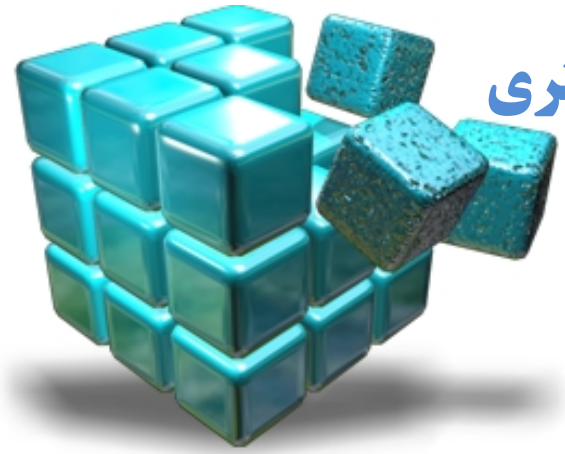
به نام خدا



اسـرار رجـیـسـتری

کاری از جواد سلطانی

وی سعی دارم طی این مقاله به منظور افزایش سطح دانش خوانندگان در زمینه رجیستری برای راه‌بران ویندوز در سطوح متوسط و پیشرفته و بخصوص برنامه‌نویسان مطالبی را ارایه دهم که کمتر درباره آنها می‌دانید.

البته با توجه به اینکه بیشتر خوانندگان این مقاله باید برنامه‌نویسان باشند, سمت و سوی عناوین را طوری انتخاب کرده‌ام که بیشتر مطالب برای برنامه‌سازان مناسب باشند. این مقاله شامل چند فصل است که هر فصل با یک سوال آغاز شده و در طول مباحث به سوال ابتدای فصل, پاسخ داده می‌شود. لازمه دریافت کامل مطالب بعضی از قسمت‌ها آگاهی شما از فصول قبل تر آن عنوان می‌باشد. پس همه فصل‌ها را با دقت مطالعه کنید.

سیستم عامل هدف این مقاله در ۹۰ درصد موارد ویندوز XP است و در قسمت‌های لازم به ویندوزهای قبل از XP نیز اشاره‌هایی شده است. با امید مفید بودن این مطالب برای شما ...

جواد سلطانی (مرداد ۱۳۸۴)

مقدمه

گر چند افسانه‌های زیادی راجع به رجیستری وجود دارد. سخنانی که گاه هر کس را ترغیب می‌کند تا با کنکاش در رجیستری خود, که شاید بتواند به سوال‌هایی که در ذهنش به وجود آمده جواب دهد. اما رجیستری چیزی جز یک پایگاه داده برای نگهداری تنظیمات برنامه‌های کاربردی و بخصوص ویندوز نیست. شرکت مایکروسافت نیز مستندات زیادی برای آگاهی کاربران در رابطه با رجیستری ویندوز ارایه نکرده است. دلیل این امر کاملا روشن است. رجیستری نقش ظریفی را در ویندوز ایفا می‌کند. با هر بار کلیک شما در هر قسمت فضای کاریتان دهها ارجاع و سوال از رجیستری صورت می‌گیرد. در واقع رجیستری سناریوی پشت پرده ویندوز است. به همین دلایل است که رجیستری حکم سنگ سرطان را دارد زیرا وجود آن اگر چه کار را برای برنامه‌سازان و همه برنامه‌های کاربردی ساده کرده اما عدم وجود آن نیز منجر به فاجعه می‌شود.

همه برنامه‌های کاربردی حتی کوچک ترین و کم کارترین آنها بدون گرفتن و نوشتن اطلاعات‌شان در رجیستری نمی‌توانند کار کنند. اما به این دلیل نیست که برنامه‌سازان طی مراحل فراگیری و کامل کردن یادگیری اصول زبان مورد علاقه‌شان به کنکاش در چون و چرای رجیستری میپردازند بلکه برنامه‌سازان به وسیله رجیستری کار خود را با داده‌های تنظیمات برنامه‌شان ساده کرده و برنامه خود را قادر می‌سازند که برای هر کاربر تنظیمات شخصی ایجاد کنند. به این ترتیب با بالا رفتن ضریب عملکرد برنامه‌شان همین طور افزایش ایمنی ارتباط هر کاربر با برنامه خود باعث شوند که با خیال آسوده به کدهای اصلی و هدف برنامه بپردازند. رجیستری این امکان را برای برنامه‌سازان فراهم می‌کند تا آنها در یک محیط کاملا سلسله مراتبی نظم را برای برنامه‌ای که می‌نویسند به ارمغان آورند.

رجیستری قلب و روح ویندوز است و یک برنامه‌نویس حرفه‌ای برنامه‌نویسی است که محیط کار خود (یعنی سیستم عاملش) را به خوبی بشناسد. در صورتی که شما جزء دسته کاربران, در سطح حداقل متوسط قرار ندارید یا اینکه تاکنون با رجیستری ویندوز کار نکرده‌اید در همین شروع کار موارد اساسی و لازم را به صورت فشرده توضیح می‌دهم در غیر این صورت می‌توانید از مطالعه ادامه این بخش صرف نظر کنید.

سخنگوی رجیستری RegEdit.exe

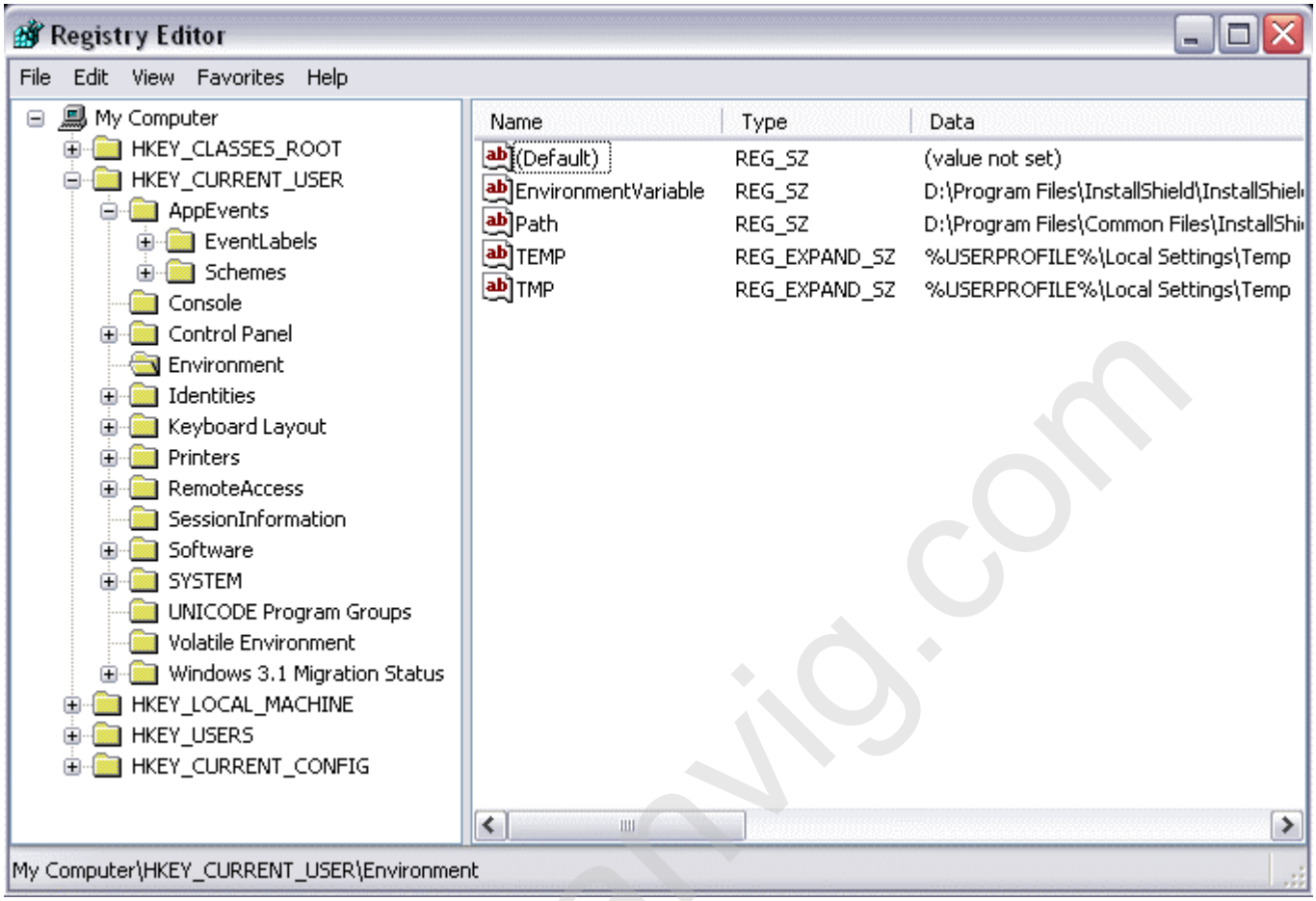
برای مشاهده محتوای رجیستری سیستم خود کافیست در منوی Run فرمان regedit را تایپ و اجرا کنید. این برنامه رابط کاربری بین اطلاعات موجود در رجیستری و شماست. Regedit اطلاعات رجیستری را همانند Microsoft Explore به نمایش در می‌آورد.

در سمت چپ ویرایشگر رجیستری Regedit پوشه‌هایی را می‌بینید که به آنها کلید Key گفته می‌شود و همان طور که مشاهده می‌کنید چند کلید هستند که بقیه کلیدها زیر مجموعه آنها می‌باشند. به این‌ها کلیدهای اصلی Sub Key می‌گوییم. البته تصویر شماره یک نمایان گر Sub key های موجود در Win XP می‌باشد که در اینجا از پنج عدد تجاوز نمی‌کنند. هر کلید اصلی مجموعه‌ای از تنظیمات یک بخش می‌باشد که در جدول به طور کل کارایی‌های هر کدام عنوان شده است. از آنجایی که نام آنها کمی بلند است ما از مخفف نام آنها استفاده خواهیم کرد.

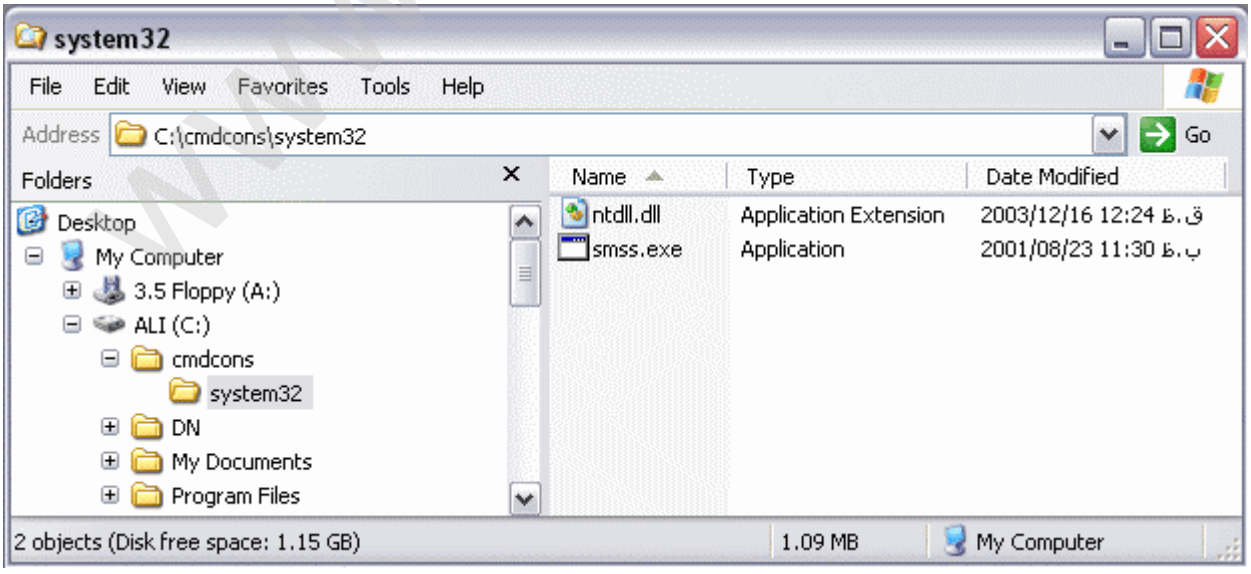
کلید اصلی	مخفف	شرح
HKEY_CLASSES_ROOT	HKCR	انواع مختلف فایل را به برنامه مربوطه ارتباط داده و ثبت کلاس‌ها برای اشیا Com را نیز در بردارد
HKEY_CURRENT_USER	HKCU	تنظیمات خاص کاربر مربوط به کاربر کنسول*
HKEY_LOCAL_MACHINE	HKLM	تنظیمات خاص کامپیوتر مربوط به همه کاربران و داده‌های سیستمی و سخت افزاری است
HKEY_USERS	HKU	حداقل شامل سه زیر کلید است که تنظیمات هر کاربر را در شاخه مربوط به آن نمایش می‌دهد
HKEY_CURRENT_CONFIG	HKCC	نوعی ارتباط با داده‌های پیکر بندی برای پروفایل‌های سخت افزاری است

* کاربر کنسول کاربری است که با صفحه کلید کار می‌کند

تصویر ۱



تصویر ۲



به تصویر شماره ۲ توجه کنید. ساختار رجیستری یک ساختار سلسله مراتبی است که بسیار مشابه ساختار داده‌های موجود در دیسک سخت شما است.

در پنجره سمت راست ویرایشگر رجیستری خود داده‌های هر کلید را می‌بینید. در واقع هر کلید یک یا چند مقدار دارد. این داده‌ها را مقادیر Values می‌نامیم. هر مقدار در رجیستری دارای سه خصیصه اصلی می‌باشد:

۱- نام Name که حداکثر ۵۱۲ کاراکتر ANSI یا ۲۵۶ کاراکتر یونی کد به غیر از کاراکترهای "\" , "*" و "?" می‌توانید در نام گذاری استفاده کنید در ضمن نام‌هایی که با "." نقطه آغاز می‌شود برای استفاده خود ویندوز هستند.

۲- نوع Type که نوع هر مقدار فرمت و نوع داده‌های آن را مشخص می‌کند که پر استفاده ترین آنها عبارتند از:

REG_BINARY : این نوع متغیر داده نوع باینری خام را ذخیره می‌کند

REG_DWORD : این نوع متغیر برای نمایش داده‌های ۴ بایتی و همچنین برای ذخیره کردن مقادیر منطقی درست یا غلط TRUE OR FALSE به کار می‌روند. بدین ترتیب که برای نمایش غلط از عدد "0" و برای نمایش درست از عدد "1" استفاده می‌شود.

REG_SZ : این نوع متغیر رشته ای استاندارد می‌باشد که برای ذخیره کردن متن قابل خواندن توسط کاربر استفاده می‌شود.

REG_EXPAND_SZ : این یک نوع متغیر رشته ای قابل گسترش می‌باشد که نرم‌افزارها جهت بعضی اعمال خود از این متغیر استفاده می‌کنند. مثلا به جای عبارت ٪SYSTEM ROOT٪ با مقدار واقعی خود یعنی مسیر نصب ویندوز (مثلا C:\WINDOWS) جایگزین می‌شود. باید توجه داشته باشید که این نوع متغیر نوعی رشته است و مادامی که نرم‌افزاری برای تبدیل آن به مقدار اصلی آن وجود نداشته باشد چیزی جزء REG_SZ نمی‌باشد.

Name	Type	Data
(Default)	REG_SZ	(value not set)
BainaryValueSample	REG_BINARY	52 2e 65 2e 67 2e 69 2e 73 2e 74 2e 72 2e 79
DWordValueOn	REG_DWORD	0x00000001 (1)
DWordValueSample	REG_DWORD	0x0000a12c (41260)
MultiSz	REG_MULTI_SZ	Line1 Line2 Line3 . . . LineN
ProcessSZ	REG_EXPAND_SZ	%UserProfile%\Application Data\Help
StringValueSample	REG_SZ	Strings & all Chars ... 123

✓ البته نوع‌های داده‌ای دیگری هم وجود دارند که برخی از آنها بخصوص برای برنامه‌نویسان سخت افزار بسیار مورد علاقه می‌باشند. در رابطه با آنها در به زودی بحث خواهم کرد ولی برای شروع کار تا همین حد کفایت می‌کند.

۳- داده‌ها Data که هر مقدار می‌تواند خالی, تهی یا حاوی داده‌ای باشد. داده هر مقدار می‌تواند حداکثر ۳۲۷۶۸ بایت باشد. اما حجم آن در عمل 2KB است.

هر کلید حداقل یک مقدار Value دارد که به آن مقدار پیش فرض Default گفته می‌شود. اگر بر روی یکی از کلیدها راست کلیک کنید و گزینه New سپس Key را انتخاب کنید کلیدی (به عنوان زیر کلید) ساخته می‌شود که Default در آن وجود دارد نوع مقدار پیش فرض همیشه باید از نوع رشته Reg_SZ باشد اما برنامه‌هایی که به درستی کار نمی‌کنند می‌توانند نوع مقدار پیش فرض را عوض کنند. داده مقدار پیش فرض هم تهی است تا مادامی که در آن چیزی نوشته شود. ویرایشگر رجیستری تهی را با علامت (value not set) نشان می‌دهد.

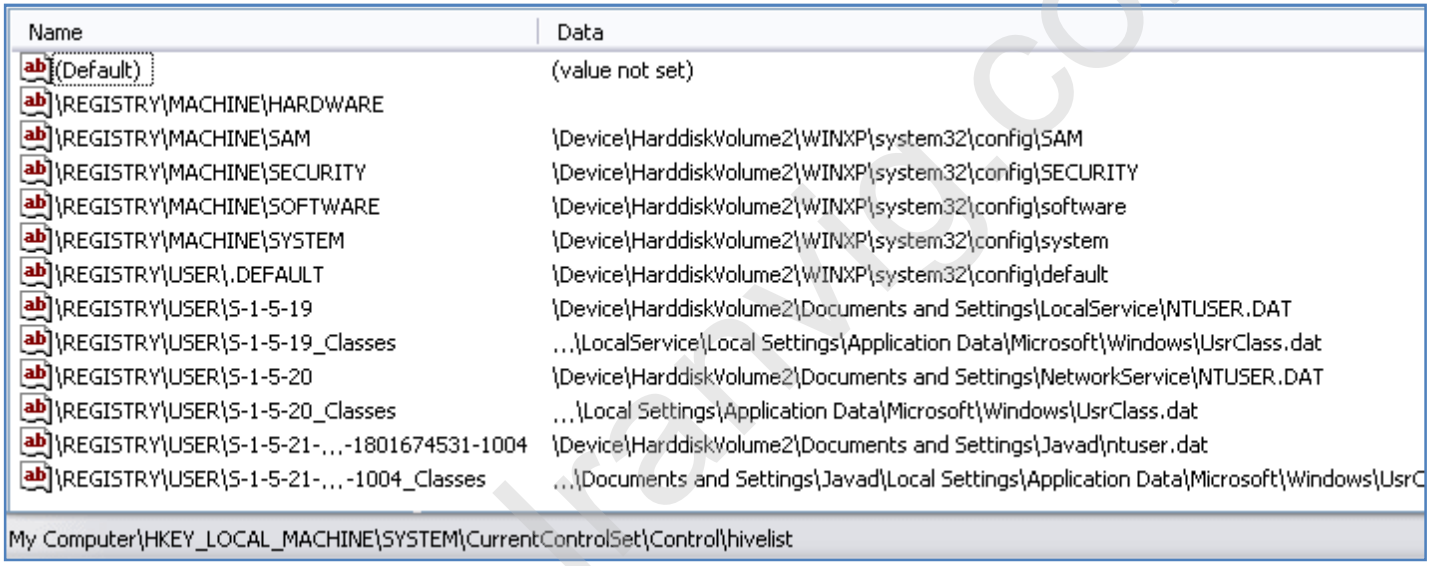
▪ فصل اول : فایل‌های Hive

آیا میدانید ویندوز محتوای رجیستری را کجا ذخیره می‌کند؟

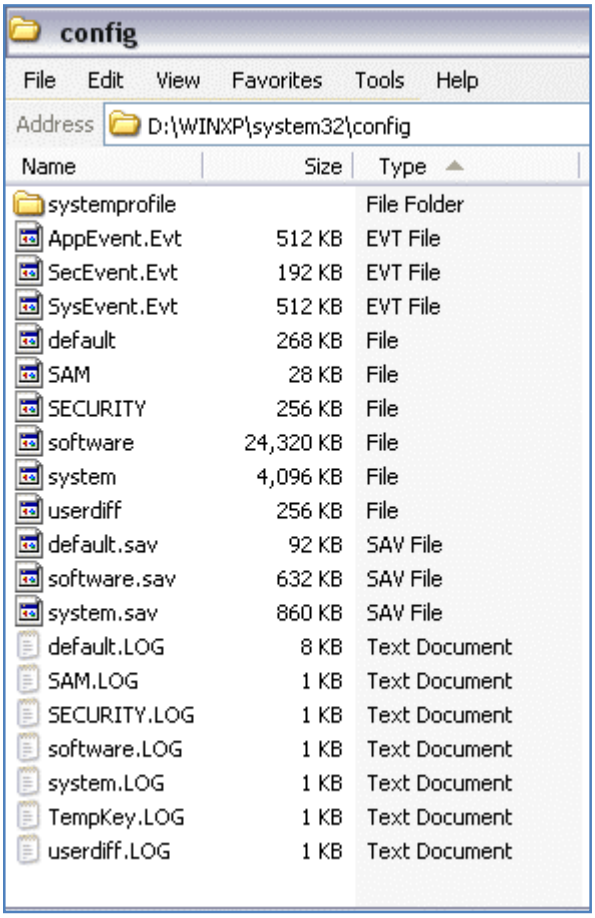
فایل‌های داده‌ای با عنوان Hive وظیفه نگه داری داده‌های رجیستری را برعهده دارند.

ویندوز فقط محتوای کلیدهای HKLM و HKU را ذخیره می نماید در مقاله‌های بعدی علت این عمر را شرح خواهم داد.

برای مطلع شدن از مسیر فایل‌های Hive به شاخه HKLM\SYSTEM\CurrentControlSet\Control\hivelist موجود در رجیستری مراجعه نمایید.



با کمی‌کنکاش در محتوای کلید فوق در خواهید یافت که کلید HKLM\HARDWARE در هیچ جایی ذخیره نمی‌شود این بدان علت است که ویندوز محتوای کلید فوق را هر بار به هنگام بالا آمدن ایجاد می‌کند. در واقع به وسیله فایل‌های inf و درایورهایی که قبلا برای سخت افزار نصب شده‌اند همچنین با سوال از خود قطعات سخت افزاری در رابطه با اطلاعات مورد نیاز برای ایجاد ارتباط بهینه با آنها, محتوای کلید HKLM\HARDWARE را در رجیستری مقدار دهی می‌کند.



اکثر فایل‌های Hive که به کلید HKLM مربوط می‌شوند در مسیر:

%WinDir%\System32\config

قرار دارند. برای دست یابی به این مسیر کافیست فرمان config را در منوی Run اجرا کنید!

فرمت تمام فایل‌های Hive باینری می‌باشد. ویندوز تمام اطلاعات موجود در کلیدها را صرف نظر از فرمت خواندن و نوشتاری آن‌ها توسط نرم‌افزارهای مختلف فقط به شکل باینری در فایل‌های Hive ذخیره می‌کند.

توجه داشته باشید که فایل‌های Hive را نمی‌توانید تغییر دهید یا آنها را پاک و جابه‌جا کنید, زیرا ویندوز به محض در خواست برای بالا آمدنش این فایل‌ها را باز می‌کند و امکان انجام عملیات های نام برده بر روی فایل‌های Hive باز را غیر ممکن می‌سازد.

در فصل‌های بعدی (بارگذاری فایل‌های Hive) روش‌هایی را به منظور پشتیبان گیری از رجیستری به وسیله فایل‌های Hive و چگونگی بارگذاری فایل‌های Hive روی یک ماشین یا سیستم بومی دیگر, ارائه خواهم داد.

جدول زیر انواع فایل‌هایی را معرفی می‌کند که به نوعی با فایل‌های Hive رابطه دارند.

انشعاب	شرح
.log	گزارش تغییرات یک فایل Hive
بدون انشعاب	خود فایل Hive است
.alt	در ویندوز XP مورد استفاده قرار نمی‌گیرد. فایل System.alt در ویندوز ۲۰۰۰ یک نسخه پشتیبان از فایل Hive اصلی سیستم یعنی System است
.sav	نسخه‌ای از یک فایل Hive که در پایان فاز "مد Text" برنامه نصب ویندوز ایجاد می‌شود. در صورت عدم موفقیت برنامه نصب ویندوز در فاز "مد graphics" این فایل‌ها به کمک برنامه نصب می‌شتابند تا در Restart بعدی برنامه نصب بتواند کار خود را از سر بگیرد

✓ **یک نکته:** ویرایشگر رجیستری خود را باز کنید (با اجرای فرمان Regedit از طریق منوی Run)

با کمی دقت در نام کلیدها متوجه می‌شوید که Regedit نام بعضی کلیدها را کاملا با حروف بزرگ نمایش داده مانند HKLM\SYSTEM تعداد این موارد زیاد نیست اما اکثر کلیدها با همان روش معمول نگاشته شده‌اند مانند HKCU\Console نکته همین جا است !

کیله کلیدهایی که با حروف بزرگ نام‌نویسی شده‌اند دارای فایل Hive مختص به خود هستند مثلا در این مورد HKLM\SYSTEM در فایل:

%WinDir%\System32\config*system*

ذخیره میگردد (دقت کنید که نام فایل system است و هیچ پسوندی ندارد).

این فصل مقدمه‌ای بر فایل‌های Hive بود.

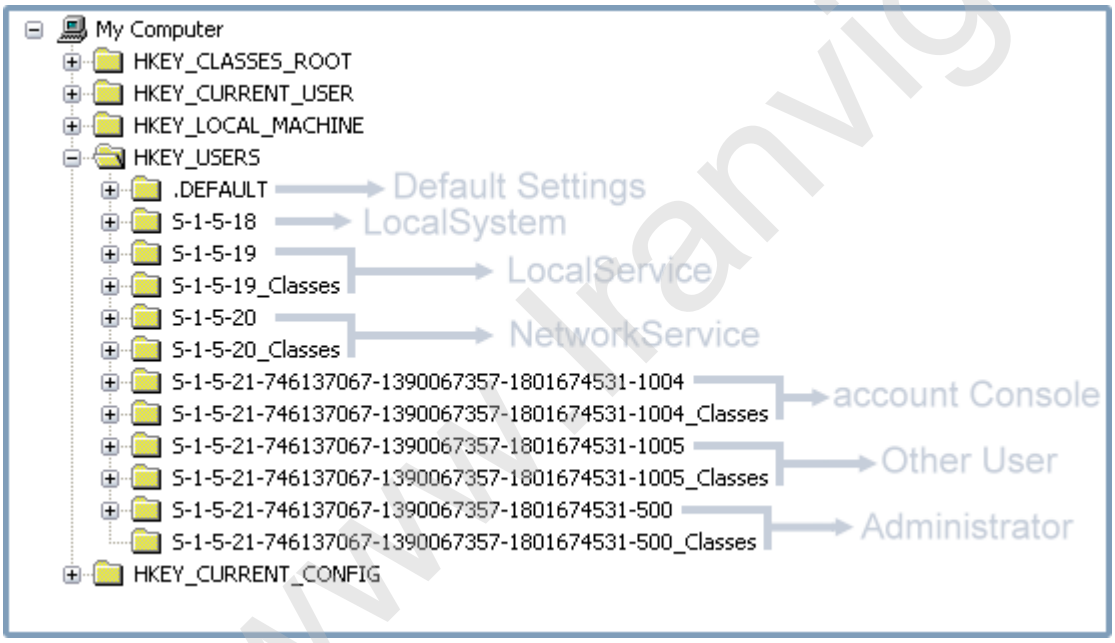
▪ فصل دوم : شناسه‌های امنیت SID

آیا می‌دانید هر کاربری که با ویندوز ارتباط برقرار می‌کند داری یک شناسه امنیت می‌باشد !

SID چیست ؟

اکانت‌های کامپیوتر, اکانت‌های کاربری, گروه‌ها و سایر شی‌های مرتبط با امنیت, اصول امنیتی به شمار می‌آیند که ویندوز برای هر کدام از آنها یک شناسه امنیت SID در نظر می‌گیرد. SID ها در محدوده خودشان منحصر به فرد هستند مثلا SID شخص من در ویندوزی که با آن کار می‌کنم برابر "S-1-5-21-746137067-1390067357-1801674531-1004" می‌باشد.

یک SID همیشه با حرف S آغاز می‌شود, عدد بعدی نشان دهنده نگارش SID می‌باشد که در این مورد ۱ است و مابقی اعداد شناسه مربوط به حوزه آن شی می‌باشند.



SID های مربوط به هر شی در حوزه فعالیت آن منحصر به فرد هستند حتی اگر شما کاربری باشید که با ویندوز ارتباط برقرار کرده و اقدام به پاک کردن اکانت اتصال خود کنید, SID که قبلا ویندوز آن اکانت را با آن می‌شناخت دیگر هیچ وقت برای هیچ شی دیگری در آن حوزه در نظر گرفته نمی‌شود.

برخی از SID ها کوتاه‌تر از مثال بالا هستند مانند S-1-5-18 (به تصویر توجه کنید) این‌ها SID های متداول هستند و در تمام کامپیوترها و حوزه‌ها یکسان می‌باشند. دلیل جالب بودن این SID ها آن است که یک هکر (کاوشگر) را کمتر به زحمت می‌اندازند و بارها و بارها در رجیستری و جاهای دیگر با آنها مواجه خواهید شد.

در زیر فهرست معروفترین آنها را می‌بینید:

SID	نام کاربر یا گروه
S-1-5-1	Dialup
S-1-5-2	Network
S-1-5-13	Terminal Service User
S-1-5-14	Remote Interactive Logon
S-1-5-18	System یا LocalSystem
S-1-5-19	LocalService
S-1-5-29	NetworkService
S-1-5-domain-500	Administrator
S-1-5-domain-501	Guest
S-1-5-domain-520	Group Policy Creator Owners
S-1-5-domain-545	Users
S-1-5-domain-546	Guests
S-1-5-domain-547	Power User

در این فهرست SID مربوط به Administrator را به صورت "S-1-5-domain-500" می‌بینید. منظور از domain در اینجا همان طور که گفته شد شناسه مربوطه در حوزه می‌باشد. به طور مثال SID مربوط به Administrator در ویندوز من برابر:

S-1-5-21-746137067-1390067357-1801674531-500

می‌باشد. SID بقیه کاربران هم به همین ترتیب قابل تشخیص است یعنی اگر ویندوز شما فقط به وسیله یک کاربر فراخوانی شده باشد شما می‌توانید خیلی راحت از طریق کلید HKU در رجیستری SID خود را ببینید.

برای برنامه‌نویسان هم یک برنامه که بوسیله آن می‌توانید SID های یک سیستم را از طریق کد به دست آورید در بسته ضمیمه همراه این مقاله وجود دارد.

چگونه می‌توانید با یک بار Logon کردن از امکانات کاربری چندین کاربر استفاده کنید ؟

برای بهتر روشن شدن موضوع مثال زیر را عنوان میکنم.

زمانی بود که میخواستم یکی از برنامه‌های مدیریت سیستم را با کاربری که در گروه Power User قرار داشت اجرا کنم اما ویندوز اجازه اجرای آن برنامه را فقط برای کاربران گروه Admin آزاد گذاشته بود بنابراین, به من که با نام کاربری Javad به ویندوز Logon کرده بودم اجازه اجرای آن را نمی‌داد !
اما, کلید Shift را پائین نگه داشتم و روی آیکن برنامه مورد نظرم راست کلیک کردم و از منویی که ظاهر شد گزینه Run As را انتخاب نمودم سپس کاربر Administrator را از لیست کاربران انتخاب کردم و بعد از وارد کردن کلمه عبور منو را تایید کردم به این ترتیب آن برنامه اجرا شد !

در چنین شرایطی وقتی به سراغ کلید HKU در رجیستری برویم علاوه بر تنظیمات SID خود SID کاربر Administrator را نیز ملاحظه خواهیم کرد. در واقع ویندوز این امکان را فراهم می‌آورد که شما فقط با یک Logon بتوانید از تنظیمات چندین کاربر استفاده کنید.

من از این امکان بهره‌های دیگری نیز برده ام. وقتی روی پروژه برنامه Active Start JSP کار می‌کردم, این برنامه‌ای بود که برای هر کاربر تنظیمات مختص به آن کاربر را ارائه می‌داد یعنی شما می‌توانستید منوها و تنظیمات دلخواه خود را بر روی آن برنامه کاملا مطابق سلیقه خود تعریف کنید و این مجزا از تنظیمات بقیه کاربران بود. بنابراین در زمان برنامه‌نویسی آن مجبور بودم با هر تغییر و اصلاح کد بین حداقل دو کاربر موجود رفت و برگشت کنم اما با روشی که شرح دادم سرعت کار خود را به رقم قابل توجهی افزایش دادم. این تکنیک برای جلوگیری از بروز خطاهای انسانی و ویروس‌های احتمالی نیز کارا می‌باشد. (بسیاری از ویروس ها فعالیت خود را درست زمان Logon آغاز می‌کنند. با این روش می‌توانید تنظیمات کاربری چند کاربر را به راحتی در یک محیط مشاهده و ویرایش نمایید).

البته این نکته را هم خاطر نشان کنم که وضعیت فوق (ظاهر شدن تنظیمات چند کاربر در رجیستری) در شرایط دیگری مانند زمانی که به دو یا چند کاربر Switch می‌کنید نیز به وجود می‌آید.

در حالت عادی یعنی زمانی که شما فقط با یک کاربر در حالت Logon هستید شناسه امنیت SID تنها همین کاربر در شاخه HKU قابل مشاهده است.

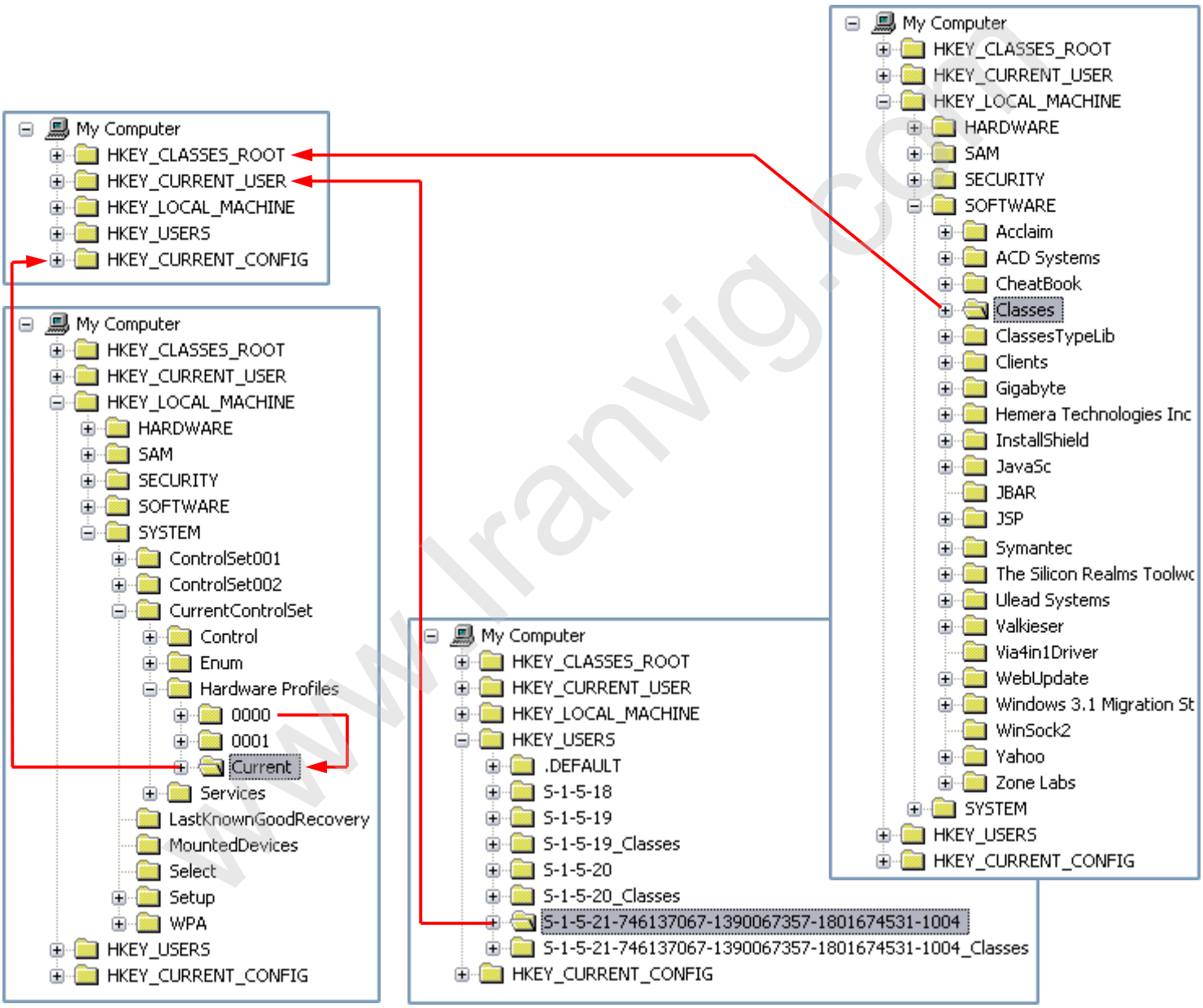
- ✓ **اصطلاح:** کاربر کنسول- کاربری است که در حال کار با صفحه کلید می‌باشد.

بنابراین در یک لحظه فقط یک کاربر کنسول وجود دارد حتی اگر با چندین کاربر Login کرده باشید یا حتی تنظیمات چندین کاربر را فراخوانی کرده باشید.

آیا میدانید رجیستری فقط به دو کلید HKLM و HKU خلاصه می‌شود !

بقیه کلیدهایی که هم ردیف این دو Sub key می‌بینید در واقع نوعی ارتباط (Link) با این دو کلید هستند.

به تصویر توجه کنید. می‌بینید که بقیه کلیدها خود یک کلید یا زیر مجموعه‌ای از دو کلید اصلی نامبرده شده هستند. این تصویر جریان مفصلی از ارتباطات را در رجیستری نمایش می‌دهد.



البته در برخی موارد ویندوز یک کلید را از ادغام چندین کلید دیگر به وجود می‌آورد مانند کلید HKEY_Classes_ROOT که خود ترکیبی از حداقل سه کلید دیگر می‌باشد !

GUID ها شناسه‌های منحصر به فرد عمومی

شناسه‌های منحصر به فرد عمومی بیشتر تحت عنوان GUID (با تلفظ گو آی دی) شناخته می‌شوند.

GUID ها اعدادی هستند که ویندوز به هر چیزی اختصاص می‌دهد. در واقع ویندوز به کمک GUID ها می‌تواند اشیا موجود در صحنه را از هم تمیز دهد. شی‌ها از جمله کامپیوترها, برنامه‌ها, اجزاء درونی و ساخت برنامه‌ها, وسایل سخت افزاری و ... هر کدام GUID منحصر به خود را دارند به عنوان مثال GUID سطل زباله Recycle bin درون همه کامپیوتر ها برابر:

{645FF040-5081-101B-9F08-00AA002F954E}

می‌باشد.

در اینجا فقط مقدمه GUID را بیان می‌کنم در ادامه به طور مفصل این موضوع با اهمیت برای کاربران و حداقل کاوش‌گران را شرح خواهم داد. مثلا شما یاد می‌گیرید که چگونه به کمک همین GUID کوچک آیکن سطل زباله را از روی دسکتاپ بردارید. فراموش نکنید که برای این کار ویندوز هیچ رابط کاربری مهیا نکرده بنابراین تنها راه موجود رجیستری است. پس برنامه‌نویسان بخصوص فصل مربوط به اشیا شل ShellObjects را بخوانند.

فرمت تمام GUID ها یکسان است. GUID ها اعداد ۱۶ بایتی هگزادسیمال هستند که در بلوک‌های 8-4-4-4-12 مرتب می‌شوند و هر بلوک به وسیله خط تیره از بلوک مجاور خود مجزا می‌شود. کل GUID هم بین دو آکولاد باز و بسته قرار می‌گیرد.

یعنی ما ۸+۴+۴+۴+۱۲ معادل ۳۲ خانه داریم که در هر خانه اعداد ۰ تا ۹ و حروف A تا F (با توجه به ۱۶ بایتی بودن GUID ها) را می‌توان نوشت. به این ترتیب فراوانی GUID ها رقمی معادل ۱۶ به توان ۳۲ می‌باشد !!! بیخود نیست که مایکروسافت منحصر به فرد بودن آنها را تضمین کرده است. یعنی امکان ندارد ویندوز درون یک سیستم برای دو شی یک GUID را اختصاص دهد.

GUID ها بسیار مورد علاقه برنامه‌نویسان هستند چون به کمک GUID می‌توانند اجزاء مختلف برنامه‌شان را درون سیستم از هم و بقیه اجزاء سیستم جدا کنند.

برای تولید GUID های مورد نیاز برنامه‌ها می‌توانید از نرم‌افزار Guidgen.exe استفاده کنید. منظورم آن است که خود GUID های مورد نیاز برنامه‌تان را نسازید اجازه دهید این کار به وسیله یک ابزار تولید تصادفی انجام شود. مطمئن باشید GUID تصادفی تولیدی منحصر به فرد خواهد بود.

و در آخر برای غنی تر ساختن شما رگه طلایی را در معدن رجیستری معرفی می‌کنم. مسیر HKEY_CLASSES_ROOT\CLSID را کنکاش کنید.

▪ فصل سوم : پشتیبان گیری کلی از رجیستری

آیا می‌دانید محافظت از رجیستری ۹۰ درصد مشکلات ویندوز را مرتفع می‌سازد ؟

یک سیستم ویندوز همواره به وسیله ویروس‌ها, کرم‌ها, تروجان‌ها و برنامه‌های کاربردی نصب شده یا ناقص نصب شده, اشتباهات فردی افراد مبتدی و یا خراب کاری‌های سهوی, در معرض خطر می‌باشد حال آنکه ویندوزی که ما (یعنی قشر برنامه‌نویس) با آن کار می‌کنیم, به دلیل دفعات آزمون و خطاهایی که روی کدهای برنامه‌مان پیاده می‌کنیم بیش از پیش در معرض خطر قرار دارد. و این یعنی قوز بالا قوز !

پس می‌بینید که یکی از مهمترین کارها تهیه نسخه‌هایی از سیستم در زمان سلامت آن است. همان طور که قبلا هم گفته‌ام رجیستری قلب و روح ویندوز شماست بطوری که می‌توانید فقط با پشتیبان گیری از رجیستری تا ۹۰ درصد مشکلات خود را خیلی راحت برطرف کنید. حال دو راه کار در پیش روی شماست که هر یک در جای خود احتیاج به بحث دارد و آنها پشتیبان گیری کلی و جزئی می‌باشند. در این مقاله روشهای پشتیبان گیری از کل رجیستری سیستم‌تان را فرا می‌گیرید.

رویکرد های متنوعی برای گرفتن پشتیبان کلی وجود دارد. روشی را انتخاب کنید که با آن راحت‌تر هستید.

System Restore کتوله فضا خوار

زمانی که ویندوز خود را نصب می‌کنید موتور عملکرد System Restore روشن است. مگر آنکه از طریق:

Control Panel / System (System Properties) / System Restore

گزینه Turn off System Restore on All Drives را انتخاب کنید.

این سیستم به شما این امکان را می‌دهد تا بدون از دست دادن اطلاعات شخصی ویندوز خود را به وضعیت پیشین برگردانید.

طرز عمل کرد آن به اینگونه است که بر تغییرات برنامه‌ها در سیستم نظارت می‌کند و نقاط احیا از فایل‌های تغییر کرده و بخصوص رجیستری تهیه می‌کند زیرا رجیستری هر لحظه در حال تغییر و تحول است. من نام این نقاط احیا را snapshot می‌گذارم (البته خیلی جاها آن‌ها را با نام restore point می‌شناسند در مقاله‌های آینده علت این نامگذاری snapshot, را شرح خواهم داد).

snapshot ها دستورالعمل‌هایی برای لغو آخرین تغییرات هستند و زمانی احیا می‌شوند که پیکر بندی کامپیوترتان به درستی کار نکند. ویندوز به طور پیش فرض snapshot ها را به هنگام وقوع رویدادهای مهم مانند نصب یک برنامه, ایجاد می‌کند (در این مورد اگر برنامه درست نصب نشود شما می‌توانید تنظیمات سیستمی خود را به زمان قبل از نصب برنامه برگردانید).

پس قبل از انجام هر عمل کنکاش (هک کردن یا کنکاش در اینجا به معنای عام آن استفاده نمی‌شود چرا که شما Registry خود را برای برطرف کردن مشکلات و یا ساختن یک برنامه کاربردی کنکاش می‌کنید) بر روی رجیستری خود تهیه یک نسخه پشتیبان از آن به وسیله System Restore توصیه می‌شود.

✓ **طرز تهیه snapshot به طور دستی:** ابتدا System Restore را با فرمان:

%WinDir%\system32\Restore\rstrui.exe

در منوی Run اجرا کنید. از منویی که ظاهر می‌شود گزینه Create a Restore Point را انتخاب کرده و به مرحله بعد بروید. در این مرحله نیز توضیحی برای snapshot خود نوشته کلید Create را بزنید تا یک snapshot ایجاد شود.

برای احیای snapshot نیز از منوی اول برنامه rstrui.exe گزینه اول که همان احیا می‌باشد را انتخاب کنید در مرحله بعدی تاریخ روزی که snapshot را ساخته بودید و بعد با توجه به نام, توضیحی که روی آن گذاشته بودید snapshot مورد نظر را انتخاب کرده و کلید احیا را بفشارید.

✓ **برنامه‌نویسان بخوانند: طریقه تهیه snapshot به وسیله کدنویسی:**

Object شیئی که به وسیله کد زیر تهیه می‌کنید گنجینه ارزشمندی برای استفاده از System Restore به واسطه کد می‌باشد:

```
Set SysRs = GetObject("winmgmts:\\.\root\default:Systemrestore")
Call SysRs.CreateRestorePoint("Secrets of Registry", 0, 100)
```

شما می‌توانید به جای عبارت "Secrets of Registry" از نام, توضیحات مورد نظر خودتان استفاده کنید.

لازم به ذکر است System Restore به حداقل ۲۰۰ مگابایت از فضای دیسک سخت شما احتیاج دارد تا فعال باشد و به طور معمول ۱۴ درصد هر یک ازدرایوهای دیسک سخت کامپیوتر شما را اشغال می‌کند. من با مدیران IT زیادی روبرو شده ام که System Restore را به عنوان یک ابزار کوچک ولی پرهزینه (فضا خوار) نام برده‌اند. آنها به محض نصب هر سیستمی اولین کاری که می‌کنند از کار انداختن System Restore می‌باشد تا از فضاهای محدود دیسک سخت در کامپیوترهای محل کارشان بهره بیشتری ببرند !

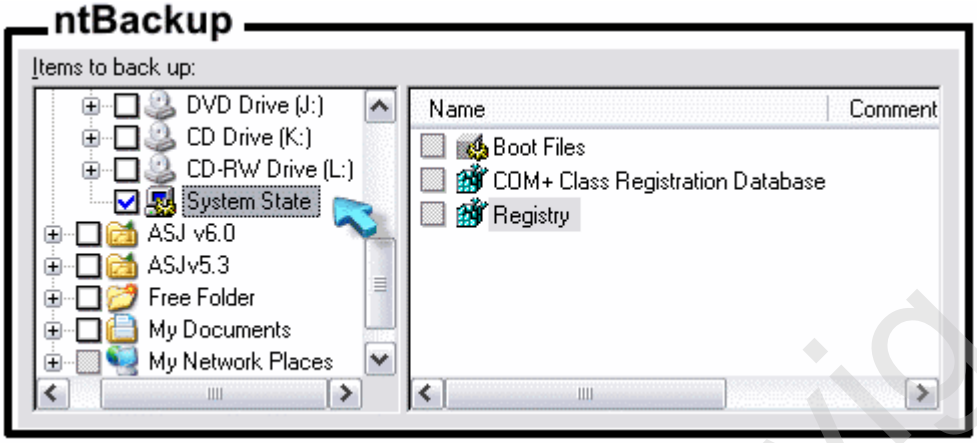
حال اینکه System Restore اطلاعات رجیستری را کجا ذخیره می‌کند و شما چگونه می‌توانید از اطلاعات نقاط احیا بدون احیای کل سیستم استفاده کنید بحثی است که در مقاله‌های بعدی به آن خواهم پرداخت.

توضیحات بیشتر در رابطه با System Restore و طرز عملکرد آن در این مقال نمی‌گنجد. در صورت تمایل برای آشنایی با ریزه کاری‌های System Restore و یا به قول خودمان گفتنی "اسرار System Restore " موضوع را با من در میان بگذارید تا مقاله‌ای مفصل در این باره تحریر کنم که شامل مطالبی است که حتی به ذهن‌تان هم خطور نمی‌کند.

پشتیبان گیری منظم از رجیستری

در این بخش برنامه خدماتی Backup ویندوز را معرفی می‌کنم. با اجرای فرمان ntBackup در منوی Run برنامه‌ای اجرا می‌شود که علاوه بر اینکه می‌توانید به وسیله آن از داده‌های روی دیسک سخت‌تان نسخه پشتیبان تهیه کنید, می‌توانید یک برنامه زمانی برای ایجاد فایل‌های پشتیبان تهیه کرده و با خیالی آسوده شب‌ها چشمان‌تان را روی هم بگذارید !

مزیت استفاده از این برنامه در آن است که شما می‌توانید حتی از فایل‌های باز که به وسیله ویندوز قفل شده‌اند (مانند فایل‌های Hive رجیستری) هم نسخه پشتیبان تهیه کنید.



همان طور که در تصویر می‌بینید در این برنامه یک گزینه مجزا از بقیه درایوها و اطلاعات با عنوان System State در اختیار شماست که بوسیله آن نه تنها از اطلاعات رجیستری بلکه از اطلاعات فایل‌های Boot و اشیا Com هم نسخه‌های پشتیبانی تهیه می‌شود. به این ترتیب شما می‌توانید در مواقع لزوم اطلاعات سیستمی خود را احیا کنید یا با باز کردن فایل‌های Backup قسمت‌هایی از رجیستری که فکر می‌کنید دچار آسیب شده است را ترمیم نمایید.

یک نکته مهم که نباید فراموش کنید آن است که با انتخاب گزینه System State تنها از تنظیمات سیستمی خاص ماشین (منظور تنظیماتی است که مربوط به همه کاربران و سیستم می‌شود) پشتیبان گرفته می‌شود و تنظیمات خاص هر کاربر را شامل نمی‌شود. به این منظور باید از طریق فایل‌های Hive تنظیمات هر کاربر, آنها را نیز در فرایند پشتیبان گیری داخل کنید. فایلی که تنظیمات خاص هر کاربر را نگه داری می‌کند:

%UserProfile%\ntuser.dat

و اطلاعات کلاس مربوط به کاربر هم در فایل زیر قرار دارد:

%UserProfile%\Local Settings\Application Data\Microsoft\Windows\UsrClass.dat

با این وجود ntBackup از همه چیز پشتیبان تهیه نمی‌کند کلید:

HKLM\SYSTEM\CurrentControlSet\Control\BackupRestore

دارای دو زیر کلید به نام های FilesNotToBackup و KeysNotToRestore است که در اولی مسیرهایی که از آنها نسخه پشتیبان تهیه نمی‌شود و دومی حاوی فهرست کلیدهایی از رجیستری است که نباید احیا شوند. شما می‌توانید مسیر یا کلیدهای دیگری را به/از این مجموعه اضافه/کم کنید.

اگر میخواهید پشتیبان گیری شما به صورت زمان بندی شده اعمال گردد وقتی فرمان ntBackup را اجرا می‌کنید در همان صفحه اول منوی ظاهر شده, گزینه متنی Advanced Mode را انتخاب کنید تا بتوانید یک برنامه زمانی برای این منظور تحریر کنید.

"من یکی از دوستاران برنامه Symantec Ghost هستم زیرا تنها ابزاری است که برای توزیع ویندوز در محیط های بزرگتر ترجیح می‌دهم. ابتدا ویندوز سپس تمام درایورهای سخت افزاری همین طور برنامه‌های کاربردی و امنیتی یک سیستم را بر روی آن نصب می‌کنم و بعد یک دیسک توزیع به وسیله برنامه Ghost تهیه می‌کنم. به این ترتیب در محل کارم وقتی سیستم خودم یا همکارم دچار مشکل (هر چند عمیق و اساسی) شود اوضاع سیستم را به وسیله یک دیسک توزیع ظرف ۱۵ دقیقه به حالت اولیه‌اش (زمان سلامت آن) باز می‌گردانم."

در فصل بعد بحث مهم‌تری با عنوان پشتیبان گیری جزئی از رجیستری را مورد بررسی قرار می‌دهیم.

▪ فصل چهارم : بار گذاری فایل‌های Hive

چگونه می‌توان فایل‌های Hive شخصی ایجاد و در یک سیستم دیگر بار گذاری کرد ؟

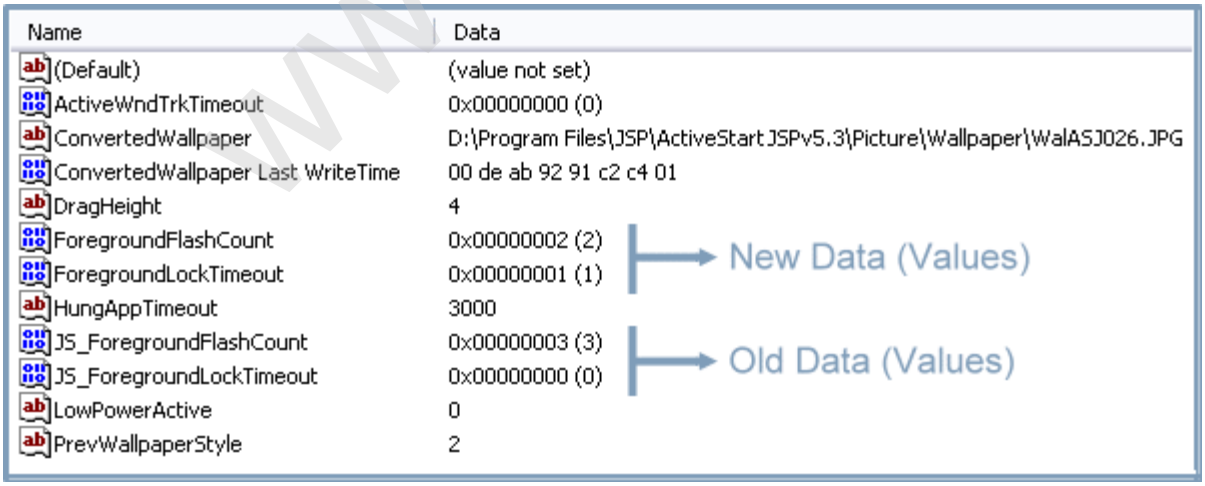
قبل از مطالعه این مقاله باید فصل اول "فایل‌های Hive" را خوانده باشید زیرا در اینجا لازم است فایل‌های Hive را بشناسید. به شما پیشنهاد می‌کنم یک پشتیبان کلی هم از رجیستری سیستم‌تان تهیه نمایید تا با خیال آسوده تمرینات و مثال های این فصل و این مقاله را دنبال کنید.

➢ روشهای پشتیبان گیری جزئی

در فصل قبل گفتیم که چطور می‌توان از کل رجیستری پشتیبان تهیه کرد. در نظر داشته باشید رویکردهای پشتیبان گیری کلی همه جا مقرون به صرفه نیستند. مثلا ممکن است شما زمان و حتی فضای حافظه کافی برای پشتیبان گیری کلی نداشته باشد به این ترتیب حتما قبل از دست کاری دادهای رجیستری سیستم خود, ابتدا به کمک روش‌هایی که در زیر معرفی می‌شود از حداقل داده‌ای که روی آن کار می‌کنید و یا کلید مربوطه پشتیبان جزئی تهیه کنید.

❖ تغییر نام مقادیر, کوتاهترین و سریع ترین راه

در اینجا با یک مثال مطلب فوق را عنوان می‌کنم. به تصویر توجه کنید که محتوای کلید HKCU\Control Panel\Desktop سیستم خود من را نشان می‌دهد.



ر این روش ساده اما کارا شما باید نام کلیدها یا مقادیر (Values) هایی را که می‌خواهید دست کاری کنید طوری تغییر دهید که اگر تغییرات شما باعث شد که اتفاق ناخوشایندی برای سیستم‌تان بی‌افتد خیلی راحت بتوانید تنظیمات قبلی را به حالت اولیه‌اش باز گردانید.

منطق حکم می‌کند که نام ها را به طور کامل عوض نکنید. مثلا من در این مورد داده ForegroundFlashCount را به JS_ForegroundFlashCount تغییر نام داده‌ام سپس دوباره داده را ایجاد کردم و مقدار جدید را به آن اختصاص داده‌ام. در اینجا من از

حروف اول نام خود در ابتدای آن استفاده کردم JS=Javad Soltani به این ترتیب هر جای دیگری در رجیستری سیستم که به JS_ برخورد کنم می‌دانم که آن قبلا یک داده با ارزش بوده.

- ✓ نکته اینجاست که ویندوز و تقریبا همه برنامه‌های کاربردی فقط داده‌ای را که با نام مورد نظر خودشان مطابقت دارد می‌خوانند. از این شیوه برای حذف کردن مقادیر نیز استفاده کنید. یعنی به جای اینکه داده‌ای را حذف کنید کافی است نام آن را تغییر دهید. البته برنامه‌های کاربردی که به درستی کار می‌کنند وقتی داده مورد مدیریت آنها از رجیستری حذف شود دوباره آن را ایجاد می‌کنند (با تغییر نام یک داده آن را از دید برنامه ها مخفی می‌کنید یا باعث می‌شوید دوباره با مقدار پیش‌فرض ایجاد شود).

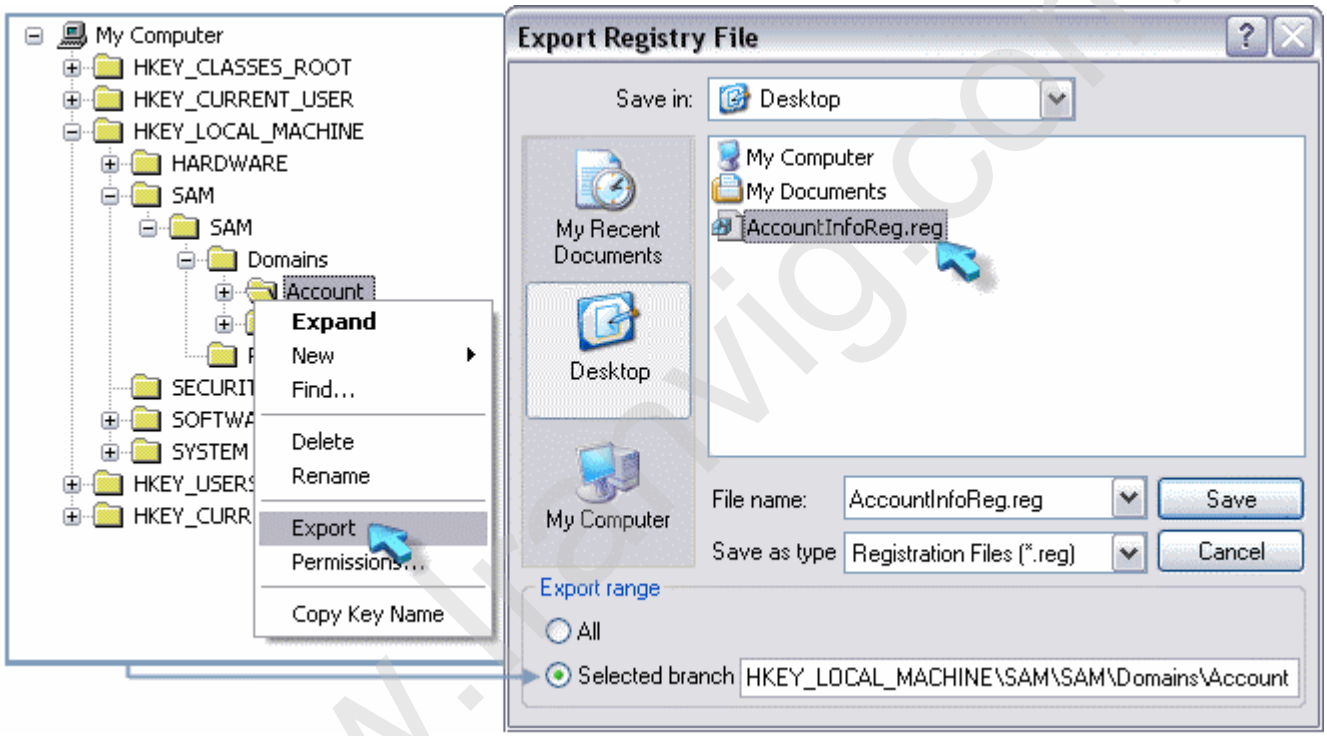
حال اگر تغییرات شما بر روی مقادیر روزانه یا لحظه‌ای می‌باشد می‌توانید علاوه بر اضافه کردن مخفف نام‌تان, تاریخ یا ساعت تغییرات را هم به ابتدای مقادیر پشتیبان اضافه کنید.

تاکید من بر اضافه کردن به اول مقادیر از آن جهت است که بعضی برنامه ها مقادیرشان را به صورت کد تعداد از رجیستری می‌خوانند مثلا من برنامه‌ای به نام Active Start JSP نوشته بودم که برای حفظ امنیت ورود و خروج اطلاعات به جای اینکه به دنبال مثلا مقدار Tnt در رجیستری بگردد, تمام مقادیری را که سه حرف اولشان Tnt بود می‌خواند. پس اگر من از داده Tnt به روش فوق پشتیبان نمی‌گرفتم برنامه‌ام همه داده‌ها را تغییر می‌داد.

باید اعتراف کنم که خود من در خصوص عمل کردن به توصیه‌های خودم چندان دقیق نیستم. فراموش کردن تهیه پشتیبان به روش فوق از مقادیر بیش از انجام این کار آسان است. اما چگونه می‌توان تشخیص داد که یک سری تغییرات ساده باعث غرق شدن کشتی نمی‌شود ؟!؟! یقینا نمی‌توان. بنابراین باید براساس گفته‌های من عمل کنید نه عمل کردم (بیش از تغییر یا حذف مقادیر از آنها پشتیبان گیری کنید).

❖ صادر کردن Export کلیدها به فایل‌های Reg

برای این کار کافیست روی کلید مورد نظرتان راست کلیک کرده و گزینه Export را انتخاب کنید. به این ترتیب منویی ظاهر می‌شود که فرمت پیش فرض برای صادر کردن آن کلید را فرمت Reg قرار داده (در قسمت Save as type منو عبارت Registration Files *.reg را ملاحظه کنید). برای فایل خروجی نامی را وارد کنید و منو را تایید کنید. در صورتی که بخواهید تمام رجیستری خود را به فایل صادر کنید گزینه رادیویی All را در کادر Export range علامت بزنید.



شما می‌توانید در محیط ویندوز اکسپلورر با راست کلیک روی فایل‌های Reg و انتخاب گزینه Edit محتوای آنها را ببینید و یا ویرایش کنید.

❖ صادر کردن Export کلیدها به فایل‌های Hive

صادر کردن به فایل‌های Hive مشابه فایل‌های Reg است با این تفاوت که در قسمت Save as type منوی Export عبارت:

Registry Hive Files (*.*)

را انتخاب می‌کنید و می‌توانید هر پسوندی برای فایل خود بگذارید یا حتی فایل را بدون پسوند ذخیره کنید. من شخصا پسوند hive. را ترجیح می‌دهم. در هر حال شما قادر به ویرایش فایل Hive ی که ساخته‌اید نخواهید بود (به فصل اول : فایل‌های Hive مراجعه کنید).

❖ وارد کردن Import فایل‌های Reg و Hive

مزیت فایل‌های Reg در آن است که با اجرای یک فایل Reg ویندوز متوجه می‌شود که شما می‌خواهید اطلاعات موجود در آن فایل را در رجیستری وارد کنید به این ترتیب بعد از دوبل کلیک (اجرا کردن فایل Reg) ویندوز از شما سوال می‌کند که آیا مایلید تغییرات اعمال شوند ؟! ... و آن فایل را وارد می‌کند.

به غیر از اجرای فایل‌های Reg می‌توانید از منوی File ویرایشگر رجیستری خود گزینه Import را انتخاب کنید و به این ترتیب نیز فایل Reg مورد نظرتان را در رجیستری وارد کنید. برای وارد کردن فایل‌های Hive هم جز Import کردن آنها از منوی File برنامه Regedit چاره دیگری ندارید.

تفاوت فایل‌های Reg با Hive

فایل‌های Hive از نظر ساختار قابل اطمینان‌ترند زیرا شما میدانید فایل Hive که از یک سیستم دیگر به دست‌تان رسیده دست کاری نشده و با خیالی آسوده آن را Import می‌کنید.

- ✓ وقتی یک فایل Reg را به سیستم‌تان وارد می‌کنید Regedit به جای جایگزینی تنظیمات موجود در فایل Reg آنها را با تنظیمات موجود در رجیستری ادغام می‌کند. این بدان معناست که Regedit مقادیر موجود در فایل Reg را ایجاد یا جایگزین می‌کند ولی مقادیری که در فایل Reg موجود نیستند اما در رجیستری وجود دارند حذف نمی‌شوند !. اما فایل‌های Hive اینطور نیستند یعنی به محض وارد کردن آنها Regedit کلیدها و مقادیر موجود در فایل Hive را به طور کلی از رجیستری پاک کرده و مقادریر جدید موجود در فایل Hive را جایگزین آنها می‌کند (همین یک دلیل کافیست که من تمام پشتیبان‌هایم را به صورت فایل Hive ذخیره کنم).
- ✓ روش‌های فوق (Import یا Export بخشهایی از رجیستری) به منظور پشتیبان گیری از راه دور هم کار می‌کنند. شما می‌توانید با انتخاب گزینه Connect Network Registry از منوی فایل File ویرایشگر رجیستری خود به یک کامپیوتر در حوزه شبکه و دسترسی خود, متصل شوید و محتوای رجیستری آن سیستم را در ویرایشگر خود ملاحظه کنید ...
- ✓ برنامه‌نویسان هم می‌توانند خیلی راحت به وسیله فایل Reg.exe کارهای بزرگی در رجیستری انجام دهند. مثلا صادر کردن یا وارد کردن یک فایل Reg یا Hive فقط به وسیله یک خط کد. البته کار با فایل Reg.exe کمی خطرناک است. پس ماجراجویی نکنید اجازه دهید در مقاله‌ای مخصوص راجع به این فایل مهم همه چیز را توضیح خواهم داد.

طریقه بار گذاری فایل‌های Load Hive

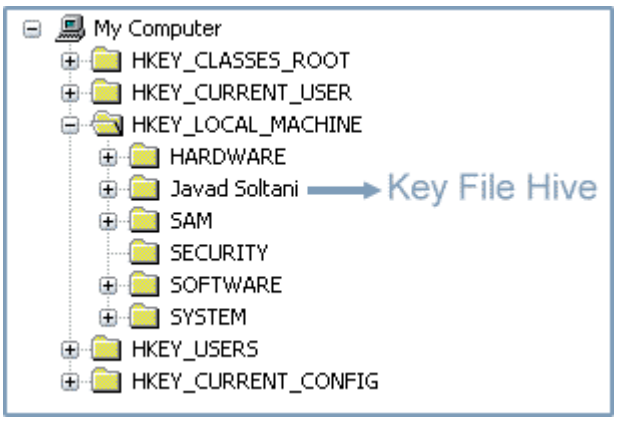
توجه کنید: بار گذاری Load کردن با وارد کردن Import فایل‌های Hive دو مقوله کاملا مجزا از هم هستند (Import کردن را توضیح دادم).

وقتی شما یک فایل Hive را بار گذاری می‌کنید ویراستار رجیستری شاخه جدیدی را ایجاد می‌کند و آن را به شما نشان می‌دهد, که حاوی محتویات فایل Load شده است.

برای بار گذاری فایل Hive:

۱. یکی از دو کلید اصلی HKLM یا HKU را در محیط Regedit انتخاب کنید.
۲. از منوی File گزینه Load Hive که هم اکنون فعال شده است را انتخاب کنید.
۳. فایل Hive مورد نظر خود را Open کنید (فرقی نمی‌کند که آن را شما ایجاد کرده‌اید یا اینکه مثلا یکی از فایل‌های Hive اصلی یک سیستم است).
۴. نامی را برای کلیدی که قرار است محتویات فایل Hive انتخابی شما, در آن به نمایش در بیاید را در منوی مورد درخواست بنویسید و آن را تایید کنید.

به این ترتیب خواهید دید که Regedit کلیدی به Sub key انتخابی شما اضافه می‌کند که اگر آن کلید را باز کنید می‌توانید محتویات فایل Hive انتخابی‌تان را در آنجا ببینید.



ویندوز از کلید جدیدی که شما ایجاد کرده‌اید هیچ استفاده‌ای نمی‌کند. اما فایل Hive که بار گذاری شده توسط سیستم قفل شده و به عنوان یکی از فایل‌های Hive اصلی شناخته می‌شود (یعنی دیگر قادر به انجام هیچ عمل کنترلی بر روی فایل Hive بار گذاری شده نخواهید بود).

به وسیله بار گذاری فایل‌های Hive می‌توانید گره‌های کور زیادی را از سیستم خود یا یک کامپیوتر بومی باز کنید. زمانی که شما یک فایل Hive را Import می‌کردید باید بی چون و چرا تنظیمات‌تان را به زمان ساخت آن فایل پشتیبان Hive برمی‌گرداندید, اما در Load کردن با ایجاد یک کلید جدید در کنار بقیه کلیدها می‌توانید مقادیر حال و گذشته, مقادیر رجیستری سیستم خود با یک سیستم دیگر را ببینید همین طور صحت داده‌ها را مقایسه کرده و به این ترتیب مشکلات را بدون خرابکاری مضاعف حل و فصل کنید.

اگر چه وارد کردن Import یک فایل Hive روش خوبی برای احیای یک شاخه است اما بارگذاری Load کردن یک فایل Hive روش مناسبی برای احیای آزمایشی تنظیمات یا صرفا بررسی و مقایسه یک مقدار است.

- ✓ **یک نکته بسیار مهم**: فراموش نکنید قبل از بستن ویراستار خود حتما فایل Hive بار گذاری Load شده را خارج یا Unload کنید در غیر این صورت ممکن است دیگر قادر به Unload کردن آن نباشید و به این ترتیب یک رجیستری مثلا با "سه گوش" یا "چند چشم" خواهید داشت. برای UnLoad کردن فایل Hive کافی است روی کلیدی که به تازگی ایجاد شده و حاوی فایل Hive است کلیک کرده و گزینه UnLoad Hive را از منوی File انتخاب کنید.

تاکنون مطالبی که در این مقاله "اسرار رجیستری" ارائه شد برای آشنایی کاربران در سطوح متوسط و پیشرفته, از ناگفته‌های رجیستری بود. از فصول بعدی وارد مباحث عملی کار می‌شویم. شما برنامه‌نویسان بهترین بهره را از این مطالب خواهید برد چرا که در آینده‌ای نه چندان دور خواهید توانست به وسیله برنامه‌ای که می‌نویسید و با کمک اسرار رجیستری برنامه‌هایی تولید کنید که قدرت بیشتری دارند و با ویندوز بهتر ارتباط برقرار می‌کنند. بنابراین مطالب ارائه شده در این ۴ فصل را با دقت بخوانید.

▪ فصل پنجم: ساختار برنامه‌های کاربردی در رجیستری

یک نرم‌افزار در کجای رجیستری می‌تواند داده ثبت و ضبط نمایید ؟

هنگامی که صحبت از تنظیمات یک برنامه به میان می‌آید باید گفت ساده‌ترین, ایمن‌ترین و شاید منطقی‌ترین راه نگه‌داری و ثبت تنظیمات یک برنامه همان رجیستری است. کمتر برنامه‌نویسی پیدا می‌شود که همه داده‌های ورودی و خروجی قابل ذخیره را به طور کلی در رجیستری ثبت کند. همان طور که گفته شد گر چند که رجیستری در حالت متداول محلی برای نگه داری تنها تنظیمات برنامه‌ها است نه همه داده‌های آنها اما تصمیم این موضوع بر عهده خود برنامه ساز می‌باشد. خود من روش ذخیره همه داده‌ها در رجیستری را در مواقع خاصی ترجیح می‌دهم زیرا حتی در صورت Uninstall کردن و یا حذف کامل آن برنامه از روی دیسک سخت, در صورت نصب مجدد برنامه, داده‌هایی در رجیستری توسط برنامه تازه نصب شده کشف می‌شوند که به روند نصب و اجرای مجدد آن کمک زیادی می‌کنند. مصداق بارز این مطلب قفل های نرم‌افزاری است.

در طرف مقابل ادعای فوق برنامه‌سازانی هستند که هیچ رابطه شفافی با رجیستری ندارند. گر چند که در فرای لایه‌های اجرا و کامپایل برنامه‌هایشان شرایط کار را طوری ترتیب می‌یابد که فایل اجرایی حاصله رابطه مستقیم و بی چون و چرایی با رجیستری سیستم عامل اجرا داشته باشد و این کاملا دور از چشمان برنامه‌نویس و مجزا از کدهایی است که برنامه‌ساز تحریر کرده است. اما توصیه برای همه برنامه‌سازان ثبت حداقل, تنظیمات برنامه در رجیستری است. اما یک نرم‌افزار در کجای رجیستری می‌تواند داده ثبت و ضبط کند ؟

➢ ساختار یک نرم‌افزار در رجیستری

بیشتر نرم‌افزارها داده‌های خود را به یک شکل در رجیستری ذخیره می‌کند با قالب :

HKCU\Software\Company\Program\Version\(settings data)

آموزنده فرمت فوق پدید آورند ویندوز یعنی مایکروسافت است. کافی است کلید HKCU\Software\Microsoft را باز کنید تا ببینید نرم‌افزارهای تولیدی این شرکت با چه نظم و ترتیبی حتی از ده سال پیش تاکنون در این محل ثبت تنظیمات شده‌اند.

در فرمت فوق داریم Company که نام پدید آورنده برنامه است. Program که نام خود نرم‌افزار می‌باشد. Version که شماره نگارش اختیاری آن برنامه است و تنظیمات نیز در زیر شاخه‌های همین مسیر قرار می‌گیرند. در واقع از این قسمت به بعد بر عهده خود برنامه‌نویس است. در نظر داشته باشید که ویندوز به اینکه شما یک داده برای ثبت یک تنظیم از برنامه‌تان را با چه فرمت, نام و حتی اینکه در کجای رجیستری ذخیره می‌کنید, دخالتی نمی‌کند چرا که ویندوز هیچ کاری با داده ثبتی شما ندارد و آن داده فقط برای برنامه‌ای که ثبتش کرده ارزشمند است.

فراموش نکنید مسیر HKCU\Software محلی برای نگه داری آن دسته از تنظیمات برنامه شما است که به هر کاربر مربوط می‌شود نه به همه کاربران یا ماشین (به عبارت HKEY_CURRENT_USER در مسیر توجه کنید). توضیح دادم که برای ثبت داده‌ها در هر جای رجیستری آزاد می‌باشید اما رعایت اصول کار باعث افزایش کیفیت می‌شود و نگاه‌ها را به نرم‌افزار شما مانند یک برنامه حرفه‌ای تغییر می‌دهد.

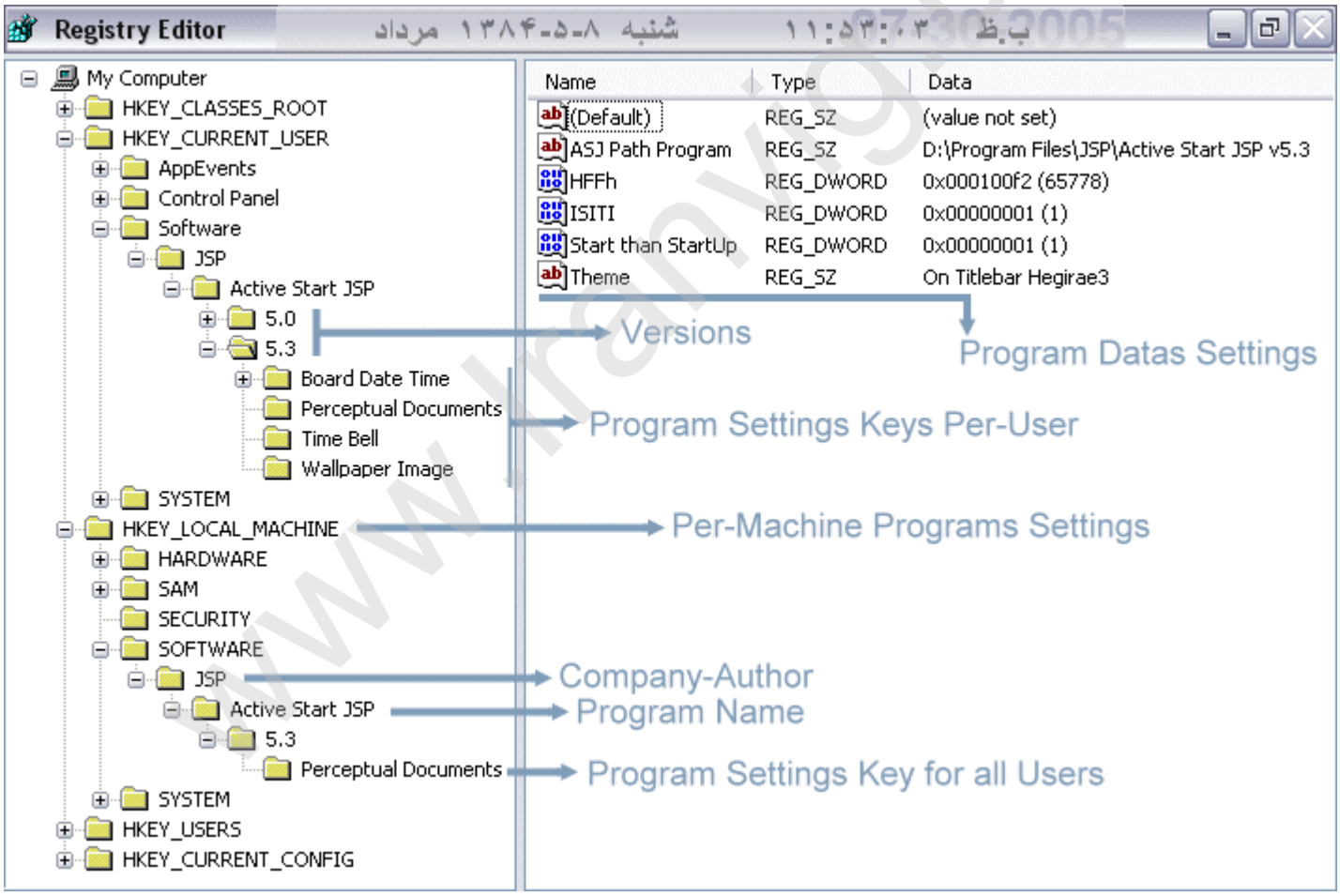
پس اگر خواستید تنظیمی از برنامه‌تان را در رجیستری ثبت کنید که مربوط به همه کاربران و یا خود ماشین بود قالب را باید به شکل زیر تغییر دهید:

HKLM\Software\Company\Program\Version\(settings data)

به عبارت HKEY_LOCAL_MACHINE در مسیر توجه کنید.

در رابطه با Version به این دلیل از کلمه "نگارش اختیاری" استفاده کردم, که بعضی از افراد شماره نگارش برنامه را که چندان مهم نیست از این قالب حذف می‌کنند اما استفاده از آن در هر صورت متداول است.

تصویر زیر تنظیمات برنامه Active Start JSP که توسط خود من ساخته شده در رجیستری را نشان می‌دهد.



تفاوت ذخیره تنظیمات در HKCU با HKLM

همان طور که در تصویر قبل دیدید کلید "Perceptual Documents" در برنامه من دو بار یکی در HKCU و دیگری در HKLM ثبت شده. Perceptual Documents قسمتی بود که کاربر در آن مشخصات اسناد دریافتی‌اش (چک‌هایش) را وارد می‌کرد و برنامه با توجه به تاریخ سررسید آن اسناد آلارم‌هایی برای کاربر صادر می‌کرد و کاربر را از اوضاع مطلع می‌نمود. اما هر کاربری می‌تواند برای سندی که خود ایجاد کرده این تنظیم را انجام دهد که سند فوق یک سند خصوصی باشد یا قابل مشاهده برای همه کاربران. برای آین قسمت من هیچ کار اضافی انجام ندادم فقط اسناد خصوصی کاربر را در کلید HKCU آن کاربر ثبت می‌کردم و در مقابل اسنادی که قابل مشاهده و ویرایش برای همه کاربران بودند را در کلید HKLM ثبت کردم.

در واقع وقتی یک کاربر برنامه را اجرا می‌کرد, برنامه فقط کلید HLCU مربوط به همین کاربر را می‌خواند و اگر داده‌ای در آن ثبت شده بود آن را به عنوان سند خصوصی آن کاربر در نظر می‌گرفت. همین طور با خواندن داده‌های کلید HKLM اسناد موجود در آن را برای همه کاربران نمایش می‌دهد.

امیدوارم با مثال بالا برای شما روشن شده باشد که, با توجه به اینکه هر کاربر دارای یک کلید HKCU مختص به خود است وقتی با یک اکانت Login می‌کنید و داده‌ای را در کلید HKCU آن کاربر ذخیره می‌کنید برای بازیابی آن داده فقط باید با آن اکانت در حالت Login باشید.

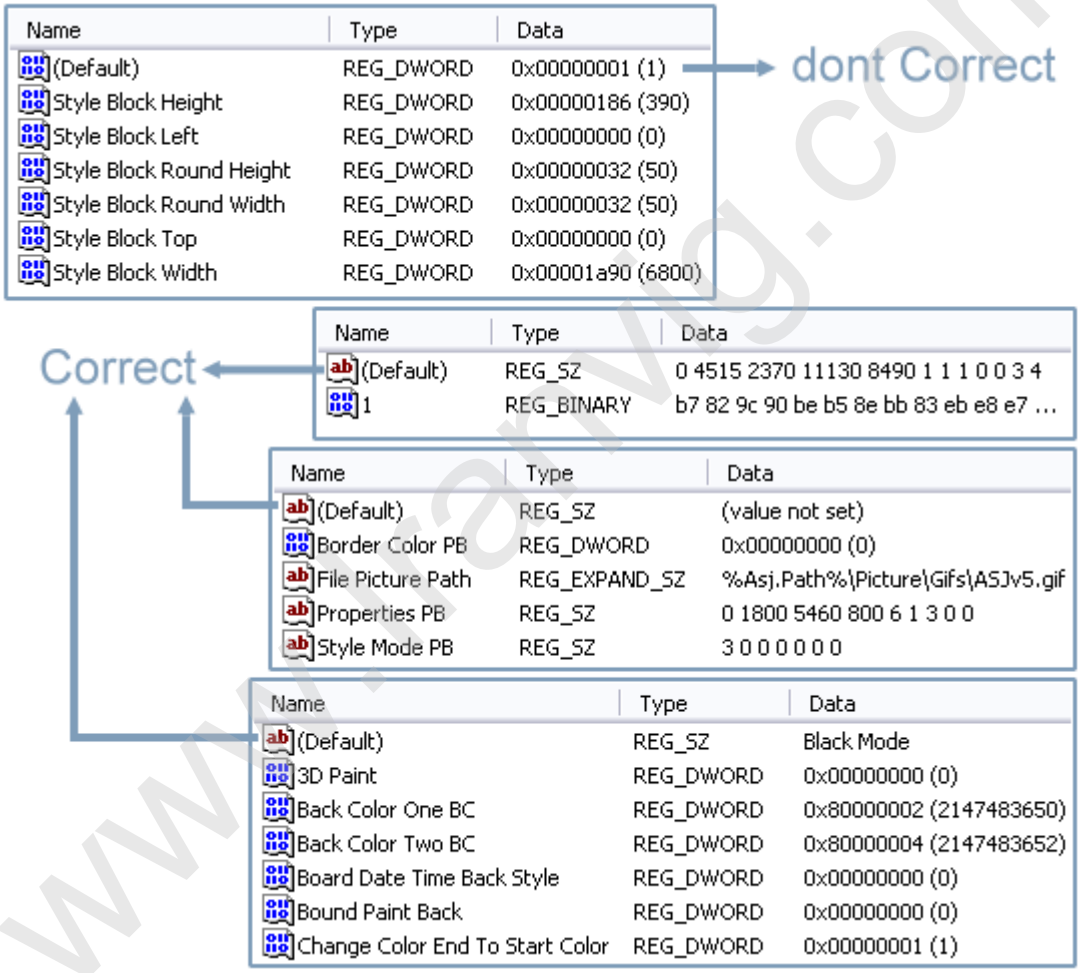
- ✓ برنامه‌هایی که به خوبی طراحی شده‌اند, تنظیمات حذف شده را از نو و اغلب با مقادیر پیش فرض, دوباره ایجاد می‌کنند. برای اینکه تنظیمات خاص کاربر یک برنامه را ری‌ست کنید کافی است کلید تنظیمات خاص-کاربر HKCU\Software\Company\Program برنامه را از رجیستری حذف کنید. پس شما برنامه‌نویسان برنامه‌هایتان همیشه باید گوش به زنگ باشند که اگر داده‌هایشان ار رجیستری حذف شد بتوانند آنها را دوباره ایجاد کنند. البته گاه با برنامه‌هایی رو به رو می‌شویم که با حذف کلید HKCU\Software\Company\Program از کار می‌افتند و در بعضی موارد نصب دوباره هم چاره ساز نیست و این موضوع واقعاً برای برنامه سازان آن نرم‌افزار جای تاسف دارد ! عموما نباید تنظیمات خاص-کامپیوتر HKLM\Software\Company\Program برنامه‌ها را حذف کنید زیرا این امر ممکن است بر عملکرد بسیاری از برنامه‌های کاربردی تاثیر بگذارد. برای اینکه حاشیه امنیت مناسبی برای خود تدارک ببینید تنظیمات خاص-کامپیوتر برنامه‌ها را جهت انجام آزمایش‌های لازم مخفی کنید (همان طور که قبلا گفته شده, برای مخفی کردن یک داده یا کلید آن را تغییر نام دهید).

- ✓ ری استارت کردن برنامه‌های کاربردی مبتنی بر ویندوز Installer آسان‌تر است. چرا که Installer دارای قابلیت ترمیم تو کار است. مجموعه آفیس مایکروسافت مثالی از این نوع می‌باشد که در آن یک قطعه از ابزار آلات را در حین کار Uninstall و یا نصب و حتی ReInstall کنید.

کار با داده Default در هر کلید

همان طور که می‌دانید با ایجاد هر کلید در رجیستری حال در محیط Regedit و یا به وسیله کدنویسی, در آن کلید یک داده با نام Default ایجاد می‌شود که از نوع Reg_SZ بوده و محتوایش تهی می‌باشد. البته خود نام Default هم نمایشی است و درون کدنویسی برای فراخوانی و یا انجام عملیات بر روی این داده باید آن را با نام تهی (هیچ یا پوچ) صدا بزنید.

شما می‌توانید هر داده‌ای را که مایل هستید در Default بنویسید و از آن در هر جایی که لازم می‌بینید استفاده کنید اما هرگز فرمت Type آن را تغییر ندهید.



برنامه‌هایی که به درستی کار نمی‌کنند می‌توانند فرمت این داده را عوض کنند اما در موارد خاصی دیده شده که خود ویندوز در برخی قسمت‌ها مانند HKLM\SAM\SAM\Domains\Builtin\Aliases فقط برای ذخیره داده‌های امنیتی, دست به تغییر فرمت داده Default زده است.

✓ وقتی یک فایل Reg را باز می‌کنید کلیه داده‌های Default کلیدها نه با این نام بلکه با علامت @ در آن فایل‌ها ذخیره شده است.

توابع API برقراری ارتباط با رجیستری

قبلا دوستان من در سایت ایرانویج www.Iranvig.com اقدام به معرفی توابع API ارتباط با رجیستری کرده‌اند اما به طور گذرا فقط آنها را معرفی می‌کنم. برای اطلاعات بیشتر در باره آنها به مقالات دوستان در همین سایت مراجعه کنید.

به وسیله توابع API که در زیر لیست کرده‌ام می‌توانید از هر جایی با رجیستری ارتباط برقرار کنید اما مسئله که در این میان وجود دارد پر هزینه بودن این توابع است (پر هزینه در اصطلاح برنامه‌نویسان در معنای مالی آن به طور مستقیم کاربرد ندارد. در واقع با افزایش تعداد خطوط کد موجب افزایش زمان برنامه‌نویسی همین طور مصرف بیشتر حافظه پویا شده و در بازه‌های زمانی بلند و حتی کوتاه مدت موجب صرف انرژی بیش از حد می‌شود و این خود با مصرف بیشتر منابع مالی در یک نرم‌افزار باعث پرهزینه شدن پروژه و در مواردی به شکست پروژه می انجامد).

```
RegCloseKey Lib "advapi32.dll"

RegCreateKey Lib "advapi32.dll" Alias "RegCreateKeyA "

RegCreateKeyEx Lib "advapi32.dll" Alias "RegCreateKeyExA"

RegDeleteKey Lib "advapi32.dll" Alias "RegDeleteKeyA"

RegDeleteValue Lib "advapi32.dll" Alias "RegDeleteValueA"

RegEnumKey Lib "advapi32.dll" Alias "RegEnumKeyA"

RegEnumValue Lib "advapi32.dll" Alias "RegEnumValueA"

RegOpenKey Lib "advapi32.dll" Alias "RegOpenKeyA "

RegOpenKeyEx Lib "advapi32.dll" Alias "RegOpenKeyExA"

RegQueryValue Lib "advapi32.dll" Alias "RegQueryValueA "

RegQueryValueEx Lib "advapi32.dll" Alias "RegQueryValueExA "

RegSetValue Lib "advapi32.dll" Alias "RegSetValueA "

RegSetValueEx Lib "advapi32.dll" Alias "RegSetValueExA "
```

توابع بهینه برقراری ارتباط یک برنامه با رجیستری

با توجه به پرهزینه بودن توابع API چند سالی است که وی آنها را به شکل چند تابع مفید, مختصر و بهینه در آورده‌ام و به طوری که به جای ده‌ها خط کد برنامه‌نویسی با یک خط کد می‌توانید آنچه را که ار رجیستری می‌خواهید محقق سازید. از آنجایی که تمام آن چیزی که در کدهای کتابخانه می‌بینید همه و همه حاصل تلاش وی است استفاده از آنها در هر جایی فقط با عنوان پدید آورنده آن مجاز می‌باشد.

اول از هر چیز خاطر نشان می‌شوم این بخش از توابع در محیط VB6.0 ویرایش شده‌اند. با این وجود آنها را به صورت یک کتابخانه Dll برای استفاده دیگر زبان‌ها در آورده‌ام. در ضمن سورس کد این فایل نیز به همراه این مقاله در بسته ضمیمه موجود است.

راجع به تمام توابع موجود در این کتابخانه به طور کامل در سند Html که همراه Pack فایل دریافت می‌کنید توضیح داده‌ام. ممکن است در آن با مواردی روبه‌رو شوید که قبلا راجع به آنها صحبت نشده مانند ACL یا کلیدهای اصلی در ویندوزهایی به جزء اکس پی. اما به زودی درباره همه آنها در مقالات جدید همه چیز را شرح خواهم داد.

در این مقاله تنها راه توابع API و توابع بهینه برای ارتباط یک برنامه با رجیستری صحبت شد در مقالات بعدی راه‌های ساده‌تر و جدیدی را معرفی خواهم کرد.

▪ فصل ششم : اشیاء شل ShellObjects

چگونه می‌توان یک دسکتاپ سفارشی داشت ؟!

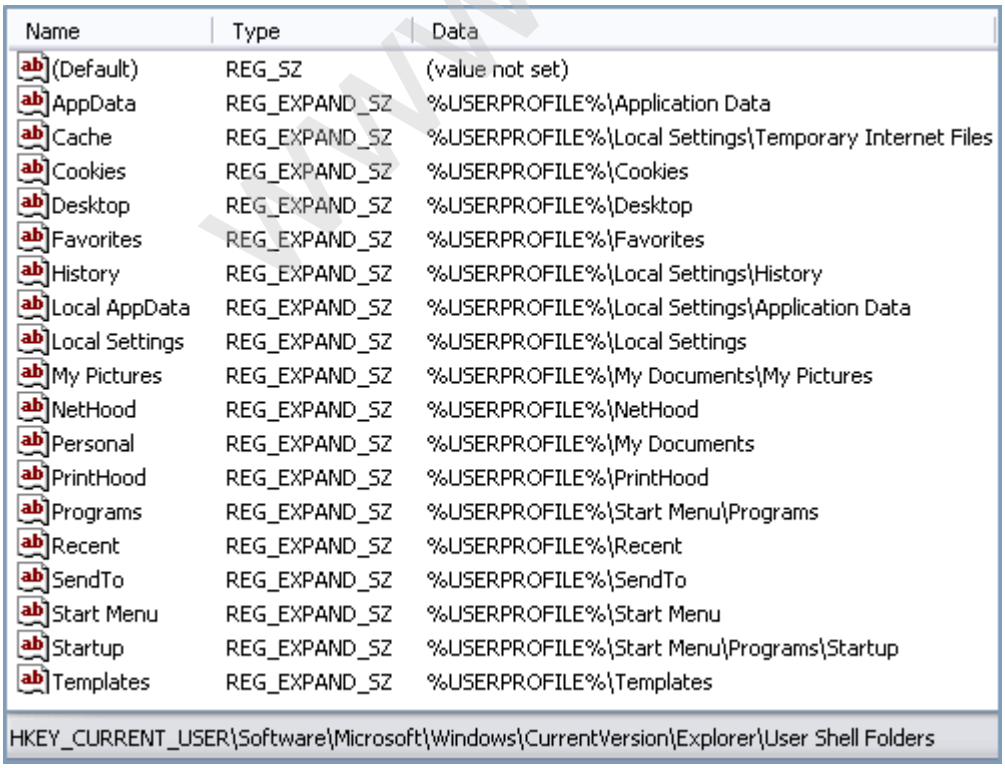
تغییر محل فولدرهای ویژه

فولدرهای ویژه شامل My Documents, My Pictures, Favorites و فولدرهای زیاد دیگری می‌باشند.

کاربران ممکن است به دلایل مختلف بخواهند محل این فولدرهای ویژه را تغییر دهند, اما تنها سه دلیل به ذهن من می‌رسد.. به عنوان مثال , کاربران ممکن است بخواهند My Documents را به درایو دیگری به جزء درایو سیستم عامل انتقال دهند تا بتوانند ویندوز خود را بدون از دست دادن سندهای خودشان از نو نصب کنند. دلیل دیگر به مواقعی مربوط می‌شود که کاربران یک شبکه بخواهند از بیش از یک کامپیوتر به سندهای خود دستیابی داشته باشند. در این گونه شرایط می‌توانند فولدرهای My Documents و Favorites را به محلی در شبکه انتقال دهند تا از هر جایی بتوانند به آنها دسترسی داشته باشند. متخصصان IT نیز اغلب تمایل دارند My Documents را به محلی در شبکه انتقال دهند تا پشتیبان گیری از سندهای کاربران آسانتر شود. اما دلیل آخر به منظور تغییر محل فولدرهای ویژه که خود من از آن بهره می‌برم استفاده از چند سیستم عامل در یک PC می‌باشد.

کلید زیر را کاوش کنید:

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders



Name	Type	Data
(Default)	REG_SZ	(value not set)
AppData	REG_EXPAND_SZ	%USERPROFILE%\Application Data
Cache	REG_EXPAND_SZ	%USERPROFILE%\Local Settings\Temporary Internet Files
Cookies	REG_EXPAND_SZ	%USERPROFILE%\Cookies
Desktop	REG_EXPAND_SZ	%USERPROFILE%\Desktop
Favorites	REG_EXPAND_SZ	%USERPROFILE%\Favorites
History	REG_EXPAND_SZ	%USERPROFILE%\Local Settings\History
Local AppData	REG_EXPAND_SZ	%USERPROFILE%\Local Settings\Application Data
Local Settings	REG_EXPAND_SZ	%USERPROFILE%\Local Settings
My Pictures	REG_EXPAND_SZ	%USERPROFILE%\My Documents\My Pictures
NetHood	REG_EXPAND_SZ	%USERPROFILE%\NetHood
Personal	REG_EXPAND_SZ	%USERPROFILE%\My Documents
PrintHood	REG_EXPAND_SZ	%USERPROFILE%\PrintHood
Programs	REG_EXPAND_SZ	%USERPROFILE%\Start Menu\Programs
Recent	REG_EXPAND_SZ	%USERPROFILE%\Recent
SendTo	REG_EXPAND_SZ	%USERPROFILE%\SendTo
Start Menu	REG_EXPAND_SZ	%USERPROFILE%\Start Menu
Startup	REG_EXPAND_SZ	%USERPROFILE%\Start Menu\Programs\Startup
Templates	REG_EXPAND_SZ	%USERPROFILE%\Templates

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders

در این کلید می‌توانید محل ذخیره فولدرهای ویژه خاص-کاربر را توسط ویندوز مشاهده کنید. هر مقدار در این کلید یک فولدر ویژه است. این مقادیر, از نوع Reg_Expand_SZ هستند, بنابراین می‌توانید متغییرهای محیطی را در آنها به کار بگیرید. مثلا با استفاده از %USERNAME% در یک مسیر می‌توانید نام کاربران را در آنها بگنجانید.

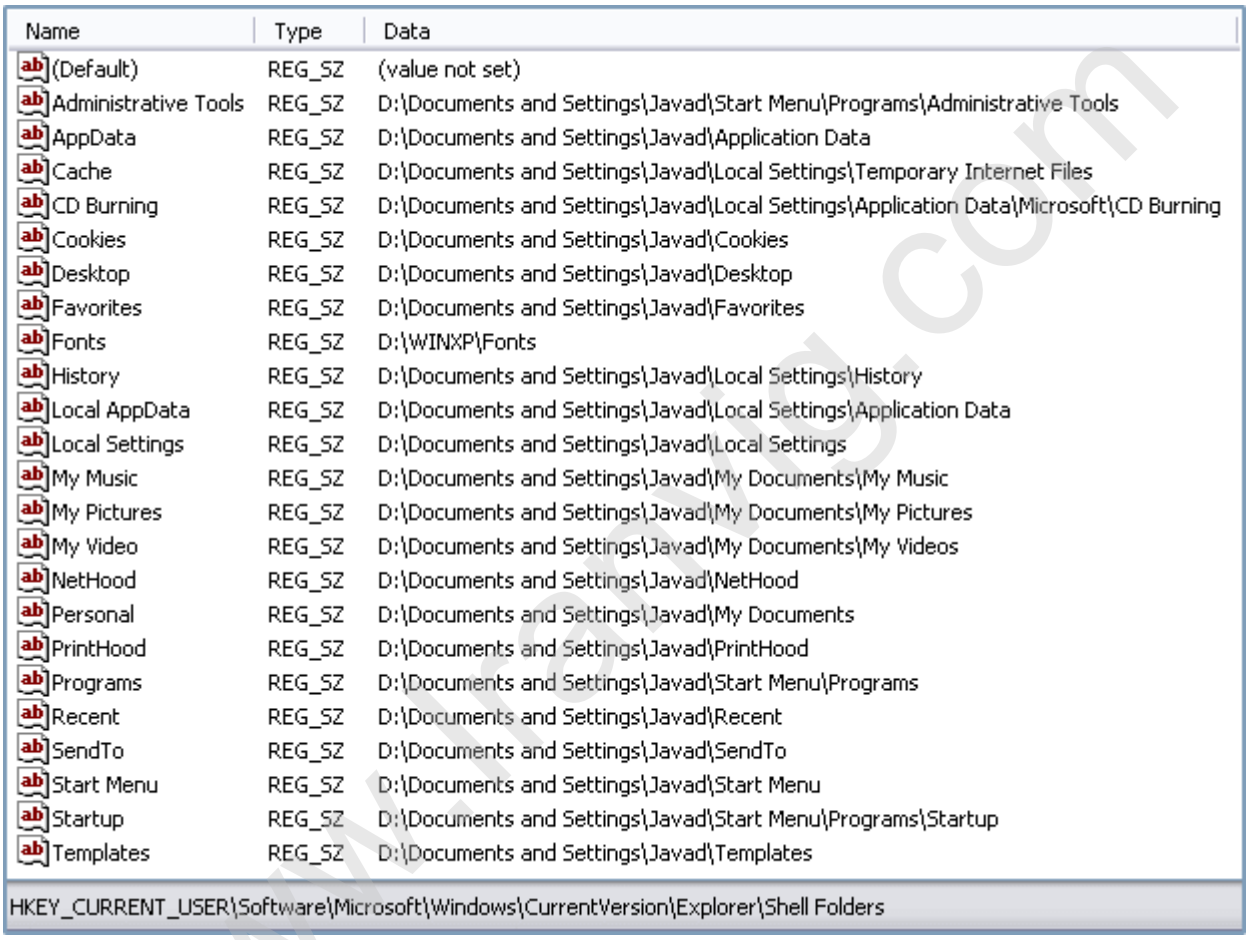
برای مثال اگر بخواهید محل فولدر Favorites کاربران به محلی در شبکه منتقل شود مقدار Favorites را در کلید مذکور به:

\\Server\Share\%USERNAME%\Favorites

تغییر دهید, که \\Server\Share نشان دهنده سرویس دهنده و محلی در آن است که فولدرها در آن قرار خواهند گرفت.

البته کلید دیگری حاوی داده‌هایی در همان مسیر وجود دارد کلید زیر را کاوش کنید:

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders



Name	Type	Data
(Default)	REG_SZ	(value not set)
Administrative Tools	REG_SZ	D:\Documents and Settings\Javad\Start Menu\Programs\Administrative Tools
AppData	REG_SZ	D:\Documents and Settings\Javad\Application Data
Cache	REG_SZ	D:\Documents and Settings\Javad\Local Settings\Temporary Internet Files
CD Burning	REG_SZ	D:\Documents and Settings\Javad\Local Settings\Application Data\Microsoft\CD Burning
Cookies	REG_SZ	D:\Documents and Settings\Javad\Cookies
Desktop	REG_SZ	D:\Documents and Settings\Javad\Desktop
Favorites	REG_SZ	D:\Documents and Settings\Javad\Favorites
Fonts	REG_SZ	D:\WINXP\Fonts
History	REG_SZ	D:\Documents and Settings\Javad\Local Settings\History
Local AppData	REG_SZ	D:\Documents and Settings\Javad\Local Settings\Application Data
Local Settings	REG_SZ	D:\Documents and Settings\Javad\Local Settings
My Music	REG_SZ	D:\Documents and Settings\Javad\My Documents\My Music
My Pictures	REG_SZ	D:\Documents and Settings\Javad\My Documents\My Pictures
My Video	REG_SZ	D:\Documents and Settings\Javad\My Documents\My Videos
NetHood	REG_SZ	D:\Documents and Settings\Javad\NetHood
Personal	REG_SZ	D:\Documents and Settings\Javad\My Documents
PrintHood	REG_SZ	D:\Documents and Settings\Javad\PrintHood
Programs	REG_SZ	D:\Documents and Settings\Javad\Start Menu\Programs
Recent	REG_SZ	D:\Documents and Settings\Javad\Recent
SendTo	REG_SZ	D:\Documents and Settings\Javad\SendTo
Start Menu	REG_SZ	D:\Documents and Settings\Javad\Start Menu
Startup	REG_SZ	D:\Documents and Settings\Javad\Start Menu\Programs\Startup
Templates	REG_SZ	D:\Documents and Settings\Javad\Templates

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders

وقتی کاربر در مرتبه آتی ارتباط برقرار می‌کند ویندوز کلید Shell Folders را با تبدیل مسیرهای موجود در User Shell Folders به روز می‌رساند. در حقیقت در مستندات مایکروسافت گفته شده که ویندوز از Shell Folders استفاده نمی‌کند و این کلید فقط برای مشاهده و استفاده کاربران می‌باشد.

پس برای جابه‌جا کردن این گونه فولدرها کافی است محتوای داده‌های موجود در کلید User Shell Folders را با توجه به نیاز خود تغییر دهید.

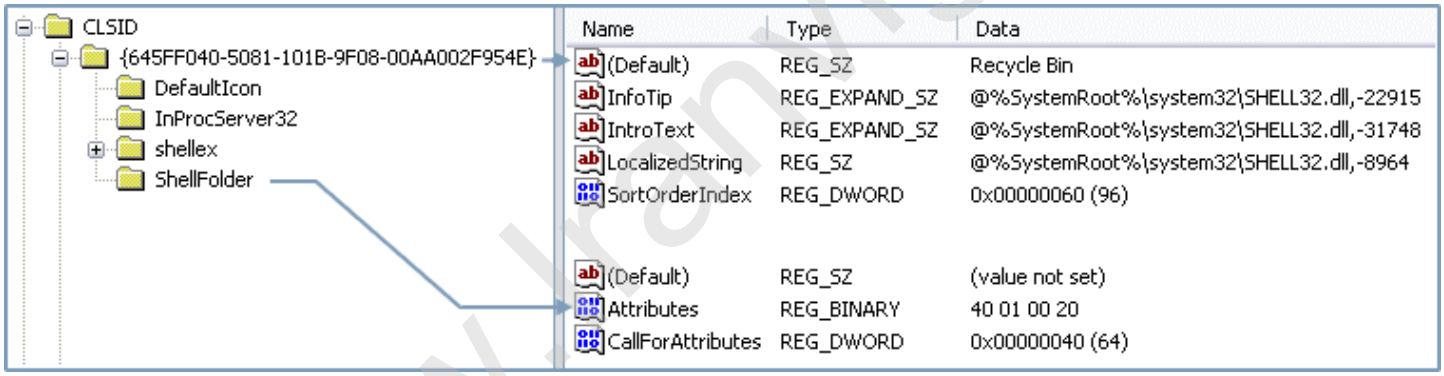
- ✓ بعد از هر گونه تغییری در محتوای کلید فوق به منظور اعمال تغییرات کافی است ارتباط خود را با اکانت کاربری‌تان قطع و مجددا وصل کنید یعنی Log off و بعد Log on کنید.

* فولدرهای ویژه این قسمت, فولدرهای خاص–کاربر هستند و در فولدرهای, پروفایل‌های کاربران قرار دارند. ویندوز فولدرهای خاص–کامپیوتر را در HKLM ذخیره می‌کند. Commom AppData, Common Desktop و Common Documents مثال‌هایی از فولدرهای خاص–کامپیوتر هستند, اما تغییر محل این فولدرها چندان مفید نیست.

تغییر فولدرهای شل با توجه به نیازهای شخصی

برخی از فولدرهایی که در ویندوز اکسپلورر, کنترل پانل یا دسکتاپ می‌بینید, واقعاً در فایل سیستم وجود ندارند. بلکه اینگونه موارد, شی‌های مبتنی بر کلاس‌های ثبت شده در کلید HKCR\CLSID هستند. برخی از فولدرها و فایل‌هایی که در فایل سیستم وجود ندارند, قابلیت ویژه‌ای دارند (مثلا فولدر History یا Briefcase) و این قابلیت‌ها نیز ناشی از کلاس‌های ثبت شده در HKCR\CLSID هستند. هر کلاس اساسا یک الگو برای ایجاد یک چیز واقعی است مثلا, یک شی در رابط کاربران. CLSID محلی است که این کلاس‌ها خودشان را در آنجا به ثبت می‌رسانند تا ویندوز از وجود آنها آگاه باشد.

برنامه‌های دیگر نیز ممکن است کلاس‌هایی را به ثبت برسانند که به آسانی می‌توانید موارد جالب را در HKCR\CLSID پیدا کنید, چرا که تمام آنها یک زیر کلید به نام ShellFolder دارند که دارای مقدار Attributes است.



کلاس‌هایی که دارای مقدار LocalizedString هستند نیز ممکن است کاندید منابعی برای تغییرات شخصی باشند, چرا که آنها این مقدار را فقط در صورتی خواهند داشت که شی‌های مبتنی بر آن کلاس‌ها در رابط کاربران ظاهر شوند. این کلاس‌ها مقاصد گوناگون دارند, و اغلب از آنها برای کاوش کردن در ویندوز استفاده خواهید کرد. جدول زیر آن دسته از کلاس‌های ثبت شده در HLCR\CLSID را نشان می‌دهد که به نظر من جالب هستند:

شی	شناسه کلاسGUID
* فولدرهای شل Folders Shell	
ActiveX Cache	{88C6C381-2E85-11D0-94DE-444553540000}
Computer Search Results	{1f4de370-d627-11d1-ba4f-00a0c91eedba}
History	{FF393560-C2A7-11CF-BFF4-444553540000}
Internet Explorer	{871C5380-42A0-1069-A2EA-08002B30309D}

شی	شناسه کلاسGUID
My Computer	{20D04FE0-3AEA-1069-A2D8-08002B30309D}
My Documents	{450D8FBA-AD25-11D0-98A8-0800361B1103}
My Network Places	{208D2C60-3AEA-1069-A2D7-08002B30309D}
Offline Files	{AFDB1F70-2A4C-11d2-9039-00C04F8EEB3E}
Programs	{7be9d83c-a729-4d97-b5a7-1b7313c39e0a}
Recycle Bin	{645FF040-5081-101B-9F08-00AA002F954E}
Search Results	{e17d4fc0-5564-11d1-83f2-00a0c90dc849}
Shared Documents	{59031a47-3f72-44a7-89c5-5595fe6b30ee}
Start Menu	{48e7caab-b918-4e58-a94d-505519c795dc}
Temporary Internet Files	{7BD29E00-76C1-11CF-9DD0-00A0C9034933}
Web	{BDEADF00-C265-11D0-BCED-00A0C90AB50F}
❋ فولدرهای کنترل پانل	
Control Panel	{21EC2020-3AEA-1069-A2DD-08002B30309D}
Administrative Tools	{D20EA4E1-3957-11d2-A40B-0C5020524153}
Fonts	{D20EA4E1-3957-11d2-A40B-0C5020524152}
Dial-up Connection	{7007ACC1-3202-11D1-AAD2-00805FC1270E}
Network Connections	{7007ACC7-3202-11D1-AAD2-00805FC1270E}
Printers and Faxes	{2227A280-3AEA-1069-A2DE-08002B30309D}
Scanners & Cameras	{E211B736-43FD-11D1-9EFB-0000F8757FCD}
Scheduled Tasks	{D6277990-4C6A-11CF-8D87-00AA0060F5BF}
❋ نمادهای تصویری کنترل پانل	
Folder Options	{6DFD7C5C-2451-11d3-A299-00C04F8EF6AF}
Taskbar and Start Menu	{0DF44EAA-FF21-4412-828E-260A8728E7F1}
User Accounts	{7A9D77BD-5403-11d2-8785-2E0420524153}
❋ سایر موارد	
Add Network Place	{D4480A50-BA28-11d1-8E75-00C04FA31A86}

شی	شناسه کلاسGUID
Briefcase	{85BBD920-42A0-1069-A2E4-08002B30309D}
Search	{2559a1f0-21d7-11d4-bdaf-00c04f60b9f0}
Help and Support	{2559a1f1-21d7-11d4-bdaf-00c04f60b9f0}
Windows Security	{2559a1f2-21d7-11d4-bdaf-00c04f60b9f0}
Run	{2559a1f3-21d7-11d4-bdaf-00c04f60b9f0}
Internet	{2559a1f4-21d7-11d4-bdaf-00c04f60b9f0}
E-mail	{2559a1f5-21d7-11d4-bdaf-00c04f60b9f0}
Network Setup Wizard	{2728520d-1ec8-4c68-a551-316b684c4ea7}

هر شی در جدول فوق دارای یک شناسه کلاس یا GUID است. این GUID درون همه کامپیوترها برای اشیا نام برده یکسان می‌باشند. برای کسب اطلاعات بیشتر در رابطه با GUID ها به فصل دوم مراجعه کنید.

من این جدول را به چهار قسمت تقسیم کرده‌ام. همان طور که در جدول می‌بینید بخش نخست یعنی فولدرهای شل که فولدرهایی هستند با مقاصد ویژه مثلا My Computer یا Network Places و غیره. پس از مسلح شدن به اطلاعات این جدول می‌توانید کارهای زیادی انجام دهید. به عنوان مثال می‌توانید فولدرهایی که در My Computer نمایش داده می‌شوند را با توجه به نیازهای شخصی خود تعیین کنید. و یا می‌توانید نمادهای تصویری که در دسکتاپ نمایش داده می‌شوند را تغییر دهید, و یا تعیین کنید که اصلا کدام نماد تصویری بر روی دسکتاپ نمایش داده شود. به عنوان مثال, مدیران برای دستیابی سریعتر به فولدر Administrative Tools اغلب ترجیح می‌دهند نماد تصویری Administrative Tools را بر روی دسکتاپ قرار دهند.

برای آگاهی هر چه بیشتر شما با این موارد, چند بخش عملی را در ادامه عنوان می‌کنم.

تغییر نام نمادهای تصویری دسکتاپ

نمادهای تصویری همچون My Computer, My Documents, Internet Explorer را می‌توانید خیلی راحت با راست کلید بر رویشان و انتخاب گزینه Rename تغییر نام دهید. اما تغییر نام نمادهای تصویری دیگر همچون Recycle Bin, چندان آسان نیست. این گونه نمادها فاقد فرمان Rename هستند !

برای تغییر نام یک نماد تصویری بدون استفاده از فرمان Rename می‌توانید از طریق ویرایش اطلاعات ثبت کلاس آن عمل کنید. مقدار LocalizedString را تغییر دهید. به مثال ذیل توجه کنید:

GUID کلاس Recycle Bin در جدول بالا {645FF040-5081-101B-9F08-00AA002F954E} است. برای اینکه نام این نماد را به Trash Can تغییر دهید مقدار LocalizedString را در کلید:

HKCR\CLSID\{645FF040-5081-101B-9F08-00AA002F954E}

به Trash Can تغییر دهید. پس از انجام این کار دسکتاپ را با ماوس برگزینید و کلید F5 (یا همان فرمان Refresh) را برای نوسازی محتوای آن نماد فشار دهید. در اینجا مقدار LocalizedString برای GUID نام برده, در حالت معمول چیزی برابر:

@%SystemRoot%\system32\SHELL32.dll,-8964

می‌باشد که معنای آن این است که ویندوز رشته را با ID شماره 8964 از فایل SHELL32.dll به کار می‌برد. کافی است آن را با نام جدید جایگزین کنید.

- ✓ LocalizedString یک مقدار Reg_EXPAND_SZ می‌باشد, بنابراین می‌توانید در آن از متغییرهای محیطی استفاده کنید. به عنوان مثال نام آن را به "%USERNAME% Garbage" تغییر دهید تا برای کاربری به نام Javad عبارت " Javad's Garbage" در زیر نماد تصویری نمایش داده شود (حتی می‌توانید حروف فارسی را نیز در آن به کار بگیرید).

یا مثلا GUID کلاس My Computer برابر {20D04FE0-3AEA-1069-A2D8-08002B30309D} است. شما باید LocalizedString را در:

HKCR\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D}

به "کامپیوتر %USERNAME%" تغییر دهید تا برای کاربری مثلا با نام جواد عبارت "کامپیوتر جواد" در زیر نماد تصویری My Computer نمایش داده شود.

به منظور بهتر روشن شدن موضوع فوق, یک برنامه که موجب تغییر نام برخی از نماد های تصویری دسکتاپ می‌شود را در بسته ضمیمه همراه این مقاله قرار داده‌ام. البته سورس کد به زبان VB6 است تا برای همه قابل استفاده باشد..

- ✓ مقدار LocalizedString را در اطلاعات ثبت بعضی از کلاس‌ها نخواهید دید. عدم وجود این مقدار نشانگر آن است که مایکروسافت تمایلی به نمایش نام آن شی‌ها در رابط کاربر نداشته است. برای تغییر نام کلاسی که فاقد این مقدار است, مقدار پیش فرض HKCR\CLSID\classID را تغییر دهید, و یا حتی بهتر است LocalizedString را به آن کلاس بی‌افزایید. وقتی ویندوز نام یک شی را جستجو می‌کند, ابتدا برای یافتن LocalizedString اقدام می‌کند, و بعد برای مقدار پیش فرض اطلاعات ثبت کلاس.

استفاده از تصاویر شخصی برای نمادهای تصویری

تمام کلاس‌های مطرح در جدول بالایی این مقاله دارای زیر کلید DefaultIcon هستند. مقدار پیش فرض این زیر کلید, نماد تصویری است که ویندوز به هنگام نمایش شی‌های مبتنی بر آن کلاس به کار می‌برد. به عنوان مثال, مقدار پیش فرض DefaultIcon در:

HKCR\CLSID\{20D04FE0-3AEA-1069-A2D8-08002B30309D}

نماد تصویری است که ویندوز به هنگام ایجاد شی My Computer در رابط کاربر (مثلا ویندوز اکسپلورر بر روی دسکتاپ), نمایش می‌دهد.

برای اینکه از یک نماد تصویری دیگر استفاده کنید, مقدار پیش فرض DefaultIcon را تغییر دهید. برای این کار می‌توان از مسیر و نام فایل یک نماد تصویری, با انشعاب ico , استفاده کنید, و یا حتی از یک مسیر مبداء resource استفاده نمایید. مسیر مبداء به شکل Name,Index یا Name,-resID است Name, همان مسیر و نام فایل حاوی نماد تصویری است که معمولا یک فایل Dll یا EXE است. بیشتر نماد های تصویری مورد استفاده در ویندوزهای۳۲ بیتی از:

%SystemRoot%\System32\Shell32.dll

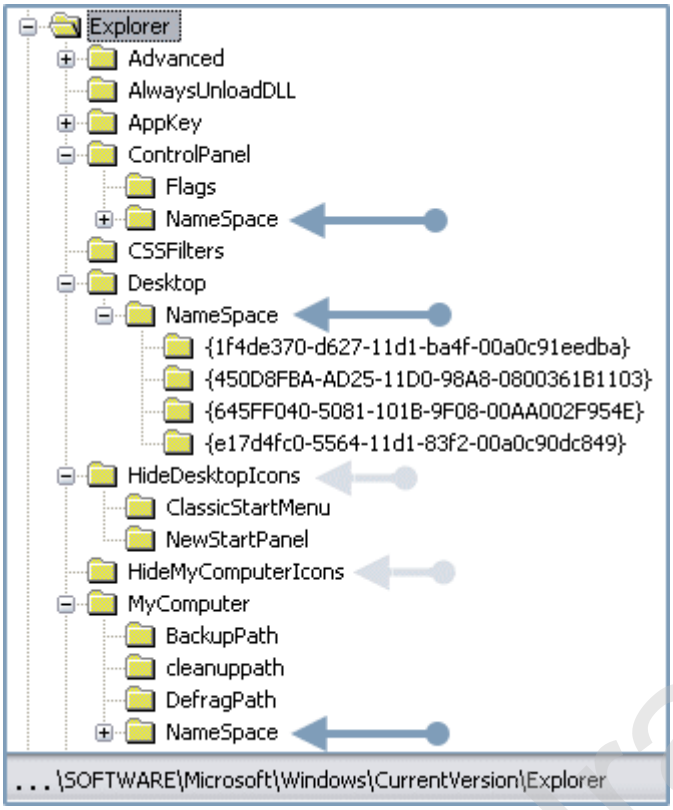
به دست می‌آیند. Index, شماره ایندکس نمادهای تصویری است که از صفر آغاز می‌شوند. resID, شناسه منبع نماد تصویری است. برنامه سازان ID هایی را به منابعی که فایل‌های برنامه‌ای, از جمله نمادهای تصویری, رشته‌ها, کادرهای مکالمه و غیره, را در آنها ذخیره می‌کنند, تخصیص می‌دهند.

همکنون برنامه‌های زیادی وجود دارند که نمادهای تصویری موجود در یک فایل برنامه, را پیکر بندی می‌کنند. به عنوان نمونه Microangelo یا PEexplorer را معرفی می‌کنم. این ابزارآلات حتی نمادهای تصویری را از یک فایل Dll یا EXE استخراج می‌کنند تا بتوانید آنها را به طور مجزا به کار ببندید.

افزودن نمادهای تصویری به دسکتاپ

دسکتاپ ویندوز به خصوص در نسخه XP آن بسیار منظم‌تر از نگارش‌های پیشین ویندوز است. طبق پیش فرض تنها نماد تصویری Recycle Bin را در آن خواهید دید. اما می‌توانید نمادهای تصویری متداول را به آن بی‌افزایید. شما قادر به اضافه کردن نمادهای تصویری Internet Explorer, My Computer, My Documents و My Network Places از طریق رابط کاربری Display به دسکتاپ خواهید بود (بحث فوق کاملا مجزا از ایجاد و توزیع Shortcut ها می‌باشد. در اینجا ایجاد کیفی و اصولی اشیا مد نظر است. آموزش مطالب فوق به خصوص برای شما برنامه‌نویسان روشن می‌کند که ویندوز چگونه می‌تواند یک عملیات ایجاد یا تغییر را فقط به وسیله اطلاعات رجیستری انجام دهد, از این مطالب در آینده, به طور مستقیم در برنامه‌هایی که می‌نویسید استفاده خواهید کرد).

در صورتی که نماد تصویری مورد نظرتان جزء چهار مورد بالا نیست, چنانچه می‌خواهید نمادهای تصویری را به فولدرهای ویژه اضافه کنید, می‌بایست رجیستری را ویرایش کنید. تمام روش‌هایی که در این قسمت یاد خواهید گرفت به شاخه Software\Microsoft\Windows\CurrentVersion\Explorer مربوط می‌شوند. این شاخه را در کلید HKLM تغییر دهید تا تغییرات برای تمام کاربران باشند; چنانچه تغییرات مورد نظر را در HLCU اعمال کنید, در آن صورت تغییرات برای یک کاربر خاص خواهند بود. به شکل زیر توجه کنید.



همان طور که در تصویر می‌بینید زیر کلیدهای NameSpace مربوط به Explorer\ControlPanel, Explorer\Desktop و Explorer\MyComputer محتوای هر یک از فولدرهای متناظر با این موارد را تعیین می‌کند. با ویرایش زیر کلیدهای فهرست شده در جدول ابتدایی این مقاله, قادر خواهید بود نمادهای تصویری را به کنترل پانل, دسکتاپ و غیره بیافزایید. به عنوان مثال برای افزودن یک نماد تصویری به دسکتاپ جهت باز کردن کادر مکالمه Run یک زیر کلید جدید به نام {2559a1f3-21d7-11d4-bdaf-00c04f60b9f0} , (که معادل GUID کادر مکالمه Run در جدول فوق الذکر است) به شاخه Explorer\Desktop بیافزایید.

سپس با Refresh کردن دسکتاپ می‌توانید نتیجه را ملاحظه کنید. با اجرای فرمان زیر در منوی Run قادر به انجام این کار به طور خودکار خواهید بود:

```
Reg Add "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Desktop\NameSpace\{2559a1f3-21d7-11d4-bdaf-00c04f60b9f0}"
```

برای معکوس کردن فرمان بالا عبارت زیر را در Run اجرا کنید.

```
Reg Delete "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Desktop\NameSpace\{2559a1f3-21d7-11d4-bdaf-00c04f60b9f0}" /f
```

فقط بعد از اجرای هر فرمان, Refresh روی دسکتاپ را فراموش نکنید.

- ✓ تنها فولدرهای ویژه کاندیدهایی برای اضافه شدن به My Computer هستند. دلیل این امر کاملا روشن است. پس فقط فولدرهای شل موجود در بخش اول جدول, ابتدایی همین مقاله, را می‌توانید به شاخه Explorer\MyComputer\NameSpace اضافه کنید.

فرمان زیر موجب اضافه شدن نماد تصویری Internet Explorer به My Computer می‌شود :

```
Reg Add
"HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\MyComputer\NameSpace\{871C5380-42A0-1069-A2EA-08002B30309D}"
```

با فرمان زیر عملکرد عملیات بالا را به حالت قبل برگردانید:

```
Reg Delete
"HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\MyComputer\NameSpace\{871C5380-42A0-1069-A2EA-08002B30309D}" /f
```

پنهان کردن نمادهای تصویری از دسکتاپ

در نگارش‌های ویندوز تا قبل از ویندوز XP برای برداشتن نمادهای تصویری از دسکتاپ مجبور هستید که زیر کلیدهای آنها را از NameSpace حذف کنید. این امر اغلب مشکل ساز می‌شود, به ویژه به هنگام برداشتن نماد تصویری Network Neighborhood از دسکتاپ.

از نگارش ویندوز XP به بعد, روش‌های ویژه‌ای برای پنهان کردن نمادهای تصویری دسکتاپ به وجود آمده است. برای برداشتن آنها از دسکتاپ یا My Computer می‌توانید شاخه Software\Microsoft\Windows\CurrentVersion\Explorer را در کلیدهای HKLM و HKCU ویرایش کنید. برای اینکه نمادهای تصویری را در MY Computer پنهان کنید, یک مقدار نوع Reg_DWORD در کلید:

SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\HideMyComputerIcons

ایجاد کنید و بهتر است نام آن, GUID کلاس نماد تصویری باشد که می‌خواهید پنهان کنید, و مقدار 0x01 را به آن اختصاص دهید. و در آخر برای مشاهده نتیجه کار ویندوز اکسپلورر را نوسازی Refresh کنید.

پنهان سازی نمادهای تصویری دسکتاپ قدری پیچیده‌تر است. دو زیر کلید به نام های ClassicStartMenu و NewStartPanel در مسیر:

SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\HideDesktopIcons

وجود دارند. زیر کلید اول برای زمانی که ویندوز XP از منوی Start قدیمی یا کلاسیک خود, استفاده می‌کند, در نظر گرفته شده است. و در آن با اضافه کردن مقدارهای جدید مطابق دستورالعملی که ذکر شد می‌توانید تعیین کنید کدام نمادهای تصویری در دسکتاپ نمایش داده شوند. به همین ترتیب موارد ذکر شده برای NewStartPanel یعنی منوی Start جدید ویندوز اکس پی صادق می‌باشد (به تصویر قبلی توجه کنید).

به عنوان مثال فرمان زیر باعث مخفی شدن نماد تصویری Recycle Bin سطل زباله از دسکتاپ در شرایط استفاده از منوی Start جدید, در ویندوزهای XP به بعد می‌شود (فراموش نکنید برای مخفی کردن یا برداشتن Recycle Bin از دسکتاپ در ویندوز هیچ رابط کاربری وجود ندارد):

```
Reg Add "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\HideDesktopIcons\NewStartPanel" /v "{645FF040-5081-101B-9F08-00AA002F954E}" /t REG_DWORD /d "0x01" /f
```

فرمان زیر را در منوی Run اجرا کنید تا Recycle Bin باز به دسکتاپ باز گردد:

```
Reg Add "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\HideDesktopIcons\NewStartPanel" /v "{645FF040-5081-101B-9F08-00AA002F954E}" /t REG_DWORD /d "0x00" /f
```

همان طور که در این مقاله مشاهده کردید کارهای عملی در این سلسله مقالات رو به افزایش است. به این منظور از شما می خواهم قبل از انجام هر عمل کنکاش در رجیستری خود تهیه نسخه‌های پشتیبان, همان طور که در فصل‌های قبل معرفی گردیدند, را فراموش نکنید.

در این مقاله چند روش عملی با استفاده از فایل Reg.exe به انجام رسید برای کسب اطلاعات بیشتر در رابطه با این رابط ارزشمند رجیستری, منتظر مقالات بعدی باشید, اما اگر عجله دارید به شما پیشنهاد می‌کنم فرمان Reg را با سویچ "؟" در جلوی اعلان داس, محیط Command Prompt اجرا کنید.

▪ فصل هفتم : کسب مالکیت فایل‌ها

چگونه می‌توان تعیین کرد یک نوع فایل در سیستم فقط با برنامه ما اجرا شود ؟

بخش عمده محتوای رجیستری در کلید HKCR است. یعنی محلی که ویندوز اطلاعات مربوط به ثبت کلاس‌ها و ارتباطات فایل‌ها را ذخیره می‌کند. این تنظیمات انواع مختلف فایل‌ها را با برنامه‌هایی مرتبط می‌کنند که می‌توانند آنها را باز, ویرایش و چاپ کنند. مثلا فایلی از نوع text که با پسوند ".txt" شناخته می‌شود را در نظر بگیرید. اطلاعات ثبت شده در کلید HKCR و شاخه مربوطه به نوع txt مشخص می‌کند که یک فایل با این قالب به وسیله چه برنامه‌ای باز شود و یا چه برنامه‌هایی کاندید برای ویرایش این نوع هستند.

حجم بالایی از تغییرات شخصی که من به طور مرتب اعمال می‌کنم, تغییرات ساده‌ای هستند در کلید HKCR. به عنوان مثال, من همیشه دوست دارم که فرامین خاصی را به ارتباطات مربوط به فولدرها تخصیص دهم تا در مواقعی که فولدری به عنوان دایرکتوری کاری جاری انتخاب می‌شود, بتوانم پنجره اعلان فرامین ام اس–داس CMD را باز کنم. همچنین فرامینی را به شی My Computer افزوده‌ام تا به سرعت بتوانم به ویراستار رجیستری Regedit دسترسی داشته باشم و حتی برای مشاهده سریع محتوای هر فایل گزینه EditText را به منوی میانبر هر فایل افزوده‌ام تا بتوانم خیلی سریع محتوای هر نوع فایلی را در محیط Notepad ببینم و در مواقع لزوم اقدام به ویرایش آنها نمایم. اگر با محتوای HKCR به خوبی آشنا باشید, در آن صورت فرصت‌های بسیار زیادی برای تغییر ویندوز متناسب با نیازها و خواسته‌های خود خواهید داشت.

➢ الگوریتم ادغام

کلید اصلی HKCR تا پیش از ویندوز ۲۰۰۰ نوعی ارتباط با کلید HKLM\SOFTWARE\Classes بود, اما از ویندوز ۲۰۰۰ تا هم اکنون بسیار پیچیده‌تر شده. در حال حاظر ویندوز کلیدهای HKLM\SOFTWARE\Classes و HKCU\Software\Classes را ادغام می‌کند. داده‌های موجود در HKLM همان داده‌های پیش فرض ثبت کلاس‌ها و ارتباطات فایل‌ها هستند. این بدان معناست که با ایجاد یک اکانت کاربری جدید تنظیمات موجود در HKLM برای کاربر جدید در نظر گرفته شده و یک کپی از آنها در کلید HKCU مربوط به آن کاربر, در همان زمان ایجاد اکانت جدید نوشته و در نظر گرفته می‌شود.

این در حالی است که تنظیمات موجود در کلید HKCU, اطلاعات خاص کاربر برای ثبت کلاس‌ها و ارتباطات فایل‌ها هستند. البته HKCU\Software\Classes واقعی نوعی ارتباط با HKU\SID_Classes است که ویندوز به هنگام بارگذاری Hive پروفایل در کلید HKU\SID, بارگذاری می‌کند. (برای مطلع شدن از تعریف و مقدار صحیح SID مورد نظر سیستم خود به فصل دوم : شناسه‌های امنیت SID مراجعه نمائید).

حال سوالی که پیش می‌آید آن است که ویندوز به هنگام استفاده از داده‌های موجود در HKCR با توجه به الگوریتم ادغام از کدام Link استفاده می‌کند ؟

اگر مقادیر مشابهی در هر دو شاخه نام برده شده وجود داشته باشد, مقدار موجود در HKCU\Software\Classes از اولویت بالاتری برخوردار است و بر مقدار موجود در HKLM\SOFTWARE\Classes ارجح خواهد بود. الگوریتم بالا مزایای زیادی دارد. این الگوریتم به کاربران امکان می‌دهد تا برنامه‌های کاربردی را نصب کنند و از آن دسته از ارتباطات فایل‌ها که تاثیری بر کاربران دیگر ندارند, استفاده نمایند. از این رو, دو کاربری که از یک کامپیوتر به طور مشترک استفاده می‌کنند, می‌توانند از دو برنامه مختلف برای ویرایش فایل‌های همنوع استفاده کنند.

هنگامی که کلید جدیدی را در HKCR به طور دستی در Regedit و حتی از طریق یک برنامه یا کد و یا یک رابط کاربری در ویندوز, ایجاد می‌نمایید, ویندوز در واقع آن را در HKLM\SOFTWARE\Classes ایجاد می‌کند. توجه داشته باشید ویندوز ثبت تغییرات در HKCU\Software\Classes را به عهده برنامه‌سازان گذاشته و کاربران عادی نیز جزء از طریق ویرایشگر رجیستری, به طور مستقیم در HKCU\Software\Classes , از هیچ راه دیگری نمی‌توانند به این شکل تنظیماتی مختص به خود در سیستم اعمال کنند. در واقع در اسناد مایکروسافت آمده فراهم کردن امکان ثبت کلاس‌های خاص کاربر (منظور در HKCU\Software\Classes) تنها برای برنامه‌ها فراهم آمده. اما وقتی کلاسی از یک برنامه را ویرایش می‌کنید, بسته به اینکه برنامه از پیش موجود باشد یا خیر, تغییرات در HKLM یا HKCU منعکس می‌شوند. اگر برنامه در هر دو محل موجود باشد ویندوز تنها نسخه موجود در HKCU را به روز می‌رساند.

این جدول موضوع را روشن تر می‌سازد.

داده‌ای در HKLM وجود دارد	داده‌ای در HKCU وجود دارد	داده‌ها از HKLM خوانده می شوند	داده‌ها از HKCU خوانده می شوند	داده‌های موجود در HKCR از کلید ... می آید	با توجه به وجود داشتن داده, داده جدید در کلید ... ثبت می‌شود
خیر	خیر	خیر	خیر	-	HKLM
بله	خیر	بله	خیر	HKLM	HKLM
خیر	بله	خیر	بله	HKCU	HKCU
بله	بله	خیر	بله	HKCU	HKCU

کلیدهای انشعاب فایل‌ها

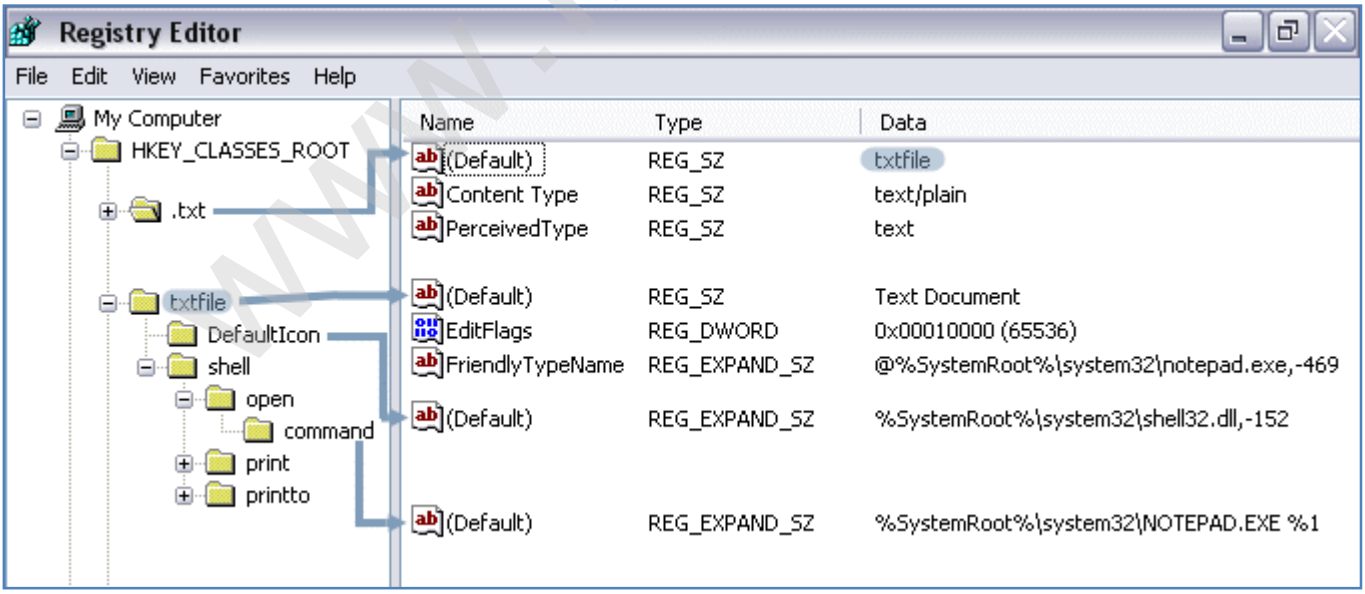
فایل‌هایی که حاوی داده‌های بخصوصی هستند, معمولا دارای انشعاب یکسان می‌باشند. به عنوان مثال, برای سندهای Word از انشعاب ".doc" استفاده می‌شود. اگر چه انشعاب‌های سه کاراکتری متعارف می‌باشند, اما انشعاب فایل‌ها می‌تواند بیش از سه کاراکتر نیز باشند. فایل‌های دارای انشعاب یکسان به یک کلاس از فایل‌ها تعلق دارند. این کلاس, مشخص کننده رفتار مشترک تمام فایل‌هایی است که از یک انشعاب استفاده می‌کنند. با اعمال تغییرات شخصی در ارتباطات فایل‌ها می‌توانید مشخص کنید که چه برنامه‌ای می‌تواند یک فایل را باز کند, فرامینی را به منوی میانبر فایل‌ها بیافزایید, یا حتی یک نماد تصویری icon شخصی مشخص کنید تا ویندوز اکسپلورر از آن برای نوع فایل‌ها استفاده کند.

ارتباطات فایل‌ها از دو بخش تشکیل می‌شود. نخست, کلید انشعاب فایل‌ها است که HKCR\.ext می‌باشد. وقتی ویندوز به اطلاعاتی درباره یک نوع فایل نیاز دارد, این کلید را جستجو می‌کند.

- ✓ مقدار پیش فرض کلید انشعاب فایل‌ها, نام کلاس برنامه مرتبط با آن فایل‌ها است, یعنی بخش دوم ارتباطات فایل‌ها.

ارتباطات کلاس برنامه‌ها در HKCR\progid ذخیره می‌شود. که progid همان ID برنامه کاربردی است. مقدار پیش فرض progid, نام رایج برنامه کاربردی است.

به عنوان مثال, ویندوز برای اینکه بفهمد یک فایل از نوع متن Text که با پسوند ".txt" ذخیره شده را باید با کدام برنامه باز کند ابتدا به سراغ کلید HKCR\.txt می‌رود. داده پیش فرض در این کلید مقدار txtfile را دارد. بنابراین ویندوز به سراغ کلید HKCR\txtfile می‌رود. در واقع txtfile کلاس مشترک بین تمام فایل‌ها با پسوند ".txt" می‌باشد. تصویر زیر مسئله را روشن تر می‌سازد:



همان طور که در شکل بالا می‌بینید ویندوز برای باز کردن یک فایل با انشعاب ".txt" از برنامه Notepad استفاده می‌کند. در واقع ذکر Open در منوی آن این مسئله را تعیین می‌نماید. در این باره به طور مفصل صحبت خواهد شد.

کلیدهای انشعاب فایل‌ها, زیر کلیدها و مقادیر گوناگون دارند. متداول ترین آنها در زیر شرح داده شده است:

- **PerceivedType** – این مقدار Reg_SZ نوع عمومی فایل را نشان می‌دهد. ویندوز XP تنها نگارشی است که (تا کنون) از آن استفاده می‌کند. این "نوع" مشابه انواع فایل‌ها است (انواع فایل‌ها مانند bmp , wav , mp3 , jpg , txt و ...) با این تفاوت که به جای نوع خاصی از فایل‌ها, به گروه گسترده‌ای از انواع فایل‌ها اطلاق می‌شود. این نوع را همچون انواع فوق العاده در نظر بگیرید. این نوع فایل‌ها شامل تصاویر, فایل‌های متنی, فایل‌های صوتی و فایل‌های فشرده می‌شود. در ویندوز XP می‌توانید این نوع را به یک نوع فایل مرتبط کنید. به عنوان مثال انشعاب‌های Jpg , Bmp , Gif , Png تماما فایل‌های تصویری در نظر گرفته می‌شوند. ویندوز XP انواع زیادی از این نوع فایل‌ها را تعریف می‌کند. در کلید انشعاب فایل, یکی از موارد ذیل را به مقدار PerceivedType تخصیص می‌دهید:

-Image

-Video

-Text

-Compressed

-Audio

-System

- **Content Type** – این مقدار Reg_SZ تعیین کننده نوع MIME می‌باشد. در آینده راجع به MIME توضیح خواهم داد.
- **OpenWithProgids** – این زیر کلید فهرستی از برنامه‌های جایگزین مرتبط با انشعاب است. ویندوز این برنامه‌ها را در قسمت Other Programs کار مکالمه Open With نمایش می‌دهد.
- **OpenWithList** – این زیر کلید حاوی یک یا چند کلید است که نام برنامه‌های کاربردی که باید در قسمت Recommended Programs کادر مکالمه Open With ظاهر شوند در آنها ذخیره می‌شود. گاهی اوقات کاربران می‌خواهند فایل‌ها را با برنامه‌های کاربردی که با کلاس فایل‌ها مرتبط نیستند, باز کنند.

به عنوان مثال, ممکن است کاربری بخواهد یک سند را به جای Word با برنامه Wordpad که همراه ویندوز توزیع می‌گردد باز کند. در مواقع دیگر کاربران ممکن است بخواهند که فایل‌هایی را باز کنند که با هیچ برنامه‌ای مرتبط نیستند. کادر مکالمه Open With پاسخگوی هر دو حالت است.

برنامه‌های کاربردی که در کادر مکالمه Open With می‌بینید, به طور حتم در کلید HKCR\Applications ثبت شده‌اند. این کلید برای هر برنامه کاربردی یک زیر کلید دارد و نام زیر کلید همان نام فایل اجرایی برنامه کاربردی است.

✓ با افزودن مقدار NoOpenWith از نوع Reg_SZ به کلید HKCR\Applications\Program.exe می‌توانید از نمایش یک برنامه کاربردی در کادر مکالمه Open With جلوگیری کنید.

تمرین – فرمان زیر را در کادر مکالمه Run اجرا کنید تا دیگر برنامه Notepad در کادر مکالمه Open With مربوط به یک فایل نمایش داده نشود:

```
Reg Add "HKCR\Applications\Notepad.exe" /v NoOpenWith /t REG_SZ /f
```

همچنین فرمان زیر عملیات بالا را معکوس می‌نماید:

```
Reg Delete "HKCR\Applications\Notepad.exe" /v NoOpenWith /f
```

در این تمرین برای مخفی کردن دیگر برنامه‌ها کافی است نام فایل اجرایی آن برنامه را همراه با پسوند (که معمولا exe است) را به جای Notepath.exe در فرمان‌های بالا جایگزین کنید. برای اینکه نتیجه این تمرین را ببینید هر بار بعد از اجرای فرمان ابتدا روی پنجره دسکتاپ Refresh کنید سپس با راست کلیک برروی یک فایل گزینه Open With را انتخاب کنید تا نتیجه را ملاحظه نمایید.

- **ShellNew** – این زیر کلید برای یک انشعاب یک الگو template را تعریف می‌کند که وقتی کاربران این نوع فایل را در منوی New انتخاب می‌کنند, ویندوز فایل جدیدی را از روی آن الگو ایجاد می‌کند. وقتی کاربران در یک قسمت خالی از یک فولدر راست کلیک می‌کنند و New را انتخاب می‌کنند, فهرستی از فایل‌های الگو را می‌بینند که می‌توانند در آن فولدر ایجاد نمایند. الگوهای دیگر را می‌توانید به منوی New بیافزایید حتی در صورت بلند بودن لیست منوی New می‌توانید مواردی که اصلا از آنها استفاده‌ای نمی‌کنید را از این لیست حذف نمایید.

برای این منظور نخست اطمینان حاصل کنید که HKCR کلیدی برای انشعاب نوع فایل مورد نظرتان را داشته باشد.

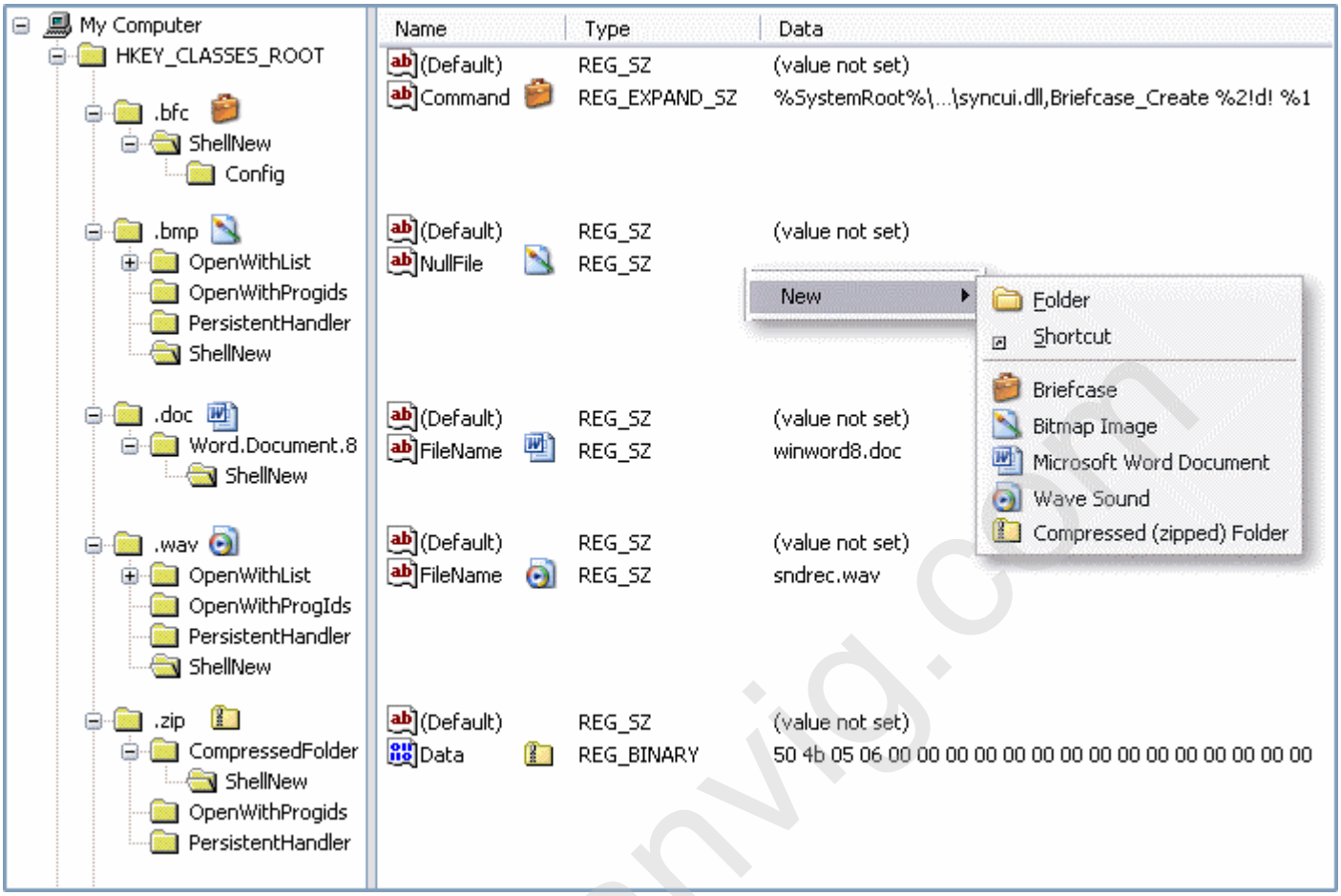
به عنوان مثال, برای اینکه الگویی برای فایل‌های دارای انشعاب ".inf" تعریف کنید ابتدا کلید HKCR\.inf\ShellNew را ایجاد نمایید. سپس یکی از مقادیر ذیل را در ShellNew ایجاد کنید:

Command – که از نوع Reg_SZ بوده و موجب اجرای یک برنامه کاربردی با (یا بی) سوئیچینگ مخصوص آن برنامه می‌باشد. و این در واقع همان الگویی است که درباره‌اش صحبت شد. به عنوان مثال, از یک فرمان برای اجرای یک ویزارد, بعد از انتخاب این مورد در منوی New, استفاده کنید.

Data – در این مورد بعد از اجرای فرمان و ایجاد فایل جدید, فایلی متشکل از داده‌های مشخص ایجاد می‌شود. یک مقدار Reg_BINARY که حاوی داده‌های فایل جدید است. توجه داشته باشید چنانچه در کلید HKCR\(.filetype)\ShellNew داده‌های NullFile یا FileName وجود داشته باشند این مقدار در نظر گرفته نخواهد شد.

File Name – فایلی ایجاد می‌کند که نسخه‌ای از یک فایل مشخص است. و یک مقدار از نوع Reg_SZ می‌باشد که حاوی مسیر و نام فایلی است که باید کپی شود و با نام فایل جدید در مکان جدید در نظر گرفته شود. البته اگر فایل در فولدر Templates پروفایل کاربر باشد, می‌توانید مسیر را مشخص نکنید و فقط نام فایل را ذکر نمایید.

NullFile – یک فایل خالی ایجاد می‌کند و مقداری از نوع Reg_SZ است که هیچ داده‌ای در آن نیست. اگر NullFile موجود باشد ویندوز Data و FileName را نادیده می‌گیرد.



به عنوان تمرین می‌خواهیم گزینه Briefcase در منوی New را که در ۹۹.۹۹ درصد مواقع استفاده‌ای نمی‌شود را از این منو حذف کنیم برای این کار کافی است نام کلید HKCR\.bfc\ShellNew را به HKCR\.bfc\JS_ShellNew تغییر دهید تا این مورد از New حذف شود و برای برگرداندن آن نیز تنها کافیست JS_ShellNew را به نام قبلی آن باز گردانید. در این تمرین بعد از تغییر در محتوای رجیستری به طور مستقیم تغییرات مشاهده نمی‌شوند تا زمانی که ویندوز را مجبور کنید که خود را در این ناحیه به روز کند علاوه بر قطع و وصل مجدد اکانت کاربری می‌توانید با انتخاب یکی از گزینه‌های منوی New تغییرات را مشاهده کنید.

جمع بندی آنچه در بالا گفته شد:

هر نوع فایلی دارای یک پسوند یا انشعاب مختص به خود است مانند فایل‌های Text که از انشعاب ".txt" برای این گونه فایل‌ها استفاده می‌شود. که در ویندوز هر انشعاب شناخته شده‌ای مانند ".txt" در مسیر HKCR\.txt نگه داری تنظیمات می‌شود. حال هر انشعاب نیز داری یک کلاس است که می‌توانید کلاس یک انشعاب را از داده HKCR\.txt\default (در اینجا مثال txt. می‌باشد) بخوانید. ضمنا کلاس هر انشعاب هم در کلید اصلی HKCR ذخیره شده, مانند کلاس فایل‌های متنی که txtfile است و در مسیر HKCR\txtfile نگه داری تنظیمات می‌شود. مطالعه کلاس فایل‌ها از آن جهت مهم است که, رفتارهای یک فایل در سیستم عامل ویندوز شما از کلاس آن ناشی می‌شود. در ادامه به بررسی بیشتر کلاس فایل‌ها, داده‌های آنها همین طور زیر کلیدهای موجود در آنها, همان طور که برای خود انشعاب صورت گرفت, خواهیم پرداخت.

زیر کلیدها و مقادیر کلاس برنامه ها

کلاس برنامه‌ها, یک برنامه را تعریف و رفتارهای مرتبط با آن را مشخص می‌کند.(ذکر عبارت برنامه در اینجا بر انواع فایل‌ها و فولدرها و همین طور اشیا خاص دلالت دارد) این کلاس‌ها در HKCR\progid قرار دارند که progid شناسه برنامه است. و همان طور هم که قبلا گفته شد به عنوان مثال به کلاس HKCR\txtfile اشاره می‌کنم که نمایانگر یک کلاس از برنامه‌ها است. ویندوز کلیدهای انشعاب فایل‌ها را از طریق مقادیر پیش فرض آنها به کلاس برنامه‌ها مرتبط می‌کند. مقادیر پیش فرض کلاس برنامه نیز حاوی نام عمومی کلاس است. فرمت درست ID هر برنامه به صورت application.components.version است به عنوان مثال, Word.Document.8 یک ID درست می‌باشد. اما این فرمت همیشه مورد استفاده قرار نمی‌گیرد حتی به وسیله ویندوز XP. کلاس برنامه‌ها دارای مقادیر و زیر کلیدهای زیر هستند:

- **AlwaysShowExt** – این مقدار نوع Reg_SZ نشان می‌دهد که ویندوز اکسپلورر باید همیشه انشعاب (پسوند) فایل‌ها را نشان دهد, حتی اگر کاربر آنها را پنهان کرده باشد.
- **CurVer** – مقدار پیش فرض این زیر کلید حاوی ID برنامه جدیدترین نگارش است.
- **DefaultIcon** – مقدار پیش فرض این زیر کلید نماد تصویری پیش فرضی است که ویندوز برای فایل‌های مرتبط با این کلاس از برنامه‌ها نمایش می‌دهد. این مقدار می‌تواند یک رشته Reg_EXPAND_SZ یا Reg_SZ باشد, اما باید از فرمت file,index استفاده شود که file نمایانگر مسیر و نام فایل حاوی نماد تصویری است, و index نمایانگر ایندکس نماد تصویری موجود در فایل است. اگر ID دقیق منابع را بدانید می‌توانید از فرمت file,-resource استفاده کنید .برای پیدا کردن index یا کد resource یک نماد تصویری در یک فایل منبع می‌توانید از یک ویراستار متفرقه برای منابع استفاده کنید. اما در صورتی که یک فایل با انشعاب ".ico" داشتید فقط ذکر مسیر و نام فایل کفایت می‌کند.
- **FriendlyTypeName** – این مقدار نوع Reg_SZ نشان دهنده نام عمومی کلاس برنامه است و این مقدار را در ویندوز اکسپلورر خواهید دید. این مقدار در ویندوز XP پیش از مقدار پیش فرض کلاس برنامه قرار می‌گیرد که نگارش‌های پیشین ویندوز هنوز از آن استفاده می‌کنند و ویندوز XP از آن برای حفظ سازگاری با نگارش‌های پیشین استفاده می‌کند. مقدار پیش فرض کلاس برنامه و این مقدار می‌بایست برای حفظ یکپارچگی یکسان باشند. ویندوز XP عموما به جای یک رشته از یک منبع در این مقدار استفاده می‌کند. فرمت آن به صورت @file,-resource یا @file,-index است.
- **EditFlags** – این مقدار نوع Reg_DWORD کنترل می‌کند که ویندوز چگونه فایل‌های مرتبط با این کلاس از برنامه‌ها را مدیریت می‌کند. همچنین به وسیله مقدار EditFlags می‌توانید کنترل کنید که کاربران چگونگی برخی از جنبه‌های این کلاس از فایل‌ها را اصلاح نمایند. هر بیت در این مقدار نمایانگر یکی از تنظیمات است. در زیر ماسک * هر یک از این بیت‌ها شرح داده شده است.

*ماسک	شرح
0x00000001	کلاس فایل‌ها را مستثنی می‌کند
0x00000002	آن دسته از کلاس فایل‌ها, مثلا فولدرها, را نشان می‌دهد که با یک انشعاب خاص مرتبط نیستند

ماسک*	شرح
0x00000004	مشخص می‌کند که کلاس فایل‌ها یک انشعاب خاص دارد
0x00000008	از ویرایش مقادیر مرتبط با این کلاس از فایل‌ها در رجیستری جلوگیری می‌کند. کاربران قادر به افزودن و یا تغییر مقادیر موجود نخواهند بود
0x00000010	از حذف مقادیر مرتبط با این کلاس از فایل‌ها جلوگیری می‌کند
0x00000020	از افزودن فرامین جدید به کلاس فایل‌ها جلوگیری می‌کند
0x00000040	از تغییر فرامین جلوگیری می‌کند
0x00000080	از حذف فرامین جلوگیری می‌کند
0x00000100	از تغییر شرح کلاس فایل‌ها جلوگیری می‌کند
0x00000200	از تغییر نماد تصویری مرتبط با کلاس فایل‌ها جلوگیری می‌کند
0x00000400	از تغییر زمان پیش فرض جلوگیری می‌کند
0x00000800	از تغییر فرامین مرتبط با "نام" فرامین جلوگیری می‌کند
0x00001000	از اصلاح یا حذف فرامین جلوگیری می‌کند
0x00002000	از تغییر یا حذف مقادیر مرتبط با DDE جلوگیری می‌کند
0x00004000	-
0x00008000	از تغییر نوع محتوای مرتبط با کلاس فایل‌ها جلوگیری می‌کند
0x00010000	به کاربران امکان می‌دهد تا از فرمان "Open" کلاس فایل‌ها برای فایل‌های download شده استفاده کنند
0x00020000	کادر انتخاب Never Ask Me را غیر فعال می‌کند
0x00040000	مشخص می‌کند که انشعاب فایل‌ها همیشه نشان داده شود, حتی اگر کاربر انشعاب (پسوند) فایل‌ها را در کادر مکالمه Folder Option پنهان نماید
0x00080000	-
0x00100000	مشخص می‌کند که اعضای این کلاس از فایل‌ها به فولدر Recent Documents افزوده شوند

* **ماسک بیت bit Masks** – اگر با سیستم‌های عدد نویسی باینری و هگزادسیمال آشنا هستید لازم است ماسک بیت را نیز فرا بگیرید. در آینده شرح کامل آن را تحریر خواهم کرد اما اکنون:

اگر جایی با یک دستورالعمل ماسک بیت (مانند ماسک 0x00000001 عدد 256) روبرو شدید, منظور آن است که عدد مورد نظر را به مبنای باینری برده (۲۵۶ به باینری برابر ۱۰۰۰۰۰۰۰ می‌شود) و بیت متناظر با ماسک (که در اینجا اولین خانه از سمت راست عدد ۱۰۰۰۰۰۰۰ که برابر ۰ می‌شود) را در اصطلاح "On" کنید (در مورد مثال حاصل ۱۰۰۰۰۰۰۰۱ می‌شود و با برگرداندن آن به مبنای دسیمال حاصل ۲۵۷ خواهد شد).

یک مثال دیگر برای ماسک بیت؛ مثلا در جدول بالا نگاه کنید اگر خواستید که کاربران قادر به تغییر تنظیمات مربوط به کلاس برنامه‌تان نباشند کافیست ماسک 0x00000008 را به داده Editflag, موجود در کلاس برنامه‌تان, اعمال کنید. در صورتی که Editflag قبلی مثلا ۲۲ بود با ماسک بیت جدید باید نتیجه ۳۰ را ذخیره نمایید.(در واقع کافیست برای اینکه بفهمید کدام خانه را باید on کنید عدد معادل ماسک را از مبنای اکتان هشت هشتی به باینری ببرید و عدد حاصله را با باینری عدد مورد نظرتان or بولی کنید) برای ساده تر شدن کارتان می‌تونید از مد Scientific ماشین حساب ویندوز استفاده نمایید.

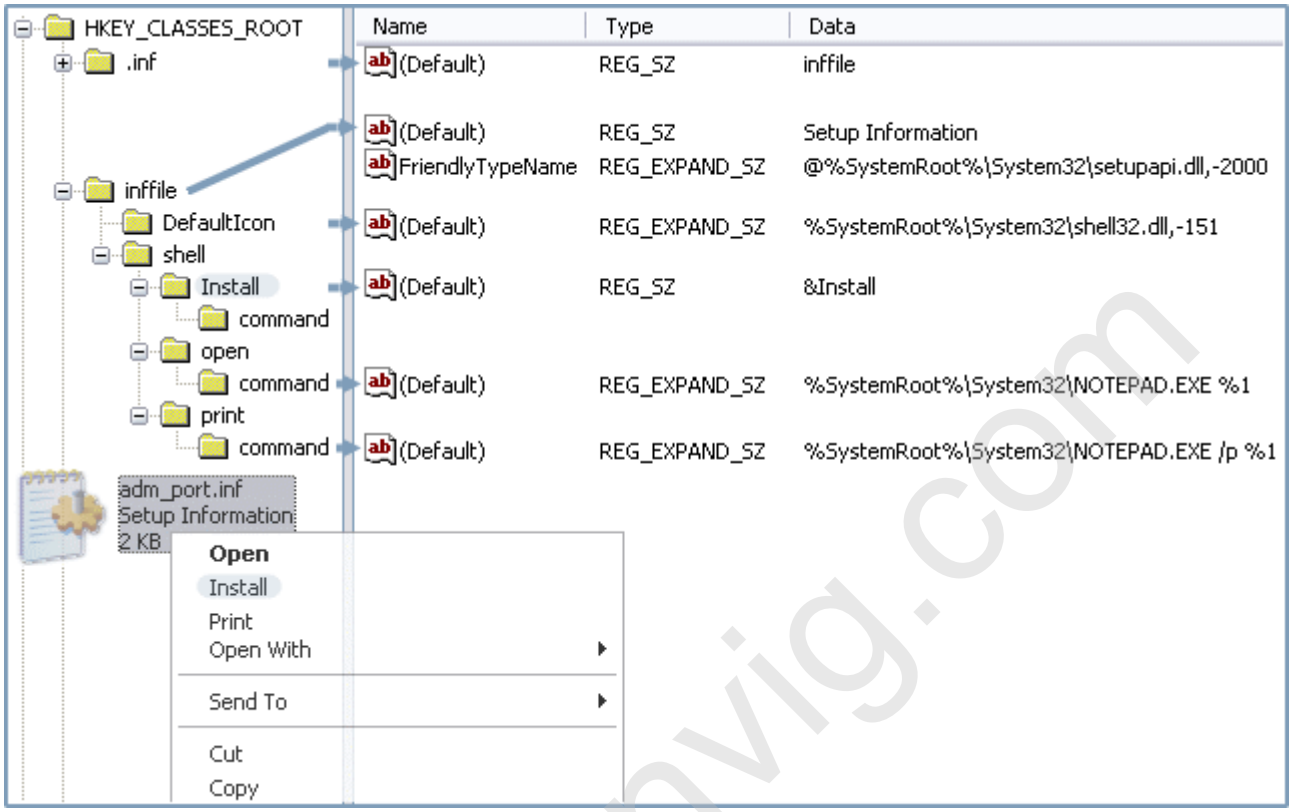
- ✓ یک نکته مهم آنکه در همه جای رجیستری اگر یک داده از نوع Reg_DWORD وجود داشت که به وسیله عملیات ماسک بیتی به وجود آمده بود, شما می‌توانید فقط این نوع مقادیر را با فرمت Reg_BINARY هم در رجیستری ذخیره نمایید (البته مقادیر با توجه به فرمت این دو نوع برای ذخیره متفاوت خواهد بود).

- **InfoTip** – این مقدار نوع Reg_SZ که حداقل برنامه‌نویسان آن را خوب می‌شناسند, حاوی پیام کوتاهی است که وقتی کاربران با ماوس به فایل یا فولدر مرتبط با این کلاس از برنامه ها اشاره می‌کنند, ویندوز برای این کلاس‌ها نمایش می‌دهد. همان طور که برای مقدار FriendlyTypeName شرح داده شد, این مقدار می‌تواند یک رشته با منبع باشد.
- **InShortcut** – این مقدار نوع Reg_SZ نشانگر آن است که فایل, یک میانبر است. ویندوز اکسپلورر این میانبر را با افزودن نماد تصویری میانبرها نمایش می‌دهد.
- **NeverShowExt** – این مقدار نوع Reg_SZ مشخص می‌کند که ویندوز اکسپلورر هرگز نباید انشعاب فایل را نمایش دهد, حتی اگر کاربر ویندوز اکسپلورر را به گونه‌ای پیکربندی کرده باشد که انشعاب فایل‌ها را برای انواع فایل‌های شناخته شده نمایش دهد.
- **Shell** – این زیر کلید حاوی فرامینی است که برای کلاس برنامه‌ها تعریف می‌شود. به عنوان مثال, کلاس txtfile فرامینی برای باز کردن و چاپ فایل‌های متنی تعریف می‌کند. این زیر کلید محل اعمال بیشتر تغییرات شخصی در HKCR است. کلاس فایل‌ها حاوی «افعالی» هستند که در حقیقت فرامینی می‌باشند که ویندوز برای انجام عملیات معین اجرا می‌کند. «افعال» با منوهای میانبری مرتبط هستند که به هنگام برگزیدن یک فایل با دکمه سمت راست ماوس مشاهده می‌کنید. هر یک از اقلام منوی میانبر یک «فعل» هستند.

افعال هر کلاس از برنامه‌ها در HKCR\progid\Shell هستند که حاوی یک زیر کلید برای فعل است. به عنوان مثال, HKCR\txtfile\Shell دارای زیر کلیدهای open و print است که برای افعال Open و Print می‌باشند. مقدار پیش فرض کلید Shell نشانگر نام فعل پیش فرض است (وقتی یک فعل پیش فرض است به هنگام دوبل کلیک برروی فایل, فعل مربوطه اجرا خواهد شد. مثلا در مورد کلاس txtfile فرمان open بلافاصله بعد از دوبل کلیک برروی یک فایل متنی اجرا می‌شود).

در مثالی دیگر فرض کنید مقدار پیش فرض کلید Shell یک کلاس edit باشد, این امر نشان می‌دهد که زیر کلید edit «فعل» پیش فرض است.

✓ اگر مقدار پیش فرض کلید Shell خالی (تهی) باشد, در آن صورت ویندوز از فعل open استفاده می‌کند.



افعال متعارف

افعال متعارف جزئی از سیستم عامل هستند. Open, Edit و Print مثال‌هایی از افعال متعارف می‌باشند. یک نکته جالب درباره این افعال آن است که ویندوز آنها را به طور خودکار با توجه به نیاز زبان‌های مختلف ترجمه می‌کند. فهرست ذیل افعال متعارف عمومی را نشان می‌دهد که برخی از آنها افعال ویژه‌ای هستند که کاربران در منوها نمی‌بینند:

- **Edit** – این فعل, همچون Open است اما به کاربران امکان می‌دهد که محتوای فایل‌ها را ویرایش کنند.
- **Explorer** – این فعل فولدر انتخابی را در ویندوز اکسپلورر باز می‌کند.
- **Find** – این فعل, Search Assistand را با فولدر انتخابی به عنوان محل جستجوی پیش فرض باز می‌کند.
- **Open** – این فعل عموما فعل پیش فرض است که یک فایل را در برنامه مرتبط با آن باز می‌کند.
- **Open As** – این فعل کادر مکالمه Open With را باز می‌کند.
- **Play** – این فعل مشخص می‌کند که محتوای فایل به جای اینکه صرفا باز شود و پخش آن به کاربر محول شود, پس از باز شدن فورا پخش شود.

- **Print** – این فعل سبب می‌شود که برنامه کاربردی محتوای فایل را چاپ کند و کار خود را به پایان برساند. برنامه‌های کاربردی برای اجرای این فعل باید حتی الامکان هیچ گونه رابط کاربری در اختیار کاربر قرار ندهند.
- **PrintTo** – فعل ویژه‌ای که از ویژگی کشیدن و رها کردن drag and drop, برای چاپگرها پشتیبانی می‌کند. کاربران این فعل را در منوهای میانبر نمی‌بینند.
- **Preview** – این فعل به کاربران امکان می‌دهد تا فایل‌ها را بدون باز یا ویرایش کردن, مشاهده نمایند. مشاهده تصاویر, به جای باز کردن آنها جهت ویرایش, مثالی از عملکرد این فعل است.
- **Poperties** – این فعل کادر مکالمه Name Properties را باز می‌کند.
- **RunAs** – فعل ویژه‌ای که به کاربران امکان می‌دهد تا با استفاده از account یک کاربر دیگر, فایلی را باز یا برنامه کاربردی را اجرا نمایند. کاربران با پایین نگه داشتن کلید Shift و برگزیدن فایل با دکمه سمت راست ماوس می‌توانند این فعل را در منوی میانبر تمام فایل‌ها مشاهده نمایند.

افعال مکمل را می‌توانید به هر کلاسی از برنامه‌ها بیفزایید. به عنوان مثال, فعل Edit in WordPad را می‌توانید برای فراهم کردن امکان ویرایش فایل‌های متنی در WordPad, بدون تغییر افعال پیش فرض, به کلاس txtfile بیفزایید. برای افزودن افعال به یک کلاس از برنامه‌ها, زیر کلید جدیدی برای آن در Shell ایجاد کنید. زیر کلید جدید, HKCR\progid\Shell\verb خواهد بود. سپس عبارتی را که می‌خواهید در منوی میانبر مشاهده کنید را به مقدار پیش فرض verb تخصیص دهید. برای اینکه یکی از کاراکترهای این عبارت را به عنوان hotkey معرفی کنید, "&" را پیش از آن قرار دهید. به عنوان مثال, "Open in &WordPad" سبب می‌شود که بتوان از W به عنوان hotkey برای این «فعل» استفاده شود.(این موضوع برای برنامه‌نویسان تازگی ندارد. شما در هنگام طراحی منوهای برنامه‌تان حتما تا کنون از این شیوه برای hotkey کردن بعضی حروف در فرمان‌ها استفاده کرده‌اید).

به تصویر قبل نگاه کنید. من مقدار پیش فرض کلید Install را به عمد آنجا قرار داده‌ام تا موضوع فوق برایتان روشن‌تر شود.

با ایجاد کلید verb همان طور که در بالا شرح داده شد کلمه Edit in WordPad در منوی میانبر نمایش داده می‌شود اما با انتخاب آن یک خطا اتفاق می‌افتد. هم اکنون باید تعیین کنیم که با انتخاب این فعل جدید چه فرمانی اجرا شود.
زیر کلید command را به verb بیفزایید. و فرمانی که می‌خواهید در نتیجه انتخاب فعل اجرا شود را به عنوان پیش فرض آن به کار ببرید. به تصویر قبل توجه کنید که چگونه هر یک از افعال هدف خود را با فرمانی که در مقدار پیش فرض زیر کلید command آمده, محقق می‌سازند.

مقدار پیش فرض command به قدری توضیح بیشتر نیاز دارد:

نخست, اگر در بین حروف مسیر و نام فایل برنامه از فاصله استفاده شده است, باید کل فرمان را در بین علائم نقل قول "" بنویسید.

- ✓ به برنامه‌سازان توصیه می‌شود در ضمن نوشتن اینگونه اطلاعات در رجیستری همیشه از علائم نقل قول استفاده کنند. در غیر این صورت وقتی ویندوز با مسیرهای همراه فضای خالی "فاصله" رو به رو می‌شود قسمت‌هایی از مسیر به عنوان فرمان تلقی شده به این ترتیب از طرف ویندوز خطا صادر می‌شود. در مواقعی هم فقط با یک بیپ کوچک در زمان اجرای فرمان اعلام می‌کند که یک جای کار می‌لنگد.

دوم, از 1٪ به عنوان یک گیرنده برای نام فایلی که با دکمه سمت راست ماوس برمی‌گزینید, استفاده کنید. به عنوان مثال, فرض کنید فرمان مورد نظر Notepad %1 باشد. حال اگر فایلی مثلا, C:\Salmple\TextS.txt را به وسیله دکمه سمت راست ماوس برگزینید و گزینه مورد نظر را انتخاب کنید, فرمان به شکل "Notepad C:\Salmple\TextS.txt" خواهد بود.

- ✓ در اینجا قرار دادن 1٪ در بین علائم نقل قول, می تواند خیال شما را از اجرای فرمان‌ها, برای فایل‌های با نام طولانی یا همراه فضای خالی, راحت کند (به این ترتیب مثلا به جای ذخیره Notepad %1 در رجیستری عبارت "Notepad "%1 را ذخیره نمایید).
- ✓ اگر برنامه‌نویسی می‌کنید می‌توانید در این موارد نوع داده پیش فرض کلید command را به شکل Reg_Expand_SZ در آورید تا بتوانید از متغییرهای محیطی مانند ٪SystemRoot٪ در آنها استفاده کنید.

افعال اضافی: این گونه افعال افعالی هستند که تنها زمانی که کلید Shift را به هنگام برگزیدن یک فایل با دکمه سمت راست ماوس, پایین نگه می‌دارید, مشاهده خواهید کرد. این افعال روش مفیدی برای جلوگیری از شلوغ شدن منوهای میانبر است. به عنوان مثال, افعالی که اغلب در منوهای میانبر مورد استفاده قرار نمی‌گیرند, پس از افزودن در پشت کلید Shift پنهان می‌شوند. برای این کار کافی است مقدار extended که از نوع Reg_SZ است و خالی می‌باشد را به زیر کلید Shell\verb بیفزایید.

تمرین – در این تمرین می‌خواهیم مالکیت یک انشعاب فرضی (ساختگی) را در سیستم به دست بگیریم. فرض کنید فایلی با پسوند ".not" داریم که محتوای آن متن است. می‌خواهیم شرایط را طوری ترتیب دهیم که با هر بار دوبل کلیک بر روی آن به وسیله برنامه Notepad باز و آماده ویرایش شود.

در مرحله اول باید کلید HKCR\.not را ایجاد کنیم. فرمان زیر را در منوی Run اجرا کنید تا این شاخه ساخته شود:

```
Reg Add "HKCR\.not" /v "" /t REG_SZ /d "noteText" /f
```

حال باید یک کلاس برای این انشعاب فرضی ایجاد کنیم که من noteText را در نظر گرفته‌ام. در مرحله بعد باید مقدار پیش فرض کلید HKCR\.not که در بالا ساخته شد برابر نام کلاس آن یعنی noteText شود. البته با اجرای فرمان بالا کلاس مورد نظر در انشعاب نام نویسی می‌شود.
حالا که نام کلاس فایل با انشعاب ".not" را می‌دانیم باید تنظیمات مربوط به این کلاس را در رجیستری اعمال کنیم.

فرمان زیر کلاس مورد نظر که کلید HKCR\noteText است را ایجاد می‌کند:

```
Reg Add "HKCR\noteText" /f
```

حال می خواهیم فعل open را به آن بیفزاییم فرمان‌های زیر را به ترتیب اجرا کنید:

```
Reg Add "HKCR\noteText\Shell\open" /v "" /t REG_SZ /d "Open Notepad" /f
Reg Add "HKCR\noteText\Shell\open\command" /v "" /t REG_SZ /d "notepad.exe %1" /f
```

خط اول فرمان‌های بالا, متن "Open Notepad" را به منوی میانبر فایل میفزاید و خط دوم فرمان اجرایی آن را ثبت می‌کند.

برای اینکه این تمرین را کامل انجام دهید و نتیجه را عینا مشاهده کنید روی دسکتاپ یا در هر مکان دیگری از کامپیوترتان یک فایل به نام "Test.not" ایجاد کنید تا اجرای فرمان‌ها را مشاهده کنید. اما فایل حاصله فاقد نماد تصویری منحصر به خودش است. فرمان زیر یک نماد تصویری برای فایل‌های با انشعاب فرضی ".not" در نظر می‌گیرد:

```
Reg Add "HKCR\noteText\DefaultIcon" /v "" /t REG_EXPAND_SZ /d %SystemRoot%\system32\shell32.dll,-152 /f
```

برای مشاهده تغییرات حاصل از فرمان بالا, باید ارتباط کاربری خود را با سیستم قطع و مجددا وصل کنید (یعنی Log off و بعد Log on) یا ویندوز را مجبور کنید خود را در ناحیه شناخت ایکن‌ها به روز برساند. برنامه‌نویسان می‌توانند از تابع SendMessageTimeOut استفاده کنند.

و اما شکل کامل تری از تمرین فوق را در یک فایل Reg آماده کرده‌ام. شما می‌توانید فایل Reg را با نام noteText.reg از بسته ضمیمه همراه این مقاله مشاهده و سپس به رجیستری خود وارد کنید.

➢ کلیدهای ویژه

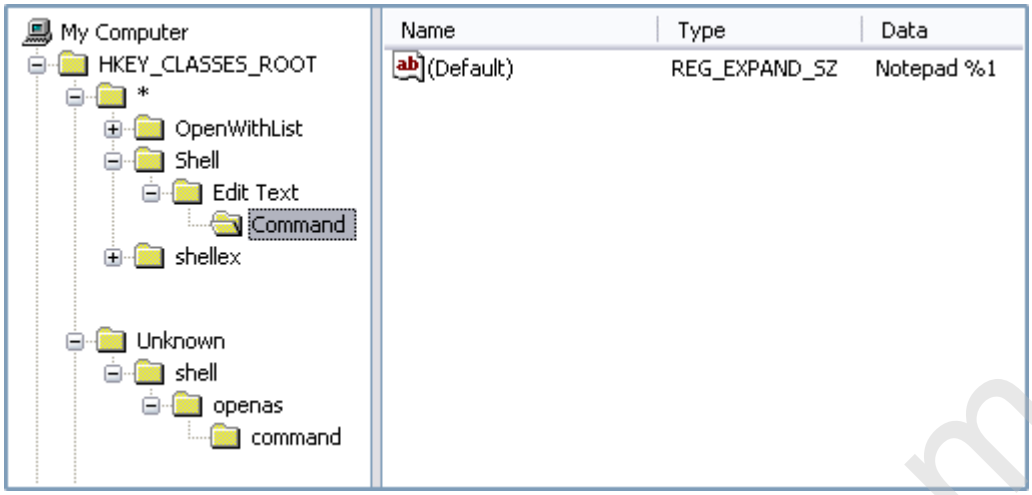
وقتی ویندوز ارتباطات فایل‌ها را جستجو می‌کند, کلیدهای زیر را به ترتیبی که نشان داده شده‌اند بررسی می‌کند. یعنی, تقدم محل‌هایی که در پایین فهرست قرار دارند, بالاتر از محل‌هایی است که در بالای فهرست قرار دارند !

HKCR\progid – این کلاس برنامه‌ای است که از طریق مقدار پیش فرض کلید انشعاب فایل با آن مرتبط است.

HKCR\SystemFileAssociations – این کلید برای تعریف انواع فایل‌های عمومی است و فرامینی را به هر یک از آنها مرتبط می‌کند. حتما نگاهی به این کلید بیندازید تا در مقاله‌های بعدی توضیح بیشتری راجع به آن بدهم.

HKCR* – این کلاس پایه برای انواع فایل‌های مختلف است. فرامین موجود در این کلید را در منوهای میانبر تمام فایل‌ها خواهید دید. کمی جلوتر یک تمرین جالب با استفاده از این کلید قرار داده‌ام.

HKCR\AllFilesystemObjects – این کلید فرامینی برای تمامی فایل‌ها و فولدرها تعریف می‌کند. در رابطه با این کلید هم در مقالات بعدی که در مورد کلیدهای مربوط به فولدرها و درایوها است, توضیحات کامل را خواهم داد.



تمرین – من قبلا در سایت ایرانویج برنامه‌ای با عنوان ویرایش همه فایل‌ها فقط با یک راست کلیک قرار داده بودم. در آن برنامه با تغییری در رجیستری این امکان برای کاربر فراهم می‌آمد تا با راست کلیک برروی هر نوع فایلی بتوانند با انتخاب گزینه Edit Text آن فایل را در محیط برنامه Notepad مشاهده و ویرایش کند. قول داده بودم روش کارش در رجیستری را برایتان شرح دهم. همان طور که در بالا مشاهده کردید برای اینکه فرمانی به منوی میانبر همه فایل‌ها بیفزایید, یک راه افزودن فرامین مربوطه, در کلید "HKCR*" است.

فرمان زیر را اجرا کنید تا EditText به منوی میانبر همه فایل‌های شما افزوده شود (به تصویر قبل توجه کنید):

```
Reg Add "HKCR\*\Shell\Edit Text\Command" /v "" /t REG_SZ /d "notepad.exe %1" /f
```

فصل هشتم : انواع پیشرفته داده

در این فصل با مفاهیم و عباراتی آشنا خواهید شد که در سطح بالاتری از داده‌های مورد نیاز یک کاربر متوسط قرار دارند. وی با طبقه بندی مطالب و فشرده سازی مفاهیم پایه برای آشنایی شما در بخش‌های قبل همین طور این فصل مهم, سعی بر آن دارم تا شما را برای فراگیری مطالب جدیدی که در آینده تحریر خواهند شد یاری نمایم. البته ممکن است برخی مطالب با سر فصل "رجیستری ویندوز" که هدف اصلی این مقاله است, رابطه‌ای نداشته باشند اما فراگیری آنها مسلما به ضرر شما نخواهد بود.

بیت ها و ماسک های بیتی bit masks

ویندوز و بسیاری از برنامه‌ها معمولا تنظیمات را در رجیستری گروه بندی و به یک عدد تبدیل می‌کنند. هر بیت در عدد حاصل, یکی از تنظیمات است. از این رو هشت مورد از تنظیمات در یک بایت, ۱۶ مورد در یک کلمه, و ... ذخیره می‌شوند. برخی اوقات در برابر تنظیمات, دسترالعمل‌هایی خواهید دید که مثلا 0x20 را به عنوان ماسک بیتی آن تنظیم معرفی می‌کنند. این بدین‌معناست که با فعال کردن بیت‌های نمایانگر 0x20 می‌توانید گزینه مورد نظر را "on" کنید. این کار به زودی مفهوم تر خواهد شد.

بیت‌های یک عدد باینری (دودویی) از راست به چپ و از صفر, شمارش می‌شوند. عدد نشان داده شده در شکل زیر 0x34 است (اعدادی که دارای پیشوند 0X صفر ایکس هستند هگزادسیمال یا شانزده‌شانزدهی می‌باشند, یعنی مبنای آنها ۱۶ است. برای محاسبه این گونه اعداد از مد Scientific ماشین حساب ویندوز استفاده کنید. اعداد هگزادسیمال در این ماشین حساب با گزینه رادیویی Hex مشخص می‌شوند).

0x34	0	0	1	1	0	1	0	0
بیت	7	6	5	4	3	2	1	0
ماسک بیت	0x80	0x40	0x20	0x10	0x08	0x04	0x02	0x01

در قسمت بالایی جدول, معادل باینری عدد فوق, و در زیر آن نیز شماره هر بیت نشان داده شده است. شماره سمت راست‌ترین بیت, صفر است. بیت‌های شماره ۲, ۴ و ۵ در این مثال یک, و سایر بیت‌ها نیز صفر می‌باشند. اگر دستورالعملی مبنی بر "on" کردن بیت شماره ۷ دیدید, عدد را از ۰۰۱۱۰۱۰۰ به ۱۰۱۱۰۱۰۰ تغییر دهید.

بسیاری از اوقات دستورالعمل‌هایی که مشاهده می‌کنید آنقدر جالب نیستند که شماره دقیق یک بیت را مشخص کنند, بنابراین باید قدری محاسبات انجام دهید. اغلب تنها چیزی که خواهید دید, یک ماسک بیتی است و می‌بایست مشخص کنید که ماسک نمایانگر چه بیت‌هایی است.

به عنوان مثال, برای اینکه بیت 0x40 را در عدد 0x34 بتوانید "on" کنید, هر دو عدد را به باینری تبدیل کنید, مشخص کنید که ماسک نمایانگر چه بیت‌هایی است. آن بیت‌ها را به یک تبدیل کنید سپس عدد را به سیستم عدد نویسی هگزادسیمال باز گردانید. همین کار را برای "off" هم انجام دهید, با این تفاوت که بیت‌های عدد مورد نظر به صفر تغییر می‌کنند. اما پس از مدتی به آسانی تشخیص خواهید داد که هر ماسک نمایانگر چه بیت‌هایی است. ماسک هر یک از بیت‌ها از راست به چپ عبارتند از:

0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x80

قسمت پایین جدول بالا این مطلب را نشان می‌دهد.

وقتی با بایت‌ها سرو کار دارید, "یک" باینری همچون "بلی" یا "درست" است, و "صفر" باینری نیز همچون "خیر" یا "نادرست" است. به عبارت دیگر مقادیر بولی هستند.

چنانچه از جبر بولی استفاده کنید, "on" و "off" کردن ماسک‌های بیتی حتی آسانتر نیز می‌شود. برای اینکه ماسک‌های متناظر با بیتی را در یک عدد "on" کنید, دو عدد را باهم OR کنید. همین طور برای اینکه آن را "off" کنید, بیت‌های موجود در ماسک را معکوس کنید, و سپس حاصل را با عدد مورد نظر AND کنید (یعنی "Not ماسک" و بعد حاصل And "عدد").

Little-Endean و Big-Endean

در عدد هگزادسیمال 0x0102 , بایت 0x01 با ارزش بالاتر و 0x02 نیز بایت با ارزش پایین‌تر است. ارزش بایت‌های سمت چپ بالاتر است, چرا که این ارقام در توان بزرگتری از ۱۶ ضرب می‌شوند. ارزش بایت‌های سمت راست پایین‌تر است, و ارزش ارقام از راست به چپ بیشتر می‌شود.

- ✓ برنامه‌ها به دو صورت اعداد را در حافظه ذخیره می‌کنند: Little-Endian یا Big-Endian .

وقتی یک برنامه, عددی را با استفاده از روش Big-Endian ذخیره می‌کند, ابتدا بایت‌های با ارزش بالاتر در حافظه ذخیره می‌شوند و سپس بایت‌های با ارزش پایین‌تر. به عنوان مثال وقتی عدد 0x01020304 با استفاده از روش Big-Endian در حافظه ذخیره می‌شود, این عدد به صورت بایت‌های Bytes: 0x01 0x02 0x03 0x04 در حافظه قرار می‌گیرد. معقول به نظر می‌رسد, این طور نیست؟

مشکل این روش در آن است که پردازنده‌های اینتل اعداد را در حافظه به این صورت ذخیره نمی‌کنند !!!

پردازنده‌های اینتل از معماری Little-Endian استفاده می‌کنند و این بدان معناست که ابتدا بایت‌های با ارزش پایین‌تر ذخیره می‌شوند, و سپس بایت‌های با ارزش بالاتر. از این رو عدد 0x01020304 به صورت Bytes: 0x04 0x03 0x02 0x01 ذخیره می‌شود.

اگر چه بیشتر ابزارهایی که به کار خواهید برد, تمام اعداد را نمایش می‌دهند Little-Endian یا Big-Endian , اما به هنگام مشاهده اعداد باینری باید توجه زیادی داشته باشید, چرا که ابزارها تغییر بایت‌ها را به طور خودکار تغییر نمی‌دهند. از این رو, اگر عدد 0x34 0x77 را در یک مقدار باینری ببینید, می‌بایست به خاطر داشته باشید که ترتیب بایت‌ها را تغییر دهید تا به 0x7734 برسید.

انواع داده‌ها Types Values

در بخش مقدمه این مقاله شما با انواع داده‌ای REG_SZ, REG_DWORD , REG_BINARY و REG_EXPAND_SZ آشنا شدید, در این بخش چند نوع داده‌ای جدید را معرفی می‌کنم:

REG_DWORD_BIG_ENDIAN : داده این نوع مقادیر double-word ای هستند که در آنها ابتدا بایت‌های با ارزش بالاتر در حافظه ذخیره می‌شوند. ترتیب بایت‌ها, عکس ترتیب ذخیره سازی آنها در مقادیر نوع REG_DWORD است. به عنوان مثال عدد 0x11220344 به صورت 0x11 0x22 0x03 0x44 در حافظه ذخیره می‌شود. این نوع داده‌ها را در معماری‌های مبتنی بر اینتل نخواهید دید.

REG_DWORD_LITTLE_ENDIAN : مقادیر double-word ای که در آنها ابتدا بایت‌های با ارزش پایین‌تر در حافظه ذخیره می‌شوند. این نوع داده‌ها مشابه REG_DWORD هستند و چون معماری‌های مبتنی بر اینتل, اعداد را با این فرمت ذخیره می‌کنند, متداولترین فرمت عدد در ویندوز است. به عنوان مثال 0x11220344 به صورت 0x11 0x22 0x03 0x44 در حافظه ذخیره می‌شود. ویراستار رجیستری قابلیت ایجاد مقادیر REG_DWORD_LITTLE_ENDIAN را فراهم نمی‌کند, چون این نوع مقادیر مشابه REG_DWORD در رجیستری هستند.

REG_FULL_RESOURCE_DESCRIPTOR : فهرست منابع برای یک وسیله سخت افزاری یا نرم‌افزاری. این نوع داده برای Plug and Play از اهمیت خاصی برخوردارند, اما در کارهایی که با رجیستری انجام می‌دهید چندان کاربرد ندارند. ویراستار رجیستری نیز روشی برای ایجاد این نوع مقادیر فراهم نمی‌کند, اما امکان نمایش آنها را برایتان فراهم می‌نماید. برای مشاهده مثالی از این نوع داده‌ها, مقادیر موجود در زیر کلیدهای مسیر HKEY_LOCAL_MACHINE\HARDWARE\DESCRIPTION را کاوش کنید.

REG_RESOURCE_LIST : فهرست مقادیر REG_FULL_RESOURCE_DESCRIPTOR. ویراستار رجیستری امکان مشاهده آنها را فراهم می‌کند, اما ویرایش آنها در ویراستار ممکن نیست.

REG_RESOURCE_REQUIREMENTS_LIST : فهرست منابعی که یک وسیله سخت افزاری نیاز دارد.

REG_LINK : یک نوع ارتباط. این نوع مقادیر را نمی‌توان ایجاد کرد چون مخصوص سیستم هستند و قالب آنها یونی‌کد می‌باشد.

REG_MULTI_SZ : مقادیر باینری که حاوی فهرستی از رشته‌ها هستند. ویراستار رجیستری در این نوع هر رشته را در یک سطر نمایش می‌دهد و امکان ویرایش این فهرست‌ها را فراهم می‌کند. هر کاراکتر تهی (0x00) در رجیستری رشته‌ها را از یکدیگر جدا می‌کند, و دو کاراکتر تهی پایان فهرست را نشان می‌دهند.

REG_QWORD : مقادیر چهار جمله‌ای (quardruple-word یا ۶۴ بیتی). این نوع داده مشابه REG_DWORD هستند, با این تفاوت که به جای ۳۲ بیت, ۶۴ بیتی می‌باشند. تنها نگارش ۶۴ بیتی ویندوز XP از این نوع مقادیر پشتیبانی می‌کند. این نوع مقادیر را می‌توانید به صورت دسیمال یا هگزادسیمال مشاهده و ویرایش کنید. 0xFE02000110010001 مثالی از این نوع مقادیر است.

REG_NONE : مقادیری که فاقد نوع تعریف شده هستند.

داده‌ها در مقادیر باینری

از بین تمامی مقادیر موجود در رجیستری, مقادیر باینری از نظر "سر راست" بودن از درجه پایین‌تری برخوردارند. وقتی یک برنامه کاربردی مقادیر باینری را از رجیستری می‌خواند, درک مفهوم و معنای آن بر عهده خود برنامه است. این بدان معناست که برنامه‌های کاربردی می‌توانند داده‌ها را با توجه به ساختارهای داده‌ای خودشان در مقادیر باینری ذخیره کنند, و این ساختارهای داده‌ای برای شما و هر برنامه کاربردی دیگر بی‌مفهوم خواهد بود.

همچنین برنامه‌های کاربردی اغلب داده‌های REG_DWORD و REG_SZ را در مقادیر REG_BINARY ذخیره می‌کنند که این امر پیدا کردن و درک مفهوم آنها را مشکل می‌سازد. در حقیقت, بسیاری از برنامه‌سازان (برنامه‌ها) مقادیر REG_DWORD و REG_BINARY را به جای یکدیگر به کار می‌برند. از این رو, با در نظر داشتن این مطلب که کامپیوترهای مبتنی بر اینتل از معماری little-endian استفاده می‌کنند, مقدار باینری 0x01 0x02 0x03 0x04 و مقدار REG_DWORD با داده 0x04030201 دقیقا یکسان هستند.

اینک می‌خواهم کارها را دشوارتر کنم. رجیستری در حقیقت همه مقادیر را به صورت باینری ذخیره می‌کند. API رجیستری هر نوع مقدار را با یک عدد شناسایی می‌کند (همان ثابت در برنامه‌نویسی) و من می‌خواهم که آن را همچون "شماره نوع" در نظر بگیرید. وقتی کلیدها را به فایل‌های REG صادر می‌کنید (موضوعی که در فصل چهارم : بارگذاری فایل‌های Hive فراگرفتید) بیشتر متوجه "شماره نوع" خواهید شد. به عنوان مثال, وقتی یک مقدار REG_MULTI_SZ را به یک فایل REG صادر می‌کنید, ویراستار رجیستری یک مقدار باینری را با "شماره نوع ۷" می‌نویسد. عموما, "شماره نوع" مرتبط با نوع مقادیر در خود ویراستار اهمیت زیادی ندارد, چرا که ارجاع به آنها از طریق نام آنهاست, اما اطلاعات جدول زیر برای برنامه‌نویسان بسیار با اهمیت است:

REG_NONE	= 0
REG_SZ	= 1
REG_EXPAND_SZ	= 2
REG_BINARY	= 3
REG_DWORD	= 4
REG_DWORD_LITTLE_ENDIAN	= 4
REG_DWORD_BIG_ENDIAN	= 5
REG_LINK	= 6
REG_MULTI_SZ	= 7
REG_RESOURCE_LIST	= 8

اسرار رجیستری

کاری از جواد سلطانی

۱۳۸۴-۱۳۸۵

به سفارش سایت ایرانویج

www.Iranvig.com