

Re: Getting logged in user from a service?

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2007-06/msg00822.html>

- *From:* "Larry Smith" <no_spam@xxxxxxxxxxxxx>
 - *Date:* Tue, 5 Jun 2007 16:26:21 -0400
-

Not at all, WMI is client/server based using DCOM, you call a service and the service executes the service call, when WMI needs to "enable" a privilege (note that I said 'enable'), it's up to the caller to ask the service to enable the required (whatever this one may be) privilege, the user doesn't need to know the "privilege" required, WMI know which one as it's stored in it's metabase.

In the exceptional case (there are only a few) that a call requires a privilege that is not held by the WMI account (say "Network Service"), then it's up to the caller to run as a more privileged user (or get a stronger logon token) and ask WMI to impersonate when executing the service call.

Part of my argument has been in response to these (sometimes) confusing comments about security. All WMI would be doing here is enabling privileges that are already present in the caller's token (not its own). That's what Ben Voight was referring to in his earlier response. It's a minor convenience but one that's expected over the native OS since WMI sits on top of it (providing higher-level services). If you don't have the required privilege(s) in the first place however then WMI can't help you. Nor should impersonation be required here since it's not needed. Your earlier comments such as "... most of these things are taken care of by the framework and it's underlying services" and "... WMI makes it possible to call OS services without YOU having the need to run with these elevated privileges" are therefore misleading. Security is still an issue that needs to be considered but you've been implying otherwise. In any case, I can't counter arguments about ease-of-use in general. Higher-level services are inherently easier to use by their very nature so it's compelling to champion their use. They're not always the right tool for the job however. The entire crux of my argument centers around that and their place in a service in particular. I find your earlier arguments about monolithic services as self-defeating in fact since there's no special threshold involved here. Nor is memory consumption (necessarily) an accurate indicator of just how big these apps really are (since memory is affected by other things such as the amount of client data currently being cached). A small footprint is therefore a relative term. Imagine how much larger and slower those services would be if they actually relied on .NET instead of C++ and the native OS (which they most likely do at the core). Do you think MSFT or most organizations would

Re: Getting logged in user from a service?

even consider .NET for most services? Not likely.