

基于 SQLDMO 的数据库定时备份的实现方法

Implementation of the Database Timing Backup Based on SQLDMO

杨 峰 陈和平 (武汉科技大学 信息科学与工程学院 湖北武汉 430081)

摘 要: SQL Server 分布式管理对象 SQLDMO 可以在程序中用命令方式操作 SQL Server,并具有停止/启动服务、建立数据库表、添加用户权限、备份/恢复数据库、数据导入/导出/发布等功能。本文在介绍 SQLDMO 基本特点的基础上,详细讨论了在 VC 环境下,采用 SQLDMO 封装 SQL Server 2000 数据库的相关操作,并用程序实现数据库定时备份的基本方法,达到了自动、安全、灵活备份数据的目的。

关键词: SQLDMO VC 定时备份 SQL Server 封装

随着电子技术的飞速发展和信息技术的广泛应用,计算机系统越来越成为企业不可或缺的数据载体,数据日益成为企事业单位日常运作中非常重要的部分和领导决策的依据。但是,计算机内的数据非常容易丢失和遭到破坏(如黑客攻击、病毒入侵、硬件故障及人为误操作等),如何利用数据备份技术来保证数据的安全已成为人们迫切需要研究的一个课题。目前数据库备份大多数是直接或间接操纵数据库的管理工具,这种方法不能达到自动进行备份的要求,并具有一定的风险性。为此,作者在 VC 环境下采用 SQLDMO 封装数据库操作,实现了自动地定时进行数据库备份,使数据更安全、操作更灵活、应用更广泛,并且不局限于专业人员,可根据管理员的要求定制个性服务来进行数据备份。

1 SQLDMO 概述

1.1 SQLDMO 介绍

SQLDMO (SQL Distributed Management Objects, SQL 分布式管理对象)可封装 Microsoft SQL Server 数据库中的对象,能完成 SQL Server 80% 的功能,是一个 COM 对象^[1]。

SQLDMO 是 SQL Server 中的 SQL Server 企业管理器所使用的应用程序接口,允许用支持自动化或 COM 的语言编写应用程序,以管理 SQL Server 安装的所有部分。SQLDMO 并不是一种新的数据库访问技术,它不关心数据库存储的信息,主要负责数据库本身的结构和维护以及对包含数据库的 SQL Server 操作,具有以下功能:

1) 能够轻松地开发适合用户环境的定制式 SQL Server 管理应用程序;

2) 应用程序可以实现 SQL Server 企业管理器的所有功能;

3) 可以增加登录帐户、控制复制、引入数据、引出数据以及执行其他管理任务;

4) 可以实现显示数据库的对象(包括数据库、表、视图、存储过程),获取局域网中的 SQL Server 数据库服务器列表,某个数据库服务器上的数据库字段属性,并提供添加、修改、删除这些对象的功能。

1.2 SQLDMO 的核心对象

SQLDMO 核心对象是按 SQL Server 企业管理器同一基本组织的分层顺序进行组织的,核心 SQL DMO 对象的层次如图 1 所示:

SQL Server 的分布式管理对象使用了一种层次体系结构,在 SQL Server 的 DMO 对象框架的顶部为应用程序,接下来是 SQL Server 对象,SQL Server 对象表示一种物理 SQL Server 系统,在 SQL Server 对象下面的各种对象集合允许使用 SQL Server 系统的不同应用,例如,RemoteServers 对象集合允许创建和管理远程服务器;同样 JobServer 对象集合允许控制 SQL Server 的任务作业和报警;在 DataBases 集合中的每一个 DataBase 对象表示一个 SQL Server 数据库,每一个数据库对象包含一个或多个表对象,以及其他类型的 SQL Server 对象,例如触发器、视图和存储过程;类似的,Table 对象层包括了列、触发器、键和索引对象;最后,Replication 对象允许建立和控制 SQL Server 的数据库复制。

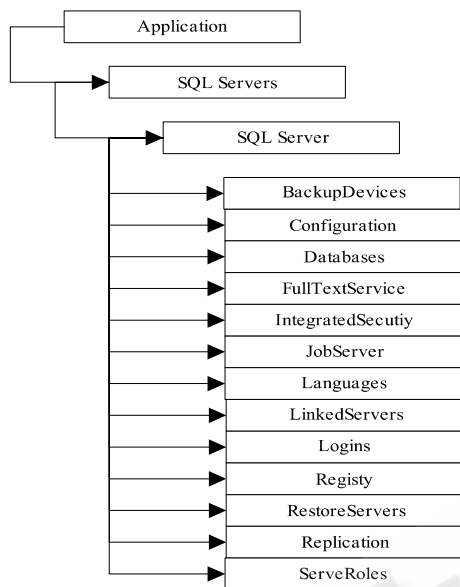


图 1 SQLDMO 的对象层次图

2 定时备份的实现方法

2.1 SQL Server 数据库备份方法

在 SQL Server 中,备份是动态的,即在进行备份的同时允许用户对数据库进行其他操作;同时备份是定期的,而不是实时的,利用备份并不能完全恢复数据库,只能将数据库恢复到制作备份的那一刻。当磁盘出现故障造成数据库损坏时,可以首先利用备份恢复数据库(大部分的数据),然后运行数据库日志,将备份后所做的操作重新再做一遍,从而将数据库完全恢复。SQL Server 备份数据库有以下四种方法^[2]:

1) 完全数据库备份 制作数据库中所有内容的副本。它在备份过程中需要花费的时间和空间最多,不宜频繁进行。恢复时,仅需要恢复最后一次全库备份即可,该备份以后的修改都将丢失。

2) 差异数据库备份 只备份最后一次全库备份后被修改的数据页。备份的时间和空间较少。恢复时,先恢复最后一次全库备份,再恢复最后一次差异备份。

3) 事务日志备份 只备份最后一次日志备份后所有的事务日志记录。备份所用的时间和空间更少。利用日志备份进行恢复时,可以指定恢复到某一个事务,这是全库备份和差异备份所不能做到的。但利用日志备份进行恢复时,需要重新执行日志记录中的修改命令来恢复数据库中的数据,所以通常恢复的时间较长。建议每周进行一次全库备份,每天进行一次差异备份,

每小时执行一次日志备份。

4) 文件或文件组备份 备份某个数据库文件或数据库文件组。必须与事务日志备份结合才有意义。例如,某数据库中有两个数据文件,一次仅备份一个文件,而且在每个数据文件备份后,都要进行日志备份。在恢复时,使用事务日志使所有的数据文件恢复到同一个时间点。

2.2 定时备份的实现

SQLDMO 提供了一个 OLE (Object Linking and Embedding, 对象连接与嵌入) 接口,使 SQL Server 的数据库管理功能易于访问。因为 SQLDMO 是作为一组 OLE 对象实现的,所以这些对象只能在 32 位客户应用程序中使用。SQL Server 的 SQLDMO 功能可以由任何 32 位、OLE 兼容的开发工具使用。采用 SQLDMO 实现数据库定时备份,主要有 6 个步骤:导入 SQLDMO 类型库、创建 SQL Server 对象、选择数据库、选择路径、选择时间以及创建 SQL Server 备份对象,其工作流程如图 2 所示:

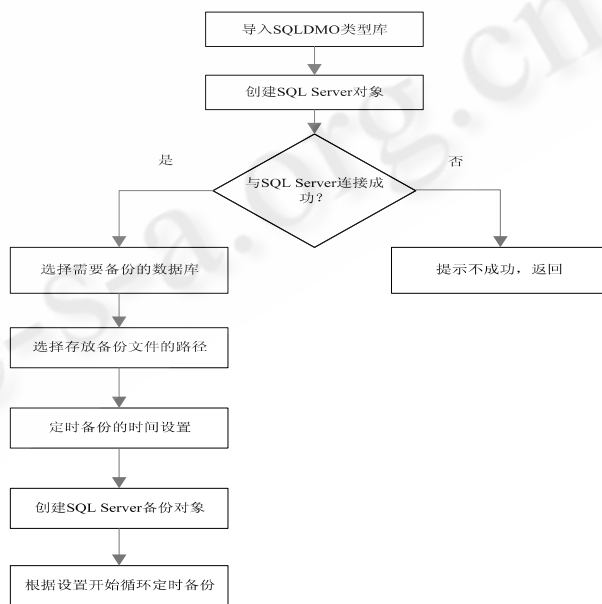


图 2 定时备份流程图

上述的定时备份功能可设为开机自动运行,只要配置好相关设置,不再需要人工操作,系统自动定时备份。下面详细介绍配置过程及实现方法。

2.2.1 定时备份前的准备工作

本文通过 SQLDMO 的资源文件 sqldmo.rll 来导入

表 1 SQLDMO 的文件

文件名	说明
Sqlqmo. hlp	DMO 帮助文件
Sqlqmo. dll	SQLDMO 进程内对象服务器的动态链接库
Sqlqmo. rll	SQLDMO 资源文件
Sqlresld. dll	SQL Enterprise Manager 资源链接库加载器
Sqlsvcs. dll	数据库服务层
Sqlsvcs. rll	数据库服务层资源

SQLDMO 类型库, 关键代码如下: #import " C: \ Program Files \ MicrosoftSQLServer \ 0 \ Tools \ Binn \ Resources \ 1033 \ sqlq

mo. rll" [3]。

2) 创建一个 SQL Server 对象并将该对象连接到 SQL Server

SQLDMO 的 SQLServer 对象 SQLDMO:: _SQLServer2Ptr 提供与 SQL Server 连接的接口, 首先创建一个 SQL Server 对象 spSqlServer2。然后连接到 SQL Server, 连接前要先通过 SQL Server 的身份验证, SQL Server 提供两种身份验证方式, Windows 身份验证和 SQL Server 身份验证。若选择 Windows 身份验证连接 SQL Server, 使用 spSqlServer2 - >

LoginSecure = (VARIANT_BOOT) (TRUE) 进行验证; 若选择

SQL Server 身份验证, 则使用 spSqlServer2 - > LoginSecure =

(VARIANT_BOOT) (FALSE) 进行验证。

通过身份验证后, SQLDMO:: _SQLServer2Ptr 提供 Connect 方法与 SQL Server 进行连接,

spSqlServer2 - > Connect ((LPCTSTR) g_sServerName, (LPCTSTR) g_sUser, (LPCTSTR) g_sPass),

g_sServerName 为 SQL Server 服务名, g_sUser 为登录 SQL Server 用户名, g_sPass 为登录密码。连接成功后即可进行数据库定时备份的配置。

2.2.2 定时备份的相关配置

1) 需要备份的数据库及其备份路径的选择

SQL Server 对象 SQLDMO:: _SQLServer2Ptr 与 SQL Server 连接成功后, 用 SQLDMO 的 Databases 对象 SQLDMO:: DatabasesPtr 列出 g_sServerName 服务下

的所有数据库, SQLDMO:: DatabasesPtr 提供 Item 方法得到数据库

名。关键代码如下: SQLDMO:: DatabasesPtr sp-Databases = spSqlServer2 - > Databases;

LONG lCount = spDatabases - > Count; // 数据库个数

For (LONG l=0; l<lCount; l++)

SQLDMO:: _Database2Ptr spDatabase2 =

spDatabases - > Item (LONG (l+1));

数据库的选择界面如图 3 所示:



图 3 选择数据库

单击图 3 的“下一步”按钮, 即可选择备份路径, 数据库和路径均可多选。

2) 备份时间的设置

本文提供的定时备份是以天为计时单位, 备份的时间精确到秒。如图 4 所示:



图 4 时间配置

由于定时备份将文件备份到了磁盘, 日积月累会占用空间, 浪费资源, 选择图 4 的“自动删除 10 次前的数据”复选框, 程序可自动删除旧的备份文件以释放空间。

2.2.3 定时备份的具体实现

完成了定时备份的准备工作和相关配置后, 开始数据库的定时循环备份。SQLDMO 的服务器对象 SQLDMO:: _Backup2Ptr 提供 (下转第 118 页)

(上接第 103 页)

SQL Server 数据库的备份接口,将其放在定时器里面循环执行。一旦定时时间到,程序会自动将备份文件放到指定的磁盘目录下,供日后数据库的恢复使用。其实现的关键代码如下:

```
SQLDMO : : _Backup2Ptr spBackup2 = NULL ;  
spBackup2 - > Action = SQLDMO : : SQLDMOBack-  
up_Database ;  
spBackup2 - > Database = ( LPCTSTR ) sData-  
baseName ; // sDatabaseName 为需要备份的数据库名  
spBackup2 - > Files = ( LPCTSTR ) sFile ; // sFile  
为备份路径
```

3 结束语

SQLDMO 的功能非常强大,利用它可以重建索引、检查页面完整性、添加索引、改变对象属性等,并可以轻松完成 SQL Server 的各项功能来简化用户的操作。

作者将基于 SQLDMO 的数据库定时备份技术应用于一某工业消防监控系统和三峡水调自动化系统等实际工程,效果明显。由于数据库能够定时备份,数据库的一些故障问题(如数据库崩溃、数据丢失)便不再可怕。实际应用表明,采用 SQLDMO 开

发 SQL Server 的应用程序是有效的和很有必要的。

参考文献

- 1 高云君,章方铭,张学杰. 基于 SQLDMO 和 ActiveX 的数据分发监控系统的设计与实现. 计算机应用研究,2004,(6):120-122.
- 2 马祖长,孙怡宁. SQL Server 2000 的数据库备份与还原. 电子技术,2002,(7):120-122.
- 3 Microsoft SQL Server 2000 联机帮助.