数据库引擎中分离出来,并通过RDO将DAO和ODBC数据库重新连接起来,即将DAO对象重新映射到RDO对象的对象接口上,对SQL Server的支持与RDO相同,这样做的目的是在创建独立于数据库的接口时使用熟悉的面向ISAM的DAO程序设计模型,

但是它打破了它与JET引擎的联系,这种联系尤其不适于远程数据访问。虽然采用DAO/ODBCDirect比使用DAO/Jet访问数据可以获得更高的性能,但是,它增加了复杂性,而且,使用DAO/Jet处理数据的许多技术都无法与ODBCDirect一起使用。它只适用于32位。DAO/ODBCDirect虽然没有过时,但是从VB5 SP3以来,该接口就没有技术更新,这就意味着在开发新的应用程序时,最好不要选择这个数据访问接口。

3.5 RDO

RemoteData控件和RDO2.0集RDO将ODBC的灵活性和绑定控件的易用性集于一身。它是在VB4中引入的32位访问接口,对SQL Server提供易于理解、高速灵活的访问接口。虽然ADO可能取代它,但它现在仍然是访问SQL Server时广泛使用的接口。

RDO模型是围绕ODBC API设计的,它支持与API使用的分层结构相平行的对象分层结构。RDO接口不需要像Jet数据库引擎那样的中间层,它依靠远程SQL Server引擎的智能,通过ODBC驱动程序管理器和SQL Server ODBC驱动程序,RDO直接连接到SQL Server。这种方法使开发人员对用户接口和后台接口有更多的控制和更大的灵活性。使用内置的RDO对象的属性和方法,开发人员就可以完成开发工作。

同样使用RDO可以用“编码”方式和RemoteData控

件绑定两种方式来访问数据。使用RDO可以获得更高的速度和性能,另外,使用RDO时,甚至可以使用异步连接来避免数据访问时的延迟。

3.6 ADO

ADO是Microsoft最新的数据访问接口,它被设计为更多类型信息的接口——而不仅仅是相关的关系数据库。在OLE DB中引入这个对象接口是为了随IIS(Internet Information Server)一起使用。它是映射了OLE DB API的面向对象的简单接口,实际支持SQL Server所有的特性。它只适用于32位系统。

ADO接口可以通过SQL Server的OLE DB提供者直接连接到SQL Server。ADO还可以通过ODBC的OLE DB提供者进行连接,而该提供者与我们在DAO和RDO使用的SQL Server驱动程序对话。ADO只有五个对象,比RDO使用更加方便,更容易掌握。ADO也有“编码”方式和ADO Data控件绑定两种数据访问方式。对于分布式结构和WEB应用程序来说,ADO无疑是最好的选择。通过HTML将ADO代码和VBScript代码嵌入ASP(Active Server Pages)页面,可以轻松实现对WEB数据的访问。

4 几种接口的比较

下面将对各数据访问接口加以比较,如下表1。阐述了各数据访问接口的特性及其状态。■

表 1

接 口	说 明	状 态
VBSQL, 16位和32位支持。	与DBLIB类似,是高速、底层的API。可访问SQL Server所有的特性。作为产品级的接口,它很难追踪并被广为接受。它只与SQL Server一起使用,使用自定义的Declare文件来访问适合VB的C函数。	已过时
ODBC API 16位和32位支持。	相关数据源的高速、底层数据库无关的接口。可访问SQL Server所有的特性,使用自定义的Declare文件来访问适合VB的C函数。没有特定的VB支持,绑定技术和32位字符串问题使得接口很复杂。虽然有广泛的支持,但正在被OLE DB所取代。	正在过时
DAO/Jet支持16位和32位。	ISAM数据库中间层的上层,有访问ODBC数据源的能力。开发速度较快,性能较满,对SQL Server功能—特别是存储过程和多结果集的支持有限。	基本停滞目前没有什么修改
DAO/ODBCirect,只支持32位。	将DAO对象重新映射到等价的RDO对象的对象接口。对SQL Server的支持与RDO相同。支持异步操作,但没有事件。比Jet快,比基于API的模型稍慢。	自VB SP3以来没有重大改进
RDO,只支持32位。	ODBC面向API的简单接口。与ODBC API同样支持SQL Server,但编程更简单,更安全。有高级的批处理模式、扩展的异步和事件驱动支持。在某种程度上,比Jet快,比基于API的模型稍慢。	自VB SP3以来没有重大改进
ADO,只支持32位。	影射了OLE DB API的面向对象的简单接口,支持广泛的数据源。2.0版提供了与RDO2.0相似的功能。实际支持SQL Server的所有特性。可以访问WEB数据。	即将代替上述所有接口

参考文献

- [美] William R. Vaughn 著《Visual Basic与SQL Server权威指南》第六版 北京大学出版社
- [美] Roger Jennings 著《Visual Basic 6 数据库开发人员指南》机械

工业出版社

- [美] M. Spenik, O. Sledge 等著 熊桂喜等译《Microsoft SQL Server 7 系统管理指南》清华大学出版社