

نحوه کار با رجیستری در C#

پیشگفتار :

سیستم عامل ویندوز تنظیمات سخت افزاری و نرم افزاری خود را بطور مرکزی در یک بانک اطلاعاتی با ساختار سلسله مراتبی ذخیره می کند که رجیستری نام دارد. رجیستری جایگزینی برای بسیاری از فایل‌های پیکربندی INI، SYS و COM است که در نسخه های اولیه ویندوز موجود بود. رجیستری ، سیستم عامل را با مهیا کردن اطلاعات مورد نیاز برای اجرای برنامه ها و load شدن component ها ، کنترل می کند.

- 1- اطلاعات موجود در رجیستری
- 2- اجزای رجیستری
- 3- استفاده از registry به وسیله C#
- 4- کلاس Registry
- 5- کلاس RegistryKey
- 6- استفاده از property های SubKeyCount و Valuecount

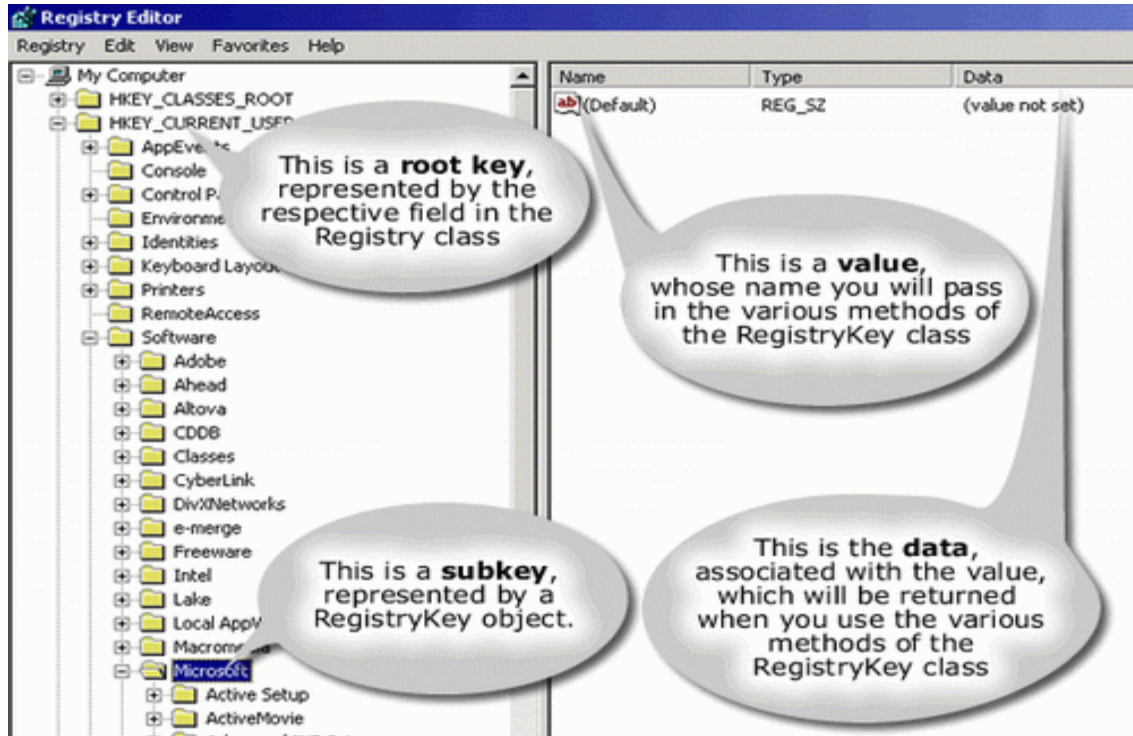
1. اطلاعات موجود در رجیستری

رجیستری شامل انواع مختلفی از اطلاعات می باشد همچون :

- اطلاعات سخت افزارهای نصب شده روی سیستم
 - اطلاعات درایورهای نصب شده روی سیستم
 - اطلاعات برنامه های نصب شده روی سیستم
 - اطلاعات پروتکل‌های شبکه ای مورد استفاده در سیستم
- ساختار رجیستری شامل چندین مجموعه رکورد است که داده های این رکوردها توسط بسیاری از برنامه ها و اجزای سیستم عامل خوانده و یا نوشته می شود .

2. اجزای رجیستری :

اجزای تشکیل دهنده رجیستری عبارتند از :



شکل 1 : اجزای تشکیل دهنده Registry

Subtree – 1 : subtree ها همانند folder های موجود در ریشه یک درایو ، هارد هستند . رجیستری

ویندوز دارای پنج subtree می باشد :

HKEY_LOCAL_MACHINE : شامل تمام داده های پیکربندی برای کامپیوتر می باشد و شامل 5

key است : Hardware ، SAM ، Security ، Software و System

HKEY_USERS : شامل داده های مربوط به تنظیمات سیستم عامل برای هر user است مثل

تنظیمات desktop و محیط ویندوز

HKEY_CURRENT_USER : شامل داده های کاربر فعلی سیستم

HKEY_CLASSES_ROOT : شامل اطلاعات پیکربندی نرم افزار است مثل داده های OLE و

داده های کلاسهایی متناظر با فایل

HKEY_CURRENT_CONFIG: شامل اطلاعات مورد نیاز برای تنظیمات داریورهای سخت

افزاري و غيره

2 – key : Key ها همانند folder ها و subfolder هاي روي هارد هستند . هر key متناظر با object هاي نرم افزاري يا سخت افزاري مي باشد . subkey ها key هايي هستند که درون یکسري key قرار دارند .

3 – Entry : هر key داراي یک يا چند entry است. هر entry داراي سه بخش مي باشد :

3.1- نام Name

3.2- نوع داده اي Data Type : مقدار هر entry يکي از انواع داده هاي زير است :

REG_DWORD ، REG_SZ ، REG_EXPAND_SZ ، REG_BINARY ،
REG_MULTI_SZ ، REG_FULL_RESOURCE_DESCRIPTOR

3.3 - مقدار Value

نکته 1 : براي مشاهده رجیستري و اعمال تغییرات در آن (لطفاً اگر هیچ تجربه اي در تنظیم کردن رجیستري ندارید اطلاعات آنرا تغییر ندهید) ، مي توانيد از برنامه regedit.exe و يا regedt32.exe موجود در ويندوز استفاده کنید . براي اينکار کافيست نام برنامه را در کادر Run وارد کنید .

3. استفاده از registry به وسيله C# :

در ابتدا با اين دید که registry اطلاعاتی بیش از آنچه در فایل است را نگهداری نمی کند و شیوه کاری NET. ذخیره سازی اطلاعات به صورت فایل های XML است موضوع را پی می گیریم .
پیش از هر کاری یک workspace در Visual Studio .NET ایجاد کنید .

File → New → Project

و در قسمت نام ، یک نام برای پروژه انتخاب کنید.

برای شروع باید به صورت صریح Microsoft.win32 را در کد خود به عنوان refrence قرار دهید تا به ابزار دستکاری Registry دست یابید.

```
using Microsoft.Win32;
```

این refrence در بین دیگر refrence های برنامه قرار می گیرد، که دستیابی به اجزای مرتبط با registry برای کنترل آن را فراهم کند.

دو کلاس در این namespace قرار دارد:

1- Registry

2- RegistryKey

این دو کلاس با هم استفاده می شوند تا بهتر بتوان عملیات های مختلف در registry را انجام داد.

4. کلاس Registry :

کلاس Registry به تنهایی نمی تواند ابزار های زیادی در اختیار کاربر قرار دهد زیرا این کلاس به ندرت هفت نود سطح بالا در registry را برای دستیابی و دستکاری در اختیار کاربر قرار می دهد وقتی این کلاس، در داخل کلاس RegistryKey استفاده می شود قدرتمند است. با این وجود با ورود به محیط Visual Studio .NET و افزودن namespace فوق و تایپ کردن " Registry." ، لیست هفت subtree زیر مشاهده خواهد شد.

Registry Class name	Registry Value
Registry.ClassesRoot	HKEY_CLASSES_ROOT
Registry.CurrentConfig	HKEY_CURRENT_CONFIG
Registry.CurrentUser	HKEY_CURRENT_USER
Registry.DynData	HKEY_DYN_DATA
Registry.LocalMachine	HKEY_LOCAL_MACHINE
Registry.PerformanceData	HKEY_PERFORMANCE_DATA

بعد از شناسایی کلاس Registry به بررسی کلاس RegistryKey می پردازیم.

: کلاس RegistryKey

این کلاس در بخش دستکاری registry شامل بخش های بیشتری است که استفاده از آنها می تواند منجر به ایجاد یا حذف داده شود.

برای استفاده از این کلاس باید یک نمونه از کلاس را ایجاد کنیم.

```
RegistryKey OurKey = Registry.Users;
```

در کد بالا یک نمونه از این کلاس با نام OurKey ایجاد کرده ایم که در داخل registry به صورت یک زیر کلید از HKEY_USERS مقدار دهی شده است.

اساسا این مقداردهی به این معناست که هر متد استفاده شده برای این کلید تاثیر مستقیم در زیر کلید (subkey) HKEY_USERS و آیتمهای داده ای آن دارد.

:OpenSubKey()

این متد یکی از مهمترین متدهای کلاس RegistryKey است زیرا اجازه دسترسی و دستکاری subkey مربوط به کلید سطح بالا را به کاربر می دهد. ممکن است عجیب به نظر برسد اگر همانند بالا از نوع کلید HKEY_USERS قرار داده باشیم. با استفاده از این متد می توان OurKey را به یک زیر کلید از نوع HKEY_USERS تبدیل کرد.

```
RegistryKey OurKey = Registry.Users;
OurKey = OurKey.OpenSubKey(".DEFAULT", true);
```

در مثال بالا یک نمونه از کلاس RegistryKey ایجاد شده است و value آن توسط کلید HKEY_USERS

مقدار دهی شده است. سپس یک subkey در داخل HKEY_USERS

که ".DEFAULT" نام دارد باز (open) شده است. پارامتر دوم متد تعیین کننده مد کلید شامل خواندنی، نوشتنی یا خواندنی/نوشتنی است. مقدار true به معنای قابلیت ویرایش کلید است.

DeleteSubKey() / CreateSubKey() & DeleteSubKeyTree()

هر سه متد فوق به مدیریت subkey های کلید جاری می پردازند و نام آنها به خوبی بیانگر عملیاتی است که انجام می دهند. کد زیر نحوه پیاده سازی آنها را نشان می دهد.

```
RegistryKey OurKey = Registry.Users;
// Create OurKey set to HKEY_USERS
OurKey = OurKey.OpenSubKey(".DEFAULT", true);
// Set it to HKEY_USERS\DEFAULT
OurKey.CreateSubKey("OurSubKey");
// Create the key HKEY_USERS\DEFAULT\OurSubKey
OurKey.CreateSubKey(@"OurSubKey\Subkey");
// Create a sub key HKEY_USERS\DEFAULT\OurSubKey\Subkey
OurKey.DeleteSubKey(@"OurSubKey\SubKey");
// Delete the subkey name "subkey"
OurKey.DeleteSubKeyTree("OurSubKey");
// Delete the whole subkey and any subkeys below it
```

: GetSubKeyNames()

این متد اجازه یافتن نام subkey های ثانویه را به کاربر می دهد. به عنوان نمونه می توان همه نام های subkey های HKEY_USERS را به دست آورد. به کمک یک الگوریتم recursive می توان همه نام ها را به صورت یکجا مشاهده کرد. قطعه کد اول یک مثال ساده از متد فوق است.

```
//The first example shows it using a foreach loop to display
each subkeyname
foreach(string Keyname in OurKey.GetSubKeyNames())
{
    MessageBox.Show(Keyname);
}
//The second example shows how to tranfer the names into a
string array
string[] Keynames = OurKey.GetSubKeyNames();
```

```

private void GetSubKeys(RegistryKey SubKey)
{
    foreach(string sub in SubKey.GetSubKeyNames())
    {
        MessageBox.Show(sub);
        RegistryKey local = Registry.Users;
        local = SubKey.OpenSubKey(sub,true);
        GetSubKeys(local); // By recalling itself it makes sure it
get all the subkey names
    }
}
//This is how we call the recursive function GetSubKeys
RegistryKey OurKey = Registry.Users;
OurKey = OurKey.OpenSubKey(@"\DEFAULT\test",true);
GetSubKeys(OurKey);

```

در قطعه کد بعدی مشاهده می شود که می توان از توابع بازگشتی برای بدست آوردن همه نام های subkey ها استفاده کرد.

متد های GetValue() / GetValuesNames() :

هنگام استخراج اطلاعات از registry، از متد های فوق استفاده می شود. پس از استفاده از متد های دیگر برای دسترسی به subkey ها، به کمک این دو متد می توان اطلاعات subkey ها را از آن ها استخراج کرد. متد اول GetValue() اجازه دسترسی به مقدار نگهداری شده در بخش registry value را با استفاده از نام یک subkey به کاربر می دهد. به عنوان مثال در زیر کلید "test" ، یک داده از نوع (REG_SZ) string، که نام آن "Testvalue" و مقدار آن "This is a test" است، قرار دارد. برای دسترسی به مقدار داده از قطعه کد زیر استفاده می شود.

```

RegistryKey OurKey = Registry.Users;

OurKey = OurKey.OpenSubKey(@"\DEFAULT\test",true);

MessageBox.Show(OurKey.GetValue("Testvalue").ToString());

```

زمانی که اطلاعات توسط کاربر از registry دریافت می شود، اطلاعات باید تبدیل شوند. برای این کار از متد ToString() استفاده شده است. علت این تبدیل object بودن مقادیر و subkey های داخل registry است. این تبدیل به کمک استفاده از متد ToString() یا casting انجام می شود.

اگر نیاز به دسترسی به مقادیر مجموعه ای از value ها در یک subkey وجود داشته باشد، به کمک متد GetValuesNames() می توان نام هر value در یک subkey را در قالب collection دریافت و به کمک کد زیر مقدار هر value را در آن subkey به دست آورد.

```
RegistryKey OurKey = Registry.Users;
OurKey = OurKey.OpenSubKey(@"\DEFAULT\test",true);
foreach(string valuenam in OurKey.GetValueNames())
{
    MessageBox.Show(valuenam + " " + OurKey.GetValue(valuenam).ToString());
}
```

: SetValue()

این متد یک value را در داخل registry value مقدار دهی می کند. برای استفاده از این متد شما باید نام و مقدار value را به متد ارسال کنید. در صورت عدم وجود value، متد فوق آن را ایجاد خواهد کرد.

```
RegistryKey OurKey = Registry.Users;
OurKey = OurKey.OpenSubKey(@"\DEFAULT\test",true);
OurKey.SetValue("Testvalue",0);
```

6- استفاده از property های SubKeyCount و Valuecount :

property اول تعداد subkey هایی که مستقیماً در زیر یک کلید قرار دارند را مشخص می کند و property دوم تعیین کننده تعداد value ها در یک subkey مشخص است.

```
RegistryKey OurKey = Registry.Users;

OurKey = OurKey.OpenSubKey(@"\DEFAULT\test",true);

MessageBox.Show(OurKey.SubKeyCount.ToString());

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

باید به این نکته اشاره کرد که وقتی کار با registry به پایان رسید، باید به کمک متد Close()، به عملیات خود خاتمه دهید مانند کد زیر:

```
RegistryKey OurKey = Registry.Users;

OurKey.Close()
```

حال به ارائه مثالی از registry می پردازیم. فرض کنید که می خواهیم Location یک فرم را توسط registry کنترل کنیم. برای این کار از یک کلید استفاده می کنیم که از نوع HKEY_CURRENT_USER تعریف شده است. برای ذخیره Location که از نوع point است نیاز به دو value به نامهای X و Y می باشد.

```
// Attempt to open the key
RegistryKey key = Registry.CurrentUser.OpenSubKey(
"Software\Play\WindowPos" );

// If the return value is null, the key doesn't exist
if ( key == null )
{
    // The key doesn't exist; create it / open it
    key = Registry.CurrentUser.CreateSubKey(
"Software\Play\WindowPos" );
}

// Attempt to retrieve the value X; if null is returned, the

value
// doesn't exist in the registry.
if ( key.GetValue( "X" ) != null )
{
```

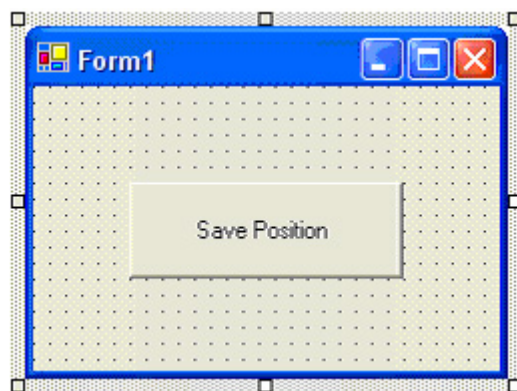
```

// The value exists; move the form to the coordinates stored
in the

// registry.
Location = new Point( (int)key.GetValue( "X" ),
(int)key.GetValue( "Y" ) );
}

```

در این کد ما سعی کرده ایم مقادیر X و Y را از کلید فوق بازیابی کنیم اگر این دو value وجود داشته باشند فرم را به مکان مناسب انتقال می دهیم که توسط تنظیم Location فرم این کار انجام می شود. Location از نوع point است پس ابتدا یک object از نوع point تعریف می کنیم و آن را با value های مناسب از registry-lookup مقداردهی می کنیم و آن را به عنوان مقدار جدید Location، قرار می دهیم. مکان فرم هنوز save نشده است پس value های X و Y در کلید registry وجود ندارند. برای اینکه اطلاعات ذخیره شود، یک button به فرم اضافه می کنیم. برای این کار یک button در بخش designer به فرم اضافه کرده و قطعه کد زیر را در کد button آن می نویسیم.



```

// Open the key
RegistryKey key = Registry.CurrentUser.OpenSubKey(
"Software\\Play\\WindowPos", true );

// Set the registry values to correspond to the form's
coordinates on the
// screen.
key.SetValue( "X", Location.X );
key.SetValue( "Y", Location.Y );

```

نکته ای که در این کد باید مورد توجه قرار گیرد اهمیت پارامتر دوم متد `openSubKey` است که از نوع `boolean` می باشد. اگر این پارامتر `true` باشد، `key` در مد نوشتاری باز می شود و به کاربر اجازه می دهد از متد `SetValue` استفاده کند در غیر این صورت (در صورت `false` بودن این پارامتر) فراخوانی این متد سبب بروز خطای فوق می شود:

error: UnauthorizedAccessException

حال با زدن کلید F5 می توان برنامه را اجرا کرده و با تغییر مکان فرم و زدن button طراحی شده، Location فرم را ذخیره می کنیم.

کلمات کلیدی :

- رجیستری - داده های پیکر بندی ویندوز - داده های تنظیمات سیستم عامل - داد های کاربر فعلی -
- اطلاعات پیکر بندی نرم افزار - ابزار دستکاری رجیستری - کلاس Registry - کلاس RegistryKey -
- متد OpenSubkey() - مدیریت subkey - متد GetSubKeyNames() - نام subkey های ثانویه -
- استخراج اطلاعات از رجیستری - متد GetValueNames() - خصوصیت SubKeyCount و
- خصوصیت Valuecount -

- HKEY_LOCAL_MaCHINE(Hardware,SAM,Security,Software,System)
- Entry - HKEY_CLASSES - HKEY_CURRENT_USER - HKEY_USERS
- REG_BINARY - REG_EXPAND_SZ - REG_SZ - REG_DWORD - Key
- regedit.exe - REG_FULL_RESOURCE_DESCRIPTOR - REG_MULTI_SZ
- GetValue - DeletSubRKeyTree() - CreateSubKey() - DeleteSubKey() - OurKey