

# How can I log users off after a period of inactivity, rather than merely locking the workstation? Is there a “logoff” screen saver?

 [devblogs.microsoft.com/oldnewthing/20190723-00](https://devblogs.microsoft.com/oldnewthing/20190723-00)

July 23, 2019



Raymond Chen

Aaron Margosis had a customer who wanted to set up something like a “logoff” screen saver.

Specifically, they have a large number of machines that are shared by multiple users. By default, when the session goes idle, the workstation locks. If a new user wants to use the computer, that new user logs in, but the previous user’s session continues running. Repeat for a dozen cycles, and the system now has a dozen idle sessions sitting around. They were looking for a way to configure the system so that instead of locking an idle user, the system logs them off.

The system is special-purpose: Each user is signed in for only a few minutes, fifteen tops. If the user walks away from the computer after starting their task and before completing it, they can just sign back in and start over. Therefore, forcibly logging the user off will not result in significant loss of data.

One idea that didn’t work was setting up a scheduled task that triggers on idle. This doesn’t work because the definition of “idle” used by the task scheduler requires the system to be close to 0% CPU in order to be considered idle, in addition to the lack of user input.

Another idea was writing a custom screen saver that logs the user off, but Aaron was hoping for a solution that didn’t involve writing code, because that means somebody has to build it, deploy it, maintain it, recompile it for new architectures, all the stuff that comes with writing code.

One of my colleagues came up with a solution that relies only on things found lying around the house:

- Use policy to enforce a screen saver with the desired idle timeout. It doesn’t matter what the screen saver is, as long as it’s a screen saver. The *Blank* screen saver works fine for this purpose.

- In Security Settings, Advanced Audit Policy Configuration, System Audit Policies, Logon/Logoff, configure “Other Logon/Logoff Events” to audit Success events.
- Define a scheduled task as follows:
  - When running the task, use the following user account: **Users**.
  - Triggers: Begin the task: **On an event**; Settings: **Basic**; Log: **Security**; Source: **Microsoft Windows security auditing**; Event ID: **4802**.
  - Actions: **Start a program**; Program/script: **C:\Windows\System32\shutdown.exe**; Add arguments: **/l /f**.

This exploits the ability to trigger a process to run based on an entry in the event log. We specify that we want audit events to be logged for successful *Logon/Logoff* events. When event 4802 (“The screen saver was invoked”) occurs, we launch the `shutdown.exe` process with the `/l /f` command line. The `/l` option triggers a logoff, and the `/f` option forces the logoff, so the user cannot block the logoff by, say, leaving an unsaved Notepad document on screen.

I thought this was a really clever solution, exploiting the ability to trigger a program based on events in the event log.

Raymond Chen

**Follow**

