

文章编号: 1005-0523(2004)01-0055-03

# 基于 VC++ 应用程序的 ODBC 数据源自动登录

张 龙<sup>1</sup>, 熊国良<sup>2</sup>, 李 焱<sup>1</sup>

(华东交通大学 1. 机电工程学院; 2. 教务处, 江西 南昌 330013)

**摘要:** 以一个 Access 数据库为例, 分析了 Windows 注册表中的键值, 介绍了如何在 VC++ 编制的应用程序中实现 ODBC 数据源的自动登录.

**关键词:** ODBC; VC++; Windows 注册表; 数据源

**中图分类号:** TP311

**文献标识码:** A

## 1 问题的提出

一个数据库应用程序通常包括后端数据库和前端数据库应用系统两部分组成. VC++ 开发的前端应用系统与后端数据库的连接可由 ODBC 来实现.

ODBC (OpenDatabaseConnectivity) 是开放式数据库连接的缩写, 它提供了极大的互操作性, 使得一个应用程序可访问不同的数据库管理系统 DBMS, 开发者可以针对某一类 DBMS 进行开发. 因此, 程序通过 ODBC 接口可以使用任何提供 ODBC 驱动程序的产品. 而 ODBC 与数据库之间的连接通过 DSN (DataSourceName) 来描述, 程序员通过控制 DSN 就可以实现与数据库的连接.

通常, 为了实现 ODBC 与后端数据库的连接, 需要用户在控制面板中手动设置 ODBC 项目来登录与数据库实现连接的 DSN. 这对于不了解 ODBC 的一般用户来说是有一定的难度的. 另外, 有时由于重新安装驱动程序, 或其它程序也会破坏 ODBC 驱动程序原有的参数, 使应用程序与数据库的连接断开. 那么能否让应用程序本身自动地实现数据源的连接呢? 解决办法就是编制程序代码实现数据源

的自动登录.

## 2 解决方法

Windows 的系统注册表中包含了关于系统配置及运行的重要信息. DSN 的信息全部存储在注册表中, 通过修改注册表就可以实现 ODBC 的登录.

### 2.1 ODBC 数据源参数分析

ODBC 数据源管理程序设置的 DSN 信息在 Windows 注册表中的存放位置为 `HEKY_CURRENT_USER \ Software \ ODBC \ ODBC.INI \ .ODBC.INI` 子键及其下部分子键层中, 存放着用户设置的系统 DSN 信息列表值. 下面以一个名为“Sample”的 Access 类型数据源为例来说明用户设置的 DSN 信息的存放位置, 含义和数据类型.

另外还有一条键值位于 `HEKY_CURRENT_USER \ Software \ ODBC \ ODBC.INI \ ODBC Data Sources \ 下`. 值名为 Sample, 值为 Microsoft Access Driver (\* .mdb), 数据类型为 String.

### 2.2 功能模块的编制

要实现数据源的自动登录就必需在程序中编制功能模块使得应用程序能够在 Windows 系统注册

收稿日期: 2003-09-22

作者简介: 张 龙(1980-), 男, 江西奉新人, 华东交通大学机电工程学院在读研究生.

表中创建和修改对应于表 1 (MS Access 类型的名为 Sample 的数据源的参数列表) 的所有注册表键值. Windows API 函数是最底层和最基础的应用程序的

编程接口. 所以下面就以 Win<sup>32</sup> API 函数来编制功能模块, 编程环境为 VC++6.0.

表 1 MS Access 类型的名为 Sample 的数据源的参数列表

参数名称	参数所在的子键层及其常规值		数据类型
	..... \ ODBC.INI \ Sample	Sample \ Engines \ Jet	
DBQ	数据库的路径及名称		String
Driver	驱动程序路径及名称		String
DriverID	驱动程序标识号(对应 Access 为 25)		Integer
FIL	对应 Access 类型固定值为(MS Access)		String
SafeTransactions	支持事务操作的个数(0)		Integer
UID	用户名(可为空)		String
ImplicitCommitSync		(Yes)	String
MaxBufferSize		缓冲区大小(2048)	Integer
PageTimeout		页超时(5)	Integer
Threads		最大线程数(3)	Integer
UserCommitSync		(Yes)	String

```
//以下创建 ODBC 数据源在注册表中的子键
CString strSubKey = "SOFTWARE \ ODBC \ ODBC.INI \
Sample";
HKEY hKey;
DWORD dw;
LONG return1 = RegCreateKeyEx (HKEY _ CURRENT _
USER, (LPCTSTR)strSubKey, 0, NULL, REG _ OPTION _ NON _
VOLATILE, KEY _ WRITE, NULL, &hKey, &dw);
if(return1! = ERROR _ SUCCESS)
return false;
//以下设置数据源的各项参数
CString strDbq="G: \ \ Sample.mdb";
RegSetValueEx (hKey, "DBQ", 0, REG _ SZ, (CONST
BYTE *)((LPCTSTR) strDbq), strDbq.GetLength());
CString strDriver = "C: \ \ WINDOWS \ \ SYSTEM \ \
ODBCJT32.DLL";
RegSetValueEx (hKey, "Driver", 0, REG _ SZ, (CONST
BYTE *)((LPCTSTR) strDriver), strDriver.GetLength());
CString strFIL="MS Access";
RegSetValueEx (hKey, "FIL", 0, REG _ SZ, (CONST
BYTE *)((LPCTSTR) strFIL), strFIL.GetLength());
CString strUID="";
RegSetValueEx (hKey, "UID", 0, REG _ SZ, (CONST
BYTE *)((LPCTSTR) strUID), strUID.GetLength());
DWORD dwDriverId=25;
RegSetValueEx (hKey, "DriverId", 0, REG _ DWORD,
(CONST BYTE *)(& dwDriverId), sizeof(dwDriverId));
DWORD dwSafeTransactions=0;
RegSetValueEx (hKey, "SafeTransactions", 0, REG _
```

```
(dwSafeTransactions));
RegCloseKey(hKey);
//以下创建 ODBC 数据源的 Jet 子键
strSubKey += "\ \ Engines \ \ Jet";
LONG return2 = RegCreateKeyEx (HKEY _ CURRENT _ US-
ER,
(LPCTSTR)strSubKey, 0, NULL, REG _ OPTION _ NON _
VOLATILE, KEY _ WRITE, NULL, &hKey, &dw);
if(return1! = ERROR _ SUCCESS)
return false;
//以下设置 Jet 子键下的各项键值
CString strImplicit="";
RegSetValueEx (hKey, "ImplicitCommitSync", 0, REG _ SZ,
(CONST BYTE *)((LPCTSTR) strImplicit), strImplicit.GetLength
());
CString strUserCommit="Yes";
RegSetValueEx (hKey, "UserCommitSync", 0, REG _ SZ,
(CONST BYTE *)((LPCTSTR) strUserCommit), strUserCommit.
GetLength());
DWORD dwPageTimeout=5;
RegSetValueEx (hKey, "PageTimeout", 0, REG _ DWORD,
(CONST BYTE *)(& dwPageTimeout), sizeof(dwPageTimeout));
DWORD dwThreads=3;
RegSetValueEx (hKey, "Threads", 0, REG _ DWORD, (CON-
ST BYTE *)(& dwThreads), sizeof(dwThreads));
DWORD dwMaxBufferSize=2048;
RegSetValueEx (hKey, "MaxBufferSize", 0, REG _ DWORD,
(CONST BYTE *)(& dwMaxBufferSize), sizeof(dwMaxBuffer-
Size));
RegCloseKey(hKey);
```

```
//以下设置 ODBC 数据库引擎名称
return1 = RegOpenKeyEx ( HKEY _ CURRENT _ USER,
“SOFTWARE \ \ ODBC \ \ ODBC . INI \ \ ODBC Data
SourceS”, 0, KEY _ WRITE, & hKey );
if (return1! = ERROR _ SUCCESS)
return false ;
CString strDbType = “Microsoft Access Driver ( * . mdb ) ” ;
RegSetValueEx ( hKey, “Sample”, 0, REG _ SZ, ( CONST
BYTE * ) (( LPCTSTR ) strDbType, strDbType . GetLength ( ) ) ;
```

### 3 结束语

本文实现了在 VC++ 6.0 环境下应用程序

ODBC 数据源的自动登录. 免去了在控制面板中手动设置数据源的过程, 有利于提高所编制的应用程序的质量. 对应于其他的开发软件和不同的数据库管理系统 DBMS 同样可以利用 Win32 API 函数中的注册表操作函数来实现. 以上源代码在 Win98/2000+VC++6.0 环境下编译通过.

### 参考文献:

- [1] 同志工作室. Visual C++ 6.0 数据库开发实例[M]. 北京: 人民邮电出版社, 2001.
- [2] 张双民, 等. Visual C++ 6.0 实战与精通[M]. 北京: 清华大学出版社, 2000.

## Auto-login of ODBC Data Source in a VC++-based on Application Program

ZHANG Long<sup>1</sup>, XIONG Guo-liang<sup>2</sup>, LI Yi<sup>1</sup>

(1. School of Mechanical Eng.; 2. Office of Educational Administration, East China Jiaotong Uni., Nanchang 330013, China)

**Abstract:** Taking a database of Access for example, the values of the Windows registration list are analyzed and the method to realize the auto-login of the ODBC data source in a VC++-based application program is investigated in this paper.

**Key words:** ODBC; VC++; Windows registration; data source

(上接第 46 页)

## Discussion on Exhaust Test and Exhaust Maintenance of I/M System of Our Country

LIN Yu, YU Hui-nian, SHUANG Ju-rong, LI Hai-jing

(Guangzhou Environmental Monitoring Center, Guangzhou 510030, China)

**Abstract:** There exists a common phenomenon that ignorance of exhaust test and exhaust maintenance in vehicle maintenance industry of our country. In this paper, the reasons and development of I/M system of our country are discussed.

**Key Words:** I/M; ignorance; mode