**[C#操作注册表](http://www.cnblogs.com/txw1958/archive/2012/08/01/csharp-regidit.html)**

下面我们就来用.NET下托管语言C#注册表操作，主要内容包括：注册表项的创建，打开与删除、键值的创建(设置值、修改)，读取和删除、判断注册表项是否存在、判断键值是否存在。  
准备工作：  
1：要操作注册表，我们必须要引入必要的命名空间：

using Microsoft.Win32;

在这个命名空间里面包含了许多注册表相关的类，足够我们使用了~~  
2：命名空间里面提供了一个类：RegistryKey 利用它我们可以定位到注册表最开头的分支：  
ClassesRoot，CurrentUser，Users，LocalMachine，CurrentConfig  
如：RegistryKey key = Registry.LocalMachine;  
3：在操作的过程中涉及到子分支，要用\\进行深入，单个\会报错！  
4：最后要调用RegistryKey对象的Close()关闭对注册表的修改~~~  
5：以下我们的例子都是在LocalMachine分支下，请注意。

一：C#注册表项的创建，打开与删除  
1：创建  
创建注册表项主要用到RegistryKey 的CreateSubKey()方法。如：

RegistryKey key = Registry.LocalMachine;

RegistryKey software = key.CreateSubKey("software\\test");

//在HKEY\_LOCAL\_MACHINE\SOFTWARE下新建名为test的注册表项。如果已经存在则不影响！

2：打开  
打开注册表项主要用到RegistryKey 的OpenSubKey()方法。如：  
注意，如果该注册表项不存在，这调用这个方法会抛出异常

RegistryKey key = Registry.LocalMachine;

RegistryKey software = key.OpenSubKey("software\\test",true);

//注意该方法后面还可以有一个布尔型的参数，true表示可以写入。

3：删除  
删除注册表项主要用到RegistryKey 的DeleteSubKey()方法。如：

RegistryKey key = Registry.LocalMachine;

key.DeleteSubKey("software\\test",true); //该方法无返回值，直接调用即可

key.Close();

注意，如果该注册表项不存在，这调用这个方法会抛出异常

二：键值的创建(设置值、修改)，读取和删除  
1：创建(设置值、修改)  
对键值的创建修改等操作主要用到RegistryKey 的SetValue()方法

[复制代码](javascript:void(0);)

RegistryKey key = Registry.LocalMachine;

RegistryKey software = key.OpenSubKey("software\\test",true); //该项必须已存在

software.SetValue("test", "博客园");

//在HKEY\_LOCAL\_MACHINE\SOFTWARE\test下创建一个名为“test”，值为“博客园”的键值。如果该键值原本已经存在，则会修改替换原来的键值，如果不存在则是创建该键值。

// 注意：SetValue()还有第三个参数，主要是用于设置键值的类型，如：字符串，二进制，Dword等等~~默认是字符串。如：

// software.SetValue("test", "0", RegistryValueKind.DWord); //二进制信息

Key.Close();

[复制代码](javascript:void(0);)

2：读取

[复制代码](javascript:void(0);)

string info = "";

RegistryKey Key;

Key = Registry.LocalMachine;

myreg = Key.OpenSubKey("software\\test");

// myreg = Key.OpenSubKey("software\\test",true);

info = myreg.GetValue("test").ToString();

myreg.Close();

[复制代码](javascript:void(0);)

info结果为：博客园

3：删除

RegistryKey delKey = Registry.LocalMachine.OpenSubKey("Software\\test", true);

delKey.DeleteValue("test");

delKey.Close();

细心的读者可能发现了第二个例子中OpenSubKey()方法参数与其他例子的不同。  
如果你要修改键值，包括创建、设置、删除键值等都要在方法后面加个布尔参数，设置为true，表示可写可改；如果仅仅只是读取键值可以不加，此时可写关闭，你不能再往里写值(当然，你要加也可以true)！  
还有读者提到读写默认键值的问题，主要在设置、读取的方法中将键名置空则就是对默认键值的操作。  
如：  
software.SetValue("", "博客园"); // 在HKEY\_LOCAL\_MACHINE\SOFTWARE\test修改默认键值的值为“博客园”。读取类似！  
另外，默认的键值是不能删除的，所以不要用DeleteValue()方法去删除，会抛出异常的！

三：判断注册表项是否存在

[复制代码](javascript:void(0);)

private bool IsRegeditItemExist()

{

string [] subkeyNames;

RegistryKey hkml = Registry.LocalMachine;

RegistryKey software = hkml.OpenSubKey("SOFTWARE");

//RegistryKey software = hkml.OpenSubKey("SOFTWARE", true);

subkeyNames = software.GetSubKeyNames();

//取得该项下所有子项的名称的序列，并传递给预定的数组中

foreach (string keyName in subkeyNames)

//遍历整个数组

{

if (keyName == "test")

//判断子项的名称

{

hkml.Close();

return true ;

}

}

hkml.Close();

return false;

}

[复制代码](javascript:void(0);)

四：判断键值是否存在

[复制代码](javascript:void(0);)

private bool IsRegeditKeyExit()

{

string[] subkeyNames;

RegistryKey hkml = Registry.LocalMachine;

RegistryKey software = hkml.OpenSubKey("SOFTWARE\\test");

//RegistryKey software = hkml.OpenSubKey("SOFTWARE\\test", true);

subkeyNames = software.GetValueNames();

//取得该项下所有键值的名称的序列，并传递给预定的数组中

foreach (string keyName in subkeyNames)

{

if (keyName == "test") //判断键值的名称

{

hkml.Close();

return true;

}

}

hkml.Close();

return false;

}

[复制代码](javascript:void(0);)

C#注册表操作的相关知识就介绍到这里，希望大家喜欢。

其他参考： <http://hi.baidu.com/redflower1004/item/19ecf85fbee5a9474eff20e6>