

Troubleshooting With DiskProbe

DiskProbe is a sector editor tool for Microsoft® Windows® 2000 that allows users with Administrator rights to directly edit, save, and copy data on a physical hard drive, which is not accessible any other way.

The need for documenting and preserving disk structures, such as the master boot record (MBR) and boot sector is clear. The ability to edit and repair these sectors on a byte-by-byte basis after corruption has occurred is invaluable. The DiskProbe tool is perfect for this job.

You can use DiskProbe to replace the MBR, repair damaged partition table information and to repair or replace damaged boot sectors or other file system data. The program can also save MBRs and boot sectors as backup binary files in case the original sectors become damaged. These on-disk data structures are not accessible through the file system, and so are not saved by any other backup programs. This document discusses the following aspects of using DiskProbe:

- Troubleshooting the Master Boot Record with DiskProbe
- Troubleshooting Boot Sectors with DiskProbe

Because sector editors allow direct access to a physical drive, they make it possible to damage or permanently overwrite critical on-disk data structures that might make all data on a drive or volume permanently inaccessible. Backup all critical data before using any low-level tool such as DiskProbe.

Troubleshooting the Master Boot Record

When troubleshooting partition table problems, the first step is to look at the partition table in the MBR. For information about finding the extended partition table for a logical volume within an extended partition, refer to the section “Walking an Extended Partition” later in this document.

Reading the MBR with DiskProbe

You can use DiskProbe to display the MBR.

To use DiskProbe to display the MBR

1. Start DiskProbe.
2. On the **Drives** menu, click **Physical Drive**.
3. In the **Open Physical Drive** dialog box, double-click the disk that contains the MBR that starts your computer (typically PhysicalDrive0) from the **Available Physical Drives** list box , as shown in Figure 3.

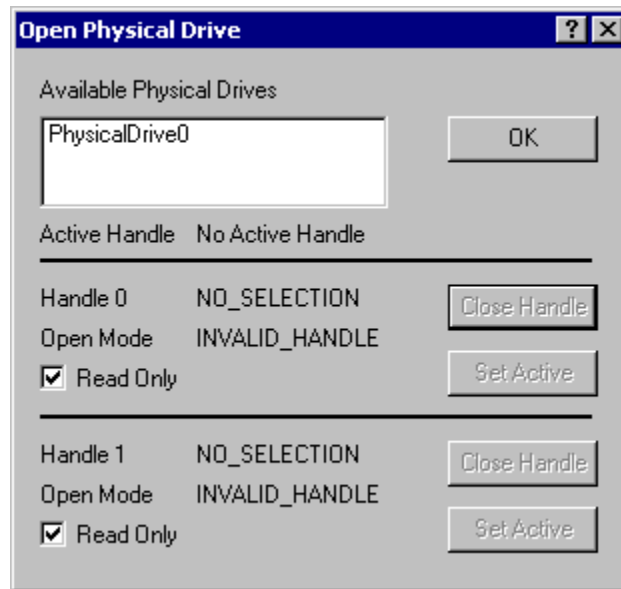


Figure 3 Open Physical Drive Dialog Box

4. In the Handle 0 area of the **Open Physical Drive** dialog box, click **Set Active**, and then click **OK**.
5. On the **Sectors** menu, click **Read**.
6. In the **Read Sectors** dialog box, shown in Figure 4, enter **0** in the **Starting Sector** text box, enter **1** in the **Number of Sectors** dialog box, and then click **Read**. (These values might already be listed by default.)

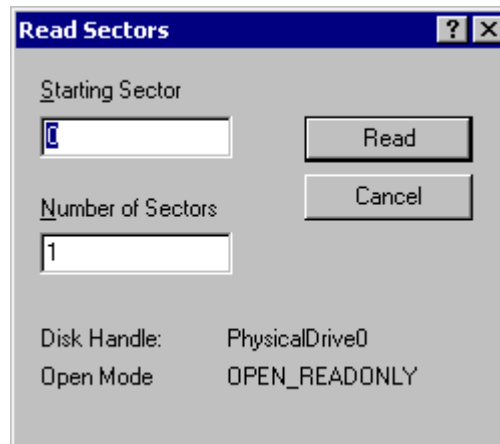


Figure 4 Read Sectors Dialog Box

The available physical drives are listed as PhysicalDrive n , where $n=0$ for the first hard disk, $n=1$ for the second hard disk, and so on. The startup disk is typically PhysicalDrive0. As such, the MBR of PhysicalDrive0 is one of the most important structures on the disk.

To see more sectors at a time, enter a higher number in the **Number of Sectors** text box in Step 6. Then, use the navigation buttons on the toolbar to step through each sector in sequential order. The sector number listed in the title bar is the starting sector of the sequence, but the number of the sector currently displayed is shown in the status bar. The sector that you read in the procedure above should look like an MBR, as shown in Figure 5.

MBR Problems

Figure 5 shows what you typically see at Cylinder 0, Side 0, Sector 1, which is the location of the MBR. The MBR has two distinct parts: Part one, from 0x00 (byte 0) to 0x01BD (byte 445) of the sector, is the master boot code, which includes the disk signature (0x6F9036FA in line 0x1B0). Part two, starting at 0x01BE (byte 446), is the partition table. The signature word for the MBR (always recorded as 0x55AA) makes up the last two bytes of the MBR.

The master boot code in the MBR shown in Figure 5 might not match the code in the MBR on your computer. Differences in partition schemes will affect the way the MBR is constructed from computer to computer. Additionally, some third-party startup and disk partitioning tools modify the master boot code in the MBR.

```
Physical Sector: Cyl 0, Side 0, Sector 1
00000000: 33 C0 8E D0 BC 00 7C FB - 50 07 50 1F FC BE 1B 7C  3.....|.P.P...|
00000010: BF 1B 06 50 57 B9 E5 01 - F3 A4 CB BD BE 07 B1 04  ...PW.....
00000020: 38 6E 00 7C 09 75 13 83 - C5 10 E2 F4 CD 18 8B F5  8n.|.u.....
00000030: 83 C6 10 49 74 19 38 2C - 74 F6 A0 B5 07 B4 07 8B  ...It.8,t.....
00000040: F0 AC 3C 00 74 FC BB 07 - 00 B4 0E CD 10 EB F2 88  ..<.t.....
00000050: 4E 10 E8 46 00 73 2A FE - 46 10 80 7E 04 0B 74 0B  N..F.s*.F..~.t.
00000060: 80 7E 04 0C 74 05 A0 B6 - 07 75 D2 80 46 02 06 83  .~.t....u..F...
00000070: 46 08 06 83 56 0A 00 E8 - 21 00 73 05 A0 B6 07 EB  F...V...!.s....
00000080: BC 81 3E FE 7D 55 AA 74 - 0B 80 7E 10 00 74 C8 A0  ..>.)U.t...~.t..
00000090: B7 07 EB A9 8B FC 1E 57 - 8B F5 CB BF 05 00 8A 56  .....W.....V
000000A0: 00 B4 08 CD 13 72 23 8A - C1 24 3F 98 8A DE 8A FC  ....r#..$?.....
000000B0: 43 F7 E3 8B D1 86 D6 B1 - 06 D2 EE 42 F7 E2 39 56  C.....B..9V
000000C0: 0A 77 23 72 05 39 46 08 - 73 1C B8 01 02 BB 00 7C  .w#r.9F.s.....|
000000D0: 8B 4E 02 8B 56 00 CD 13 - 73 51 4F 74 4E 32 E4 8A  .N..V...sQOtN2..
000000E0: 56 00 CD 13 EB E4 8A 56 - 00 60 BB AA 55 B4 41 CD  V.....V.`..U.A.
000000F0: 13 72 36 81 FB 55 AA 75 - 30 F6 C1 01 74 2B 61 60  .r6..U.u0...t+a`
00000100: 6A 00 6A 00 FF 76 0A FF - 76 08 6A 00 68 00 7C 6A  j.j..v..v.j.h.|j
00000110: 01 6A 10 B4 42 8B F4 CD - 13 61 61 73 0E 4F 74 0B  .j..B....aas.Ot.
00000120: 32 E4 8A 56 00 CD 13 EB - D6 61 F9 C3 49 6E 76 61  2..V.....a..Inva
00000130: 6C 69 64 20 70 61 72 74 - 69 74 69 6F 6E 20 74 61  lid partition ta
00000140: 62 6C 65 00 45 72 72 6F - 72 20 6C 6F 61 64 69 6E  ble.Error loadin
00000150: 67 20 6F 70 65 72 61 74 - 69 6E 67 20 73 79 73 74  g operating syst
00000160: 65 6D 00 4D 69 73 73 69 - 6E 67 20 6F 70 65 72 61  em.Missing opera
```

Part 1 Part Head

```

00000170: 74 69 6E 67 20 73 79 73 - 74 65 6D 00 00 00 00 00    ting system....
00000180: 00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00    .....
00000190: 00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00    .....
000001A0: 00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00    .....
000001B0: 00 00 00 00 00 2C 44 63 - 32 31 37 31 35 0C 80 01    .....,Dc21715...
000001C0: 01 00 07 FE 7F 7D 3F 00 - 00 00 BF A3 5D 00 00 00    ...}?)?.....]...
000001D0: 41 7E 06 FE BF 62 FE A3 - 5D 00 A5 22 38 00 00 00    A~...b..].."8...
000001E0: 81 63 0B FE FF 7A A3 C6 - 95 00 18 A3 44 00 00 00    .C...z.....D...
000001F0: C1 7B 05 FE FF FF BB 69 - DA 00 E2 48 D4 00 55 AA    .{.....i...H..U.

```

Figure 5 The MBR for a computer with three primary partitions and one extended partition

You can use DiskProbe to display the MBR of your computer and compare it with the MBR shown in Figure 5. When comparing the MBRs, check the following:

- The error message strings on the screen matches the error message strings (shown in the right column) of Figure 5.
- Only zeros appear between the text “Missing operating system” and the location 0x01B8.
- A four-byte disk signature might be present starting at location 0x01B8. However, the absence of a disk signature does not necessarily indicate a problem.

Viruses can damage the MBR and make it impossible to start the computer from the hard disk. Because the master boot code in the MBR runs before the operating system starts, a corrupted MBR can prevent the operating system from starting from the infected hard disk.

Backing Up and Restoring the MBR

The MBR on the hard disk that you use to start your computer is the most important sector on the disk. A corrupted MBR on the disk containing the boot partition might prevent the computer from starting. MBRs on other, non-bootable disks are usually not as critical. However, if the partition table information in the other MBRs is not correct, you might not be able to access some or all of the partitions, preventing access to applications and data on those drives.

Backing up the MBR

You should always save the MBR of the startup disk and of any other hard disks that you use, because the MBR of each disk contains the partition table for the disk. DiskProbe supports long file names, which helps in identifying the contents and origin of the saved file.

To backup the MBR

1. Start DiskProbe.

2. On the **Drives** menu, click **Physical Drive**.
3. In the **Open Physical Drive** dialog box, double-click the disk that contains the MBR that you want to backup.
4. In the **Handle 0** area of the **Open Physical Drive** dialog box, click **Set Active**, and then click **OK**.
5. On the **Sectors** menu, click **Read**.
6. In the **Read Sectors** dialog box, enter **0** in the **Starting Sector** text box, enter **1** in the **Number of Sectors** text box, and then click **Read**. (These values might already be listed by default.)
7. On the **View** menu, click **Partition table** to verify that the sector that you read looks like the sample MBR shown in Figure 5, and that the **System ID** list box does not identify all items in the **Partition table index** list box as **Unknown**).

If you have a partition formatted with FAT32 or have upgraded a disk to dynamic disk, the **System ID** list box will display **Unknown** for those partition types, as DiskProbe does not recognize those types of partitions.

If the disk has been upgraded to dynamic disk and new volumes have been created, these new volumes will not appear in the partition table. Any changes to existing volumes, such as extending or spanning multiple disks, will also not appear in the partition table.

8. On the **File** menu, click **Save As**, enter the file name, then click **OK**.

Enter the full path and file name of the MBR to save it to disk. For example, to save the MBR of the startup hard disk to a floppy disk, enter a descriptive file name such as:

A:\ Mbrdisk0.dsk

9. Save the file to the Emergency Repair Disk (ERD).

Consider saving the MBR to more than one floppy disk. Do not save the backup to the hard disk. If the partition table becomes corrupted, you might not have access to the volume.

Restoring the MBR

If you can start the computer, using DiskProbe to restore a saved MBR is a simple process. However, if the startup disk's MBR is damaged, disk partitions and logical volumes might not be accessible, and you might not be able to start Windows 2000 or run DiskProbe, depending upon the extent of the damage.

In cases where the corruption of the startup disk's MBR only affects the master boot code, you can correct the problem by using the Recovery Console and running the **Fixmbr** command. As **Fixmbr** only rewrites the master boot code of the MBR, corruption in the partition table will not be corrected by using the Recovery Console.

There are third-party tools available that can fully restore a backed-up copy of the startup disk's MBR. In addition, if you have any of the Windows NT version 4.0 Resource Kits, a tool supplied in those kits called DiskSave can do the job. See DiskSave help for more information.

The following procedure can also be used to restore the MBR only if the MBR of secondary, non-bootable disks is damaged and a valid backup exists.

To restore the MBR when the computer is startable

1. Start DiskProbe.
2. On the **Drives** menu, click **Physical Drive**.
3. In the **Open Physical Drive** dialog box, double-click the disk that contains the MBR you want to restore.
4. In the **Handle 0** area of the **Open Physical Drive** dialog box, click **Set Active**, clear the **Read Only** check box, and then click **OK**.
5. On the **File** menu, click **Open File**, enter the file name, and then click **Open**. If the file size is not a multiple of 512, DiskProbe displays a message asking if you wish to proceed. If you see this message, click **Cancel**. The file is not the correct size for a saved MBR, is likely corrupted, and is not usable for restoring the MBR on the hard disk.
6. On the **View** menu, click **Bytes** to verify that the file looks like the sample MBR in Figure 5 and that it is 512 bytes long (View the file's properties in Windows Explorer to confirm the size).
7. On the **Sectors** menu, click **Write** to display the **Write Sector** dialog box, shown in Figure 6.

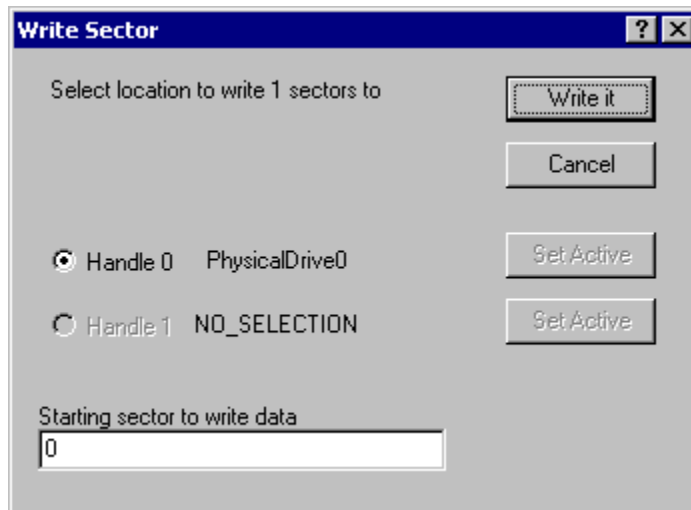


Figure 6 Write Sector Dialog Box

Caution

8. In the **Starting sector to write data** text box, enter **0**, then click **Write it**. DiskProbe displays a dialog box asking you to verify the action before it is executed.
9. Click **Yes** to overwrite the existing MBR.

Be sure that the MBR you restore is from the same disk to which it is being applied. Also, the saved MBR must exactly match the current disk configuration of your computer. If not, partitions and the data they contain might be permanently lost. When you make changes that affect the MBR, such as repartitioning the disk, converting between basic and dynamic disk, or converting to a new file system, use DiskProbe to save updated backup copies of the MBR.

Troubleshooting the Partition Table

The first step in troubleshooting partition table problems is to read the MBR and check the partition table. This section describes viewing the partition table in the MBR. For information about finding the partition table for an extended partition, refer to the section “Walking an Extended Partition” later in this document.

To display the partition table

1. Start DiskProbe.
2. On the **Drives** menu, click **Physical Drive**.
3. In the **Open Physical Drive** dialog box, double-click the disk that contains the partition table that you want to display.
4. In the **Handle 0** area of the **Open Physical Drive** dialog box, click **Set Active**, and then click **OK**.
5. On the **Sectors** menu, click **Read**.
6. In the **Read Sectors** dialog box, enter **0** in the **Starting Sector** text box, enter **1** in the **Number of Sectors** text box, and then click **Read**. (These values might already be listed by default.)
7. On the **View** menu, click **Partition** table. Information about the partition table is displayed, as shown in Figure 7
8. In the **Partition table index** list box, double-click a partition number to see information about a specific partition table entry. To view the boot sector in hexadecimal format, on the **View** menu, click **Bytes**.

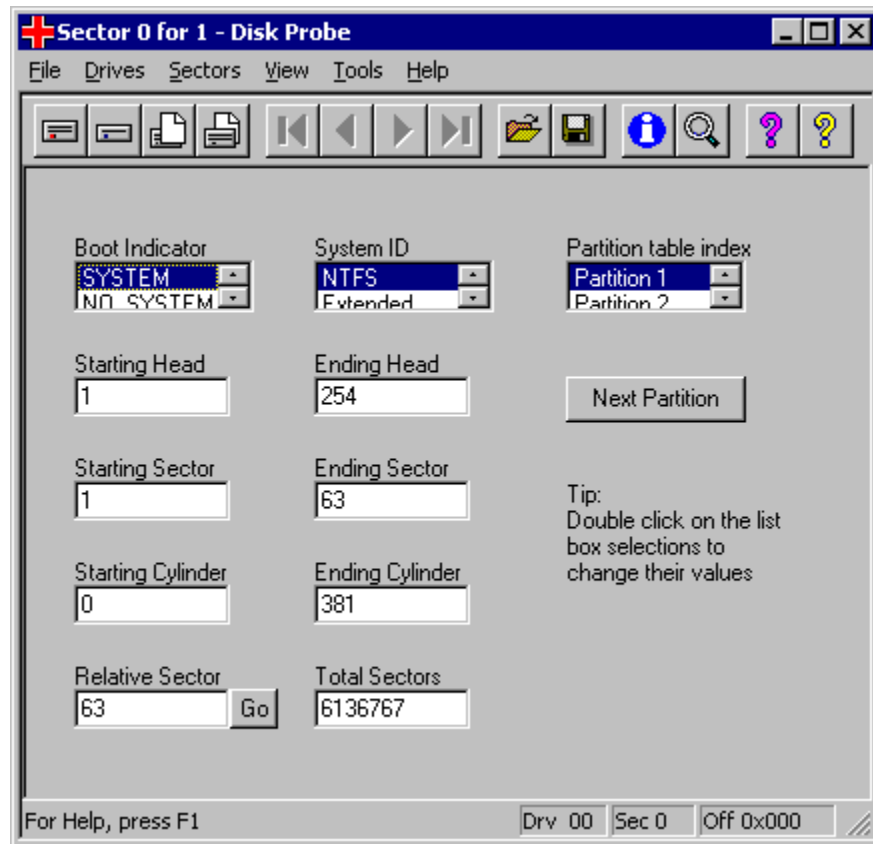


Figure 7 Disk Probe Partition Table Dialog Box

Navigating the Partition Table

When you use DiskProbe to view the partition table, the action that occurs when you click either **Next Partition** or **Go** depends on the type of partition that you are currently viewing. DiskProbe requires manually changing the views of the disk data to be useful. Table 2 provides a description of the **Next Partition** and **Go** buttons.

Table 2 Actions Controlled by Next Partition and Go Buttons

| Original partition | Next Partition button | Go button |
|--------------------|--|---|
| Primary Partition | If the next partition in the MBR is a primary partition, DiskProbe reads that partition's boot sector. Click the View menu, and then click either the NTFS BootSector or the FAT BootSector view, to see the correct display. | DiskProbe moves to the offset listed in the Relative Sector field and reads the boot sector of the current partition. Click the View menu, and then click either the NTFS BootSector or the FAT BootSector view, to see the |

| | | |
|---|--|--|
| | If the next partition in the MBR is an extended partition, DiskProbe reads the partition table in the extended boot record of the first logical volume. | correct display. |
| Extended Partition | Reads the partition table in the extended boot record (EBR) of the first logical volume in the extended partition. | DiskProbe moves to the offset listed in the Relative Sector text box and reads the EBR of that logical volume in the extended partition. |
| Logical volume within an extended partition | Reads the partition table in the EBR of the next logical volume in the extended partition. Click this button again continue to show the extended partition tables of subsequent logical volumes. | DiskProbe moves to the offset listed in the Relative Sector text box and reads the boot sector of the current logical volume. Click the View menu, and then click either the NTFS BootSector or the FAT BootSector view, to see the correct display. |

The **Go** button is located to the immediate right of the **Relative Sectors** text box.

Repairing the Partition Table

If you need to change the values of the fields within the MBR, you can do so within the Partition table view of DiskProbe. This method of editing the data fields is much more accurate than doing so from the Bytes view, in which you have to count the bytes against the offset addresses of the data fields to make sure you are changing the correct field.

To repair the partition table

1. Start DiskProbe.
2. On the **Drives** menu, click **Physical Drive**.
3. In the **Open Physical Drive** dialog box, double-click the disk that contains the partition table you want to repair.
4. In the **Handle 0** area of the **Open Physical Drive** dialog box, click **Set Active**, clear the **Read Only** check box, then click **OK**.
5. On the **Sectors** menu, click **Read**.
6. In the **Read Sectors** dialog box, enter **0** in the **Starting Sector** text box, enter **1** in the **Number of Sectors** text box, and then click **Read**. (These values might already be listed by default.)
7. On the **View** menu, click **Partition table**.
8. In the **Partition table index** list box, double-click the partition number for which you want to change information.

9. Enter the new values on the **Partition Table** view of DiskProbe.
10. On the **Sectors** menu, click **Write**.
11. In the **Write Sector** dialog box, make sure the **Starting** sector to write data is 0, then click **Write it** to save your changes.
12. Click **Yes** to verify your decision to overwrite the existing MBR.

The partition table structure is at the end of the MBR, between bytes 0x01BE (446) and 0x01FD (510). For information about finding the partition table entry for logical volumes in an extended partition, see “Walking an Extended Partition” later in this document.

If none of the information in your partition table is correct, assuming the computer is still running, it might be safer and faster to back up all volumes and restore the data on a freshly formatted disk instead of repairing the partition table. You can then recreate the volumes, reformat the hard disk, and restore all data.

When you view the information for an extended partition in the partition table area of the MBR, the **Total Sectors** text box includes the entire extended partition, which is typically larger than the first logical volume in the partition.

Walking an Extended Partition

The user must create logical volumes before the space in an extended partition can be used. The partition system used to locate and read each logical volume is a variation of the partition table found in the MBR. Each logical volume has its own extended partition table, which lists its own volume first, and the next logical volume in the series is listed second. The third and fourth extended partition table entries are not used. The last logical volume in the extended partition does not have a second entry. The logical volumes link to each subsequent volume as in a chain, and each extended partition table must be read for the system to locate and access all of the configured drives in the computer. This concept is shown in Figure 8.

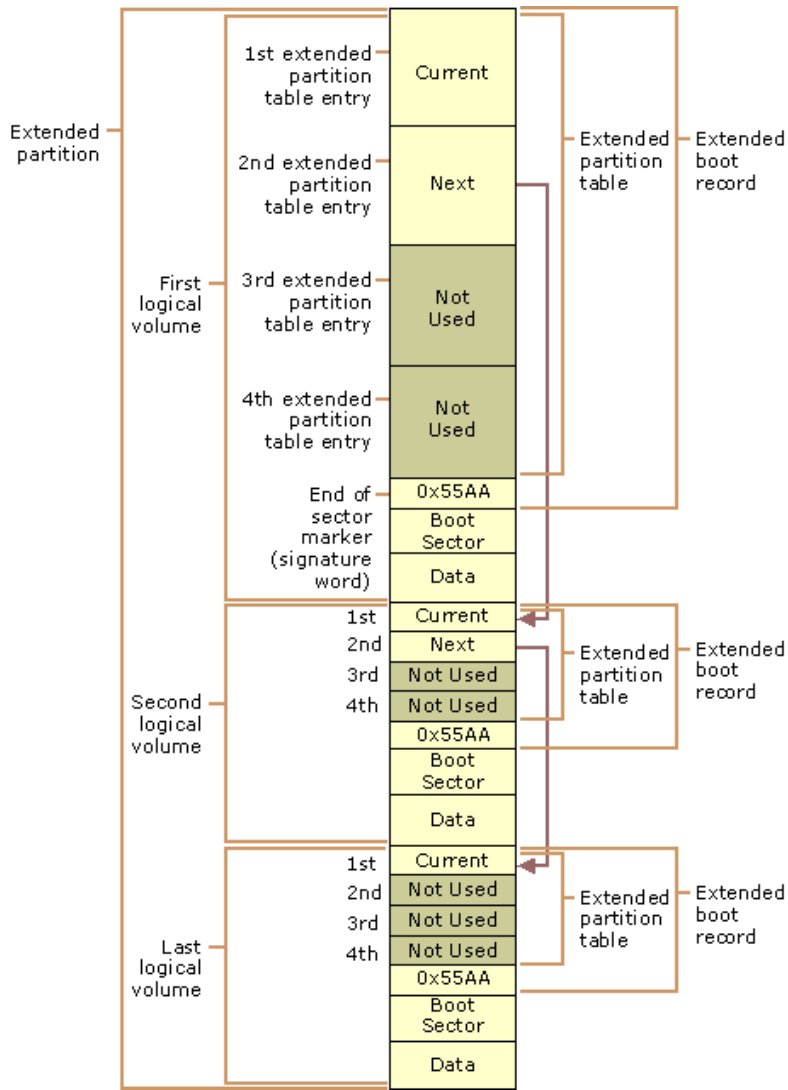


Figure 8 Extended Partition Structure

The term “Walking an extended partition” refers to using the information defined in the extended partition table for one logical volume in the chain to find the next logical volume in the chain. Walking an extended partition is necessary to repair a corrupted link in the chain. If one of those links in the chain becomes corrupted, Windows 2000 can no longer access that volume or the volumes that follow it. Instead of manually searching each extended partition table for pointers to the next volume, you can use DiskProbe.

Manually Walking the Extended Partition

Figure 9 shows a view from Disk Management. The disk scenario defined in this view of Disk Management is used to describe the features and functions of DiskProbe.

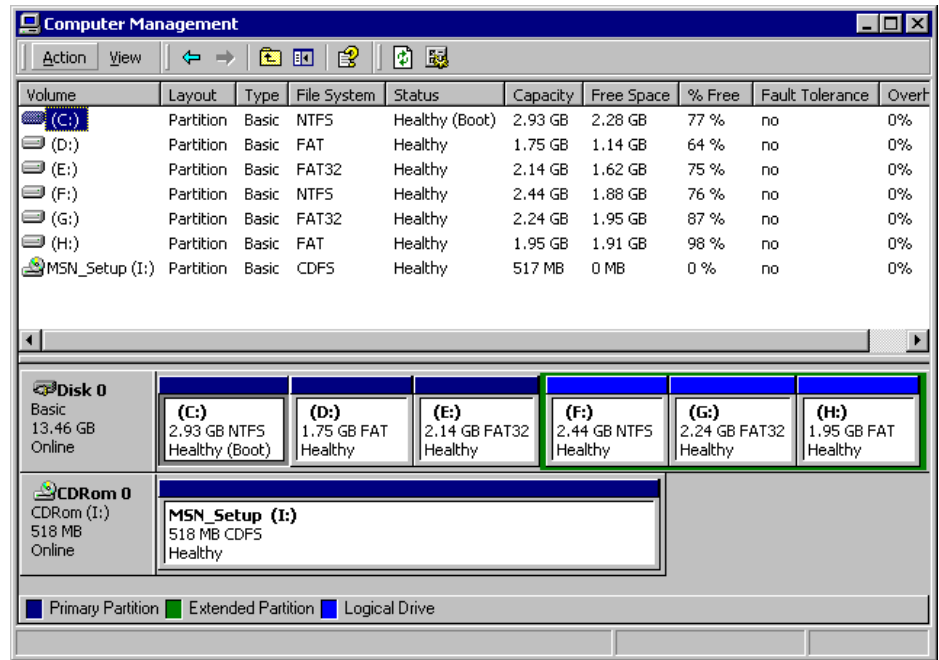


Figure 9 Disk Management view of Disk 0

To manually walk the extended partition in this example, start with the information on Relative Sector 0 in the partition table. The partition table entry for the extended partition points to the first logical volume. By following the links in the partition table entry for each logical volume, you can find the information for all logical volumes.

Table 3 identifies sectors that contain information about the extended partition and the logical volumes. It also describes the buttons used to walk the extended partition.

Table 3 Extended Partition and Logical Volume Sectors

| Relative Sector | Description of Contents |
|-----------------|--|
| 0 | The fourth entry in the MBR's partition table is an extended partition, which is a pointer to the EBR of the first logical volume. To read the extended partition table in the EBR of the first logical volume, drive F, click Next Partition . |
| 14313915 | Partition 1 of the first EBR describes logical volume 1, drive F. Partition 2 is an extended partition with information for the second logical volume, drive G. Partitions 3 and 4 contain only zeros. To read the extended |

Note

partition table for drive G, click **Next Partition**. To read the boot sector of drive F, click **Go**.

5124735 Partition 1 of the second EBR describes logical volume 2, drive G. Partition 2 is an extended partition with information for the third logical volume, drive H. Partitions 3 and 4 contain only zeros. To read the extended partition table for drive H, click **Next Partition**. To read the boot sector of drive G, click **Go**.

9831780 Partition 1 of the third EBR describes logical volume 3, drive H. Partition 2 contains only zeros because drive H is the last logical volume of the extended partition. Partitions 3 and 4 contains only zeros. To read the boot sector of drive H, click **Go**.

Table 4 describes the information in partition 4 of the MBR and each of the partition 1 entries for the extended partition as listed in DiskProbe.

Table 4 Partition Table Definitions for the Extended Partition

| Field Name | MBR Entry | Logical Volume F | Logical Volume G | Logical Volume H |
|------------------------|-----------|------------------|------------------|------------------|
| System ID | Extended | NTFS | Unknown | FAT Large |
| Boot Indicator | NO_SYSTEM | NO_SYSTEM | NO_SYSTEM | NO_SYSTEM |
| Start Head | 0 | 1 | 1 | 1 |
| End Head | 254 | 254 | 254 | 254 |
| Start Sector | 1 | 1 | 1 | 1 |
| End Sector | 63 | 63 | 63 | 63 |
| Start Cylinder | 891 | 891 | 1023 | 1023 |
| End Cylinder | 1023 | 1023 | 1023 | 1023 |
| Relative Sector | 14313915 | 63 | 63 | 63 |
| Total Sectors | 13912290 | 5124672 | 4706982 | 4080447 |

The System ID for volume G is listed as **Unknown** in DiskProbe because volume G is formatted with FAT32. DiskProbe does not directly support FAT32, although you can still edit sectors on and gather relevant information about FAT32-formatted volumes.

The only information in the EBR is the extended partition table. The **System ID** entry for each partition 2 entry should be identified as **Extended** unless it is the last logical volume in the extended partition, in which case it will be identified as **Unknown**. Partition 3 and 4 entries will always be identified as **Unknown** in EBRs. To read the first sector of the next logical volume when the value in the **System ID** list box for partition 2 is **Extended**, click **Next Partition**.

The **Relative Sector** text box for each partition 2 entry in an extended partition is the offset from the beginning of the extended partition, not the offset from the beginning of the partition table associated with the partition 1 entry. Therefore, do not use **Go** to try to read the boot sector for partition 2. Instead, click **Next Partition** to jump to the next logical volume, and then click **Go** while partition 1 is selected.

Occasionally the **System ID** list box for NTFS volume sets and stripe sets does not have the value 0x87, which DiskProbe displays as **NTFS FT** (fault tolerant). Instead, the value might be 0x86, which DiskProbe displays as **FAT FT**. This incorrect value causes no problems, because Windows 2000 uses information in the boot sector to determine which file system to use. However, it can be confusing when trying to understand the data on the disk. To be certain which file system is used for a volume, check the **OEM ID String** text box for a valid value in the boot sector view.

Editing and Repairing the MBR

Attempting to repair a damaged MBR is risky and should not be attempted by a user unfamiliar with the procedure. Making changes to the partition table that are not valid or altering the data in the master boot code might permanently prevent the computer from accessing that disk until it is reformatted. In that case, all data on the inaccessible disk will be lost.

Repairing a damaged MBR is relatively simple if the disk contains only primary partitions and no spanned, striped, mirrored, or RAID-5 volumes or extended partitions.

If the wrong partition is set as the active partition, or if you do not have an active partition on the disk, use DiskProbe to set the **Boot Indicator** list box for the correct partition.

The active partition is always a primary partition with its **Boot Indicator** list box set to **SYSTEM**.

To change or set the Boot Indicator status

1. Start DiskProbe.
2. On the **Drives** menu, click **Physical Drive**.
3. In the **Open Physical Drive** dialog box, double-click the disk that contains the partition table you want to edit.
4. In the **Handle 0** area of the **Open Physical Drive** dialog box, click **Set Active**, clear the **Read Only** check box, and then click **OK**.
5. On the **Sectors** menu, click **Read**.
6. In the **Read Sectors** dialog box, enter **0** in the **Starting Sector** text box, enter **1** in the **Number of Sectors** text box, and then click **Read**. (These values might already be listed by default.)

7. On the **View** menu, click **Partition table**.
8. To identify the system partition, or to specify that no partition is set as the system partition, double-click each partition number in the **Partition table index** list box and watch the resulting display in the **Boot Indicator** list box.
9. To clear the **Boot Indicator** list box for a partition, in the **Boot Indicator** list box, double-click its partition number, and then double-click **NO_SYSTEM**.
— Or —
To set the partition as active, in the **Boot Indicator** list box, double-click its partition number, and then double-click **SYSTEM**.
10. On the **Sectors** menu, click **Write**.
11. In the **Write Sector** dialog box, make sure the **Starting sector to write data** is **0**, and then click **Write it** to save your changes.
12. Click **Yes** to overwrite the existing MBR.

When a volume set was created in Windows NT, it set the FT bit of the **System ID** byte in the partition table. The FT bit indicated that Windows NT needed to use the registry subkey HKEY_LOCAL_MACHINE\SYSTEM\DISK to find all of the members of the volume.

Windows 2000 only allows Windows NT-style FT volumes to be used without upgrading to dynamic disk if they exist at the time of the operating system upgrade. However, Windows 2000 disks must be converted to dynamic disk before you can create new FT volumes. Since dynamic disk uses the disk management database at the end of the disk to create and manage all dynamic volumes rather than recording entries in the partition table, the registry subkey HKEY_LOCAL_MACHINE\SYSTEM\DISK is no longer used in Windows 2000.

In a multiple-boot configuration, the other operating system cannot use a partition with the FT bit set.

To clear the fault tolerant flag for a primary partition

1. Start DiskProbe.
2. On the **Drives** menu, click **Physical Drive**.
3. In the **Open Physical Drive** dialog box, double-click the disk that contains the partition table that you want to edit.
4. In the **Handle 0** area of the **Open Physical Drive** dialog box, click **Set Active**, clear the **Read Only** check box, and then click **OK**.
5. On the **Sectors** menu, click **Read**.
6. In the **Read Sectors** dialog box, enter **0** in the **Starting Sector** text box, enter **1** in the **Number of Sectors** text box, and then click **Read**. (These values might already be listed by default.)
7. On the **View** menu, click **Partition table**.

8. In the **Partition table index** list box, double-click the partition number that you want to change.
9. To remove the FT flag from a file allocation table (FAT)16 partition, in the **System ID** list box, double-click either **FAT 12 bit** (for partitions smaller than 16 MB), **FAT 16 bit** (for partitions between 16 and 32 MB), or **FAT Large** (for partitions larger than 32 MB).
— Or —
To remove the FT flag from an NTFS partition, in the **System ID** list box, double-click **NTFS**.
10. On the **Sectors** menu, click **Write**.
11. In the **Write Sector** dialog box, make sure the **Starting sector to write data** is **0**, and then click **Write it** to save your changes.
12. Click **Yes** to overwrite the existing MBR.

DiskProbe displays a message box to verify that the FT bit is removed from the partition.

If you use DiskProbe to search for the areas on multiple disks that are members of a single striped volume set, the first disk of the set will use the **System ID** 0x87 for an NTFS-formatted FT volume. Subsequent disks within the set will use the **System ID** 0x86, typically reserved for a FAT16-formatted FT volume. This incorrect value causes no problems, because Windows 2000 uses information in the boot sector to determine which file system to use. However, it can be confusing when trying to understand the data on the disk. To be certain which file system is used for a volume, check the **OEM ID String** text box for a valid value in the **Boot Sector** view.

Since the **System ID** 0x87 is only used once per FT volume, it serves as a marker to help you determine which disk was first in the volume set.

All the disks in FAT16-formatted FT volumes use the **System ID** 0x86.

If a primary partition or logical volume extends beyond cylinder 1023, set the following partition table fields to their maximum values: **Starting Sectors** and **Ending Sectors** (63), **Starting Heads** and **Ending Heads** (255), and **Starting Cylinders** and **Ending Cylinders** (1023). Maximum values tell Windows 2000 not to use the information in the partition table to access the volume. Remember that your partitions should not go beyond cylinder 1023 if you have set up your computer to multiple-boot to MS-DOS, Windows 95, or Windows 98.

To edit the MBR

1. Start DiskProbe.
2. On the **Drives** menu, click **Physical Drive**.
3. In the **Open Physical Drive** dialog box, double-click the disk that contains the partition table you want to edit.
4. In the **Handle 0** area of the **Open Physical Drive** dialog box, click **Set Active**, clear the **Read Only** check box, and then click **OK**.
5. On the **Sectors** menu, click **Read**.

6. In the **Read Sectors** dialog box, enter **0** in the **Starting Sector** text box, enter **1** in the **Number of Sectors** text box, and then click **Read**. (These values might already be listed by default.)
7. On the **View** menu, click **Bytes** and make changes to the MBR.

When editing in **Bytes** view, only hexadecimal numerals are permitted. DiskProbe will not allow you to enter non-hexadecimal characters.

8. On the **Sectors** menu, click **Write**.
9. In the **Write Sector** dialog box, make sure the **Starting sector to write data** is **0**, then click **Write it** to save your changes.
10. Click **Yes** to overwrite the existing MBR.

When editing the MBR in **Bytes** view, keep in mind that several of the data structures stored within the partition table fields are of odd size. **Starting Sector** and **Ending Sector** are 6-bit fields, and **Starting Cylinder** and **Ending Cylinder** are 10-bit fields. Editing these structures in **Bytes** view, one byte at a time, can have erroneous and unpredictable results. The contents of these fields should only be edited within the **Partition Table** view of DiskProbe.

When you change partition table information for a primary or an extended partition, the address number of the sector edited is always 0.

If you mistakenly make changes to the master boot code section of the MBR, this can be next to impossible to manually repair, as the master boot code is binary executable programming code. It is recommended that you correct the problem by using the Recovery Console and running the **Fixmbr** command.

Troubleshooting Boot Sectors

Several viruses are known to cause problems with the boot sector, even if the volume is formatted by using the NTFS file system. Infection can occur either by running an infected MS-DOS-based program on a multiple-boot computer or by starting up MS-DOS on a computer with an infected startup floppy disk. Most viruses use BIOS INT 13h calls to transfer themselves to an absolute sector on the disk. Windows 2000 uses protected mode disk drivers instead of BIOS routines to access disk drives. This renders most boot sector viruses harmless and protects computers running Windows 2000 from this type of virus infections, but this protection only applies when Windows 2000 is running. Windows 2000 cannot protect the computer from this type of infection when the computer is running another operating system. In some cases, the damage to the boot sector can cause the computer to lock up after displaying a Stop message (also known as a “blue screen”) with the following message:

```
STOP 0x0000007B INACCESSIBLE_BOOT_DEVICE.
```

Another symptom of a boot sector problem occurs when the computer locks up without displaying any message and the screen remains black.

To display the boot sector by using the Physical Drive option

1. Start DiskProbe.
2. On the **Drives** menu, click **Physical Drive**.
3. In the **Open Physical Drive** dialog box, and then double-click the disk that contains the partition table you want to display.
4. In the **Handle 0** area of the **Open Physical Drive** dialog box, click **Set Active**, and then click **OK**.
5. On the **Sectors** menu, click **Read**.
6. In the **Read Sectors** dialog box, enter **0** in the **Starting Sector** text box, enter **1** in the **Number of Sectors** text box, and then click **Read**. (These values might already be listed by default.)
7. On the **View** menu, click **Partition table**.
8. In the **Partition table index** list box, double-click the partition number containing the boot sector you want to display.
9. Click **Go** located next to the **Relative Sector** text box.
10. To view the boot sector in hexadecimal format, on the **View** menu, click **Bytes**.
— Or —
11. To view formatted information, on the **View** menu, click **NTFS BootSector** or **FAT BootSector**.

Because the boot sector contains several fields that are computer-specific, not every byte in your boot sector will be identical to the information shown in Figure 10 or Figure 11. However, the three following areas on your computer’s boot sector should match what you see in the sample boot sector text given in Figures 10 and 11:

- The first three bytes are the x86 CPU jump instruction.
- The next 11 bytes are the OEM ID string.
- Error message strings appears near the end of the sector.

The samples in Figures 10 and 11 are similar to what you can expect to see on any computer running Windows 2000. You can use the boot sectors to determine the value of the partition table entries. If you do not know the values for the number of sectors per track, number of tracks per cylinder, or number of cylinders, click **Volume Information** on the **Drives** menu for the appropriate volume.

```
00000000: EB 3C 90 4D 53 44 4F 53 - 35 2E 30 00 02 40 01 00  .<.MSDOS5.0..@..
...
000001A0: B0 4E 54 4C 44 52 20 20 - 20 20 20 20 0D 0A 4E 54  .NTLDR      ..NT
000001B0: 4C 44 52 20 69 73 20 6D - 69 73 73 69 6E 67 FF 0D  LDR is missing..
000001C0: 0A 44 69 73 6B 20 65 72 - 72 6F 72 FF 0D 0A 50 72  .Disk error...Pr
000001D0: 65 73 73 20 61 6E 79 20 - 6B 65 79 20 74 6F 20 72  ess any key to r
000001E0: 65 73 74 61 72 74 0D 0A - 00 00 00 00 00 00 00 00  estart.....
000001F0: 00 00 00 00 00 00 00 00 - 00 00 00 AC BF CC 55 AA  ....U.
```

Figure 10 Partial Bytes View of a FAT16 Boot Sector

```

00000000: EB 52 90 4E 54 46 53 20 - 20 20 20 00 02 08 00 00  .R.NTFS      .....
...
00000180: EB F2 C3 0D 0A 41 20 64 - 69 73 6B 20 72 65 61 64  ....A disk read
00000190: 20 65 72 72 6F 72 20 6F - 63 63 75 72 72 65 64 00  error occurred.
000001A0: 0D 0A 4E 54 4C 44 52 20 - 69 73 20 6D 69 73 73 69  ..NTLDR is missi
000001B0: 6E 67 00 0D 0A 4E 54 4C - 44 52 20 69 73 20 63 6F  ng...NTLDR is co
000001C0: 6D 70 72 65 73 73 65 64 - 00 0D 0A 50 72 65 73 73  mpresed...Press
000001D0: 20 43 74 72 6C 2B 41 6C - 74 2B 44 65 6C 20 74 6F  Ctrl+Alt+Del to
000001E0: 20 72 65 73 74 61 72 74 - 0D 0A 00 00 00 00 00 00  restart.....
000001F0: 00 00 00 00 00 00 00 00 - 83 A0 B3 C9 00 00 55 AA  ....U.

```

Figure 11 Partial Bytes View of an NTFS Boot Sector

Although other corruption problems are possible, if any strings are incorrect or missing you should assume that the sector is corrupted. Even if there is no obvious damage, the boot sector might not work correctly.

To check the status of the boot sector, rename the file Ntldr, and then restart Windows 2000 from the hard disk.

Ntldr usually has the system, hidden, and read-only file attributes set. Because Windows 2000 cannot start when the boot sector on the boot partition is corrupted, if the partition is formatted with FAT16, you can start the computer from an MS-DOS startup floppy disk to change the attributes of the file. To change the attributes, use the MS-DOS **Attrib** command. Sample syntax for removing the file attributes (assuming the external MS-DOS commands on the bootable floppy are in the A:\Dos folder and Ntldr is in the C:\ folder) is as follows:

```
a:\dos\attrib -s -h -r c:\ntldr
```

Sample syntax for reapplying the proper file attributes after renaming Ntldr is as follows:

```
a:\dos\attrib +s +h +r c:\ntldr
```

If the boot sector that you check is on a FAT16 primary partition and if the boot sector is okay, but the Ntldr file is corrupted or missing (or in the case of this test, renamed), this error message appears:

```

NTLDR is missing
Press any key to restart

```

If the boot sector that you check is on an NTFS primary partition and if the boot sector is okay, but the Ntldr file is corrupted or missing (or in the case of this test, renamed), this error message appears:

```

NTLDR is missing
Press Ctrl+Alt+Del to restart

```

These errors, generated by the executable boot code of the boot sector, indicate that the boot sector is okay and that the problem lies with Ntldr. You can restore a damaged or deleted Ntldr by using the ERD.

If you do not see one of the preceding error messages, the problem is probably a corrupted boot sector. For information about replacing the boot sector, refer to the section “Backing Up and Restoring the Boot Sector” later in this document.

If you conducted the boot sector test described earlier, be sure to rename the changed file name back to Ntldr so that the boot sector can find it again.

Viewing the Boot Sector

You can use DiskProbe to view formatted information in the BIOS parameter block and the extended BIOS parameter block.

To display a boot sector by using the Logical Volume option

1. Start DiskProbe.
2. On the **Drives** menu, click **Logical Volume**.
3. In the **Logical Volumes** list box of the **Open Logical Volume** dialog box, and then double-click the drive letter of the volume that contains the boot sector that you want to display.
4. In the **Handle 0** area of the **Open Logical Volume** dialog box, click **Set Active**, then click **OK**.
5. On the **Sectors** menu, click **Read**.
6. In the **Read Sectors** dialog box, enter **0** in the **Starting Sector** text box, enter **1** in the **Number of Sectors** text box, and then click **Read**. (These values might already be listed by default.)
7. On the **View** menu, click either **FAT BootSector** or **NTFS BootSector** (as appropriate to the selected partition or logical volume) to view the boot sector information.

Clicking **NTFS BootSector** from the **View** menu displays a dialog box like the one in Figure 12. For an example of the **FAT BootSector** view, see Figure 13 later in this document.

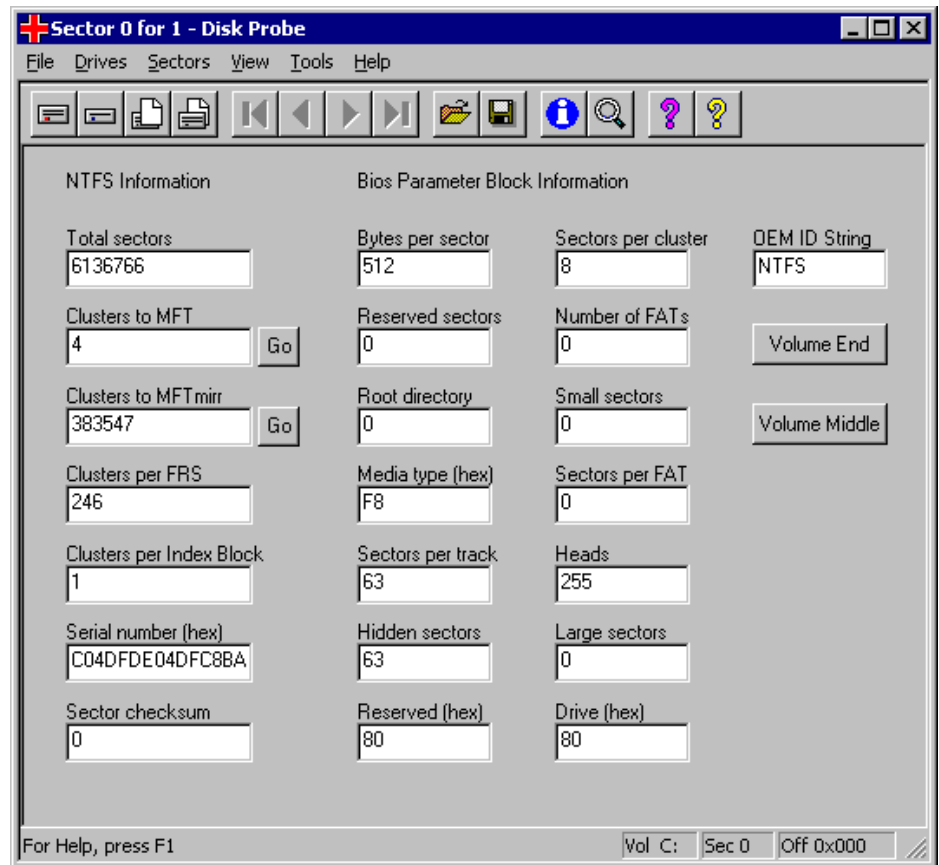


Figure 12 DiskProbe View of an NTFS Boot Sector

When viewing an NTFS boot sector, click **Go**, located next to the text boxes **Clusters to MFT** and **Clusters to MFT mirr**, to go to the first sector of the master file table (MFT) or the MFT mirror, respectively.

DiskProbe requires that you use the **Bytes** view of the disk data to see the MFT or the MFT mirror.

A backup of NTFS boot sectors are recorded, either at the logical center of the volume (on drives formatted with Microsoft Windows NT version 3.51 and earlier) or at the end of the volume (on drives formatted with Windows NT 4.0 and Windows 2000). Click **Volume Middle** to view the backup NTFS boot sector on volumes formatted with Windows NT 3.51 and earlier. Click **Volume End** to view the backup NTFS boot sector on volumes formatted with Windows 2000.

When converting a volume from FAT16 to NTFS or when extending a volume set on Windows NT 3.51 or earlier, the backup boot sector is not written in the correct location. Two such situations are:

- Converting a volume from FAT16 to NTFS.
- Extending a volume set.

Make sure that the sector looks like an NTFS boot sector, as shown in Figure 11, and that the information is accurate for the volume you are viewing.

When you view a FAT16 boot sector, the **Volume End** button is displayed, but the FAT16 boot sector on the volume has no backup. To find where volumes begin and end so that you can repair a partition table, use **Volume End** for both FAT16 and NTFS primary partitions and logical volumes. **Volume End** allows you to read the sector immediately before the beginning of the next primary partition or logical volume.

When a backup NTFS boot sector is displayed, ignore the **Go**, **Volume End**, or **Volume Middle** buttons. These buttons read the correct sectors only when you view the original boot sector at the beginning of the volume. Using them against a backup NTFS boot sector will cause the program to display erroneous data.

Backing Up and Restoring the Boot Sector

The boot sector contains information that the file system uses to access the volume. The MBR uses the boot sector on the system partition to load the operating system kernel files, or in the case of Windows 2000, the boot loader.

There are several procedures that you can use to backup and restore the boot sector. The procedure that you choose will depend on which of the following that you are backing up or restoring the boot sector for:

- A primary partition.
- A logical volume within an extended partition.
- Several boot sectors in an extended partition.
- A spanned, striped, mirrored, or RAID-5 volume.

When you view a boot sector with the procedures presented later in this section, verify that the boot sector displayed on the screen resembles the sample boot sectors displayed in Figures 10 and 11.

To backup a boot sector for a primary partition by using the Physical Drive option

1. Start DiskProbe.
2. On the **Drives** menu, click **Physical Drive**.
3. In the **Open Physical Drive** dialog box, and then double-click the disk that contains the boot sector that you want to backup.
4. In the **Handle 0** area of the **Open Physical Drive** dialog box, click **Set Active**, and then click **OK**.
5. On the **Sectors** menu, click **Read**.

Tip

6. In the **Read Sectors** dialog box, enter **0** in the **Starting Sector** text box, enter **1** in the **Number of Sectors** text box, and then click **Read**. (These values might already be listed by default.)
7. On the **View** menu, click **Partition table**.
8. In the **Partition table index** list box, double-click the partition number for the boot sector that you want to save.
9. Click **Go** located next to the **Relative Sector** text box.
10. On the **View** menu, click either **FAT BootSector** or **NTFS BootSector** to verify that the sector displayed looks like a boot sector (check the **OEM ID String** text box for a valid value).
11. On the **File** menu, click **Save As**, enter the file name, and then click **OK**.
12. Save the file to the ERD.

Consider saving the boot and backup boot sectors to more than one floppy disk. DiskProbe supports the use of long file names, which helps in identifying the contents, disk origin and sector address of the saved file. It is recommended that you include the sector number of the boot sector as part of a descriptive file name. This simplifies the process of restoring the boot sector to the correct location on disk.

To backup the boot sector by using the Logical Volume option

1. Start DiskProbe.
2. On the **Drives** menu, click **Logical Volume**.
3. In the **Logical Volumes** list box of the **Open Logical Volume** dialog box, and then double-click the drive letter of the volume that contains the boot sector that you want to backup.
4. In the **Handle 0** area of the **Open Logical Volume** dialog box, click **Set Active**, then click **OK**.
5. On the **Sectors** menu, click **Read**.
6. In the **Read Sectors** dialog box, enter **0** in the **Starting Sector** text box, enter **1** in the **Number of Sectors** text box, and then click **Read**.
7. On the **View** menu, click either **FAT BootSector** or **NTFS BootSector** to verify that the sector displayed looks like a boot sector (check the **OEM ID String** text box for a valid value).
8. On the **File** menu, click **Save As**, enter the file name, and then click **OK**.
9. Save the file to the ERD.

Backing up a Boot Sector in an Extended Partition

The following procedure requires walking an extended partition. For more information, refer to the “Walking an Extended Partition section earlier in this document.

To backup a boot sector in an extended partition by using the Physical Drive option

1. Start DiskProbe.
2. On the **Drives** menu, click **Physical Drive**.
3. In the **Open Physical Drive** dialog box, and then double-click the disk that contains the boot sector you want to backup.
4. In the **Handle 0** area of the **Open Physical Drive** dialog box, click **Set Active**, and then click **OK**.
5. On the **Sectors** menu, click **Read**.
6. In the **Read Sectors** dialog box, enter **0** in the **Starting Sector** text box, enter **1** in the **Number of Sectors** text box, and then click **Read**. (These values might already be listed by default.)
7. On the **View** menu, click **Partition table**.
8. In the **Partition table index** list box, double-click the entry for the extended partition that you want to back up.
9. Click **Next Partition** to step sequentially through the logical volumes until you reach the volume that contains the boot sector that you want to backup
10. Click **Go** located next to **Relative Sector** to read the boot sector.
11. On the **View** menu, click either **FAT BootSector** or **NTFS BootSector** to verify that the sector displayed looks like a boot sector (check the **OEM ID String** text box for a valid value).
12. On the **File** menu, click **Save As**, enter the file name, and then click **OK**.
13. Save the file to the ERD.

You can also use the **Search** function on the **Tools** menu to find each boot sector. For more information about finding a boot sector with **Search**, see “Finding a Boot Sector” later in this document.

You can also print a map of the disk by using the DiskMap tool, which provides you with the address of the extended partition table entry for each logical volume. You can then use DiskProbe to read the first sector of the logical volume that contains the boot sector that you want to backup, and use **Go** to read the boot sector.

Restoring a Boot Sector on a Primary Partition

If a boot sector has been corrupted on a non-startup volume, and a backup of that boot sector was made prior to the corruption, you can use DiskProbe to restore the backed up boot sector and regain access to the volume.

To restore the boot sector for a non-startup primary partition on a bootable computer by using the Physical Drive option

1. Start DiskProbe.
2. On the **Drives** menu, click **Physical Drive**.
3. In the **Open Physical Drive** dialog box, double-click the disk that contains the boot sector you want to restore.
4. In the **Handle 0** area of the **Open Physical Drive** dialog box, click **Set Active**, clear the **Read Only** check box, and then click **OK**. If you know the address of the sector that you need to restore, skip to step 11. Otherwise, continue with the following step.
6. On the **Sectors** menu, click **Read**.
7. In the **Read Sectors** dialog box, enter **0** in the **Starting Sector** text box, enter **1** in the **Number of Sectors** text box, and then click **Read**. (These values might already be listed by default.)
8. On the **View** menu, click **Partition table**.
9. In the **Partition table index** list box, double-click the partition number for the boot sector that you want to replace.
10. In the **Relative Sector** text box, record the sector number for the boot sector of the selected partition.
11. On the **File** menu, click **Open File**, enter the name of the file that contains the boot sector, and then click **Open**.
12. On the **View** menu, click either **FAT BootSector** or **NTFS BootSector** to verify that the sector displayed looks like a boot sector (check the **OEM ID String** text box for a valid value). Also verify that the file is 512 bytes long. (The size can be confirmed by viewing the file properties in Windows Explorer.)
13. On the **Sectors** menu, click **Write**.
14. In the **Starting sector to write data** dialog box, enter the sector number recorded from step 10, and then click **Write it**.
15. Click **Yes** to overwrite the existing boot sector.

DiskProbe displays an error if the size of the opened file is not a multiple of 512.

To restore the boot sector for a non-startup primary partition on a bootable computer by using the Logical Volume option

1. Start DiskProbe.

2. On the **Drives** menu, click **Logical Volume**.
3. In the **Logical Volumes** list box in the **Open Logical Volume** dialog box, double-click the drive letter of the volume that contains the boot sector that you want to restore.
4. In the **Handle 0** area of the **Open Logical Volume** dialog box, click **Set Active**, clear the **Read Only** check box, and then click **OK**.
5. On the **File** menu, click **Open File**, enter the name of the file that contains the boot sector, and then click **Open**.
6. On the **View** menu, click either **FAT BootSector** or **NTFS BootSector** to verify that the sector displayed looks like a boot sector (check the **OEM ID String** text box for a valid value). Also verify that the file is 512 bytes long. (The size can be confirmed by the viewing the file properties in Windows Explorer.)
7. On the **Sectors** menu, click **Write**.
8. In the **Starting sector to write data** text box, enter **0**, then click **Write it**.
9. Click **Yes** to overwrite the existing boot sector.

If the boot sector on the startup volume is damaged, you will not be able to start Windows 2000, and you will not be able to run DiskProbe. In this case, you can recreate the boot sector by using either the **Fixboot** command of Recovery Console or the ERD.

If you know the address of the boot sector that you want to restore (which you will know if the sector number is part of the file name), use the procedure “To restore the boot sector for a primary partition on a bootable computer by using the Physical Drive option” earlier in this document, but skip steps 6 through 10.

If you do not know the address of the boot sector that you want to restore, you can find it by walking either the partition table for primary partitions or the extended partition table for an extended partition, depending upon on which type of partition the volume is located. Follow steps 1 through 9 in the procedure “To backup boot sectors in an extended partition by using Physical Drive” earlier in this document and record the figure in the **Relative Sector** text box. You can also run DiskMap to find the boot sector addresses of existing partitions. For primary partitions, use the **Relative Sector** address for that partition’s boot sector. For logical volumes within extended partitions, add the address from **Relative Sector** to the extended boot record (EBR) address to get the boot sector address.

Restoring a Boot Sector in an Extended Partition

To restore the boot sector in an extended partition by using the Physical Drive option

1. Start DiskProbe.
2. On the **Drives** menu, click **Physical Drive**.
3. In the **Open Physical Drive** dialog box, double-click the disk that contains the boot sector that you want to restore.

4. In the **Handle 0** area of the **Open Physical Volume** dialog box, click **Set Active**, clear the **Read Only** check box, and then click **OK**. If you know the address of the sector that you need to write, skip to step 13. Otherwise, continue with the following step.
6. On the **Sectors** menu, click **Read**.
7. In the **Read Sectors** dialog box, enter **0** in the **Starting Sector** text box, enter **1** in the **Number of Sectors** text box, and then click **Read**. (These values might already be listed by default.)
8. On the **View** menu, click **Partition table**.
9. In the **Partition table index** list box, double-click the entry for the extended partition.
10. Click **Next Partition** to step sequentially through the logical volumes until you reach the volume that contains the boot sector that you want to replace.
11. Click **Go** located next to **Relative Sector** to read the boot sector.
12. Record the sector number for the boot sector of the selected volume as shown in both the title bar and the status bar of DiskProbe.
13. On the **File** menu, click **Open File**, enter the name of the file that contains the boot sector that you want to restore, and then click **Open**.
14. On the **View** menu, click either **FAT BootSector** or **NTFS BootSector** to verify that the sector displayed looks like a boot sector (check the **OEM ID String** text box for a valid value). Also verify that the file is 512 bytes long. (The size can be confirmed by viewing the file properties in Windows Explorer.)
15. On the **Sectors** menu, click **Write**.
16. Enter the sector number recorded from step 12 in the **Starting sector to write data** text box, and then click **Write it**.
17. Click **Yes** to overwrite the existing boot sector.

Restoring a Backup NTFS Boot Sector as the Primary Boot Sector

If you format an NTFS volume while running Windows NT 4.0 or Windows 2000, a backup of the boot sector for NTFS volume is copied to the end of the volume.

Earlier versions of Windows NT put a backup NTFS boot sector at the center of the volume. If you did not make a backup of the boot sector, find the backup sector and copy it to the first sector of the volume.

“Finding a Boot Sector,” later in this document, describes different methods for finding the boot sector. It is always advisable to save backup boot sectors to a disk. When you find a boot sector within the boundaries of an NTFS volume, make sure that the data is applicable for the volume. If you have deleted volumes, or extended an NTFS volume set, there might be old, invalid boot sectors on the disk.

To restore the backup NTFS boot sector of a non-active, primary partition as the primary boot sector

This procedure assumes that the primary partition requiring boot sector restoration is not the active, primary partition, since damage to the active, primary partition's boot sector would prevent the system from starting, and prevent the user from starting DiskProbe. If the damaged boot sector was that of the active, primary partition, you can recreate that boot sector by using the Recovery Console.

1. Start DiskProbe.
2. On the **Drives** menu, click **Physical Drive**.
3. In the **Open Physical Drive** dialog box, double-click the disk that contains the backup NTFS boot sector that you want to restore.
4. In the **Handle 0** area of the **Open Physical Volume** dialog box, click **Set Active**, clear the **Read Only** check box, and then click **OK**.
5. On the **Sectors** menu, click **Read**.
6. In the **Read Sectors** dialog box, enter **0** in the **Starting Sector** text box, enter **1** in the **Number of Sectors** text box, and then click **Read**. (These values might already be listed by default.)
7. On the **View** menu, click **Partition table**.
8. In the **Partition table index** list box, double-click the partition number containing the backup NTFS boot sector that you want to copy.
9. Record the sector number for the primary boot sector as shown in the **Relative Sector** text box.
10. Click **Go** located next to **Relative Sector** to read the boot sector.
11. On the **View** menu, click **NTFS BootSector**.
12. If the volume was formatted by Windows NT 4.0 or Windows 2000, click **Volume End**. If it was formatted by an earlier version of Windows NT, click **Volume Middle** to access the backup NTFS boot sector.
13. Verify that the backup NTFS boot sector displayed looks like a boot sector (check the **OEM ID String** text box for a valid value).
14. On the **File** menu, click **Save As**, enter the file name, and then click **OK**.
15. On the **File** menu, click **Open File**, enter the name of the file that contains the backup NTFS boot sector that was just saved, and then click **Open**.
16. On the **Sectors** menu, click **Write**.
17. In the **Starting sector to write data** text box, enter the sector number recorded from step 9, and then click **Write it**.
18. Click **Yes** to overwrite the existing boot sector.

If the startup disk's damaged boot sector was not properly backed up by a tool such as DiskProbe prior to the corruption, you can try using a third-party, MS-DOS-based tool that can perform disk editing at the sector level to locate and copy the backup NTFS boot sector to the address of the primary boot sector. However, it might be easier to use the Recovery Console's **Fixboot** command or the ERD to resolve the problem.

To restore the backup NTFS boot sector of an extended partition as the primary boot sector

1. Start DiskProbe.
2. On the **Drives** menu, click **Physical Drive**.
3. In the **Open Physical Drive** dialog box, double-click the disk that contains the backup NTFS boot sector you want to restore.
4. In the **Handle 0** area of the **Open Physical Volume** dialog box, click **Set Active**, clear the **Read Only** check box, and then click **OK**.
5. On the **Sectors** menu, click **Read**.
6. In the **Read Sectors** dialog box, enter **0** in the **Starting Sector** text box, enter **1** in the **Number of Sectors** text box, and then click **Read**. (These values might already be listed by default.)
7. On the **View** menu, click **Partition table**.
8. In the **Partition table index** list box, double-click the entry for the extended partition.
9. Click **Next Partition** to step sequentially through the logical volumes until you reach the volume that contains the backup NTFS boot sector that you want to copy.
10. Click **Go** located next to the **Relative Sector** text box to read the boot sector.
11. Record the sector number for the primary boot sector of the selected volume as shown in both the title bar and the status bar of DiskProbe.
12. On the **View** menu, click **NTFS BootSector**.
13. If the volume was formatted by Windows NT 4.0 or Windows 2000, click **Volume End**. If it was formatted by any earlier version of Windows NT, click **Volume Middle**.
14. Verify that the backup NTFS boot sector displayed looks like a boot sector (check the **OEM ID String** text box for a valid value).
15. On the **File** menu, click **Save As**, enter the file name, and then click **OK**.
16. On the **File** menu, click **Open File**, enter the name of the file that contains the backup NTFS boot sector that was just saved, and then click **Open**.
17. On the **Sectors** menu, click **Write**.
18. In the **Starting sector to write data** text box, enter the sector number recorded from step 11, and then click **Write it**.

19. Click **Yes** to overwrite the existing boot sector.

Finding a Boot Sector

When you need to search for the address of a boot sector to repair a damaged partition table, it is possible that one or more bytes of data in the MBR have become corrupted. Verify that the information you use to find a boot sector, or a partition table in an extended partition, is accurate. For example, if you use the figure in the partition table view's **Total Sectors** text box for one primary partition and add the **Volume Information** dialog box's **Sectors Per Track** figure to calculate the location of the boot sector for the next partition, confirm there is a valid boot sector at that address before editing the corrupted partition table.

The easiest way to find the start of a primary partition is to find the partition's boot sector. For an extended partition, subtract the **Sectors per Track** figure, found in both the **FAT16 BootSector** (as shown in Figure 13) and **NTFS BootSector** views and the **Volume Information** dialog box (available from the **Drives** menu), from the sector address of the extended boot sector (this information is found in both the title bar and the status bar of DiskProbe), to find the EBR for the logical volume.

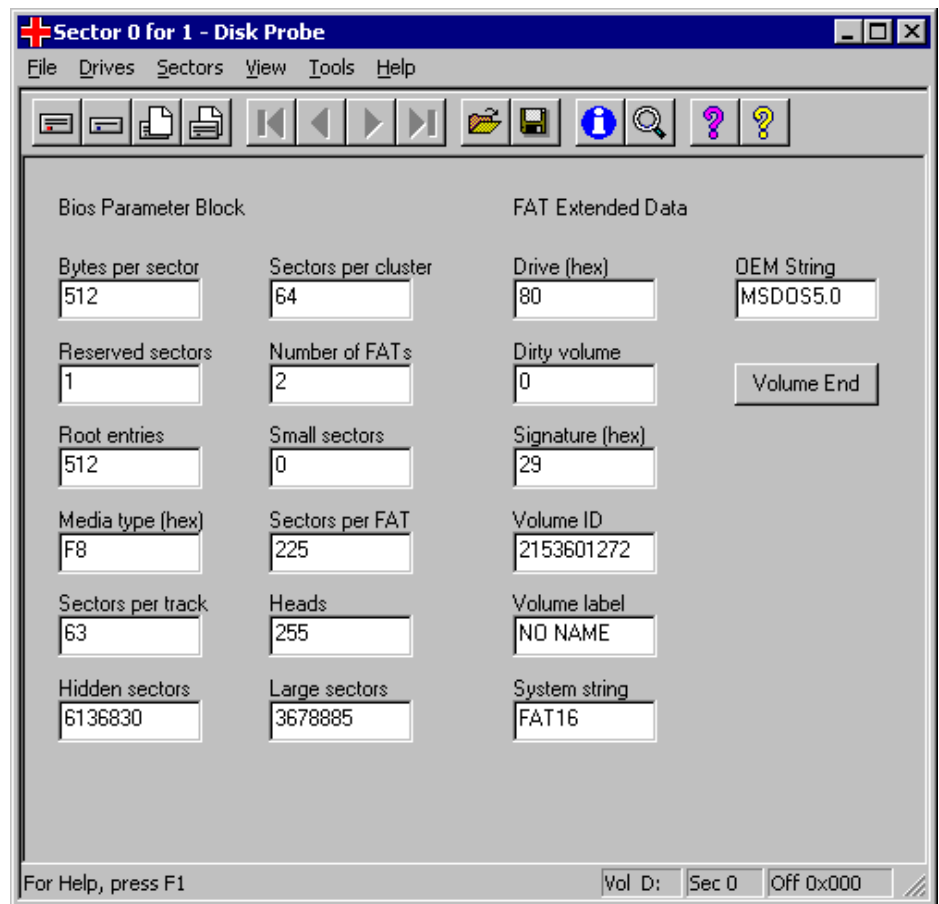


Figure 13 DiskProbe View of a FAT16 Boot Sector

The MBR for your hard disk is always the first sector on head 1, cylinder 0. Because disks typically have either 32 or 63 sectors per track, you can usually find the boot sector for the first partition at either relative sector 32 or relative sector 63.

In the **Partition Table** view, click **Go** located next to the **Relative Sectors** text box to allow DiskProbe to read the boot sector of the currently selected partition table entry.

The **Partition Table** view includes a **Total Sectors** text box. If you know that the value in this field is accurate, use the value to calculate the start of the next primary partition or logical volume.

The **FAT BootSector** and **NTFS BootSector** views have a field that contains the total number of sectors. If you know where a primary partition or logical volume starts, you can add the number of sectors to the starting sector to determine where the next partition or logical volume starts.

The total number of sectors for an NTFS boot sector displays in the **Total Sectors** text box when you click the option **NTFS BootSector** on the **View** menu. The total number of sectors for a FAT16 volume displays in either the **Small Sectors** or **Large Sectors** text box when you click **FAT BootSector** on the **View** menu. If the size of the primary partition or logical volume is fewer than 65,536 sectors (fits within 16-bit representation), the **Small Sectors** text box contains the size and **Large Sectors** is 0. Otherwise, the size is in the **Large Sectors** text box and the **Small Sectors** text box is 0.

Finding Boot Sectors Elsewhere on the Volume

Following are three situations in which you might find a boot sector that is not at the beginning of a volume. You need to know about these situations so that you do not use the information about these boot sectors to repair the partition table:

- If you format an NTFS volume, or convert a volume from FAT16 to NTFS when running either Windows NT 4.0 or Windows 2000, the NTFS volume has a backup of its NTFS boot sector at the end of the volume. Earlier versions of Windows NT put the backup NTFS boot sector at the middle of the volume when you format it, but do not create a backup when you convert the volume.
- When you extend an NTFS volume set created by Windows NT 3.51 or earlier, Windows 2000 puts a new backup copy of the boot sector at the end of the extended volume set. It does not remove the old copy of the boot sector.
- If you have deleted volumes, old, invalid boot sectors might remain on the disk.

Searching for Boot Sectors using DiskProbe

You can also use DiskProbe to search for boot sectors. As you find each one, write down its sector address number and use the **Partition Table** view in DiskProbe to enter the information.

To search for boot sectors

1. Start DiskProbe.
2. On the **Drives** menu, click **Physical Drive**.
3. In the **Open Physical Drive** dialog box, double-click the disk that you want to search for boot sectors.
4. In the **Handle 0** area of the **Open Physical Volume** dialog box, click **Set Active**, and then click **OK**.
5. On the **Tools** menu, click **Search** to display the **Search Sectors** dialog box.
6. In the **Search Sectors** dialog box, select the appropriate options in the **Search Type** and **Character Type** group boxes.
7. To search a specific location within the sectors, in the **Offset in hex** text box, select the **Search at offset** check box and specify an address.

— Or —

To set the range of sectors searched, enter values in the **First Sector to Search** and **Last Sector to Search** text boxes.

— Or —

Keep the default values in the **First Sector to Search** and **Last Sector to Search** text boxes.

The first and second options will make the search run much faster than searching each entire disk. You can search the entire disk by choosing the third option. Entering **0** in the **First Sector to Search** text box will accomplish the same task.

8. Enter a string of printable characters to search for in the **Enter characters to search for** text box. You can use up to 64 characters.
9. Click **Search** to begin the search procedure.

The default value for the **First Sector to Search** is set by using the sector you are in when you open the **Search** function. The above procedure, as it does not include a step to read any particular sector prior to initiating the search, begins by default at sector 0 on the disk. The **Last Sector to Search** is always the last sector on disk.

Figure 14 is an example of a search for an NTFS boot sector.

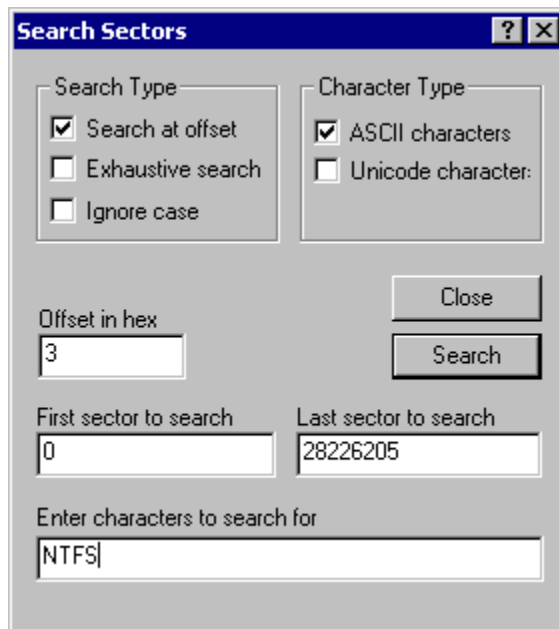


Figure 14 Search Sectors Dialog Box

The content of the boot sector for an NTFS volume depends on the version of Windows that you run when formatting the volume. If you specify a value in the **Offset in hex** text box, you might need to search at a different offset depending on the data and the version of Windows used to format the volume. If you do not know

which version of Windows was used to format a volume, search for the string “NTFS.” DiskProbe might find matches that are *not* boot sectors.

If you use the **Search at offset** option, note that each sector is only checked for the match of the first character in the string at the offset address specified. If the **Search at offset** option is selected but nothing is entered in the **Offset in hex** text box, an offset of 0 is used by default. While this type of search can be much faster than the **Exhaustive search** option, you need to know the offset address of the data you are seeking in order to find a match. Use the **Ignore case** option if you are unsure of the exact characters used in the string.

ASCII characters are one byte in length and are primarily used within text documents.

Use the **ASCII characters** option to search for characters used in text documents.

Unicode characters, which are the default character set used in Windows 2000, are two bytes in length. US English Unicode characters have the high byte set to null.

Since Windows 2000 uses Unicode to store NTFS file names in the File Record Segment, a data structure used within NTFS, use the **Unicode characters** option to search for file names on NTFS partitions.

If DiskProbe finds a match, it displays the address of the sector. Be sure to verify that each sector DiskProbe finds is a valid boot sector. The **View** menu enables you to look at the sector in **Bytes** view or as an **NTFS BootSector** or a **FAT BootSector**.