



در هنگام نوشتن یک برنامه در مواقعی بایستی مقداری را در یک مکان از سیستم کاربر ذخیره کنیم .
انتخاب برنامه نویسان در این گونه موارد معمولاً از بین سه گزینه ی زیر می باشد:

1- DataBase

2- فایل های INI

3- رجیستری (تنها در برنامه نویسی برای سیستم های ویندوزی کاربرد دارد)

استفاده از : DataBase به منظور نگهداری تعداد زیادی از مقادیر ، بهترین گزینه است . همچنین در صورتی که پای امنیت اطلاعات ذخیره شده در میان است ، در انجام این روش شک نکنید . در این روش میتوانید از قابلیت Password Protection که در تمامی بانک های اطلاعاتی وجود دارد ، استفاده کنید.

استفاده از فایل های :INI در بعضی موارد هم میتوان از این روش استفاده کرد . البته این روش همراه با پیچیدگی و امنیت پایین میباشد.

استفاده از رجیستری : روشی که قصد توضیح آن را داریم ، همین روش است . این روش دارای امنیت نسبتا پایینی ست . پس امنیت اطلاعات خود را با این روش به خطر نیاندازید . البته میتوانید اطلاعات خود را به صورت Encrypt شده در رجیستری قرار دهید (مثلا مواقعی که قصد ذخیره یک رمز عبور ، در یک برنامه ساده را دارید) . یکی از مزایای این روش سادگی و راحتی آن است.

اگر موافق باشید ، به سراغ مراحل عملی کار برویم:

یک پروژه ی جدید سی شارپ باز کنید و اسم آن را **Registry in CS** بگذارید.

حالا وارد قسمت کد نویسی شده و Name Space زیر را در قسمت General تعریف کنید:

```
using Microsoft.Win32;
```

حالا میتوانید خیلی راحت در هر جایی از برنامه‌تان از کلاس Registry استفاده کنید.

در قسمت زیر اجزای این کلاس را مورد بررسی قرار خواهیم داد:

1- **Registry.ClassesRoot**

از این قسمت میتوانید برای انجام عملیات مختلف بر روی شاخه **HKEY_CLASSES_ROOT** استفاده کنید.

چون متدها به صورت Alphabetic مرتب شده اند ، بنده هم آنها را به ترتیب توضیح میدهم . البته در یک مورد استثنا وجود دارد . آن هم متد **OpenSubKey** هست که باید در ابتدا توضیح بدهم ، چون تعداد زیادی از متدها به این متد وابستگی دارند.

1-1 : **OpenSubKey**

بوسیله این متد میتوانید یک زیر کلید را باز کنید . برای مثال به کد زیر دقت کنید:

```
RegistryKey reg;
```

```
reg = Registry.ClassesRoot.OpenSubKey("GC",true);
```

در این کد نکات زیادی وجود دارد .

خط اول : یک متغیر از نوع RegistryKey تعریف کردیم.

خط دوم : مقدار متغیر reg را که در بالا تعریف کرده بودیم ، برابر کلید بازشده (**GC**) قرار دادیم تا بتوانیم عملیات مورد نظر خود را روی کلید باز شده انجام دهیم.

البته اگر خواستید زیر کلیدهای بیشتری را باز کنید ، باید از کد زیر استفاده کنید:

```
RegistryKey reg;
```

```
reg = Registry.ClassesRoot.OpenSubKey("GCCLsid",true);
```

در همان خط دوم یک کلمه **true** می بینید . شاید پرسید این کلمه چه معنایی دارد؟! این کلمه به برنامه میگوید که کاربر دسترسی به این کلید را تایید میکند (در بعضی مواقع دسترسی باید تایید شود.)

1- 2 : **Close**

این متد کلید قبلی را که توسط متد `OpenSubKey` باز شده بود ، میبندد . بهتر است بعد از هر بار که متد `OpenSubKey` را به کار میگیرید ، در پایان کارتان متد `Close` را اجرا کنید.

1- 3 : **CreateSubKey**

وظیفه این متد ساخت یک زیر کلید است . کد زیر را ببینید:

```
RegistryKey reg;
```

```
reg = Registry.ClassesRoot.OpenSubKey("GC",true);
```

```
reg.CreateSubKey("NewSubKey");
```

1- 4 : **DeleteSubKey**

اگر خواستید یک زیر کلید را حذف کنید ، میتوانید از این متد استفاده کنید . کد زیر همان کلید قبلی را که ساختیم پاک میکند:

```
RegistryKey reg;
```

```
reg = Registry.ClassesRoot.OpenSubKey("GC",true);
```

```
reg.DeleteSubKey("NewSubKey");
```

1- 5 : **DeleteSubKeyTree**

این متد یک کلید با تمام زیر کلیدهایش و مقادیر زیر مجموعه اش را پاک میکند:

```
RegistryKey reg;
```

```
reg = Registry.ClassesRoot.OpenSubKey("GC",true);
```

```
reg.DeleteSubKeyTree("NewSubKey");
```

1- 6 : **DeleteValue**

این متد هم که از نامش پیداست . پاک کردن یک مقدار کار این متد است . به کد زیر توجه کنید:

```
RegistryKey reg;
```

```
reg = Registry.ClassesRoot.OpenSubKey("GC",true);
```

```
reg.DeleteValue("Your Value");
```

1- 7 : **Flush**

خوب است در هر قسمتی که کارتان با رجیستری تمام شد ، یک بار این متد را فراخوانی کنید . البته متد Close کار این متد (Flush) را هم انجام خواهد داد.

```
Registry.ClassesRoot.Flush();
```

1- 8 : **GetAccessControl**

به MSDN مایکروسافت مراجعه کنید.

1- 9 : **GetSubKeyName**

با استفاده از این کد میتوانید زیر کلیدهای یک کلید رجیستری را بدست بیاورید . فقط قبل از اینکه کد زیر را اجرا کنید ، یک لیست باکس به فرمتان اضافه کنید و خاصیت Name آن را برابر با listBox قرار دهید:

```
RegistryKey Reg;
```

```
Reg = Registry.CurrentUser.OpenSubKey("Software");
```

```
//-----
```

```
foreach (string strSubKeysName in Reg.GetSubKeyNames())
```

```
{
```

```
listBox.Items.Add(strSubKeysName);
```

```
}
```

1- 10 : **GetValue**

این متد میتواند مقدار یک Value را برگرداند . کد زیر را ببینید:

```
RegistryKey reg;
```

```
reg = Registry.ClassesRoot.OpenSubKey("GC",true);
```

```
MessageBox.Show(reg.GetValue("Your Value").ToString());
```

1- 11 : **GetValueKind**

با این متد میتوانید نوع یک Value را دریافت کنید.

انواع Value عبارتند از:

1- String

2- Binary

3- DWORD

4- Multi-String

5- Expandable

به این کد توجه کنید:

```
RegistryKey reg;
```

```
reg = Registry.ClassesRoot.OpenSubKey("GC",true);
```

```
MessageBox.Show(reg.GetValueKind("Your Value").ToString());
```

1- 12 : **GetVlaueNames**

با این کد هم میتوانید مقادیر موجود در یک کلید را بدست بیاورید . کد زیر:

```
RegistryKey Reg;  
  
Reg = Registry.CurrentUser.OpenSubKey("SoftwareMy APPSettings");  
  
//-----  
  
foreach (string strSubKeysName in Reg.GetValueNames())  
  
{  
  
listBox1.Items.Add(strSubKeysName);  
  
}
```

1- 13 : **Name**

این یکی هم مسیر جاری رجیستری را برمیگرداند:

```
RegistryKey reg;  
  
reg = Registry.ClassesRoot.OpenSubKey("GC",true);  
  
MessageBox.Show (reg.Name);
```

1- 14 : **SetAccessControl**

به MSDN مایکروسافت مراجعه کنید.

1- 15 : **SetValue**

بوسیله این متد میتوانید یک مقدار را در رجیستری بنویسید:

```
RegistryKey Reg;
```

```
Reg = Registry.LocalMachine.OpenSubKey("SOFTWARE", true);
```

```
//-----
```

```
Reg.SetValue("My Value", "Program",RegistryValueKind.String);
```

یک نکته در این کد وجود دارد . در خط سوم کدها آنجا که نوشته شده **RegistryValueKind.String** ، اینجا نوع متغیر تنظیم شده است . انواع متغیرها را میتوانید در راهکار **11-1** مشاهده کنید .

1- 16 : **SubKeyCount**

این یکی هم تعداد زیر کلیدهای یک کلید رجیستری را بازگشت خواهد داد:

```
RegistryKey Reg;
```

```
Reg = Registry.LocalMachine.OpenSubKey("SOFTWARE", true);
```

```
//-----
```

```
MessageBox.Show(Reg.SubKeyCount.ToString());
```

1- 17 : **ValueCount**

و اما آخرین مورد . بوسیله این تابع میتوانید تعداد متغیرهای داخل یک کلید را بدست بیاورید.

```
RegistryKey Reg;
```

```
Reg = Registry.LocalMachine.OpenSubKey("SOFTWARE", true);
```

```
//-----
```

```
MessageBox.Show(Reg.ValueCount.ToString());
```