

C# 系统应用之注册表使用详解

原创 Eastmount 最后发布于2014-05-11 21:35:31 阅读数 4324 ☆ 收藏

展开



Python+TensorFlow人工智能

该专栏为人工智能入门专栏，采用Python3和TensorFlow实现人工智能相关算法。前期介绍安装流程、基础语法...

Eastmount

¥9.90

去订阅

在平时做项目时,我们有时会遇到注册表的操作,例如前面我们需要获取IE浏览器地址栏的信息、获取“我的电脑”地址栏输入的文件夹信息、USB最近使用信息等.注册表项是注册表的基本组织单位,它包含子表项和值条目.简言之,注册表项相当于注册表里的文件夹.它们存储计算机安装程序的信息,如颜色设置、屏幕大小、历史记录等.

举个以前使用注册表最多的例子:当我们玩魔兽或dota时,总需要打开“运行”->输入“regedit”打开注册表,在HKEY_CURRENT_USER->Software->Blizzard Entertainment->Warcraft III->Video中修改reswidth(屏幕宽)和resheight(屏幕高).使其屏幕分辨率与游戏一致全屏显示.

下面就详细讲解注册表的使用.

一.注册表的结构

注册表(registry)是Windows系统中一个重要的数据库,它用于存储有关应用程序、用户和系统信息.注册表的结构就像一颗树.树的顶级节点(hive)不能添加、修改和删除.

如下图所示是Windows注册表的顶级节点:



在C#中对注册表进行操作,需要引用命名空间:using Microsoft.Win32.

RegistryKey类表示注册表中的顶级结点,此类是注册表的封装.

Registry类提供表示Windows注册表中的根项RegistryKey对象,并提供访问项/值的static方法.常用的Registry对象的顶级节点(蜂窝,hive)的属性如下表所示:

Registry属性	注册表顶级节点	说明
ClassessRoot	HKEY_CLASSES_ROOT	定义文档的类型\类以及与类型关联的信息以及COM组件的配置数据
CurrentConfig	HKEY_CURRENT_CONFIG	包含有关非用户特定的硬件的配置信息
CurrentUser	HKEY_CURRENT_USER	包含当前登录到Windows的用户的配置信息
LocalMachine	HKEY_LOCAL_MACHINE	包含与计算机相关的配置信息,不管用户是否登录
Users	HKEY_USERS	包含有关默认用户配置的信息

注册表中常用的数据类型有:

REG_SZ字符串数据的主要类型,用于存储固定长度的字符串或其他短文本值.我们在实际程序中通常只用这种数据类型,如果要保存布尔值时,将它表示成0或1.

REG_BINARY用于存储二进制数据.

REG_EXPAND_SZ可扩展的字符串值,可以保存在运行时才解析的系统变量.

REG_MULTI_SZ以数组的格式保存多个文本字符串,每个字符串"元素"都以null字符结束.

二.注册表常用方法

1.创建注册表键

首先先确定在哪个顶级结点下创建键,然后调用CreateSubKey()方法创建.

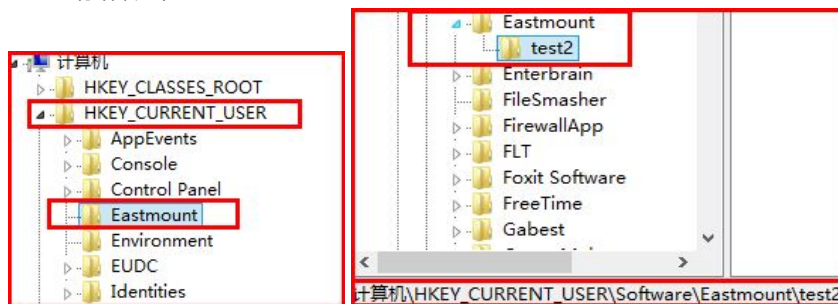
```
public RegistryKey CreateSubKey(  
    string subkey    //要创建或打开的子项的名称或路径  
)
```

例如在HKEY_CURRENT_USER下创建键Eastmount和HKEY_CURRENT_USER\Software\Eastmount\test2.其中我们最常用的是在\Software这个键下创建程序产品键,保存一些程序的配置在注册表中.如果Software中没有Eastmount键,则会先创建这个键及其子键.如果存在就不会重写.

```
// 创建键  
// 在HKEY_CURRENT_USER下创建Eastmount键  
RegistryKey test1 = Registry.CurrentUser.CreateSubKey("Eastmount");  
// 创建键结构 HKEY_CURRENT_USER\Software\Eastmount\test2  
RegistryKey test2 = Registry.CurrentUser.CreateSubKey(@"Software\Eastmount\test2");
```

注意:C#字符串中反斜杠是转义字符,前面加前缀@时,告诉字符串按字面值处理而不进行转义.@"Software\Eastmount\test2"="Software\\Eastmount\\test2".

运行结果如下:



2.打开子键

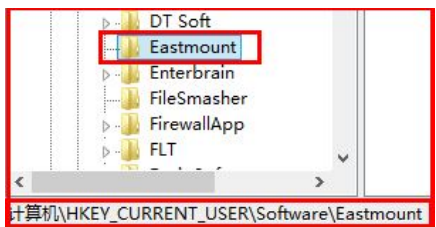
```
public RegistryKey OpenSubKey(  
    string name,        //要打开的子项名称或路径  
    bool writable       //如果需要项的写访问权限=true  
)
```

3.删除注册表的键

删除注册表的键有两个方法: DeleteSubKey()和DeleteSubKeyTree().DeleteSubKey()删除不包括子键的键及其所有值,DeleteSubKeyTree()不仅删除键及其值,还删除键下所有子键(慎用该方法).

```
// 函数原型  
public void DeleteSubKey(  
    string subkey    //要删除的子项名称  
)  
// 删除HKEY_CURRENT_USER下创建Eastmount键  
Registry.CurrentUser.DeleteSubKey("Eastmount");  
// 删除创建的子键test2  
Registry.CurrentUser.DeleteSubKey(@"Software\Eastmount\test2");
```

如果键不存在,会有异常错误提示,最好使用try..catch..异常处理.它仅仅删除Software\Eastmount中子项test2而保留Eastmount项.结果如下图所示:



注意:如果使用Registry.CurrentUser.DeleteSubKey(@"Software\Eastmount");它会提示错误"注册表具有子项,并且此方法不支持递归移除".如果使用Registry.CurrentUser.DeleteSubKeyTree(@"Software\Eastmount")则会删除Eastmount和子项test2.

4.获取键值

Registry.GetValue()检索与指定的注册表项中的指定名称关联的值.如果keyName指定的子项不存在,则返回null;否则返回与valueName关联的值;如果未找到valueName则返回defaultValue.

```
public static Object GetValue(  
    string keyName,          //以注册表顶级节点开头的键的完整路径  
    string valueName,        //名称/值的名称  
    Object defaultValue      //当valueName不存在时返回的值  
)
```

例如获取HKEY_CURRENT_USER\Environment中的path键值.

```
// 获取Environment中路径  
string strPath;  
strPath = (string)Registry.GetValue(@"HKEY_CURRENT_USER\Environment",  
    "path", "Return this default if path does not exist");  
MessageBox.Show(strPath);
```

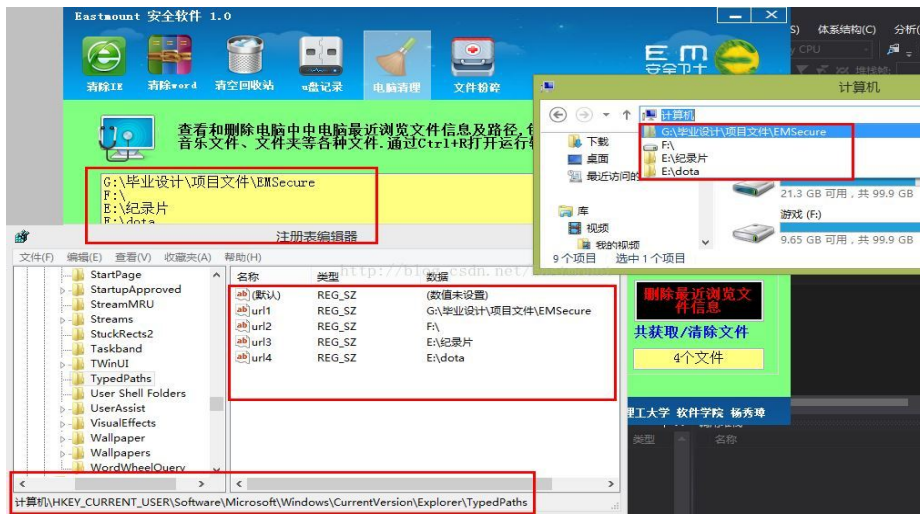


RegistryKey.GetValue方法可以检索与指定名称关联的值.函数原型public object GetValue(string name).其中参数name表示键的名称.如程序中我获取计算机地址栏输入的信息.

```
// 定义注册表顶级结点 命名空间Microsoft.Win32  
Microsoft.Win32.RegistryKey key = Microsoft.Win32.Registry.CurrentUser.OpenSubKey  
    ("Software\Microsoft\Windows\CurrentVersion\Explorer\TypedPaths", true);  
// 判断键是否存在  
if (key != null)  
{  
    // 检索包含此项关联的所有值名称 即url1 url2 url3  
    string[] names = key.GetValueNames();  
    foreach (string str in names)  
    {  
        // 获取url中相关联的值  
        listBox1.Items.Add(key.GetValue(str).ToString());  
        Num++;  
    }  
    // 显示获取文件总数  
    this.textBox1.Text = Num + "个文件";  
}
```

}

运行结果如下图所示,其实还可以获取很多东西,USB使用信息、浏览器使用信息、软件路径等等.



5.创建和设置键值

Registry.SetValue()方法设置指定的注册表项的指定名称/值对,如果指定的项不存在,则创建该项.

// 函数原型

```
public static void SetValue(  
    string keyName,        //以注册表顶级节点开头的键的完整路径  
    string valueName,      //名称/值对的名称  
    Object value           //要存储的值  
)  
// 设置键值Version=1.25  
Registry.SetValue(@"HKEY_CURRENT_USER\Software\YourSoftware", "Version", "1.25");
```

注意:每次调用SetValue方法时,它都打开一个注册表项,设置其值,然后关闭该项.如果需要大批量修改建议使用RegistryKey.SetValue方法.public object SetValue(string name,object value).

PS:举个简单的例子通过设置我的电脑-{20D04FE0-3AEA-1069-A2D8-08002B30309D}为1,即可隐藏桌面我的电脑图标.修改为0又可以显示,你知不知道把文件夹添加".{20D04FE0-3AEA-1069-A2D8-08002B30309D}"后就变成了"我的电脑",可以用来隐藏文件.

```
RegistryKey rgK = Registry.CurrentUser.CreateSubKey(@"Software\Microsoft\Windows\CurrentVersion\Explorer  
\HideDesktopIcons\NewStartPanel");  
rgK.SetValue("{20D04FE0-3AEA-1069-A2D8-08002B30309D}", 1);
```

最后希望文章对大家有所帮助,文章主要是自己这段时间对C#调用注册表信息的总结,一方面讲述了注册表相关的基础知识,另一方面通过代码结合图片格式讲述简单运用注册表.同时如果文章中有什么啰嗦或不足之处,还请海涵.

文章参考自己的项目、MSDN和[利用C#操作注册表](#)

(By:Eastmount 2014-5-11 夜9点半 原创CSDN<http://blog.csdn.net/eastmount/>)

👍 点赞 2 ☆ 收藏 📄 分享 ...



Eastmount 博客专家

发布了450 篇原创文章 · 获赞 6227 · 访问量 496万+

他的留言板

关注