

۲-۱۰ چگونه تغییرات در رجیستری را غیر فعال کنیم؟

برای اینکه از تغییرات رجیستری ویندوز توسط کاربران جلوگیری کنیم می توانیم رجیستری را ببندیم.

راه اول (با استفاده از پنجره **regedit**):

از منوی **Start** وارد **Run** شده و عبارت **regedit** را وارد نمایید ، سپس **Enter** را بزنید تا وارد ویرایشگر رجیستری شوید و به مسیر زیر بروید :

HKEY_Current_User\Software\Microsoft\Windows\CurrentVersion\Policies\System

در قسمت دیگر صفحه ، بر روی فضای خالی از صفحه راست کلیک کرده و از **New** گزینه **DWORD Value** را انتخاب کنید و نام آن را **DisableTaskMgr** قرار دهید.

بر روی **DisableRegistryTools** دوبار کلیک کرده و مقدار **Value** را بر روی **۱** تنظیم سپس **OK** کنید.

جهت بازگرداندن به حالت قبلی و فعال کردن **Regedit** کافی است عدد **۱** را به **۰** تغییر دهید.

راه دوم (با استفاده از پنجره **Group Policy**):

۱- از منوی **Start/RUN** برنامه **GPedit.msc** را اجرا کنید.

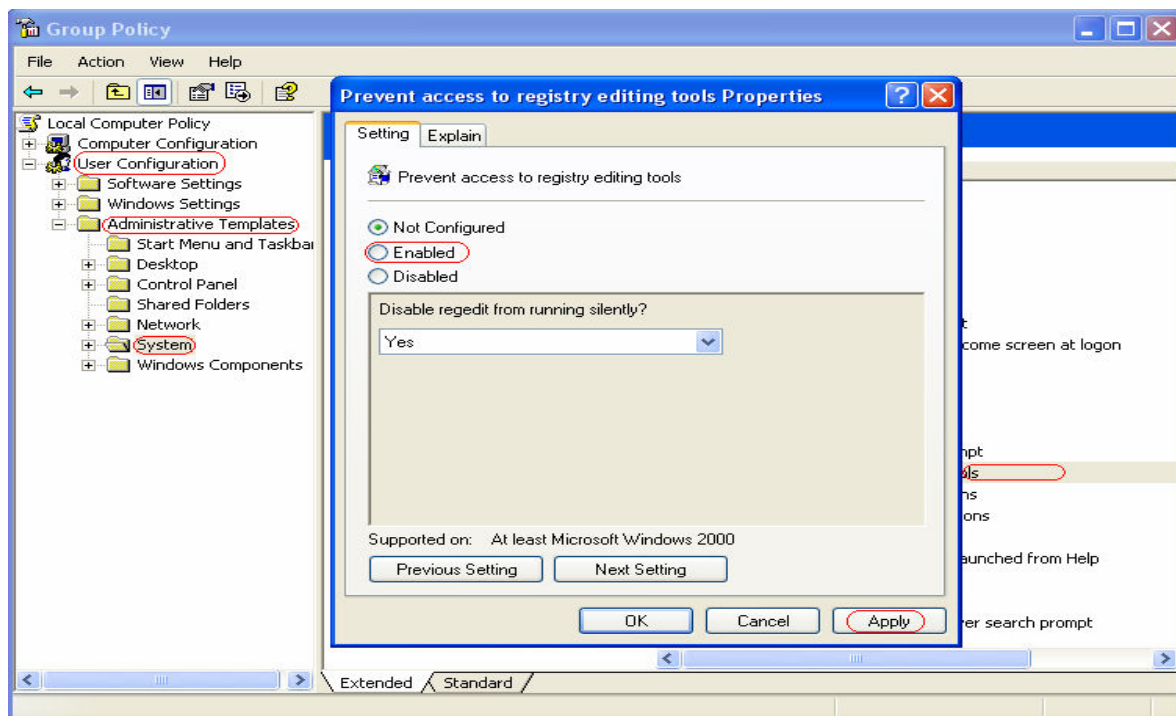
۲- از پنجره ظاهر شده مسیر زیر را پیدا کنید.

user configuration \ administrative templates \ system

۳- حال از پنجره ظاهر شده گزینه **prevent access to registry editing tools** را دابل کلیک کنید و از

پنجره ظاهر شده گزینه **Enable** را انتخاب کرده و پنجره را با زدن **Ok** ببندید. حال دیگر رجیستری باز نخواهد شد.

برای فعال کردن عکس این کار را انجام دهید.



شکل ۲-۴: غیر فعال کردن تغییرات در رجیستری

۲-۱۱ چند مثال از تغییرات در رجیستری

۲-۱۱-۱ غیر فعال کردن Task Manager

حتما شما نیز با پنجره Task Manager آشنا هستید. پنجره ای که در مواقع بحرانی با فشردن سه کلید **Ctrl + Alt + Delete** به کمک شما می آید. بنا به هر دلیلی ممکن است دوست داشته باشید Task Manager را در ویندوز غیرفعال کنید .

راه اول (با استفاده از پنجره **regedit**):

از منوی **Start** وارد **Run** شده و عبارت **regedit** را وارد نمایید ، سپس **Enter** را بزنید و به مسیر زیر بروید:

HKEY_Current_User\Software\Microsoft\Windows\CurrentVersion\Policies\System

در قسمت دیگر صفحه ، بر روی فضای خالی از صفحه راست کلیک کرده و از **New** گزینه **DWORD Value** را انتخاب کنید.

سپس نام آن را **DisableTaskMgr** قرار دهید.

بر روی **DisableTaskMgr** دابل کلیک کرده و مقدار **Value** را بر روی ۱ تنظیم کنید . با **OK** کردن تنظیمات ذخیره می شود.

جهت بازگرداندن به حالت قبلی و فعال کردن **Task Manager** کافی است عدد ۱ را به ۰ تغییر دهید.

راه دوم (با استفاده از پنجره **Group Policy**):

در این تکنیک دسترسی به **Task Manager** را بدون استفاده از رجیستری غیر فعال می کنیم.

از منوی **Start** وارد **Run** شده و عبارت **gpedit.msc** را وارد نمایید ، سپس **Enter** را بزنید و به مسیر زیر بروید:

User Configuration\Administrative Templates\System\Ctrl+Alt+Delete options

حال در قسمت سمت راست پنجره روی **Remove Task Manager** دابل کلیک کرده و از پنجره باز شده چک باکس **Enabled** را انتخاب کنید.

۲-۱۱-۲ غیر فعال کردن دستور **Shut Down**

برای غیر فعال کردن فرمان **Shut Down** کلید زیر را در رجیستری پیدا کنید:

[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer]

یک مقدار جدید از نوع **DWORD** با نام **noclose** ایجاد کرده و آنرا برابر با 1 قرار دهید تا فرمان **Shut Down** غیر فعال گردد. برای فعال سازی مجدد آن ، مقدار فوق را برابر با 0 قرار دهید

۲-۱۱-۳ از کار انداختن راست کلیک در Desktop

وارد رجیستری شوید و کلید زیر را پیدا کنید :

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\Currentversion\Policies\Explorer

یک متغیر جدید از نوع DWORD بسازید و اسم آن را NoViewContextMenu را به آن بدهید با دادن مقدار ۱ به این متغیر منوی مزبور را از کار میندازید و با دادن مقدار ۰ به آن دوباره آن را به کار می اندازید .

۲-۱۱-۴ برداشتن منوی file از مرورگر ویندوز

رجیستری را باز کرده و کلید زیر را پیدا کنید :

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\Currentversion\Policies\Explorer

یک متغیر جدید از نوع DWORD بسازید و به آن نام NofileMenu بدهید اگر به این متغیر مقدار ۱ بدهید حذف و اگر مقدار ۰ بدهید به جای خود باز میگردد .

۲-۱۱-۵ نمایش یک عنوان دلخواه برای قسمت Logon در ویندوز

به کلید زیر بروید :

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\Current Version\Winlogon

داده ای از نوع String تعریف و نام آنرا LegaNoticeCaption قرار دهید .
حال مقدار آنرا متن دلخواه خود قرار دهید . این متن درون پنجره نمایش داده میشود .
داده ای دیگر از نوع String ایجاد و نام آنرا LegaNoticeText قرار دهید . این متن در بخش عنوان پنجره قرار میگیرد .

۲-۱۱-۶ تنظیم پنجره خوش آمدگویی ویندوز

به کلید زیر بروید :

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Explorer\Tips

خوب شما میتوانید در این قسمت ۴۸ نکته در ویندوز ۹۸ و ۹۵ و ۵۰ نکته در XP و NT قرار دهید که در هنگام بالا آمدن ویندوز برای شما نمایش دهد شما میتوانید متغیرهایی از نوع STRING VALUE بسازید و به آن نامی از ۰ تا ۵۰ بدهید و نکات خود را در آن قرار دهید تا با هر بار بالا آمدن ویندوز به نمایش درآید این نکته بسیار بسیار جالبی است که خیلی کارا میشه باهاش کرد مانند نکاتی که میخوایم کاربران ویندوز آنها را رعایت کنند و ...

۲-۱۲ برنامه نویسی رجیستری به وسیله C#

در ابتدا با این 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 .NET Visual Studio 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 می کند.

دو کلاس در این namespace قرار دارد:

Registry-۱

RegistryKey-۲

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

۲-۱۲-۱ 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 .NET Visual Studio 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
Registry.Users	HKEY_USERS

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

۲-۱۲-۲ 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 زیرکلید از `HKEY_USERS` مقداردهی شده است. اساساً 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.

۲-۱۲-۳ متدهای کلاس `RegistryKey` متد `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 متد کلید شامل خواندنی یا خواندنی /نوشتنی است. مقدار `true` به معنای قابلیت ویرایش کلید است.

متدهای `DeleteSubKey()` / `CreateSubKey()` & `DeleteSubKeyTree()`

هر سه متد فوق به مدیریت `subkey` های 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.OpenSubKey(".DEFAULT", true);
// Set it to HKEY_USERS\DEFAULT
OurKey.CreateSubKey(@"DEFAULT\OurSubKey");
// Create the key HKEY_USERS\DEFAULT\OurSubKey
OurKey.CreateSubKey(@"DEFAULT\OurSubKey\Subkey");
// Create a sub key HKEY_USERS\DEFAULT\OurSubKey\Subkey
OurKey.DeleteSubKey(@"DEFAULT\OurSubKey\Subkey");
// Delete the subkey name "subkey"
OurKey.DeleteSubKeyTree(@"DEFAULT\OurSubKey");
// Delete the whole subkey and any subkeys below it
```

متد :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.

```
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.OpenSubKey(".DEFAULT", 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.

متدهای :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 string (REG_SZ)، 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;
// Create OurKey set to HKEY_USERS
OurKey.OpenSubKey(".DEFAULT", true);
MessageBox.Show(OurKey.GetValue("Testvalue").ToString());
```

متد SetValue():

این متد یک Value را در داخل registry value مقداردهی 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.OpenSubKey(".DEFAULT", true);
OurKey.SetValue("Testvalue", 0);
```

استفاده از Property های Valuecount و SubKeyCount :

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 معین است.

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

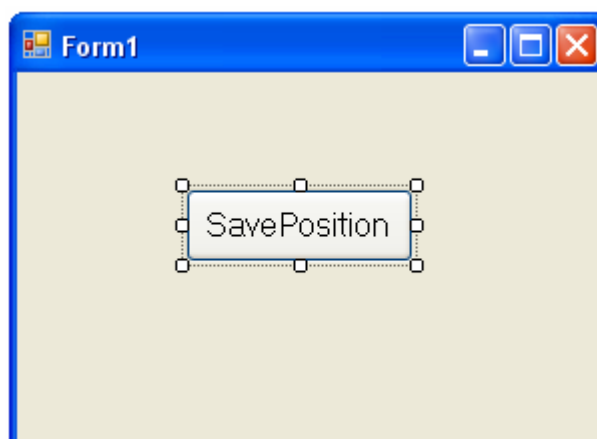
۲-۱۲-۴ مثالی از یک برنامه Registry با C#

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 Point است نیاز به دو Value با 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 بازبایی 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 هنوز 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 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 زیر 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.

۲-۱۳ دستور کار

با استفاده از تغییرات در رجیستری کاری کنید که:

۱. تصویر desktop عوض نشود
 ۲. control panel از لیست برنامه ها حذف شود
 ۳. قسمت run غیر فعال شود و یا حذف شود (سپس دوباره فعال شود)
 ۴. از تغییر پسورد توسط کاربر جلوگیری شود.
- مراحل کار را بنویسید

۲-۱۴ پروژه برنامه نویسی

برنامه ای بنویسید که

- الف- بدون مراجعه به محیط ویرایشگر رجیستری ، نام کلید، مسیر ایجاد کلید ، نام مقادیر موجود در کلید ، نوع داده مقادیر و مقدار آن را گرفته و در رجیستری ایجاد کند.
- ب- با استفاده از دستورات برنامه بالا نرم افزاری طراحی کنید که بطور اتوماتیک تنظیمات یکی از تکنیک های رجیستری را انجام دهد.

۲-۱۵ منابع

www.computing.net
www.IRITN.com
www.hamkelasy.com
www.iau-neyshabur.ac.ir