

# Is RunAsInvoker a secret, even higher UAC setting?

 [devblogs.microsoft.com/oldnewthing/20161117-00](http://devblogs.microsoft.com/oldnewthing/20161117-00)

November 17, 2016



Raymond Chen

The comments in this oldnewthing article reveal a secret, even higher UAC setting:

RunAsInvoker <https://t.co/tTajZx6FIR>

— Esme (True Any Pronouns) (@madewokherd) [August 18, 2016](#)

Actually, `RunAsInvoker` is a secret, even *lower* UAC setting.

What `RunAsInvoker` does is to ignore any elevation request in the application's manifest and treat the manifest as if it had said

```
<requestedExecutionLevel level="asInvoker" uiAccess="false" />
```

which is the default behavior. The program simply runs with the same privileges as the code that launched it. There is no attempt to elevate.

This means that if you run the program from an elevated command prompt, then the program stays elevated. If you run the program from a non-elevated command prompt, then the program stays non-elevated.

Try it. Make sure RegEdit is not already running, then open a non-elevated command prompt and set `__COMPAT_LAYER=RunAsInvoker`, and then run `regedit` from that command prompt. The resulting copy of RegEdit is running without administrator privileges. You can see this by trying to edit something in HKLM.

While it's true that `RunAsInvoker` suppresses UAC prompts, that's true because `RunAsInvoker` doesn't perform any elevation. If you aren't performing any elevation, then naturally you don't need an elevation prompt. If the resulting process is elevated, then it means that the calling process was already elevated. You were already on the other side of the airtight hatchway.

[Raymond Chen](#)

**Follow**

