

# UG 二次开发中外部数据访问方法研究\*

黄曙荣, 秦珂, 韩冻冰

(盐城工学院 CAD 中心, 江苏 盐城 224003)

**摘要:**在 UG 应用系统中,经常需要访问文本文件、Excel 表及数据库表等各种外部数据。对 UG 二次开发过程中外部数据的访问方法作了研究,介绍了各种方法的编程步骤、示例代码和使用特点,这对涉及外部数据访问的 UG 应用系统的二次开发有较大的借鉴意义。

**关键词:**UG 二次开发;外部数据访问;UG/Open API;MFC

**中图分类号:**TP391.72    **文献标识码:**A    **文章编号:**1671-5322(2007)02-0015-03

随着 UG 系统的普及应用,基于 UG 平台的二次开发工作显得越来越重要,在 UG 的二次开发应用系统中,经常需要与各种外部数据打交道,UG/Open API 提供了丰富的文件访问操作函数,利用这些操作函数及 VC++ 的 MFC 类库,可以方便地实现外部数据读写访问编程。

## 1 外部数据访问的主要方法

在 UG 二次开发应用系统中,主要访问的外部数据有文本文件、Excel 表、关系型数据库表等,通常可以通过以下方法访问这些外部数据<sup>[1]</sup>:

- (1)通过 UG/Open API 提供的文件操作函数访问外部数据。
- (2)通过 ODBC 数据源访问外部数据。
- (3)通过 ADO 访问外部数据。

## 2 外部数据访问方法的实现

### 2.1 通过 UG/Open API 提供的文件操作函数访问文本文件数据

UG/Open API 中有关文件操作的函数包含在头文件 `uf_cfi.h` 中,通过函数 `uc4500` 可以打开二进制文件,相关的文件操作函数有 `uc4510`、`uc4511`、`uc4512`、`uc4513` 及 `uc4520`、`uc4521`、`uc4522`、`uc4523`,通过函数 `uc4504` 可以打开文本

文件,相关的文件操作函数有 `uc4514`、`uc4524`、`uc4525`,函数 `uc4540` 用于关闭文件,判断文件是否存在使用函数 `UF_CFI_ask_file_exist`。

通过文件操作函数访问文本文件的程序流程如图 1 所示。

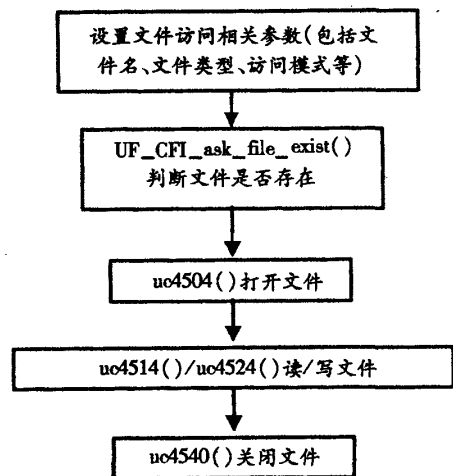


图 1 UG/Open API 中访问文本文件流程

Fig. 1 Access text file in UG/Open API

使用这种方法访问外部数据时,要注意以下问题:

\* 收稿日期:2006-10-25

作者简介:黄曙荣(1969-),女,江苏盐城人,副教授,主要研究方向为数据库应用系统开发、计算机辅助设计、PDM/PLM。

(1) UG/Manager 和 IMAN 中的文件和文件夹操作不能使用 uf\_cfi.h 中函数;

(2) UG/Open API 中定义了一些常量标识符,用以限制文件操作相关字符串的长度:

文件名长度不能超过 UF\_CFI\_MAX\_FILE\_NAME\_LEN;文件夹路径的长度不能超过 UF\_CFI\_MAX\_PATH\_NAME\_LEN;整个文件路径的长度不能超过 UF\_CFI\_MAX\_PATH\_NAME\_LEN。

## 2.2 通过 ODBC 数据源访问 Excel 表或关系型数据库表数据

ODBC(Open Database connectivity 开放数据库互连)提供了应用程序接口(API),使得外部数据访问应用程序可以通过 ODBC 驱动器与指定的外部数据相关联。可访问的外部数据可以是文本文件、Excel 表、关系数据库等。

在 Visual C++ 中,MFC 的 ODBC 类 CDatabase(数据库类)、CRecordSet(记录集类)为 UG 二次开发中外部数据访问提供了切实可行的解决方案。

用 MFC 和 ODBC 进行外部数据访问编程时,一般过程如下:

(1) 创建 ODBC 数据源 DSN(Data Source Name)。在控制面板的管理工具中可以打开 ODBC 数据源管理器,创建 DSN。

(2) 利用 MFC 的 ODBC 类 CDatabase 和 CRecordSet 实现对外部数据的访问编程。

CDatabase 类用来提供对数据源的连接,通过它可以对数据库源进行操作:调用 Open 或 OpenEx 成员函数指定一个 DSN,就建立起了与 DSN 的底层数据源的连接;成员函数 Close 用于关闭与数据源的连接。CRecordset 类提供了对表记录进行操作规程的许多功能,如查询记录、添加记录、删除记录、修改记录等<sup>[2]</sup>。

通过 ODBC 访问一个 Excel 表的实例代码段如下:

(1) 定义 CDatabase 类对象及相关变量

```
CDatabase Database; // 定义一个 MFC 的 CDatabase 数据库类对象 Database
```

```
CString SQLCommand = "select `Serial_Number`, `Outer_Diameter`, `Inner_Diameter`, `Height` from Sheet1 $ "; // SQL 查询语句
```

```
DWORD dwOptions = 0; // 设置连接的建立方式
```

```
CDBVariant temp; // 数据库数据通用类型
```

```
int result; // 定义储存返回值的变量
```

```
char buf[133];
```

```
result = Database.OpenEx( "DSN = Standard Part", dwOptions ); // 打开数据库
```

(2) 定义 CRecordset 类对象并在 UG NX 的信息窗口中显示数据

```
if( result != 0 ) // 如果成功打开
```

```
{
```

```
CRecordset rs( &Database ); // 定义记录集
```

```
if ( rs.Open( CRecordset::snapshot, SQLCommand ) != false )
```

```
{
```

```
rs.MoveFirst( );
```

```
short nFields = rs.GetODBCFieldCount( );
```

// 获取记录的字段数目

```
UF_UI_Open_listing_window( );
```

```
while( ! rs.IsEOF( ) )
```

```
{
```

```
for( short index = 1; index < nFields; index
```

```
++ )
```

```
{
```

```
rs.GetFieldValue( index, temp );
```

```
sprintf( buf, "%f", temp.m_dblVal );
```

```
UF_UI_write_listing_window( buf );
```

```
UF_UI_write_listing_window( "\t" );
```

```
}
```

```
UF_UI_write_listing_window( "\n" );
```

```
rs.MoveNext( );
```

```
}
```

```
rs.Close( ); // 关闭记录集
```

```
}
```

```
Database.Close(); // 关闭数据库连接
```

```
}
```

## 2.3 通过 ADO 访问外部数据

ADO(ActiveX Data Objects)是目前在 Windows 环境中比较流行的客户端数据库编程技术。ADO 中最重要的对象有 3 个: Connection、Command 和 Recordset,它们分别表示连接对象、命令对象和记录集对象。

ADO 编程有 3 种方式:使用预处理指令 #import、使用 MFC 中的 CDispatchDriver 和直接使用 COM 提供的 API,这 3 种方式中,第 1 种最为简便。

通过 ADO 访问数据库的一般步骤如下:

(1) 在 stdafx.h 文件中添加对 ADO 支持的代码:

```
#import "C:\Program Files\Common Files\System\ado\msado15.dll" no_namespace rename( " EOF ", " adoEOF ") rename( " BOF ", " adoBOF ")
```

代码中,预处理命令#import 是编译器将此命令中所指定的动态链接库文件引入程序中,并从动态链接库文件中抽取出其中的对象和类的信息。参数 no\_namespace 表示不使用 ADO 的名字空间。为了避免出现常量名冲突,需要将 EOF 改名为 adoEOF、BOF 改名为 adoBOF。

#### (2) 建立到数据源的连接

只有建立了与数据库服务器的连接后,才能进行其他有关数据库的访问和操作。ADO 使用 Connection 对象来建立与数据库服务器的连接,它相当于 MFC 中的 CDatabase 类,和 CDatabase 类一样,调用 Connection 对象的 Open 即可建立与服务器的连接。注意:在使用 ADO 对象之前必须调用 CoInitialize 来初始化 COM 环境。相关示例代码如下:

```
CoInitialize( 0 ); // 初始化对 COM 的调用
_ConnectionPtr pConnection = NULL; // 定义 ADO 对象 Connection 的智能指针变量
_RecordsetPtr pRecordset = NULL; // 定义 ADO 对象 Recordset 的智能指针变量
pConnection.CreateInstance( __uuidof( Connection ) ); // 初始化 Connection 指针
pRecordset.CreateInstance( __uuidof( Recordset ) ); // 初始化 Recordset 指针
CString strConnectionString( " Provider = Microsoft.Jet.OLEDB.4.0; Data Source = f:\ug\screw.mdb " ); // 连接字符串
CString strSqlCommandString( " SELECT * From screw_table " ); // 查询语句
HRESULT hr = E_UNEXPECTED;
// 连接数据库
hr = pConnection -> Open( ( _bstr_t ) strSqlCommandString, "", "", adModeUnknown );
```

#### (3) 访问数据源(查询、更新、删除等)

以下代码通过记录集将数据显示到 UG 信息

窗口中:

```
hr = pRecordset -> Open( ( _bstr_t ) strSqlCommandString,
    _variant_t( ( IDispatch * ) pConnection, true ),
    adOpenStatic,
    adLockOptimistic,
    adCmdText ); // 获取记录集
CString tmp;
UF_UI_open_listing_window();
// 输出字段名
long lFieldsCount = -1;
lFieldsCount = pRecordset -> GetFields() -> Count;
for ( long i = 0; i < lFieldsCount; i++ )
{
    tmp.Format( "%-7s", ( LPCSTR ) pRecordset -> GetFields() -> Item[ i ] -> Name );
    UF_UI_write_listing_window( ( LPTSTR ) ( LPCSTR ) tmp );
    UF_UI_write_listing_window( "\t" );
}
UF_UI_write_listing_window( "\n\n" );
// 输出所有记录
pRecordset -> MoveFirst();
while( ! pRecordset -> adoEOF )
{
    tmp = pRecordset -> GetCollect( 0L );
    UF_UI_write_listing_window( ( LPTSTR ) ( LPCSTR ) tmp );
    ..... // 输出记录中其他字段值
    pRecordset -> MoveNext();
}
pRecordset -> Close();
pRecordset = NULL;
(4) 关闭数据源
相关示例代码如下:
pConnection -> Close();
pConnection = NULL;
CoUninitialize();
```

(下转第 54 页)

**参考文献:**

- [1] 魏小龙. MSP430 系列单片机接口技术及系统设计实例[M]. 北京:北京航空航天大学出版社,2003.
- [2] 万福君. 单片微机系统设计与应用[M]. 北京:中国科学技术大学出版社,2001.
- [3] 赵负图. 传感器集成电路手册[M]. 北京:化学工业出版社,2002.
- [4] 胡大可. SP430 系列超低功耗 16 位单片机原理与应用[M]. 北京:北京航空航天大学出版社,2004.

## The Design of the Low Power Loss and High Accuracy Digital Thermometer

XU Zhi - guo

(Yancheng Civil Aviation Station, Jiangsu Yancheng 224051, China)

**Abstract:** The digital thermometer design mainly starts from the hardware shaping and software design, which can effectively reduce the power loss, lengthen the operating time, increase the measuring accuracy. The software design uses the inter mittent working pattern, which reduces the operating time of CPU in guarantee system performance so that the system always works in low - consumption patterns and the system energy consumption is very low. Meanwhile, the slope A/D converter integrated in the microcontroller, multi - point calibration technology and the linear interpolation method are used in order to improve the system precision.

**Keywords:** sensor; single - chip computer; thermometer; low power consumption; A/D

(上接第 17 页)

### 3 结论

在 UG 二次开发过程中,确定选择哪种方法访问外部数据,取决于具体的应用系统要求。通过 UG/Open API 编程较适合于对外部文本文件的访问,可方便实现用户与 UG 信息窗口交互等

功能;通过 ODBC 和 ADO 技术可以访问各种支持的数据源,包括文本文件、Excel 表及各种关系型数据库表等,ADO 与 ODBC 相比,ADO 技术使用更加简便,免去了繁琐的配置数据源工作,但有些数据库仅支持 ODBC,不直接支持 ADO 数据库技术。

**参考文献:**

- [1] 董正卫,田立中,付宜利. UG/OPEN API 编程基础[M]. 北京:清华大学出版社,2002.
- [2] David J Kruglinski. Visual C + + 技术内幕[M]. 北京:清华大学出版社,2001.

## Research of Data Access in the Second Development Based on UG

HUANG Shu - rong, QIN Ke, HAN Dong - bing

(CAD Center of Yancheng Institute of Technology, Jiangsu Yancheng 224003, China)

**Abstract:** In the UG application systems, it is important to access various external data sources, such as text file, Excel sheet, and Database table. Approaches to accessing external data, instructions for how to program, key sample source codes, and their applicable domain and characteristics, were introduced in this paper, which provide reference to the extension/customizing development based on UG.

**Keywords:** the second development based on UG; data access; UG/Open API; MFC