

Running Windows with No Services

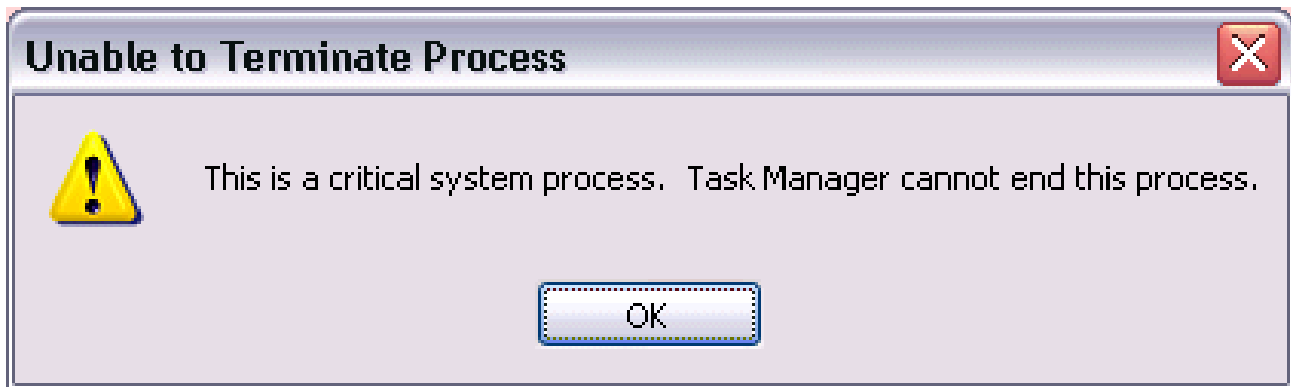
Mark Russinovich
(From Mark Russinovich Blog)

Running Windows with No Services

A Windows service provides functionality to the operating system and user accounts regardless of whether anyone is logged into a system. Windows XP comes with around four dozen services enabled by default, including ones that many people consider superfluous like Remote Registry, Alerter, and SSDP Discovery (Universal Plug and Play). A question many Windows administrators commonly have is therefore, which services can I safely disable? What if I told you that for at least basic functionality like Web surfing and application execution, Windows doesn't need any services? In fact, you can also do those things without system processes like Winlogon.exe, the interactive logon manager, and Lsass, the local security authority subsystem.

The following steps, which you must follow carefully to achieve a minimal Windows system, were derived by Dave Solomon through experimentation, and when he discovered that Windows was usable without all the core system processes we were dumbfounded. After figuring this out he and I polled senior Windows experts like the vice president of the Core Operating Systems Division, the technical lead of the Virtual PC team, and a lead Windows security architect to see if they thought that Windows would function at all, much less if Internet Explorer would work, without the support of Winlogon, Lsass, and services, and the unanimous answer was 'no'. Even after we showed them the demonstration I'm about to share with you they all thought that we'd staged some kind of trick.

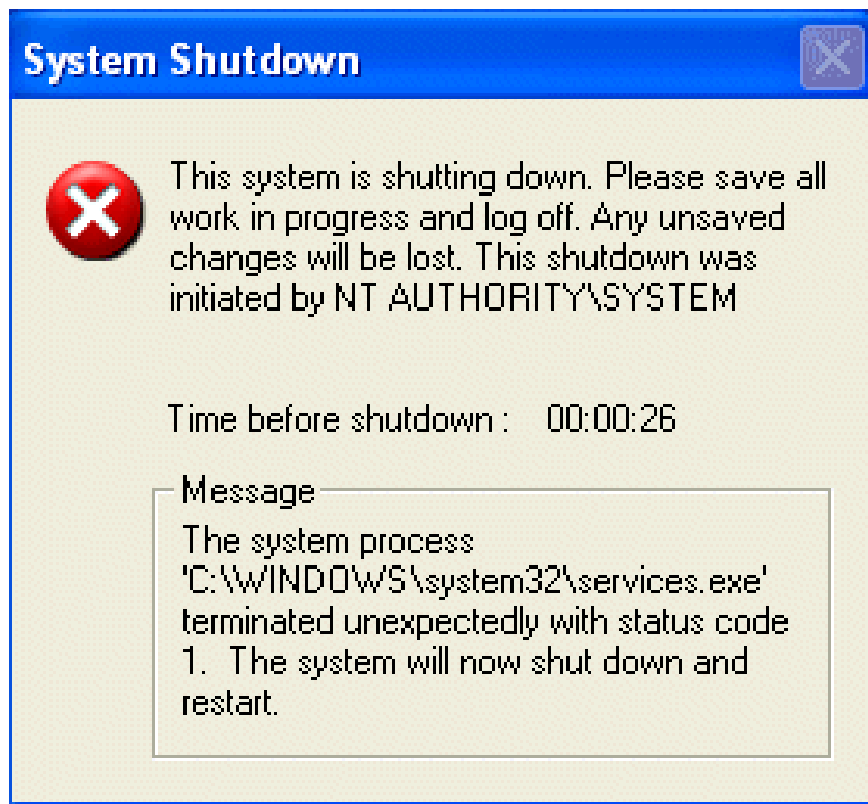
The first step to achieving a minimal Windows configuration is to kill the system processes I've mentioned. You can't use Task Manager for the job, however, because it has an internal list of processes that it considers critical and that it won't terminate. Try to kill Smss.exe, Winlogon.exe, Services.exe, Lsass.exe or Csrss.exe and you'll see this dialog:



So if you don't have it already download Process Explorer. To make things go more quickly uncheck the Confirm Kill entry in the Process Explorer Options menu. Then kill Smss.exe, the Session Manager process. The reason we start with Smss.exe is that Smss.exe watches the back of Winlogon, the process it creates during the boot, so if you terminate Winlogon first Smss.exe gets upset and blue screens the machine with an error indicating that the Windows logon process terminated unexpectedly. And if you kill Lsass or Services without killing Winlogon you'll see this dialog that Winlogon shows before it shuts down the system (you can abort the shutdown by running "shutdown -a"):

Running Windows with No Services

Mark Russinovich
(From Mark Russinovich Blog)



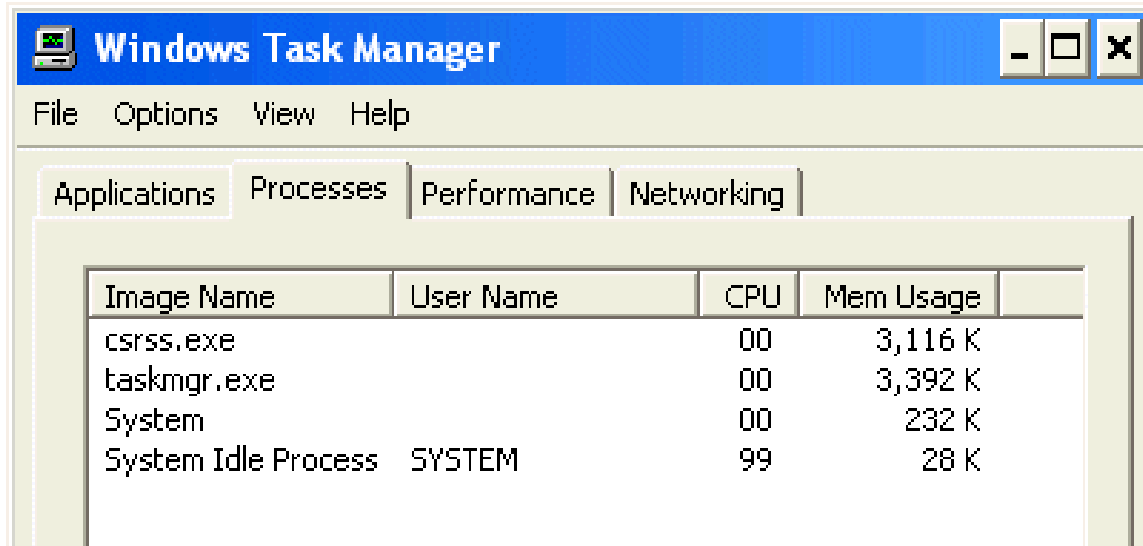
Once Smss.exe is out of the way select Winlogon and choose Kill Process Tree from in the Process menu. This terminates Winlogon.exe, Lsass.exe, Services.exe, and all the Windows service processes. We're almost done.

The next step is to kill all other standard processes except for Csrss.exe (and of course Process Explorer). Csrss.exe is the only process in the system that has the "critical process" bit set in its kernel process structure (EPROCESS) flags field. On the termination of a process with the flag set the kernel halts with a CRITICAL_PROCESS_DIED blue screen. Note that you won't be able to terminate the System Idle Process, System, Interrupts, or DPC processes. The Idle process isn't a real process and simply tracks the time when no thread is executing. The System process holds operating system kernel threads and device driver threads, and Interrupts and DPCs are artificial processes that Process Explorer uses to display interrupt and Deferred Procedure Call (DPC) activity.

Because Process Explorer shows the Interrupts and DPCs artificial processes switch to Task Manager at this point to get a real idea of what's actually running by activating the Run command in Process Explorer's File menu and entering "taskmgr". Then exit Process Explorer and look to Task Manager's Process tab. This is what you should see (themes disappear when the Svchost.exe process hosting the theming service terminates):

Running Windows with No Services

Mark Russinovich
(From Mark Russinovich Blog)



You have achieved minimal Windows: the only two processes, not including Task Manager, are System and Csrss.exe. You're now ready to start experimenting. Verify that you can surf the Internet by launching "iexplore" from Task Manager's Run command in its File menu. Then restart Explorer by running "explorer". You're done with Task Manager so you can exit it.

There will be a delay before Explorer redraws the desktop because it waits for the Service Control Manager (SCM) to signal the ScmCreatedEvent, which Services signals during its initialization. Below is the stack of the main Explorer thread waiting. The second parameter to WaitForSingleObject is a timeout value that's interpreted as milliseconds and 0xEA60 is 60,000 – 60 seconds:

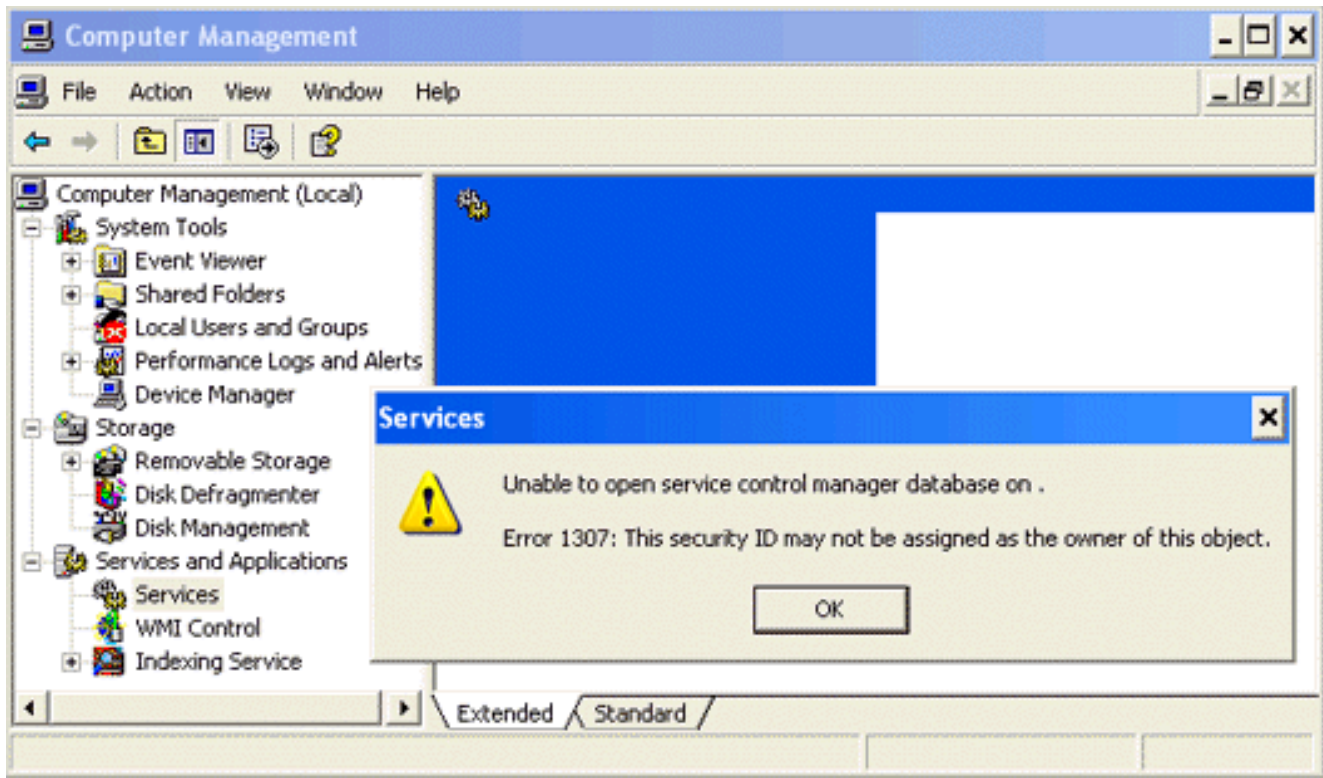
```
Args to Child
7c8025db 000000ac 00000000 ntdll!KiFastSystemCallRet (FPO: [0,0,0])
000000ac 00000000 0007feb4 ntdll!ZwWaitForSingleObject+0xc (FPO: [3,0,0])
000000ac 0000ea60 00000000 kernel32!WaitForSingleObjectEx+0xa8 (FPO: [Non-Fpo])
000000ac 0000ea60 7c8092ac kernel32!WaitForSingleObject+0x12 (FPO: [Non-Fpo])
00000000 7f016d50 7ffde000 explorer!WaitForSCMToInitialize+0x59 (FPO: [0,0,0])
7ffde000 00000000 00000000 explorer!UIInitializeWaitForSCM+0x8 (FPO: [0,0,0])
01 Timeout ) 00020648 explorer!ExplorerWinMain+0x132 (FPO: [Non-Fpo])
80000001 00120000 7ffde000 explorer!ModuleEntry+0x6d (FPO: [Non-Fpo])
0101e24e 00000000 78746341 kernel32!BaseProcessStart+0x23 (FPO: [Non-Fpo])
```

Once Explorer starts it clips the task bar off the bottom of the display so get it back by right-clicking on the barely visible task bar and applying the "Show Quick Launch" option. Notice that even though the task bar is fully visible it doesn't show the active windows.

With Explorer, the start menu and desktop back you can wander your system, trying various applications and utilities to see how they respond when there are no services running. There are many things that will work, but of course also many things that won't. For example, here's the Services node of the Computer Management MMC snapin displaying an expected error message:

Running Windows with No Services

Mark Russinovich
(From Mark Russinovich Blog)



What are the real limitations of running like this? Some will become obvious during your exploration, but a major one is that you won't be able to logoff (or shutdown) since neither Lsass nor Winlogon are running. Networking is also crippled, especially in a LAN, since accessing other computers requires the participation of Lsass in the cross-machine domain authentication process.

The bottom line is that this stripped-down Windows configuration is not practical, but makes a cool demonstration of just how little of Windows is required for basic functionality.

On a more personal note, I'm going to be in the Cape Canaveral area on Thursday, August 11, and so am calling out to any NASA employee Sysinternals fans to see if you'd be willing to arrange for a special tour of the space center.