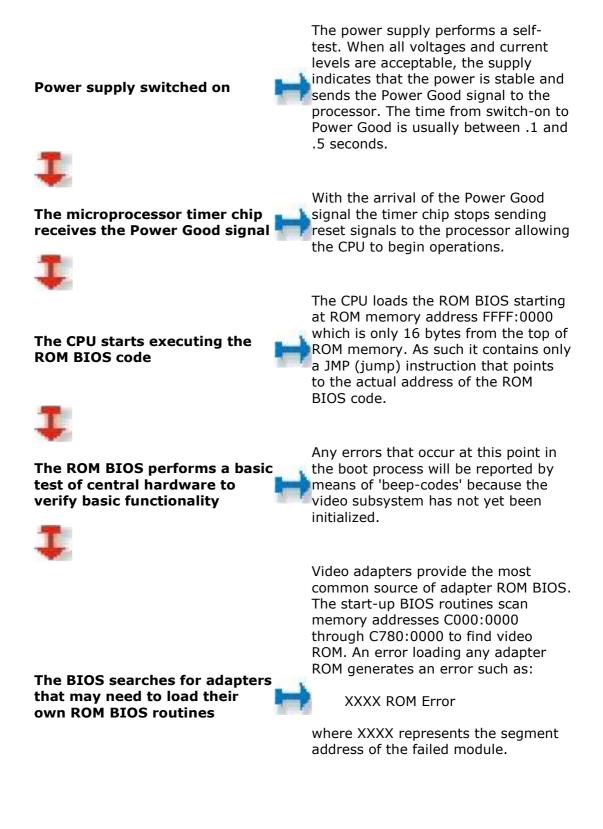
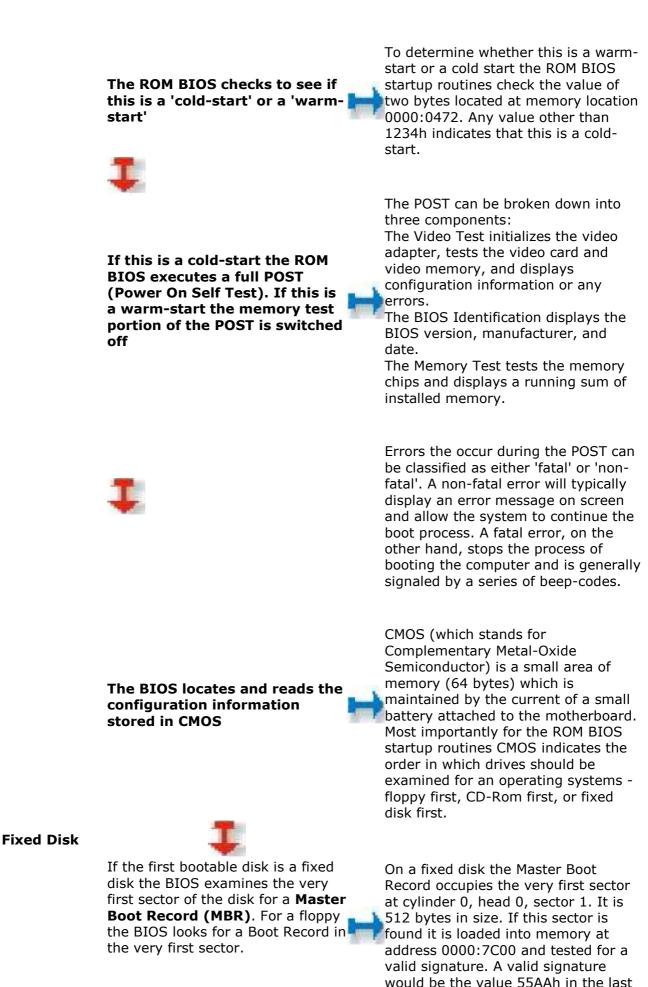
The PC Boot Process - Windows XP







The process of installing multiple With a valid MBR loaded into operating systems on a single PC memory the BIOS transfers control usually involves replacing the original partition loader code with a Boot MBR of the boot process to the partition loader code that takes up most of Loader program that allows the user the 512 bytes of the MBR. to select the specific fixed disk to load in the next step of the process The Boot Record is also 512 bytes and contains a table that describes the The partition loader (or Boot Loader) examines the partition characteristics of the partition Partition table for a **partition marked as** (number of bytes per sectors, number Table active. The partition loader then of sectors per cluster, etc.) and also searches the very first sector of the jump code that locates the first of that partition for a **Boot Record**. the operating system files (IO.SYS in DOS). Operating System The loading of Windows XP is controlled by the file NTLDR which is a hidden, system file that resides in the The active partition's boot record is root directory of the system partition. checked for a valid boot signature NTLDR will load XP in four stages: **Boot Record** and if found the boot sector code is executed as a program. 1) Initial Boot Loader Phase 2) Operating System selection 3) Hardware Detection 4) Configuration Selection During the initial phase **NTLDR** switches the processor from realmode to protected mode which places the processor in 32-bit memory mode and turns memory Windows XP supports partitions NTLDR paging on. It then loads the formatted with either the FAT-16, **Initial Phase** appropriate mini-file system FAT-32, or NTFS file system. drivers to allow NTLDR to load files from a partition formatted with any of the files systems supported by

XP.

two bytes. Lacking an MBR or a valid signature the boot process halts with an error message which might read:

NO ROM BASIC - SYSTEM HALTED

A Master Boot Record is made up of two parts - **the partition table** which describes the layout of the fixed disk and **the partition loader code** which includes instructions for continuing

the boot process.



BOOT.INI contains entries for more NTLDR than one operating system NTLDR **OS Selection** will stop the boot sequence at this **BOOT.INI** point, display a menu of choices,

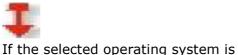
> time for the user to make a selection.

Assuming that the operating system being loaded is Windows NT, 2000, or XP pressing F8 at this stage of the boot sequence to display various boot options including "Safe Mode" and "Last Known Good Configuration"

the root directory, NTLDR will read

and wait for a specified period of

it's contents into memory. If



NTLDR Hardware Detection

NTLDR

Selection

Configuration



If this computer has more than one defined Hardware Profile the NTLDR program will stop at this point and display the Hardware **Profiles/Configuration Recovery** menu.

XP, NTLDR will continue the boot

DOS based NTDETECT.COM

program to perform hardware

process by locating and loading the



If the file BOOT.INI is not found in the root directory NTLDR will continue the boot sequence and attempt to load XP from the first partition of the first disk, typically C:\.

After each successful boot sequence XP makes a copy of the current combination of driver and system settings and stores it as the Last Known Good Configuration. This collection of settings can be used to boot the system subsequently if the installation of some new device has caused a boot failure.

NTDETECT.COM collects a list of currently installed hardware components and returns this list for later inclusion in the registry under the HKEY_LOCAL_MACHINE\ HARDWARE key.

Lacking more than one Hardware Profile NTLDR will skip this step and not display this menu.

Kernel Load

After selecting a hardware configuration (if necessary) NTLDR begins loading the XP kernel (NTOSKRNL.EXE).

During the loading of the kernel (but before it is initialized) NTLDR remains in control of the computer. The screen is cleared and a series of white rectangles progress across the bottom of the screen. NTLDR also loads the Hardware Abstraction Layer (HAL.DLL) at this time which will insulate the kernel from hardware. Both files are located in the \system32 directory.



F8

NTLDR Boot Device Drivers

Kernel

Initialization

NTLDR now loads **device drivers** that are marked as boot devices. With the loading of these drivers NTLDR relinquishes control of the computer.

NTOSKRNL goes through two

services required for the

phases in its boot process - phase

0 and phase 1. Phase 0 initializes

just enough of the microkernel and

Executive subsystems so that basic

completion of initialization become

display a graphical screen with a status bar indicating load status.

available.. At this point, the system



Every driver has a registry subkey entry under HKEY_LOCAL_MACHINE \SYSTEM\Services. Any driver that has a Start value of SERVICE_BOOT_START is considered a device to start at boot up. A period is printed to the screen for each loaded file (unless the /SOS switch is used in which case file names are printed.

XP disables interrupts during phase 0 and enables them before phase 1. The HAL is called to prepare the interrupt controller; the Memory Manager, Object Manager, Security Reference Monitor, and Process Manager are initialized.

Phase 1 begins when the HAL is called to prepare the system to accept interrupts from devices. If more than one processor is present the additional processors are initialized at this point. All Executive subsystems

are reinitialized in the following order:

- 1) Object Manager
- 2) Executive
- 3) Microkernel
- 4) Security Reference Monitor
- 5) Memory Manager
- 6) Cache Manager
- 7) LPCS
- 8) I/O Manager
- 9) Process Manager

I/O Manager

The initialization of I/O Manager begins the process of loading all the **systems driver files**. Picking up where NTLDR left off, it first finishes the loading of boot devices. Next it assembles a prioritized list of drivers and attempts to load each in turn.



SMSS

The last task for phase 1 initialization of the kernel is to launch the **Session Manager Subsystem (SMSS)**. SMSS is responsible for creating the usermode environment that provides the visible interface to NT. The failure of a driver to load may prompt NT to reboot and try to start the system using the values stored in the Last Known Good Configuration.

SMSS runs in user-mode but unlike other user-mode applications SMSS is considered a trusted part of the operating system and is also a native application (it uses only core Executive functions). These two features allow SMSS to start the graphics subsystem and login processes.



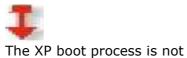




win32k.sys

SMSS loads the **win32k.sys** device driver which implements the Win32 graphics subsystem.

Shortly after win32k.sys starts it switches the screen into graphics mode. The Services Subsystem now starts all services mark as Auto Start. Once all devices and services are started the boot is deemed successful and this configuration is saved as the **Last Known Good Configuration**.



considered complete until a user has successfully logged onto the system. The process is begun by the **WINLOGON.EXE** file which is loaded as a service by the kernel and continued by **the Local Security Authority (LSASS.EXE)** which displays the logon dialog box.

This dialog box appears at approximately the time that the Services Subsystem starts the network service.

Logon