

用VisualC 实现注册表简单操作计算机等级考试 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/646/2021\\_2022\\_\\_E7\\_94\\_A8VisualC\\_c97\\_646148.htm](https://www.100test.com/kao_ti2020/646/2021_2022__E7_94_A8VisualC_c97_646148.htm)

Windows的注册表中包含了Windows的系统配置、PC机的硬件配置、Win32应用程序和用户的其他设置信息。注册表和INI文件不同，它是多层次的树状数据结构，具有六个分支(根键)，每个分支又由许多的键和键值组成，而每个键则代表一个特定的配置项目。在注册表中键可以包括子键和值。我们可以对他做个比喻，键就是目录，而子键和值可以看成文件，这样比喻可能不科学但是很实际。在Windows 98下注册表包括在两个文件内，他们是user.dat和system.dat两个文件。System.dat包括标准的系统信息，他们被保存在 HKEY\_LOCAL\_MACHINE的根键内。User.dat文件包括用户指定的信息，如用户策略，桌面设定等等。为了平时查看方便，Windows为我们准备了一个小程序regedit.exe。执行他就可以看到注册表的树状结构。在实际编程工作中，我们经常会遇到如何在Visual C 中对Windows注册表整个树状结构信息进行访问和修改的问题，如查询和修改注册表中用户姓名和公司名称的有关信息等。为此，本实例将相关的注册表操作封装到函数中，并通过一个简单的例程给出了上述函数的调用方法，该例程可以在注册表中创建、删除、显示一个键值，同时还可以清空最近阅读过的文档历史记录。

一、实现方法 对注册表的编程要用到句柄，我们需要通过一个句柄访问注册表键值，当打开或创建一个键值的时候，会返回一个该键的句柄，并且调用和分析键和创建键值，在分析和创建的同时需要传递句柄到函数。WINDOWS提供预定义

的用语---根一级键的保留句柄，如 HKEY\_CLASS\_ROOT, HKEY\_CURRENT\_USER, HKEY\_LOCAL\_MACHINE, HKEY\_USER 等，这些都是与注册表的根键相对应并且同名的句柄。当访问一个根键的时候，传递这些通用句柄。这就不用打开根键啦，因为他们总是在打开状态下，可使用默认键的句柄访问。Win32 API 提供了大约 25 个有关注册表的函数，他提供了对注册表的读取，写入，删除，以及打开注册表的所有函数，并且可以实现对注册表备份，连接和对远端注册表进行查看等功能。但是在编程的时候首先需要考虑你是在什么操作系统编辑此类程序，虽然微软的操作系统，如 NT 和 Windows 98 都是 32 位操作系统，但是有些 API 函数中并不支持 98，这点是要注意的。API 经历和发展了很多年，有些函数已经重复，比如 RegSetValue() 及 RegSetValueEx() 都是用来设置注册表键值的，两者的区别在于前者是设置注册表键的默认值，仅支持作为数据类型的字符串，而后者不仅继承了前者的所有功能而且还能对多值或类型进行操作。一般 API 对比较新的函数都会在后缀追加 "Ex" 的同样名称函数，建议在编程中均应尽可能的使用高级函数。下面介绍一些比较常用的操作注册表的 API 函数：

- 1、RegCloseKey() 原型：  
RegCloseKey(HKEY hKey) 解释：关闭指定的主册表键，释放句柄。当对一个或多个键或值操作完成以后，需要关闭其键来进行保存操作结果，关闭一个键后，句柄变为非法，此时应释放句柄。
- 2、RegCreateKeyEx() 原型：LONG  
RegCreateKeyEx( HKEY hKey, LPCTSTR lpSubKey, DWORD Reserved, LPTSTR lpClass, DWORD dwOptions, REGSAM samDesired, LPSECURITY\_ATTRIBUTES

IpSecurityAttributes, PHKEY phkResult, LPDWORD

lpdwDisposition)。解释：打开指定的键或子键。如果要打开的键不存在的话，本函数会试图建立它。提供该函数是为了向后兼容。所有的WIN32应用程序应使用函数RegCreateKeyEx( )。各参数及返回值的含义如下：各参数及返回值的含义如下：hKey为主键值，可以取下面的一些数值

：HKEY\_CLASSES\_ROOT、HKEY\_CURRENT\_CONFIG、HKEY\_CURRENT\_USER、HKEY\_LOCAL\_MACHINE、HKEY\_USER、HKEY\_PERFORMANCE\_DATA(WINNT操作系统)、HKEY\_DYN\_DATA(WIN9X操作系统)；参数lpSubKey为一个指向以零结尾的字符串的指针，其中包含将要创建或打开的子键的名称。子键不可以用反斜线(\)开始。该参数可以为NULL；参数Reserved为保留值，必须设置为0；参数lpClass为一个指向包含键类型的字符串。如果该键已经存在，则忽略该参数；参数dwOptions为新创建的键设置一定的属性。可以取下面的一些数值：

REG\_OPTION\_NON\_VOLATILE，表示新创建的键为一个非短暂性的键（数据信息保存在文件中，当系统重新启动时，数据信息恢复）；REG\_OPTION\_VOLATILE，表示新创建的键为一个短暂性的键（数据信息保存在内存中）

，Windows95忽略该数值

；REG\_OPTION\_BACKUP\_RESTORE 仅在WINNT中支持，可以提供优先级支持；参数samDesired用来设置对键访问的权限，可以取下面的一些数值：KEY\_CREATE\_LINK，表示准许生成符号键；KEY\_CREATE\_SUB\_KEY 表示准许生成子键；KEY\_ENUMERATE\_SUB\_KEYS 表示准许生成枚举子键

；KEY\_EXECUTE 表示准许进行读操作；KEY\_NOTIFY表示准许更换通告；KEY\_QUERY\_VALUE 表示准许查询子键；KEY\_ALL\_ACCESS 提供完全访问，是上面数值的组合；KEY\_READ 是下面数值的组合：KEY\_QUERY\_VALUE、KEY\_ENUMERATE\_SUB\_KEYS、KEY\_NOTIFY；KEY\_SET\_VALUE 表示准许设置子键的数值；KEY\_WRITE 是下面数值的组合：KEY\_SET\_VALUE、KEY\_CREATE\_SUB\_KEY；参数lpSecurityAttributes为一个指向SECURITY\_ATTRIBUTES结构的指针，确定返回的句柄是否被子处理过程继承。如果该参数为NULL，则句柄不可以被继承。在WINNT中，该参数可以为新创建的键增加安全的描述；参数phkResult为一个指向新创建或打开的键的句柄的指针；参数lpdwDisposition指明键是被创建还是被打开的，可以是下面的一些数值：REG\_CREATE\_NEW\_KEY 表示键先前不存在，现在被创建；REG\_OPENED\_EXISTING\_KEY 表示键先前已存在，现在被打开。如果该函数调用成功，则返回ERROR\_SUCCESS。否则，返回值为文件WINERROR.h中定义的一个非零的错误代码，可以通过设置FORMAT\_MESSAGE\_FROM\_SYSTEM标识调用FormatMessage（）函数来获取一个对错误的总体描述。

### 3、RegOpenKeyEx（）

原型：LONG RegOpenKeyEx(HKEY hKey, LPCTSTR lpSubKey, DWORD ulOptions, REGSAM samDesired, PHKEY phkResult)。解释：打开一个指定的键，并返回打开键的句柄。各参数及返回值的含义如下：参数hKey的含义同RegCreateKeyEx函数中的hKey参数；参数lpSubKey为一个指向以零结尾的字符串的指针，其中包含子键的名称，可以利

用反斜线 (\) 分隔不同的子键名。如果字符串为空，则根据hKey参数创建一个新的句柄。在这种情况下，并不关闭先前打开的句柄；参数ulOption保留，通常必须设置为0；参数samDesired的含义同RegCreateKeyEx函数中的samDesired参数；参数phkResult为一个指针，用来指向打开的键的句柄。可以通过RegCloseKey函数关闭这个句柄；函数的返回值同RegCreateKeyEx函数的返回值。

#### 4、 查询某一个键值

：RegQueryValueEx ( ) 原型：LONG

RegQueryValueEx(HKEY hKey, LPCTSTR lpValueName, LPDWORD pReserved, LPDWORD lpType, LPBYTE lpData, LPDWORD lpcbData)。解释：根据要查询的键的句柄，要返回的查询的数据。各个参数及返回值的含义如下：参数hKey为当前的一个打开的键的句柄，具体数值同RegCreateKeyEx函数的hKey参数；参数lpVauleName为一个指向非空的包含查询值的名称的字符串指针；参数lpReserved保留，必须为NULL；参数lpType为一个指向数据类型的指针，数据类型为下列类型之一：REG\_BINARY 二进制数据、REG\_DWORD 32位整数、REG\_DWORD\_LITTLE\_ENDIAN little - endian格式的数据，例如0X12345678以 ( 0X78 0X56 0X34 0X12 ) 方式保存、REG\_DWORD\_BIG\_ENDIAN big - endian格式的数据，例如0X12345678以 ( 0X12 0X34 0X56 0X78 ) 方式保存、REG\_EXPAND\_SZ 一个包含未扩展环境变量的字符串、REG\_LINK 一个Unicode类型的链接、REG\_MULIT\_SZ 以两个零结尾的字符串、REG\_NONE 无类型数值、REG\_RESOURCE\_LIST 设备驱动资源列表、REG\_SZ 一个以零结尾的字符串根据函数使用的字符集类型的不同而设置

为Unicode或ANSI类型的字符串；参数lpData为一个指向保存返回值的变量的指针。如果不需要返回值，该参数可以为NULL；参数lpcbData为一个指向保存返回值长度的变量的指针。其中长度以字节为单位。如果数据类型为REG\_SZ、REG\_MULTI\_SZ或REG\_EXPAND\_SZ，那么长度也包括结尾的零字符，只有在参数lpData为NULL时，参数lpcbData才可以为NULL；返回值同RegCreateKeyEx函数的返回值；

5、RegSetValueEx（）原型：LONG RegSetValueEx(HKEY hKey, LPCTSTR lpValueName, LPDWORD lpReserved, DWORD dwType, const BYTE \*lpData, DWORD cbData)。解释：设置注册表中的一个键值。各个参数及返回值的含义如下：参数hKey的含义同RegCreateKeyEx函数中的hKey参数；参数lpValueName为一个指向包含值名的字符串指针；Reserved保留，通常必须设置为0；参数dwType确定了设置的值的类型同RegQueryValueKeyEx的lType参数；参数lpData为一个指向包含数据的缓冲区的指针；参数cbData以字节为单位，指定数据的长度；返回值同RegCreateKeyEx函数的返回值。

6、RegDeleteKey（）原型：LONG RegDeleteKey（HKEY hKey, LPCTSTR lpSubKEY）；解释：函数RegDeleteKey删除一个键及所有的子键。各个参数及返回值的含义如下：参数hKey的含义同RegCreateKeyEx函数中的hKey参数；参数lpSubKey的含义同RegCreateKeyEx函数中的lpSubKey参数。

二、编程步骤

- 1、启动Visual C 6.0，生成一个基于对话框的应用程序RegDemo，按照上文中的图一所示设计程序界面；
- 2、使用Class Wizard为对话框中的按钮添加鼠标单击响应处理函数；
- 3、添加代码，编译运行程序。

100Test 下载频道开通，各

类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)