

## WHAT IS THE DOS BOOT SEQUENCE??

Short answer:

*Power On Self Test (POST)* which checks BIOS, CPU, RAM, Video, Keyboard, drives, etc.

Load *Operating System (OS)*

- ? *IO.SYS*
- ? *MSDOS.SYS*
- ? *CONFIG.SYS*
- ? *COMMAND.COM*
- ? *AUTOEXED.BAT*

### Specifics

1. BIOS locates the Master Boot Record (MBR) on the hard drive.
2. The partition table find the physical location of the logical boot drive and turns to the boot record of that logical drive
3. The boot record (a very short program) loads two hidden files into memory. These files are *IO.SYS* and *MSDOS.SYS*
  - o The *IO.SYS* file contains more BIOS software
  - o The *MSDOS.SYS* contains software to manage files, run applications software and interface with hardware
4. Once these two files are loaded, the boot record program is longer needed and turns control over to a file stored on *MSDOS.SYS*
5. This program looks on the hard drive for a file named *CONFIG.SYS*. This is the first OS file that you, the user, can change. This file contains commands that tell DOS how many files it can open at any one time (*FILE=*) and how many file buffers (a temporary holding area for a file) to create (*BUFFERS=*). It also contains the commands to load device drivers (small programs that tell your computer how to communicate with devices such as printers) (*DRIVERS=*) and other information. Several drivers can be loaded into memory and *CONFIG.SYS* puts them anywhere it wants unless the program requests a certain memory location.
6. When *CONFIG.SYS* is done, MSDOS looks for another file called *COMMAND.COM*. This file consists of 3 parts: more code to manage Input/Output (I/O), internal DOS commands such as COPY and DIR, and a short program that looks for *AUTOEXEC.BAT*.

7. *AUTOEXEC.BAT* stands for "automatically executed batch" program. This file holds a list of DOS commands that are automatically executed each time DOS loads. Two of these commands are:
  - *PROMPT \$P\$G* this instructs DOS to display the current directory name and the current drive name as part of the prompt.  
C:\Windows instead of C:>.
  - *PATH* Tells DOS where to look for program files. Example:  
C:\Windows\Driver Cache\Fonts
  - *AUTOEXEC.BAT* also loads TSRs (terminate and stay resident programs).
8. The boot process is completed after *AUTOEXEC.BAT* has finished executing. At this point, *COMMAND.COM* is in charge and you have the command Prompt (C:>).

## **Windows 95/98--BOOT SEQUENCES**

The Windows 95 boot sequence is as follows:

1. POST (Power-On Self Test)
2. The Plug and Play(PnP) BIOS begins by looking at the hardware devices on the system and figuring out which ones are PnP compliant. The BIOS first enables the devices that are not Plug and Play, and then tries to make the PnP devices use the leftover resources.
3. The Basic Input/Output System looks for devices (Hard Drives, Floppy Disk, CDs) containing the Operating System (OS)
4. Master Boot Record (MBR) executes the boot record on the hard drive, which looks for the initial hidden files of Windows 95, called IO.SYS.
5. IO.SYS loads. IO.SYS looks for CONFIG.SYS file, and, if found, the CONFIG.SYS file executes. *The CONFIG.SYS file is not required for Windows 95.*
6. IO.SYS searches for MSDOS.SYS. MSDOS.SYS is a hidden file with settings used to customize the boot process.
7. COMMAND.COM loads.
8. AUTOEXEC.BAT.
9. The heart of Windows 95 now loads, providing a desktop from which you can execute application software.

## **WINDOWS 98 BOOT SEQUENCE:**

Just as with Windows 95, Windows 98 goes through a boot sequence in the order listed above.

1. BIOS runs POST
2. BIOS loads a small DOS core
3. The DOS core loads Windows 98

## The Windows NT 4.0 Boot Process

Files required for a successful boot:

- ? NTLDR - The operating system loader, which must reside in the root of the boot drive. In a multi-boot environment, it will be used to start the other systems initially. This file is hidden, system, and read-only.
- ? [BOOT.INI](#) - a text file which is used to build the OS Selection menu, and gives the path to each OS available. This file also must be in the root of the boot drive, and is read-only, system.
- ? BOOTSECT.DOS - This file contains the boot sector of the Operating System that was on the hard drive previous to installing NT. NTLDR will use this to boot in a multi-boot environment if an OS other than NT was chosen from the boot menu. It too, is a hidden system file that must be in the root of the boot drive.
- ? NTDETECT.COM - This program examines the hardware on the machine, and builds a list, which it passes back to NTLDR to be used to build the Hardware Hive of HKEY\_LOCAL\_MACHINE in the Registry. This file is also a hidden, read-only system file, in the root of the boot partition.
- ? NTOSKERNEL.EXE - The kernel of the OS itself, which resides in the WINNT\SYSTEM32 directory.
- ? NTBOOTDD.SYS - This device driver file will only be used on systems that boot from a SCSI disk on which the SCSI adapter BIOS is disabled.
- ? SYSTEM - This file is located in the WINNT\SYSTEM32\CONFIG folder, and controls which drivers and services are loaded during the Windows NT startup.

Next when the machine is first powered on, it will go through a series of steps before NT actually begins booting:

- ? POST - Power On Self Test
- ? The machine locates the boot device, and loads the MBR (Master Boot Record) into memory.
- ? The MBR's program will locate the active partition and load the boot sector into memory from it.
- ? NTLDR will be loaded into memory and run.

This brings us to the actual NT booting process.

- ? NTLDR switches the processor to a 32-bit flat memory model, supporting up to 4 GB of RAM (physically installed).

- ? NTLDR starts what is called a minifile system. Windows NT can read one of three file formats: FAT, NTFS & CDFS.
- ? NTLDR reads the BOOT.INI file, and displays the operating system selections in the Boot Menu.
- ? If Windows NT is selected, NTLDR will run NTDETECT.COM.
- ? If another OS is selected, NTLDR will load and run BOOTSECT.DOS, and pass control to it, and exit. The other OS will continue as though the machine had just booted. If BOOTSECT.DOS is missing or corrupt, it must be replaced or reconstructed in order to boot to the other OS.
- ? NTDETECT.COM scans the machine's hardware (you will notice the keyboard lights and modem lights flash at this point, as it scans the various ports).
- ? NTLDR then loads NTOSKRNL.EXE and passes the hardware information to it.
- ? This technically ends the "boot phase" and begins the "load phase."

## LOAD PHASES

There are four more phases to go before NT is officially up and running. They are the Kernel Load, Kernel Initialization, Services Load, and Subsystem Start Phases.

- ? Kernel Load phase, the HAL (hardware abstraction layer) is loaded, which hides the physical hardware from applications.
- ? System hive of the registry is loaded next, and scanned for drivers and services that should be loaded.
- ? Kernel Initialization phase, the screen is blue. The drivers are initialized and loaded, and the registry's CurrentControlSet is then saved, and the Clone control set is created, but not saved. The registry hardware list is then created from the information gathered earlier.
- ? Services Load phase, the session manager is started (SMSS.EXE), which runs any programs listed in HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager: BootExecute. After this, the session manager sets up the pagefile(s). Next the Clone Control Set is written to the registry. The last thing the session manager does is load the required subsystems (by default, only Win32).

- ? Subsystem Start phase, WINLOGON.EXE is automatically started, which starts the Local Security Authority (LSASS.EXE) and brings up the logon dialog box (CTRL+ALT+DEL). The Service Controller (SCREG.EXE) is then run, which looks through the registry for services that are set to automatic load, and loads them.

### **User Logon:**

Once the user logs on successfully, a boot is considered complete, and the Clone control set is copied to the Last Known Good control set.

## The Windows 2000 Boot Process

These files must be present and not corrupted on the computer, and in the folder indicated:

NTLDR	Root of the active partition
Boot.ini	Root of the active partition
Bootsect.dos (only if dual booting)	Root of the active partition
Ntdetect.com	Root of the active partition
Ntbootdd.sys (only if booting from a SCSI partition, and SCSI BIOS is not present on the controller)	Root of the active partition
Ntoskrnl.exe	%SystemRoot%\System32
Hal.dll	%SystemRoot%\System32
SYSTEM key (part of the Registry)	%SystemRoot%\System32\Config
Device drivers (varies according to devices present)	%SystemRoot%\System32\Drivers

By default, the "Root of the active partition" means C:\, and "%SystemRoot%" means c:\winnt.

On non-Intel (RISC) based systems, NTLDR is not needed. Instead the computer's firmware performs those functions, and is controlled by OSLOADER.EXE. A successful boot process on an Intel-based computer running Win2K takes the following six steps to complete:

**Step 1 - POST** - Power On Self Test. This is something that all computers will perform, regardless of the Operating System that is installed on it. This step is only mentioned because a working hardware platform is necessary for a successful boot. During the POST, a computer will typically test its memory, verify that it has all necessary hardware, such as a keyboard. After checking itself, the computer will allow adapter cards (such as SCSI cards) to run their own POSTs. After the POST, the computer will locate a boot device, and load the MBR (Master Boot Record) into memory, which in turn locates the active partition and loads the boot sector into memory. Up to this point, the computer's hardware has played the active role. Without properly functioning hardware, the operating system doesn't matter much. At this point it is good to know that About.com also has the best site on the Internet for [PC Support](#), complete with [a great newsletter](#), an [interactive forum](#), and a [chat room](#). Even if your computer is booting and running fine right now, this site is a great resource for tips and information your computer.

**Step 2 - Choose the OS.** NTLDR is the key component of this step. It will use the files Ntdetect.com, boot.ini, and bootsect.dos (for dual-boot machines). At the beginning of this step, the screen will display the "OS Loader V5.0" message

on the screen. When NTLDR runs, it switches the processor into 32-bit flat memory mode (until this point the computer was running in real mode (just like your old 8086 or 8088 CPU). It then starts the appropriate mini-file system (e.g.:FAT, NTFS), so that it can read the files from the disk. It will then read the Boot.ini file, and display the boot menu on the screen. If an OS other than Windows 2000 is selected, NTLDR then loads the bootsect.dos file and passes control to it, which then boots the other OS. If a Windows 2000 OS is selected, then NTLDR runs Ntdetect.com to gather information about the computer's hardware. It is also in this step when you can choose to press F8 for troubleshooting and advanced startup options. Ntdetect detects the following hardware components:

- ? Computer ID
- ? Bus/adapter type
- ? SCSI adapters
- ? video adapters
- ? keyboard
- ? Com ports
- ? Parallel ports
- ? Floppy disks
- ? Mouse/pointing devices
- ? Floating-point coprocessor

Once Ntdetect.com has collected the hardware information, NTLDR will load Ntoskrnl and pass that information to it.

**Step 3 - Kernel Load.** This phase begins with the loading of ntoskrnl.exe, along with the file hall.dll. NTLDR will also read the SYSTEM registry key into memory, and select the hardware configuration and control set (from the Registry) that will be used for this boot. If you have more than one hardware profile created, you could select it at this point in the boot sequence. NTLDR will also load any device drivers that have a start value (again from the Registry) of 0x0. If you add the switch /SOS in the boot.ini, you will be able to see the drivers listed on the screen as they are loaded. At this point all of these files have been loaded into memory.

**Step 4 - Kernel Initialization.** Once Ntoskrnl.exe is initialized, it creates the Clone control set by copying the current control set. It will also create the HARDWARE key in the Registry using the information gathered by earlier by ntdetect.com. Ntoskrnl.exe will then initialize the drivers loaded earlier, and will then scan the Registry for device drivers that have a start value of 0x1.

**Step 5 - Services Load.** This step begins with the starting of the Session Manager (Smss.exe). It will run the programs listed in its BootExecute Registry entry, as well as starting the required subsystems. The Win32 subsystem will then start Winlogon.exe, which starts the Local Security Administration (Lsass.exe), and the Ctrl+Alt+Delete window appears. The Service Controller (Screg.exe) will check the Registry for services with a start value of 0x2, and will load them. Services can be loaded simultaneously, but dependent on their dependencies. Services with start values of 0x3 are started manually, and services with start values of 0x4 are disabled.

**Step 6 - Logon.** The logon prompt will appear during the previous step, but it begins the final step in the boot-up process. A boot is not considered successful or completed until a user logs in. After a successful logon, the Clone control set from Step 4 is copied to the LastKnownGood control set. This makes it available as an advanced boot option in the later half of Step 2 for the next boot. What this means is that if you install a bad device driver that renders your computer unbootable, you can press F8 during Step 2, and choose to use the LastKnownGood control set. Since this control set was created before you installed the bad driver, your system will load without trying to start the driver, allowing you to boot successfully.

## WINDOWS XP BOOT SEQUENCE

1. POST
2. NTLDR: The MBR reads the boot sector which is the first sector of the active partition. This sector contains the code that starts *Ntldr* which is the boot strap loader for Windows XP. The first role of Ntldr is to allow full memory addressing, start the file system, read boot.ini and put up the boot menu. **IMPORTANT:** Ntldr must be located in root folder of the active partition along with *Ntdetect.com*, *boot.ini*, *bootsect.dos* (for dual booting) and *Ntbootdd.sys* (needed with some SCSI adapters).
3. Ntdetect: Gets information about installed hardware. Ntldr then uses the ARC path specified in the boot.ini to find the boot partition.
4. Ntoskrnl.exe and Hal.dll.
5. Ntldr reads the registry files, selects a hardware profile, control set and loads device drivers, in that order.
6. Ntoskrnl.exe starts Winlogon.exe which starts Lsass.exe (Local Security Administration), this is the program that displays the Welcome screen (If Professional Edition-the Windows Log On dialog box), and allows the user to log on with his/her user name and password.

## References

<http://www.qvctc.commnet.edu/classes/csc277/boot-xp.html>

<http://www.qvctc.commnet.edu/classes/csc277/handouts.html>