Windows 7 Startup

Updated 7. August 2014 - 5:49 by Remah [1]

Introduction

This article is primarily aimed at interested users ... but it still helps to be more expert

This article is written for Windows users who want to know what is happening during Windows startup. My goal is to make Windows startup more easily understood by the average user. So I have provided a lot of basic information for those of you who aren't familiar with Windows terminology [2]. This article will be most useful to advanced users because it provides detailed startup sequences and examples. This could be helpful if you are trying to troubleshoot Windows startup problems so I have included some pointers for where to look for particular problems.

The article moves from background material to a simple overview which is expanded upon by looking at the various Windows processes with examples of their processing steps. Wherever possible I have confirmed the steps in a real example but I've still had to rely on other commentaries as I've only used the tools that would be used by a confident user. That's why there is no mention of advanced tools for programmers like kernel debuggers or the special debug version of Windows (a checked build).

At present this article is a work in progress with more detail to be added. It does not include Windows 8 which might be added later. At this stage, I don't intend to add information on resuming Windows from sleep or hibernation, Windows installation, safe mode and the other startup options. This keeps the guide a lot simpler.

This article only looks at Windows 7

Windows 7 is a good compromise between old and new versions of Windows. Although it is very similar to Vista but there are still very significant differences between versions and you will have to look elsewhere to understand those.

The examples use Windows 7 64-bit

The examples I am using are based on startup traces I ran on my test PC running Windows 7 64-bit. 64-bit Windows is the future and I need to highlight how 64-bit Windows handles 32-bit processes.

The traces provide some timings to give you a relative indication of the time taken by the startup phases and it also provides you with the option to compare it with your own Windows startup. Just be aware that there are several reasons why your relative timings may be considerably different to mine.

Icons highlight key issues

I have included icons to highlight various topics of interest so you can scan the detail more easily.

32-bit and 64-bit Windows have some significant differences:
32 applies to 32-bit Windows only.
64 applies to 64-bit Windows only.

On the few occasions where the Windows Edition is important:
⊞ indicates differences between Windows Editions

If you are troubleshooting then look for these icons:
⚠ indicates a known troubleshooting issue.
⚙ provides information on diagnostic tools and their output.
⚠ indicates a critical process (processes set this status themselves) which can crash or halt Windows if it fails.

How to view the larger tables and diagrams

I am trying to pack a lot of information into some of the tables and diagrams so they look better in a display that is 1600 pixels wide. If your display is smaller, particularly if it is below 1200 pixels wide, then you can use the 'Printer-friendly view [3]' to remove the sidebars so you can read them more easily.

What you should know

Kernel mode has to start before User mode

You should understand that Windows has two modes of operation which largely determine the sequence of startup activities:

- User mode is what we work with. It runs our applications programs on top of layers of services and subsystems that are mainly provided by the Windows Kernel mode.
- Kernel mode sits between the hardware and our application programs, supervises the running of the computer, and provides subsystems and services for User-mode programs to use. Kernel mode startup roughly corresponds to the time that the "Starting Windows" splash screen is

displayed.

User mode depends upon Kernel mode so the Windows Kernel has to be loaded first and only later in the startup process are the User-mode sub-systems and services loaded. During Kernel-mode startup there is very little for you to see apart from the Starting Windows screen. During User-mode startup the logon screen and the desktop screens are almost always visible.

Windows does some things to spread the startup load

Windows startup processing is more sequential at the start and becomes more parallel. An important attribute of this division is that Kernel mode is mainly sequential because there are many dependent processes and prerequisites. So the Kernel-mode sub-systems are largely built up in a specific order. Whereas User mode is a virtual explosion of processes spawning other processes and almost always running in parallel because most of the dependencies are incorporated in the Kernel mode.

Windows also has ReadyBoot and prefetch to ensure that needed components are ready to memory when needed to load or start.

To maintain responsiveness, Windows delays the starting of many programs. Services and drivers good examples of this:

- Boot start and system start drivers start during the kernel-mode phase.
- Auto start and on demand services start later in the user-mode when the Service Control Manager (SCM) is running.
- Some services have a delayed-start attribute or have dependencies so SCM delays their start until 2 minutes after the SCM starts. In the meantime, other startup processes like user logon have started more quickly.

Critical processes must keep running

Windows has many critical processes that cause Windows to crash if they fail. That is unless Windows has booted in debugging mode in which case the debugger will appear:

- ⚠ System process for the Kernel (NTOSKrnl.exe)
- ⚠ The Session Manager Sub-System (SMSS.exe)
- ⚠ Client Server Runtime Sub-System (CSRSS.exe)
- ⚠ Windows Logon (WinLogon.exe)
- ⚠ Windows Init (WinInit.exe)
- ⚠ Windows Logon User Interface Host (LogonUI.exe) - RDP only?
- ⚠ Local Security Authority Process (lsass.exe)
- ⚠ Service Control Manager (Services.exe)

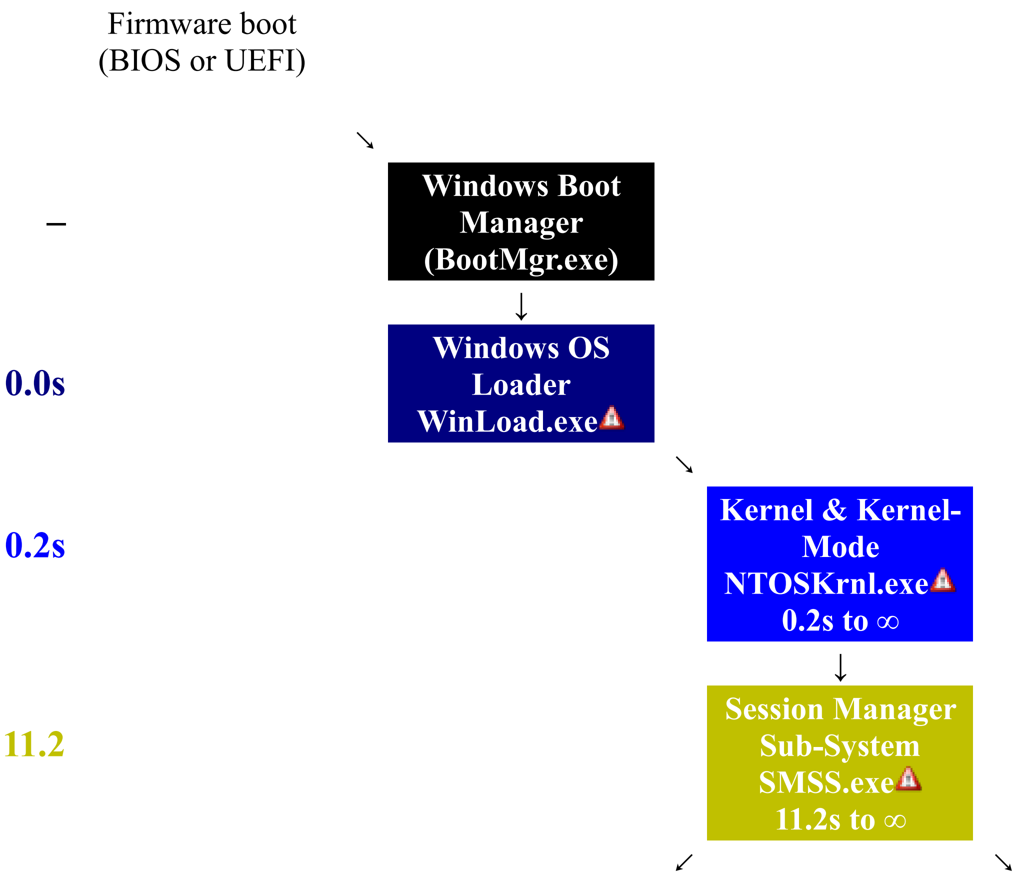
- ⚠ Service Host (svchost.exe) with RPCSS or Dcom/PnP.
- ⚠ Desktop Window Manager (DWM.exe)
- plus other optional processes such as performance monitoring or Internet Information Server (ISS),

Computer Startup

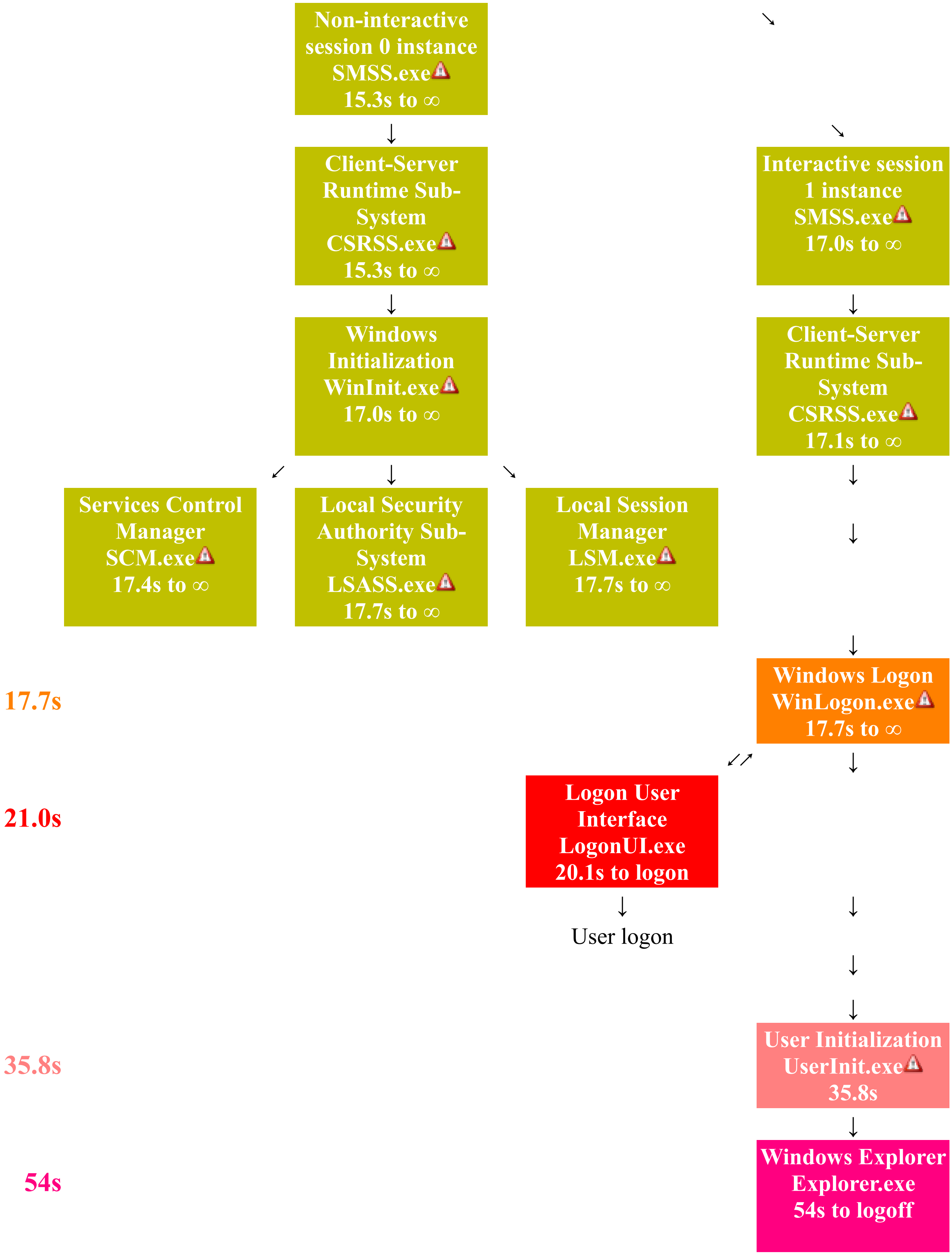
Diagram 1 is a simple flowchart of the major programs that control the sequence of a normal Windows startup. There are many more essential programs that are initialised and run by these programs. I didn't include any of them although many are listed in the more detailed startup steps later in this article.

This diagram matches Diagram 2, 'Phases of Windows Startup for immediate logon'. The colors here largely match the Boot Phase scheme in Diagram 2. As do the times on the left which startup trace times in seconds. Until the user logon screen appears at 35.8 seconds, this diagram also matches Diagram 3, 'Phases of Windows Startup for a delayed logon'.

Diagram 1 - An overview of Windows startup



Firmware boot
(BIOS or UEFI)
–
Windows Boot Manager (BootMgr.exe)
0.0s
Windows OS Loader WinLoad.exe⚠
0.2s
Kernel & Kernel-Mode NTOSKrnl.exe⚠ 0.2s to ∞
11.2
Session Manager Sub-System SMSS.exe⚠ 11.2s to ∞



Non-interactive session 0 instance
SMSS.exe⚠
15.3s to ∞

Client-Server Runtime Sub-System
CSRSS.exe⚠
15.3s to ∞

Windows Initialization
WinInit.exe⚠
17.0s to ∞

Services Control Manager
SCM.exe⚠
17.4s to ∞

Local Security Authority Sub-System
LSASS.exe⚠
17.7s to ∞

Local Session Manager
LSM.exe⚠
17.7s to ∞

Interactive session 1 instance
SMSS.exe⚠
17.0s to ∞

Client-Server Runtime Sub-System
CSRSS.exe⚠
17.1s to ∞

17.7s

Windows Logon
WinLogon.exe⚠
17.7s to ∞

21.0s

Logon User Interface
LogonUI.exe
20.1s to logon

User logon

35.8s

User Initialization
UserInit.exe⚠
35.8s

54s

Windows Explorer
Explorer.exe
54s to logoff

100s		Windows startup is complete but boot start services usually continue to load.
200s		User startup completes when there is no startup disk activity for 10 seconds.

Important points to note:

- The three processes started by WinInit - SCM, LSASS and LSM - all start about the same time (17.4 and 17.7 seconds) as WinLogon (17.7 seconds) but I have chosen to separate them to emphasize the separation of non-interactive and interactive sessions.
- Some processes end when they pass control to the next process in the flowchart. But most processes continue to run for longer and many run until Windows is shutdown - I've indicated these with the infinity symbol (∞): NTOSKrnl, SMSS, CSRSS, SCM, LSASS, LSM, WinLogon. Explorer would also be in this list except that it only runs when a user logs on to an interactive session.
- I have indicated critical processes (⚠) that must run so Windows will run. You will notice that the processes that interact directly with users (LogonUI and Explorer) are not critical so if they fail they do not automatically crash Windows.
- Much of the time until the user logon screen (LogonUI) appears looks like it is spent starting the Session Manager (SCM) and creating the non-interactive and interactive sessions. But this is also the time when many kernel-mode sub-systems, the Windows APIs and the registry are also starting.

Booting your computer

For the sake of completeness I include this section as an overview of the process that takes place when you turn on your computer but before Windows is started. It won't be discussed again in the more detailed discussion of Windows startup.

Show more...

The Windows Boot Manager

The Windows Boot Manager, bootmgr.exe, can display a boot menu but I am describing the simplest startup process so those options are not discussed here. It is also not timed so it is not included in discussions of the example trace.

Boot Manager locates the Boot Configuration Data (BCD)

There remains one distinction which is where the Boot Manager locates the Boot Configuration Data (BCD):

- The BIOS locates the configuration in "\Boot\BCD" on the system volume.
- The EFI locates the configuration in the "\EFI\Microsoft\Boot\" directory on the EFI system partition.

Boot Manager runs any boot-time utilities

The Boot Manager can run utility programs to diagnosis problems or to perform maintenance. The Microsoft memory tester, memtest.exe, is probably the most useful example.

Boot Manager displays a boot menu if required

The Boot Manager displays a boot menu if more than one option is to be presented to the user.

Make sense of the Windows startup phases

There are several schemes for describing startup phases

The starting point for the following discussion of Windows startup is after the Windows Boot Manager, bootmgr.exe, has been loaded and control has been passed from the computer firmware, whether it was the BIOS or UEFI. Up until the point where the Windows Boot Manager hands over control to the Windows Loader there is no record of the time.

Form this point there are several ways of describing the Windows startup phases. You may find that the differing intervals and terminology are a hindrance when either reading articles about Windows startup or trying to interpret the diagnostic results from various tools. The remainder of this section illustrates the similarities and differences between these schemes before I discuss the startup components in more detail.

Startup phase schemes

Here is a description of each category used in Diagrams 2 & 3. You could say that they move from user-oriented on the left to more technical on the right but really only the first category is accessible to most users.

Visible to Users is what you see on your screen. These events normally occur some time after the start of the interval.

Boot Time is used in the Windows Event Manager and those statistics are available at any time.

Windows Focus indicates the process that is currently awaiting input and is usually visible. Focus is the graphical equivalent of the text-based cursor.

Boot Interval is used in the Boot Phase analysis of Windows Performance Analysis (WPA) when summarizing trace data provided by Windows Performance Recorder (WPR). It is very similar to the Boot Phase analysis which is different only in that the Pre Session interval is divided into the OS Loader and Kernel-mode initialization.

Boot Phase is widely used in Microsoft tutorials on analysing Windows startup. It is probably the most useful to understand simply because it is often used to describe what is happens when you delve more deeply into Windows startup.

The Drivers & Services category, as I've called it, is focused on kernel mode Plug and Play (PnP) Manager which loads devices and drivers in three main phases which correspond to the three categories of devices and drivers:

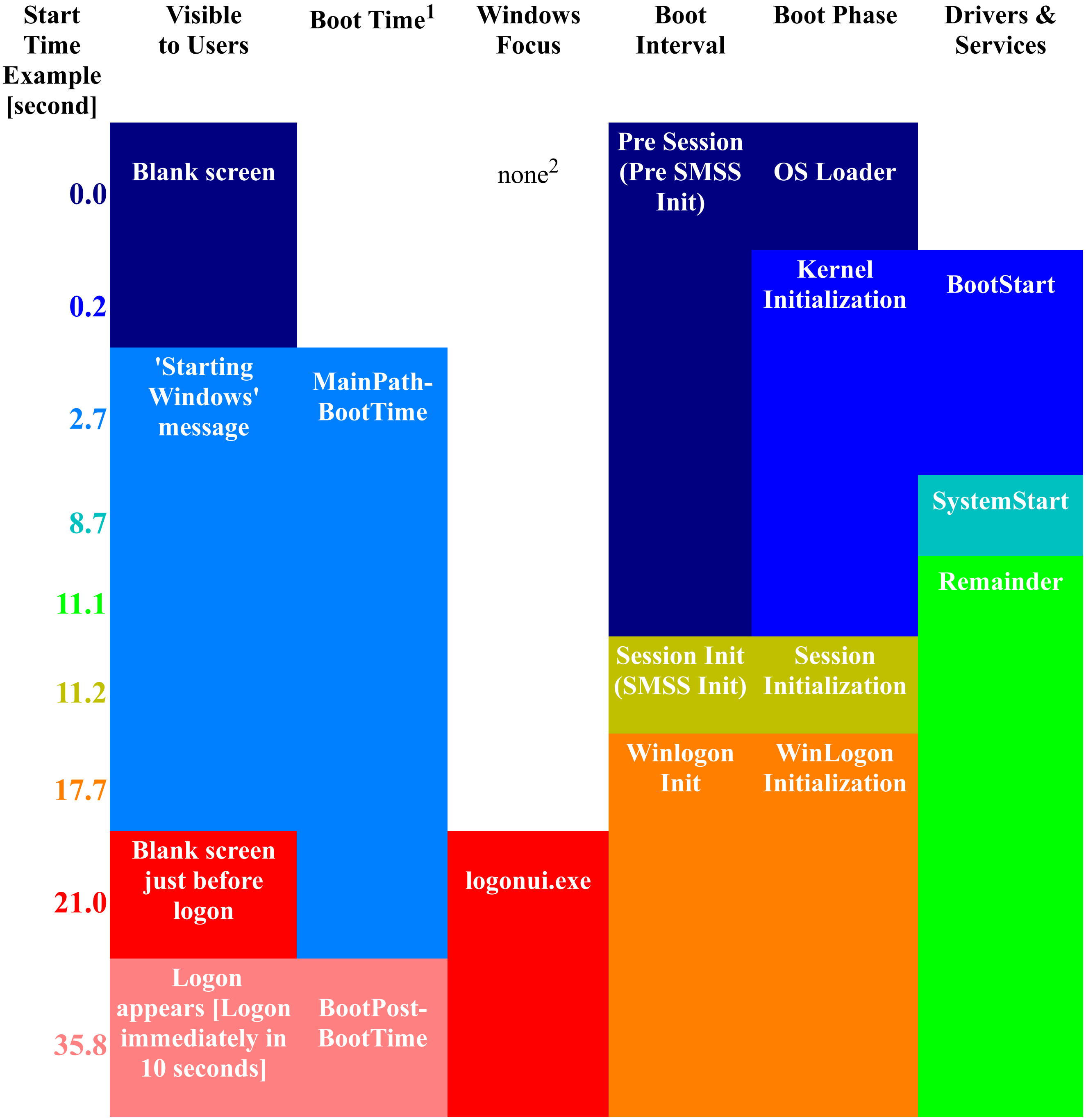
- BootStart devices and drivers are those that are run before the Windows Kernel mode is completely running.
- SystemStart devices and drivers are those that are run when the supporting Windows Kernel-mode components are running.
- AutoStart devices and drivers are all those that are started for user sessions.
- The remainder are OnDemand devices and drivers. They are run when they are required so they are not part of the startup phases.

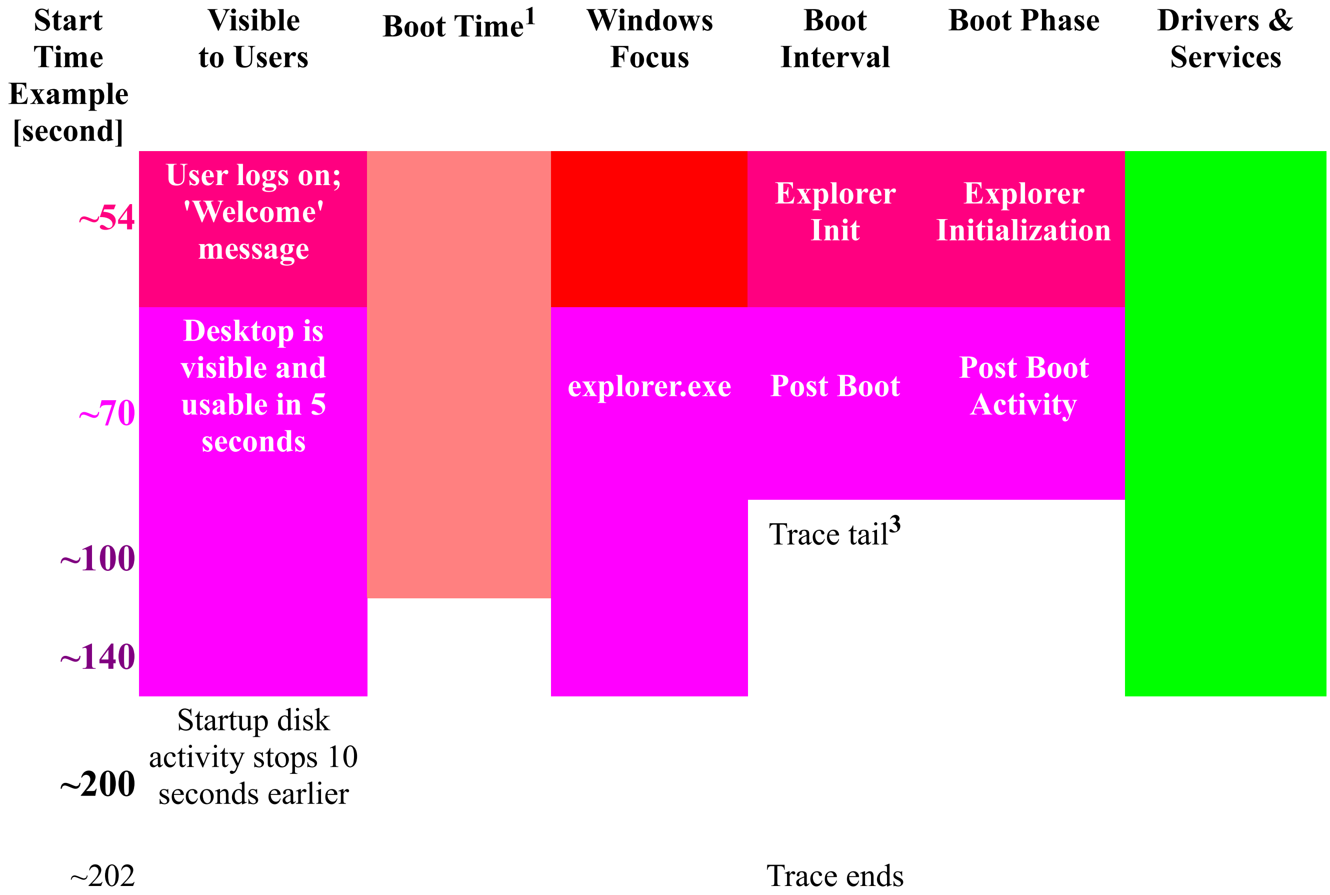
What happens when we logon immediately or use auto logon

Diagram 2, "Phases of Windows Startup - immediate logon", approximates what happens when I logon immediately the prompt appears, i.e. within 10 seconds. There is now no separation of the user logon activities from the system startup activities and the phase overlaps change.

I say the timings are approximate because under normal circumstances Windows startup changes every time it runs. Not only does Windows optimize the startup process but a slight delay in one process can cascade further delays to other activities. In practice that is exactly what happened. For some reason the MachinePolicyApplication was delayed 1.178 seconds and Windows generated an Event 107 record so I could see this in Event Viewer.

Diagram 2 - Phases of Windows Startup - immediate logon





Start Time Example [second]	Visible to Users	Boot Time[1]	Windows Focus	Boot Interval	Boot Phase	Drivers & Services
~54	User logs on; 'Welcome' message			Explorer Init	Explorer Initialization	
~70	Desktop is visible and usable in 5 seconds		explorer.exe	Post Boot	Post Boot Activity	
~100				Trace tail[3]		
~140						
~200	Startup disk activity stops 10 seconds earlier					
~202				Trace ends		

[1] TotalBootTime = MainPathBootTime + BootPostBootTime. BootPostBootTime doesn't follow the usual naming convention because that would have led to PostBootBootTime and they obviously didn't like the repetition of boot.

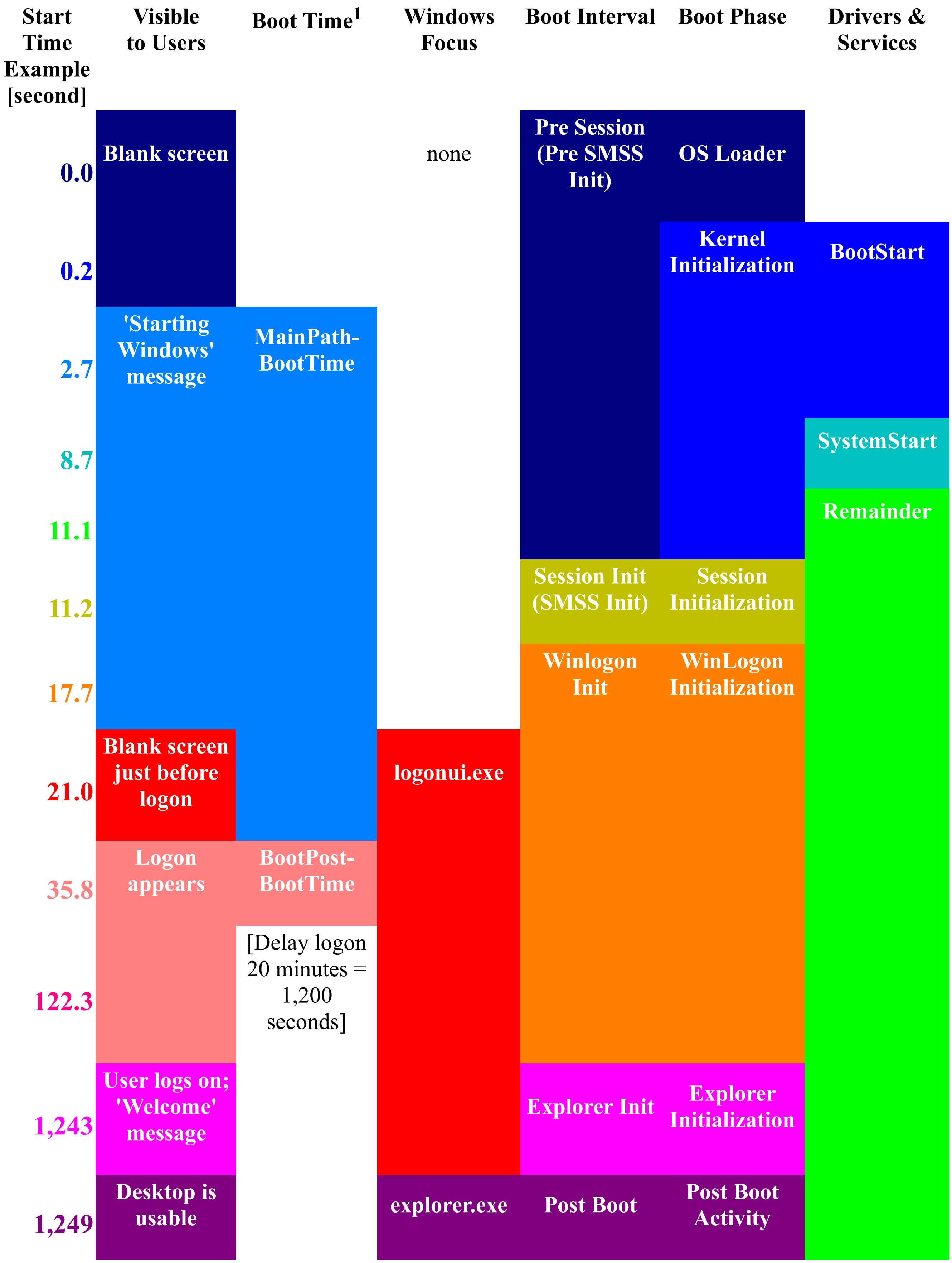
[2] Technically, there can be no focus [4] until the graphical user interface (GUI) appears.

[3] The Trace Tail is not coloured because it is not part of a normal startup. It runs from the the end of the Post Boot Phase to the end of the trace.

What happens when we don't logon immediately

If you don't logon the first time you are prompted then you allow the system startup activities to complete more quickly. There isn't much change in the startup phases and the number of steps remains the same. The major difference is that the BootPostBootTime and the auto_start driver initialization both finish before you start logon. The obvious advantage is that when you logon you will avoid waiting for other startup activities to finish so your system is more responsive right from when the desktop appears.

Diagram 3 - Phases of Windows Startup - delayed logon



Start Time Example [second]	Visible to Users	Boot Time[1]	Windows Focus	Boot Interval	Boot Phase	Drivers & Services
0.0	Blank screen		none	Pre Session (Pre SMSS Init)	OS Loader	
0.2					Kernel Initialization	BootStart
2.7	'Starting Windows' message	MainPath-BootTime				
8.7						SystemStart
11.1						Remainder
11.2				Session Init (SMSS Init)	Session Initialization	
17.7				Winlogon Init	WinLogon Initialization	
21.0	Blank screen just before logon		logonui.exe			
35.8	Logon appears	BootPost-BootTime				
122.3		[Delay logon 20 minutes = 1,200 seconds]				
1,243	User logs on; 'Welcome' message			Explorer Init	Explorer Initialization	
1,249	Desktop is usable		explorer.exe	Post Boot	Post Boot Activity	

Start Time Example [second]	Visible to Users	Boot Time[1]	Windows Focus	Boot Interval	Boot Phase	Drivers & Services
1,378	Startup disk activity stops 10 seconds earlier		🟪	Trace Trail[2]		
1.382			🟪	Trace ends		

[1] Again, the TotalBootTime = MainPathBootTime + BootPostBootTime but it is now about 12 shorter longer than in Diagram 2 due to delaying some of the user logon activities.

[2] The Trace Trail runs from the end of the Post Boot activities to the end of the boot trace that I setup to record Windows startup activities.

Other startup phase timings

There are other useful startup phases that I didn't include in Diagrams 2 & 3 these are single phases that focus on one startup process:

Services Autostart which is when most Windows services start to load just before Explorer is initialized until just before PostBoot Activity starts. In delayed logon that is from 18.9 to 49,9 seconds and it is not much different for immediate logon.

It is worth noting that various activities started and stopped while I was delayed my login. In an immediate logon they would not be included as they are configured to run after the Windows startup process. The longest delay was for Adobe Flash Player Update Service which ran 900 seconds (15 minutes) after Services Autostart ended.

ReadyBoot, the Windows startup prefetcher (prefetch gets items before they are needed), optimizes the loading of Windows components from disk drives. ReadyBoot uses the easily-confused ReadyBoost caching driver. These are turned off if your disk drive is fast enough, for example, if it is a solid-state drive (SSD). In the example, ReadyBoot in delayed logon runs from 2.1 to 102 seconds, finishing just before Session Initialization. It runs slightly longer at 105 seconds for immediate logon because there are more activities running at the same time.

The Windows startup components

Now we look at the various processes that occur during startup. I have chosen to provide a summary for each process with a more detailed table. These processes are listed in boot time sequence rather than by name.

NOTE: The current set of times are for the delayed start. I will add the autostart timings in a later update along with more detail and further phase variables used in the event log and trace reporting.

Windows Operating System (OS) Boot Loader: Winload.exe ⚠

The fist steps in loading the Windows Kernel mode are provided by the Windows Boot Loader. This program provides temporary functions that boot or start the Windows Kernel which is the first permanent component to start. The Boot Loader continues to perform further activities to support Kernel mode initialization until it has got sufficient sub-systems running to continue with its normal permanent operations.

The Boot Loader activities fall into four main areas:

- Reads the minimum configuration data from disk: the Boot Configuration Data (BCD) and the SYSTEM hive of the Registry.
- Enumerates the devices and "boot start" drivers. For the most essential drivers and itself, the boot loader also verifies their integrity and crashes the system if there are any problems.
- Initializes the system so the Windows Kernel can be loaded and executed.
- After the Kernel starts, loads into memory the configuration and enumerated drivers for the Kernel to use.

Show more...

Kernel-mode ⚠

Kernel mode processes are the core of Windows. There are a wide range of kernel-mode sub-systems that provide the basic components of the operating system which other programs rely upon. The diagram in this Overview of Windows Components [6] is for Windows 2000 but it is similar enough to Windows 7 for our purpose.

Kernel-mode processes have almost unrestricted access to resources than user-mode processes which are restricted in many ways to protect Windows. Kernel-mode processes:

- Can access hardware directly whereas user-mode processes cannot.

- Can access all of the computer's memory whereas user-mode processes are limited to assigned memory spaces.
- Can access the kernel memory whereas user-mode processes cannot.
- Are not normally paged out of RAM to virtual memory on disk.
- Run at a high priority so they don't have to wait on user-mode processes which run at a lower priority.

Kernel mode initialization performs three main functions:

- Set-up data structures.
- Load and initialize components.
- Start the Plug and Play (PnP) manager to initialize the boot start drivers that were enumerated by the Windows Boot Loader.

Show more...

User Mode: Session Manager ⚠

The Session Manager (SMSS) performs three main tasks:

- Spawns many other processes that spawn further processes. That includes spawning multiple instances of itself, concurrently (running at the same time) up to four plus one per additional CPU;
- Loads and starts the drivers other than the boot drivers; and
- Initializes the Registry.

The first instance also:

- Marks itself as critical.

Show more...

User Mode: Client Server Run-Time Sub-System (CSRSS) ⚠

The Client Server Runtime is a critical process that is used to provide the the user-mode portion of the Win32 API (Application Programming Interface). Originally it provided the entire API but Win32K now provides the kernel mode portion. However, CSRSS continues to create kernel-mode threads.

Show more...

User Mode: Windows Initialization (WinInit) ⚠

Windows Initialization or WININIT runs the first time a user logs on. It runs once to handle system tasks that do not need to run again.

Show more...

User Mode: Services Control Manager (SCM)

The Service Control Manager runs as a Windows console program.:

- Scans the registry for configured device drivers and services.
- Loads the auto class device drivers and services
- Waits for requests to start and stop services

Show more...

User Mode: Local Security Authority SubSystem (LSASS)

The Local Security Authority Sub-System that handles local (i.e. not network) system security policies. Most of its functionality is handled by the Local Security Authority service (LSASrv.dll) and its database is stored in the registry in a protected area under HKLM\Security:

- authenticates user logon by calling the appropriate authentication DLL. Authenticated users have an access token generated that contains the user security profile.
- System security auditing including sending related event messages to the Event log.

It processes any request for security authorisations that it receives through the LCM port it creates. Requests come from three sources:

- Winlogon
- network logon service process
- other user-mode processes that want to authenticate users

Show more...

User Mode: Local Session Manager (LSM)

The Local Session Manager manages terminal server sessions running on the local machine.

LSM is notified by WinLogon of:

- logon and logoff
- connect to and disconnect from session
- lock and unlock the desktop
- start and terminate the shell

Show more...

User Mode: Interactive Windows Logon ⚠

The Windows Logon Application runs under LSASS and manages interactive logon sessions. It performs the following main tasks:

- Displays the logon screen by running the Logon User Interface (LogonUI.exe);
- Services are started by the Service Control Manager (SCM)
- Identifies and authenticates users through credential provider DLLs
- Can load additional network provider DLLs
- Group policy is applied.
- Passes the username and password to LSASS for authentication.

The Secure Attention Sequence (SAS), keying Ctrl+Alt+Del, cannot be produced by a process so it differentiates users from processes.

Show more...

User Mode: User Initialization (UserInit)

User Initialization (UserInit) sets-up the user environment before starting the Windows shell which by defaults is Windows Explorer:

- runs logon scripts
- connects to the network
- applies Group Policies including running the Group Policy logon script
- creates events for some failed logon scripts

Show more...

User Mode: Windows Logon User Interface Host (LogonUI)

The Windows Logon User Interface Host provides the user interface for logging on:

- Presents users with a logon screen to using credential providers to obtain the user account name and password. Windows has default credential providers that can be replaced or supplemented by third-party providers.
- Allow alternative credential providers to be used for alternative input methods e.g. biometric scans such as thumbprints and retinas.
- Allow secondary authentication using network provider DLLs. This provides for authentication from a network server at the same time using one logon.

LogonUI is a separate process from WinLogon.exe. Any failure with third-party credential providers will not cause Windows to crash. Instead it can spawn another instance of LogonUI.

Show more...

User Mode: Network Logon (NetLogon)

Network logon (%SystemRoot%\System32\NETLOGON.DLL) is not used in the example. It is usually invisible to users as it would use the credentials authenticated by user logon. If it requires additional credentials then they are obtained using the Network Provider during LogonUI.

User Mode: Explorer

When Explorer starts the Desktop Window Manager also starts and displays the desktop.

The default desktop appears when the shell is ready to display something, or after thirty seconds, whichever is first.

Show more...

User Mode: Trace trail

In my example, Windows Performance Reporting (WPR) started with Explorer initialization at 1,243s, started the trace tail at 1,378.69 when Explorer has booted to the desktop, and ended the trace at 1,382s:
C:\Program Files (x86)\Windows Kits\8.0\Windows Performance Toolkit\WPRUI.EXE

Answers to more questions about Windows startup

Where does the kernel run?

The kernel mode runs in the system process On every started Windows system there are two permanent processes that must be running:

- Idle is always process 0 which is used when the processor has nothing to process
- System is always process 4 which includes most of the kernel mode threads and only kernel-mode subsystems. This process owns all system threads but take note that device drivers can create system threads in any process.

Why can't I find a service?

Services can have three different names which can confuse you:

- The process name which is most visible
- The registry name which is used internally
- The display name visible in the Services Administration Tool. If this is blank then it defaults to the registry name.

How does running with lower privileges affect startup?

The answer to this question will be added in a later update.

What difference does the Windows edition make to startup?

Remember this article only looks at Windows 7.

Client versus server versions

Desktops need fast desktop response times so interactive users are not waiting. Whereas servers generally need high performance for the applications but have little need to respond quickly to interactive users. That's why desktop and server versions are optimized differently and have different specifications particularly the limits for CPUs, memory and storage.

You can determine the edition by looking in the registry to see some of the supported features. The product policy details are a copy of Tokens.dat.

HKLM\System\CurrentControlSet\Control\ProductOptions\ProductType =
Client, WINNT
ServerL ServerNT, LANMan.NT

Related Links

- Windows Startup Terminology [2] summarises terms that are used in this article. The terms generally relate to running programs so there is section on how programs start and run.
- What Everybody Should Know About the Windows Registry [9]

Microsoft references:

- Windows Internals [10] book is recommended by Microsoft if you want to look 'under the hood' of Windows. I agree having bought my own copies of the boos after I published the first version of this article. They are cheap at the moment but the next edition which covers Windows 8 should be available soon in 2014.