

Microsoft Knowledge Base Article - Q174018

Description of the Windows 95 Startup Process

The information in this article applies to:

- Microsoft Windows 95
-

Summary

This article describes the Windows 95 startup process.

More Information

The Windows 95 startup process can be broken into the following steps:

- The read-only memory (ROM) Basic Input-Output (BIOS) bootstrap process
- The master boot record (MBR) and boot sector
- The Io.sys file
- Real-mode configuration
- The Win.com file and the Windows 95 Environment

Step 1 - The ROM BIOS Bootstrap Process

When you start your computer, the ROM BIOS bootstrap loads from the FFFF0h memory address. The following steps occur during the ROM BIOS bootstrap process:

1. The Power On Self-Test (POST) occurs.
2. The A drive is checked for the existence of a boot disk.
3. If a boot disk is not found in the A drive, the ROM BIOS bootstrap checks for a hard disk. If a hard disk is found, the ROM loader transfers control to the operating system loader.
4. The master boot record and partition table are read.

Microsoft and several original equipment manufacturers (OEMs) have defined a Plug and Play BIOS specification. This specification defines the interactions between the Plug and Play BIOS, Plug and Play devices, and option ROMs. If your computer has a Plug and Play BIOS, the following additional steps are performed:

5. The Plug and Play BIOS checks non-volatile random access memory (RAM) for input/output (I/O) port addresses, interrupt request lines (IRQs), direct memory access (DMA) channels, and other settings needed to configure Plug and Play devices on the computer.
6. All Plug and Play devices found by the Plug and Play BIOS are disabled.
7. A map of used and unused resources is created.
8. The Plug and Play devices are configured and re-enabled, one at a time.

Windows 95 Configuration Manager queries the Plug and Play BIOS for device information, and then queries each Plug and Play device for its configuration.

If your computer does not have a Plug and Play BIOS, Plug and Play devices are initialized using their default settings when you start your computer. These devices may be reconfigured dynamically when Windows 95 starts.

Step 2 - The Master Boot Record and Boot Sector

The master boot record determines the location of the boot partition by reading the partition table located at the end of the master boot record. Once the location of the boot partition is determined, the master boot record passes control to the boot sector in that partition. The boot sector contains the disk boot program and a table of disk characteristics. The boot sector checks the BIOS Parameter Block (BPB) to find the location of the root directory, and then copies the Io.sys file from the root directory into memory.

Step 3 - The Io.sys File

The following steps occur when the Io.sys file loads into memory:

1. A minimal file allocation table (FAT) file system is loaded.
2. The Msdos.sys file is read.
3. The "Starting Windows 95" message is displayed for <n> seconds, or until you press a Windows 95 function key. The amount of time the message is displayed is determined by the BootDelay=<n> line in the Msdos.sys file. The default is 2 seconds.
4. If you have multiple hardware profiles in Windows 95, you receive the following message and must choose a hardware configuration to use:
Windows cannot determine what configuration your computer is in.
5. The Logo.sys file is loaded and displays a startup image on the screen.
6. If the Drvspace.ini or Dblspace.ini file exists, the Drvspace.bin or Dblspace.bin file is loaded into memory.
7. The Io.sys file checks the system registry files (System.dat and User.dat) for valid data.
8. The Io.sys file opens the System.dat file. If the System.dat file is not found, the System.da0 file is used for startup. If Windows 95 starts successfully, the System.da0 file is copied to the System.dat file.
9. The Dblbuff.sys file is loaded if the "DoubleBuffer=1" is in the Msdos.sys file, or if double buffering is enabled under the following registry key:

```
HKLM\System\CurrentControlSet\Control\WinBoot\DoubleBuffer
```

Windows 95 Setup automatically enables double buffering if it detects that it is required.

10. If you have multiple hardware profiles in Windows 95, the hardware profile you chose is loaded from the registry.
11. The Io.sys file processes the Config.sys file.

Step 4 - Real-Mode Configuration

Some hardware devices and programs require that drivers or files be loaded in real-mode in order for them to work properly. To ensure backwards compatibility with these types of hardware devices or programs, Windows 95 processes the Config.sys and Autoexec.bat files if they exist.

1. The Config.sys file loads drivers into memory. If the Config.sys file does not exist, the Io.sys file loads the following required drivers:

The Io.sys file obtains the location of these files from the "WinBootDir=" line of the Msdos.sys file. These files must reside on the hard disk.
2. Windows 95 reserves all global upper memory blocks (UMBs) for Windows 95 operating system use or for expanded memory support (EMS).
3. The Autoexec.bat file loads files and terminate and stay resident (TSR) programs into memory.

Step 5 - The Win.com File and the Windows 95 Environment

1. After the Autoexec.bat file is processed, the Win.com file is run.
2. The Win.com file accesses the Vmm32.vxd file. If there is enough available RAM, the Vmm32.vxd file loads into memory, otherwise, it is accessed from the hard disk. This may result in a slower startup time. The Vmm32.vxd file is similar to the Win386.exe file used in earlier versions of Windows.
3. The real-mode virtual device driver loader checks for duplicate virtual device drivers (VxDs) in the Windows\System\Vmm32 folder and the Vmm32.vxd file. If a VxD exists in both the Windows\System\Vmm32 folder and the Vmm32.vxd file, the duplicate VxD is "marked" in the Vmm32.vxd file so that it is not loaded.
4. Real-mode VxDs can be loaded into memory in any of the following ways:
5. The real-mode virtual device driver loader checks that all required VxDs loaded successfully. If not, it attempts to load the drivers again.
6. Once the real-mode virtual device driver loading is logged, driver initialization occurs. If there are any VxDs that require real-mode initialization, they begin their process in real-mode.

7. Vmm32 switches the computer's processor from real-mode to protected- mode.
8. A three-phase VxD initialization process occurs in which the drivers are loaded according to their InitDevice instead of the order in which they are loaded into memory. The VxDs are carried out in the following sequence:

- a. SYS_CRITICAL_INIT (SYSCRITINIT):

Interrupts are disabled during this phase. This gives VxDs time to prepare for device initialization without being interrupted by the system. No file I/O is allowed during SYSCRITINIT, so all SYSCRITINITs are not written to the Bootlog.txt file until after SYSCRITINIT is complete for all VxDs.

- b. SYS_DEVICE_INIT (DEVICEINIT)

The bulk of the VxD initialization takes place during this phase. File I/O is allowed during DEVICEINIT, so each VxD's DEVICEINIT is logged as it occurs. The one exception is during Ifsmgr's DEVICEINIT. Ifsmgr takes over the real-mode file system, and disk I/O is not allowed until Ifsmgr's DEVICEINIT succeeds. For this reason, Ifsmgr does not appear in the DEVICEINIT phase.

When a DevLoader VxD is called, it loads other drivers it is responsible for, regardless of their InitDevice order. The DevLoader examines the Registry and finds drivers (for example, portdrivers [such as.mpd files]) and any associated support drivers. It then initializes the device associated with these drivers. During this phase, if a VxD failed to initialize, it was unable to properly communicate with the hardware or service it drives. Typically, this is due to incorrect hardware settings or the service not being installed.

The remaining static VxDs continue with the initialization phase. Also, dynamic VxDs may begin initializing during this phase. They do not have a SYSCRITINIT phase. However, a dynamic VxD may also load anytime after Windows 95 has started.

- c. SYS_INIT_COMPLETE (INITCOMPLETE)

VxDs that successfully pass the InitComplete phase should be working properly. If a VxD was listed in one of the previous phases but is not successful in this phase, that VxD is unloaded from memory.

GUI Components:

After all the static VxDs are loaded, the Krnl32.dll, Gdi.exe, User.exe, and Explorer.exe (the default Windows 95 shell) files are loaded.

Network Environment and Multi-User Profiles:

The next step in the startup process is to load the network environment. Once this occurs, the user is prompted to log on to the network that is installed.

Windows 95 allows multiple users to save their custom desktop settings. When a user logs on to Windows 95, their desktop settings are loaded from the registry. If the user does not log on, the desktop configuration uses a default desktop.

StartUp Group and RunOnce Programs:

Programs in the StartUp group and the RunOnce registry key are run during the last phase of the startup process. After each program in the RunOnce registry key is started, the program is removed from the key.

First Published: Sep 17 1997 3:30PM

Keywords: win95 kbinfo boot w95susdfaq