ADO编程总结（一）

1. 在stdafx.h中添加代码：

#import "C:\Program Files\Common Files\System\ado\msado15.dll" no\_namespace rename("EOF","rsEOF")

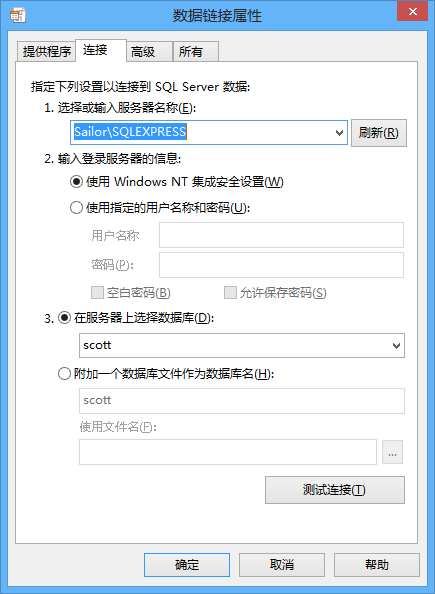
1. 在APP.cpp的InitInstance中加入：

::CoInitialize(NULL);

在ExitInstance()中加入：

::CoUninitialize();

1. 工程目录下用记事本新建一文件，另存为扩展名为.udl的文件,如这里保存为mydata.udl，双击出现数据链接属性对话框，参考下图进行设置，确定。



1. 定义以下变量：

\_ConnectionPtr m\_pConnection;

\_RecordsetPtr m\_pRecordset;

1. 初始化连接函数

BOOL CADOTestDlg::InitConnection()

{

HRESULT hr;

try

{

①hr=m\_pConnection.CreateInstance(\_uuidof(Connection));

if (SUCCEEDED(hr))

{

②m\_pConnection->ConnectionString="File Name=mydata.udl";

m\_pConnection->ConnectionTimeout=20;

hr=m\_pConnection->Open("","","",adConnectUnspecified);

if (FAILED(hr))

{

AfxMessageBox(\_T("打开连接失败"));

return FALSE;

}

}

else

{

AfxMessageBox(\_T("实例化连接失败"));

return FALSE;

}

return TRUE;

}

catch(\_com\_error e)

{

\_bstr\_t bstrSource(e.Source());

\_bstr\_t bstrDescription(e.Description());

AfxMessageBox(bstrSource+bstrDescription);

return FALSE;

}

}

对以上程序主要结果是一个try catch语句，其中

①hr=m\_pConnection.CreateInstance(\_uuidof(Connection));

实例化连接对象，固定格式

②m\_pConnection->ConnectionString="File Name=mydata.udl";

设置连接串属性为UDL文件

m\_pConnection->ConnectionTimeout=20;

连接等待时间为20秒

hr=m\_pConnection->Open("","","",adConnectUnspecified);

使用Connection对象打开连接，其中adConnectUnspecified表示同步模式，若为adAsyncConnect表示异步模式。

该函数是比较固定，可以在以后的应用中复制经少量修改即可应用。

1. 查询数据库表中的信息并输出：

void CADOTestDlg::OnBnClickedBtnShowAll()

{

HRESULT hr;

①\_RecordsetPtr pShowAllRecordset;

hr=pShowAllRecordset.CreateInstance(\_uuidof(Recordset));

if (FAILED(hr))

{

AfxMessageBox(\_T("Create Instance of Recordset failed!"));

return;

}

CString strSql;

\_variant\_t var;

CString strValue;

int curItem=0;

strSql=\_T("select \* from emp");

try

{

②hr=pShowAllRecordset->Open(

\_variant\_t(strSql),

m\_pConnection.GetInterfacePtr(),

adOpenDynamic,

adLockOptimistic,

adCmdText

);

if (SUCCEEDED(hr))

{

③while (!pShowAllRecordset->rsEOF)

{

curItem=m\_DataList.GetItemCount();

④var=pShowAllRecordset->GetCollect(\_T("EMPNO"));

if (var.vt!=VT\_NULL)

strValue=(LPCSTR)\_bstr\_t(var);

m\_DataList.InsertItem(curItem,strValue);

var=pShowAllRecordset->GetCollect(\_T("ename"));

if (var.vt!=VT\_NULL)

strValue=(LPCSTR)\_bstr\_t(var);

m\_DataList.SetItemText(curItem,1,strValue);

var=pShowAllRecordset->GetCollect(\_T("hiredate"));

if (var.vt!=VT\_NULL)

strValue=(LPCSTR)\_bstr\_t(var);

m\_DataList.SetItemText(curItem,2,strValue);

var=pShowAllRecordset->GetCollect(\_T("sal"));

if (var.vt!=VT\_NULL)

strValue=(LPCSTR)\_bstr\_t(var);

m\_DataList.SetItemText(curItem,3,strValue);

var=pShowAllRecordset->GetCollect(\_T("comm"));

if (var.vt!=VT\_NULL)

strValue=(LPCSTR)\_bstr\_t(var);

m\_DataList.SetItemText(curItem,4,strValue);

⑤pShowAllRecordset->MoveNext();

}

}

}

catch (\_com\_error\* e)

{

AfxMessageBox(e->ErrorMessage());

return;

}

⑥pShowAllRecordset->Close();

pShowAllRecordset=NULL;

}

该函数也比较固定，通过设置命令文本字符串，对数据库进行操作，主要有以下几步

①\_RecordsetPtr pShowAllRecordset;

hr=pShowAllRecordset.CreateInstance(\_uuidof(Recordset));

定义Recordset对象，并实例化，格式固定。

②hr=pShowAllRecordset->Open(

\_variant\_t(strSql),

m\_pConnection.GetInterfacePtr(),

adOpenDynamic,

adLockOptimistic,

adCmdText

);

尝试打开记录，其中Open函数参数意义如下：

第1个参数：记录源，可以是下列内容之一：Command对象变量，SQL语句，存储过程，表名或完整的路径名，这里是命令文本，一般也是用命令文本。

第2个参数：指定在哪个连接中打开记录集。m\_pConnection.GetInterfacePtr()：Returns the encapsulated interface pointer，返回封闭接口指针，不懂。

第3、4个参数：详细说明见《Visual C++6.0 开发指南》714，715页，不懂。

第5个参数：表示第1个参数的各类，这里是adCmdText，表示命令文本，其它取值见《Visual C++6.0 开发指南》715页。

③while (!pShowAllRecordset->rsEOF)循环。

当记录未到末尾时，执行循环，这里有可能会报错：

: “rsEOF”: 不是“\_Recordset”的成员

请确认在stdafx.h中添加了以下代码：rename("EOF","rsEOF")

var=pShowAllRecordset->GetCollect(\_T("EMPNO"));

if (var.vt!=VT\_NULL)

strValue=(LPCSTR)\_bstr\_t(var);

m\_DataList.InsertItem(curItem,strValue);

④var=pShowAllRecordset->GetCollect(\_T("EMPNO"));

if (var.vt!=VT\_NULL)

strValue=(LPCSTR)\_bstr\_t(var);

m\_DataList.InsertItem(curItem,strValue);

比较固定的结构，从记录中读取列名为"EMPNO"的数据，保存到var结构中，再转换到CString对象。

数据类型的转换：

COM中的VARIANT、BSTR两种数据类型与C++中CStrng类型的转换分别通过两个类来实现：

\_variant\_t 和\_bstr\_t。

⑤pShowAllRecordset->MoveNext();

记录指针向前移动。

⑥pShowAllRecordset->Close();

pShowAllRecordset=NULL;

关闭记录对象。