

## Re: Registry access via WMI in C#

**Source:**

<http://www.derkeiler.com/Newsgroups/microsoft.public.dotnet.framework.aspnet.security/2003-01/2448.html>

---

**From:** Alek Davis ([stop\\_spam\\_alek.davis@intel.com](mailto:stop_spam_alek.davis@intel.com))

**Date:** 01/14/03

From: "Alek Davis" <[stop\\_spam\\_alek.davis@intel.com](mailto:stop_spam_alek.davis@intel.com)>

Date: Tue, 14 Jan 2003 09:29:31 -0800

Michael,

I am just trying to understand why you are hooked on WMI. In general, WMI is not recommended if you can accomplish the same operation using some other methodology, because WMI is really slow. For example, getting a serial number of a hard drive using WMI is 4000 (four thousand) times slower than using native Win32 API via P/Invoke. So, unless you need WMI for something you cannot accomplish otherwise, do not use it. If this is how you did it before, it does not mean that you must continue doing it if there is a better way.

I don't really know what WMI uses behind the scene. Most likely it is DCOM, so you may be right. OpenRemoteBaseKey probably uses standard Win32 registry API (e.g. RegOpenKey, or whatever it is) via P/Invoke (which many of .NET classes do), and this API may or may not use DCOM. I just do not know. And why would you care?

— Alek

"Dr. Strangedub" <[strangedub@netscape.net](mailto:strangedub@netscape.net)> wrote in message news:3e243846.516920842@news.rsvl.unisys.com...

> *Thanks Alex –*

>

> *Well, after doing a bit of searching and tinkering yesterday, I did get*

> *OpenRemoteBaseKey to work for me. As mentioned before, our existing .asp pages*

> *use WMI (StdRegProv), so I figured we needed to code up analogous logic in C#...*

> *Now, OpenRemoteBaseKey doesn't appear to use WMI, so I'm not exactly sure how it*

> *works. (I know WMI uses DCOM under the covers – so perhaps this must also rely*

> *on DCOM?)*

>

> *–Michael Rose*

> Unisys Corp  
>  
> On Mon, 13 Jan 2003 17:21:22 -0800, "Alek Davis"  
> <[stop\\_spam\\_alek.davis@intel.com](mailto:stop_spam_alek.davis@intel.com)> wrote:  
>  
>>  
>> Why don't you simply use RegistryKey.OpenRemoteBaseKey method to open a  
key  
>> on remote machine? This sounds pretty straight-forward.  
>>  
>> -- Alek  
>>  
>> "Dr. Strangedub" <[strangedub@netscape.net](mailto:strangedub@netscape.net)> wrote in message  
>> news:3e233198.449674677@news.rsvl.unisys.com...  
>>> We are in the process of converting some specialized ASP web  
applications  
>> from  
>>> asp (using VBScript / Javascript) to ASP.NET (using C#). One of the  
>> functions  
>>> of our applications is to update the Windows registry on a remote host  
>> (either  
>>> the host where the web exists, or another host). Today the server-side  
>> VBScript  
>>> uses WMI to access the registry via the StdRegProv class as follows:  
>>> ==  
>>> Set objRegistry = GetObject("winmgmts:root\default:StdRegProv")  
>>> lAccess = KEY\_NOTIFY + KEY\_QUERY\_VALUE + KEY\_SET\_VALUE  
>>> Err.Clear ' Initialize error status  
>>> IRC = objRegistry.CheckAccess(HKEY\_LOCAL\_MACHINE, shortPath,  
>> lAccess,bGranted)  
>>> ==  
>>>  
>>> I would appreciate any help (code samples, links, other docs) in coding  
>> this  
>>> logic in C#. We picked up the string "winmgmts:root\default:StdRegProv"  
>> used on  
>>> the GetObject call from an example at the online MSDN site -- and there  
>> seemed  
>>> to be no real explanation of how this string gets us to a Namespace.  
>>>  
>>> I know we could simply use the Microsoft.Win32 Namespace if our  
registry  
>> access  
>>> was limited to the localhost (i.e., the web server host), but we really  
do  
>> need  
>>> to do remote updates as well.  
>>>  
>>> Thanks in advance for any help on this.  
>>>  
>>> -Michael G. Rose

microsoft.public.dotnet.framework.aspnet.security: Re: Registry access via WMI in C#

> >> *Unisys Corp.*  
> >> *michael.rose@unisys.com*  
> >>  
> >>  
> >  
> >  
>