

```
        pRs->MoveNext();
    }
}
else
{
    srand((unsigned) time(NULL));
    j = rand();
    j = (j * 20) % (k - 27);
    while(j > 0)
    {
        pRs->MoveNext();
        j--;
    }
    while(!pRs->ADOEOF)
    {
        strcpy(m_strSql, "Insert Testprocdetail(SID, QNO,
        QType, Q) values(");
        strcat(m_strSql, m_strSID);
        strcat(m_strSql, ", ");
        _itoa(i, m_strBuf, 10);
        strcat(m_strSql, m_strBuf);
        strcat(m_strSql, ", ");
        strcat(m_strSql, "1");
        strcat(m_strSql, ", ");
        pRs->Fields->GetItem("QID")->
        get_Value(&m_varData);
        _itoa(m_varData.iVal, m_strBuf, 10);
        strcat(m_strSql, m_strBuf);
        strcat(m_strSql, ")");
        pConn->Execute(m_strSql, &m_varNum, -1);
        i++;
        if (i > 54)
            break;
        pRs->MoveNext();
    }
}
```

```

    }
}
pRs->Close();
/*产生多选题*/
pRs = pConn->Execute("SELECT QID FROM MULTIPLY", &m_varNum, -1);
k = 0;
while (!pRs->ADOEOF)
{
    k = k + 1;
    pRs->MoveNext();
}
pRs->MoveFirst();
if (k < 28)
{
    while (!pRs->ADOEOF)
    {
        strcpy(m_strSql, "Insert Testprocdetail(SID, QNO,
        QType, Q) values(");
        strcat(m_strSql, m_strSID);
        strcat(m_strSql, ", ");
        _itoa(i, m_strBuf, 10);
        strcat(m_strSql, m_strBuf);
        strcat(m_strSql, ", ");
        strcat(m_strSql, "2");
        strcat(m_strSql, ", ");
        pRs->Fields->GetItem("QID")->
        get_Value(&m_varData);
        _itoa(m_varData.iVal, m_strBuf, 10);
        strcat(m_strSql, m_strBuf);
        strcat(m_strSql, ")");
        pConn->Execute(m_strSql, &m_varNum, -1);
        i++;
        pRs->MoveNext();
    }
}

```

```
}  
}  
else  
{  
    srand((unsigned) time(NULL));  
    j = rand();  
    j = (j * 20) % (k - 27);  
    while(j > 0)  
    {  
        pRs->MoveNext();  
        j--;  
    }  
    while(!pRs > ADOEOF)  
    {  
        strcpy(m_strSql, "Insert into Testprocdetail(Sid,  
QNO, QType, Q) values(");  
        strcat(m_strSql, m_strSID);  
        strcat(m_strSql, ", ");  
        _itoa(i, m_strBuf, 10);  
        strcat(m_strSql, m_strBuf);  
        strcat(m_strSql, ", ");  
        strcat(m_strSql, "2");  
        strcat(m_strSql, ", ");  
        pRs->Fields->GetItem("QID")->  
get_Value(&m_varData);  
        _itoa(m_varData.iVal, m_strBuf, 10);  
        strcat(m_strSql, m_strBuf);  
        strcat(m_strSql, ")");  
        pConn->Execute(m_strSql, &m_varNum, -1);  
        i++;  
        if (i > 80)  
            break;  
        pRs->MoveNext();  
    }  
}
```

```
    }  
    pRs->Close();  
    pConn->Close();  
    *x_Success=::SysAllocString(L"Operate Succeed");  
}  
catch(...)  
{  
    if (pConn != NULL)  
    {  
        pConn->Close();  
    }  
    *x_Success = ::SysAllocString(L"Error Occured");  
    return S_OK;  
}  
return S_OK;  
}
```

IcaculateScore 接口的实现。在这个接口中只有一个方法 **CaculateScore()**。**CaculateScore** 是根据学生的做题答案与标准答案对照从而计算学生的分数。**CaculateScore** 中调用了组件 **QuestionInfo** 中的 **GetQuestion** 方法。为了实现这个调用，进行了以下几个步骤：

(1) 注册组件 **QuestionInfo**。注册组件的方式有两种，一种是在命令行中执行 **regsvr32 QuestionInfo.dll**；另一种使用 Win2000 自带的组件服务。组件服务的使用方法在前面介绍 COM+ 时已介绍了，在此不再重复。

(2) 拷贝文件。将 **QuestionInfo** 组件生成的 **QuestionInfo.tlb** 文件拷贝到 **Exam** 的工作目录下。

(3) 引入文件。在 **Stdafx.h** 文件中插入如下语句：

```
#import "QuestionInfo.tlb" no_namespace
```

(4) 在这着重说明一下 **IcaculateScore** 接口的生成，它和前面的接口生成有些不同。在 **ATL OBJECT Wizard** 中选择 **MSTransaction Server Component**，如图 11-7 所示，然后在 **MTS** 属性页中选择 **Support IobjectControl**，如图 11-8 所示。这主要是为了保证组件支持事务特性。

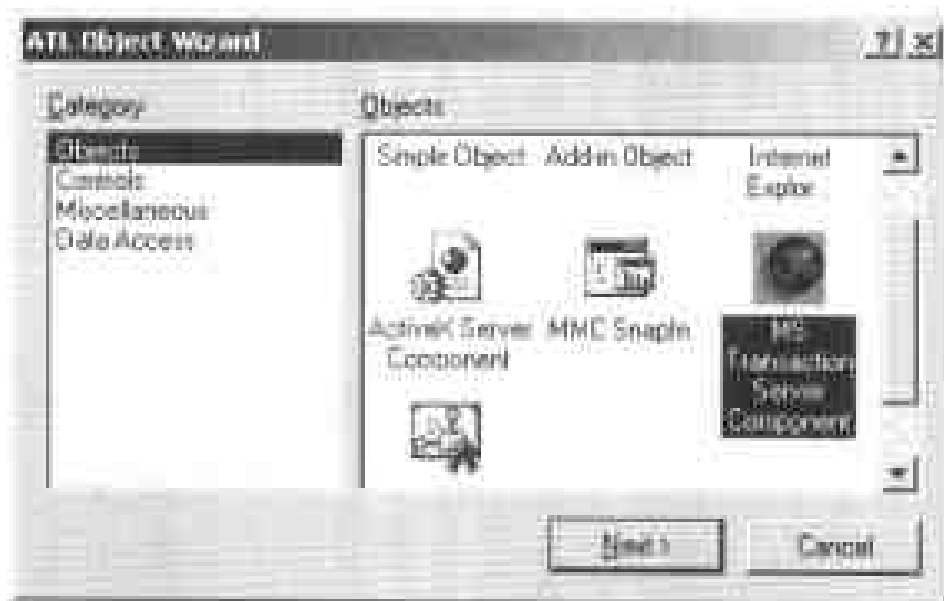


图 11-7 ATL Object Wizard 对话框



图 11-8 “ALT Object Wizard 属性”对话框

(5) 实现 CaculateScore 方法。首先增加成员变量 m_pQuestion:

```
IQuestionPtr m_pQuestion;
```

然后在 CaculateScore () 函数中增加如下实现代码 (程序 11-3-8):

程序 11-3-8:

```
STDMETHODIMP CCaculateScore::CaculateScore(int x_nSID,  
BSTR *x_Success)  
{  
    // TODO: Add your implementation code here
```

```

try
{
    if
    (FAILED(pConn.CreateInstance(__uuidof(Connection))))
    /*建立 ADO 的 Connection 对象*/
    {
        *x_Success = ::SysAllocString(L"Couldn't create
        connection component!");
        return S_OK;
    }

    if (FAILED(pConn->Open("Provider=MSDASQL.1;
    Password=;Persist Security Info=True;User ID=SA;
    Data Source=EXAM;Initial Catalog=EXAM", "", "", -1)))
    /*建立连接*/
    {
        *x_Success = ::SysAllocString(L"Couldn't open
        connection!");
        return S_OK;
    }
}
/* 查询学号为 x_nSID 考试过程记录集 */
char m_strBuf[10];
char m_strSql[256];
CComBSTR m_bstrSql;
CComVariant m_varNum;
CComVariant m_varData;
strcpy(m_strSql, "select * from testprocdetail where SID=");
_itoa(x_nSID, m_strBuf, 10);
strcat(m_strSql, m_strBuf);
pRs = pConn->Execute(m_strSql, &m_varNum, -1);
if (pRs->ADOEOF)
{
    *x_Success = ::SysAllocString(L"No such records for the

```

```

        specified student!");
        return S_OK;
    }
    CComVariant m_varData1;
    pRs->MoveFirst();
    int i=1;
    m_nScore=0;
/*开始计算分数*/
    while(!pRs->ADOEof)
    {
        char m_strFieldName[20];
        int m_iQtype, m_iQID; /*存放题目类型和题目 ID*/
        short m_iAnswer;
        strcpy(m_strFieldName, "QTYPE");
        pRs->Fields->GetItem(m_strFieldName)->
            get_Value(&m_varData);
        if(m_varData.vt != VT_NULL)
        {
            m_iQtype=m_varData.iVal;
            strcpy(m_strFieldName, "Q");
            pRs->Fields->GetItem(m_strFieldName)->
                get_Value(&m_varData);
            if (m_varData.vt != NULL)
            {
                m_iQID=m_varData.iVal;
                if(!::CoInitialize((LPVOID)0)) //代码 1
                {

*x_Success=::SysAllocString(L"Initianlize failed");
                    return S_OK;
                }
            if (FAILED(m_pQuestion.CreateInstance(__uuidof(Question)))) //代码 2
            {

```

```

*x_Success=::SysAllocString(L"Failed in creating Instance");
        return S_OK;
    }
    m_pQuestion->GetQuestion(m_iQtype, m_iQID); //代码3
    m_pQuestion->get_Answer(&m_iAnswer); //代码4
        strcpy(m_strFieldName, "A");
        pRs->Fields->GetItem(m_strFieldName)->
        get_Value(&m_varData);
        if(m_varData.vt != VT_NULL)
        {
            if(m_varData.iVal == m_iAnswer )
            {
                m_nScore += 2; /*当答案正确时加两分*/
            }
        }
    }
    else
    {
        *x_Success=::SysAllocString(L"one question no
        answer.");
    }
}

pRs->MoveNext();

}

/*将计算分数的结果插入到 TestResult 中*/
    strcpy(m_strSql, "INSERT INTO TESTRESULT(SID, TESTDATE,
SCORE) VALUES(" );
    _itoa(x_nSID, m_strBuf, 10);
    strcat(m_strSql, m_strBuf);
    strcat(m_strSql, ", GETDATE(), ");
    _itoa(m_nScore, m_strBuf, 10);
    strcat(m_strSql, m_strBuf);
    strcat(m_strSql, ")");
    pConn->Execute(m_strSql, &m_varNum, -1);

```

```

strcpy(m_strSql, "UPDATE TESTPRCC SET FINISHFLAG='Y' WHERE
SID=");
_itoa(x_nSID, m_strBuf, 10);
strcat(m_strSql, m_strBuf);
pConn->Execute(m_strSql, &m_varNum, -1);
pConn->Close();
m_spObjectContext->SetComplete();//代码 5
*x_Success = ::SysAllocString(L"Successfully!");
CoUninitialize();
}

catch(...)
{
    *x_Success = ::SysAllocString(L"Error occurs!");
    if (pConn != NULL)
    {
        pConn->Close();
    }
    m_spObjectContext->SetAbort();//代码 6
    ::CoUninitialize();
    return S_OK;
}

::CoUninitialize();//代码 7
return S_OK;
}

```

代码 1 是初始化组件环境。代码 2 是实例化组件 QuestionInfo，建立“Question”对象。代码 3 是调用接口 IQuestion 的 GetQuestion 方法。代码 4 是调用接口 IQuestion 的 GetAnswer 方法，获得题目的正确答案。这一步需要注意的是 GetQuestion 操作一般将记录中的所有值赋给其相应的成员变量，而 GetAnswer 是将成员变量 Answer 中的值返回给客户应用程序。因而，在执行 GetAnswer 方法时，一般先执行 GetQuestion 方法。代码 5 是表明组件对所属事务提交表示同意。代码 6 是表明组件对所属事务提交表示不同意。在前面介绍 COM+ 的部分中，对组件的事务环境已进行了介绍，在此就不重复了。代码 7 通过调用 CoUninitialize 退出 COM 库服务。

ISaveCurrentAnswer 接口的实现。ISaveCurrentAnswer 主要实现 SaveAnswer 方法。SaveAnswer 有五个输入参数：学生 ID、题目号、答案、剩余时间、读题标志（读下一题或读上一题）。根据学生 ID 定位 TestProc 表，用剩余时间更新 Timeleft 字段，根据读题标志更新 CurrentQuestion 字段。然后再根据学生 ID、题目号定位 TestProcDetail 表，将答案更新到记录中。实现代码如程序 11-3-9 所示：

程序 11-3-9:

```
STDMETHODIMP CSaveCurrentAnswer::SaveAnswer(int x_nSID, int
x_nQID, int x_nAnswer, int x_nTimeLeft, int x_nForward, BSTR
*x_Success)
{
// TODO: Add your implementation code here
try
{
    if
(FAILED(pConn.CreateInstance(__uuidof(Connection))))
/*初始化 Connection 对象*/
{
        *x_Success = ::SysAllocString(L"Couldn't create
connection component!");
        return S_OK;
    }
    if (FAILED(pConn->Open("Provider=MSDASQL.1;
Password=;Persist Security Info=True;User ID=SA;
Data Source=EXAM;Initial Catalog=EXAM", "", "", -1)))
/*建立数据库连接*/
{
        *x_Success = ::SysAllocString(L"Couldn't open
connection!");
        return S_OK;
    }
/*修改 CurrentQuestion、timelefe 字段 */
    char m_strSql[256];
    char m_strBuf[10];
```