CRegKey类的使用

注册表操作，使用MFC提供的CRegKey类，封装了相应的操作，但是需要在对话框源文件

中引用atlbase.h头文件  
主要方法  
  注册表是个树结构，由根键，子键和项组成，其中项是子键下的数据，包括项的名称，数据类型和数据  
1：Create:用于创建所标志的键，声明如下  
 LONG Create(  
   HKEY hKeyParent, //表示打开的父键句柄；  
   LPCTSTR lpszKeyName,表示键的名称；  
   LPTSTR lpszClass=REG\_NONE,表示键的类型  
   DWORD dwOptions=REG\_OPTION\_NON\_VOLATILE,表示键的打开方式  
   REGSAM samDesired=KEY\_ALL\_ACCESS,表示访问权限  
   LPSECURITY\_ATTRIBUTES lpSecAttr=NULL,表示句柄安全属性  
   LPDWORD lpdwDispotion=NULL//表示返回函数执行的动作)  
2：Open:用于打开注册表的键值  
LONG Open(HKEY hKeyParent,//父键句柄  
         LPCTSTR lpszKeyName，打开的键名称  
         REGSAM samDesired=KEY\_ALL\_ACCESS//访问权限)  
3：SetValue:用于设置指定键的项信息，有三种方法  
LONG SetValue(DWORD dwValue,LPCTSTR lpszValueName）  
LONG SetValue(LPCTSTR lpszValue,LPCTSTR lpszValueName)  
 其中 dwValue,//表示设置的整数值  
  lpszValueName//表示设置的项的名称  
  lpszValue//表示设置的字符串数据  
LONG SetValue(HKEY hKeyParent,LPCTSTR lpszKeyName,LPCTSTR

lpszValue,LPCTSTR lpszValueName=NULL)  
  其中hKeyParent表示父键句柄  
    lpszKeyName表示键的名称；  
     lpszValue表示设置的项的数据  
     lpszValueName表示项名称

4：QueryValue:获取指定键的数据  
LONG QueryValue(LPCTSTR lpszValueName,DWORD \*pdwType,void \*pData,ULONG

                \*pnBytes);  
ATL\_DEPRECATEDLONG QueryValue(DWORD &dwValue,LPCTSTR lpszValueName);  
ATL\_DEPRECATEDLONG QueryValue(LPSTR szValue,LPCTSTR lpszValueName,DWORD

\*pdwCount);  
   lpszValueName表示查询的项的名称  
   pdwType整型指针，用于返回项的数据类型  
    pData表示数据缓冲区，用于存储函数返回的数据  
    pnBytes表示数据缓冲区pData的大小  
    dwValue用于存储函数返回的整型数据  
     szValue表示缓冲区，  
    pdwCount 表示字符串数据的大小  
5：Close关闭

void CRegDlg::OnRead()   
{  
// UpdateData(false);UpdateData(true);  
 UpdateData(true);  
  CRegKey reg;  
  reg.Open(HKEY\_CURRENT\_USER,m\_key);  
  DWORD size=1000;  
  CString value;  
  reg.QueryValue(value.GetBuffer(0),m\_value1,&size);  
  MessageBox(value,"提示");

}

void CRegDlg::OnWrite()   
{  
 //UpdateData(false);UpdateData(true);  
 UpdateData(true);  
  CRegKey reg;  
 reg.Create(HKEY\_CURRENT\_USER,m\_key);  
 reg.SetValue(m\_value1,m\_value2);  
 reg.Close();   
}