

Debugging Windows Vista - The Boot.ini File Bug

(Microsoft Corporation)

Introduction: The Boot.ini file bug

If you try to set up a kernel debugger on the latest builds of Microsoft Windows Vista, you will notice a major change. Depending on which build you have, the Boot.ini file is either missing from the build or it is present but is ignored.

This tip explains how to configure kernel debugging on a computer that is running Windows Vista. These instructions are designed for the simplest case: a computer on which Windows Vista is the only installed operating system. Instructions for computers that have both Windows Vista and versions of Windows earlier than Windows Vista are only slightly more complex and will be described in detail in the Windows Driver Kit (WDK) for Windows Vista.

Note: Early beta test builds of Windows Vista still use the Boot.ini file and, on those builds, you can edit the Boot.ini file to set up debugging. You can identify builds that still use Boot.ini by the "NOBCD" boot parameter in the Boot.ini file. The NOBCD parameter was supported only in these early builds of Windows Vista. The NOBCD parameter and the Boot.ini file are ignored in later builds, and the Boot.ini file is not created at all in builds 5253 and later.

What has Changed?

Windows Vista introduces a new boot loader architecture; a new firmware-independent boot configuration and storage system called Boot Configuration Data (BCD); and a new boot option editing tool, BCDEdit (BCDEdit.exe). These components are designed to load Windows more quickly and more securely.

The traditional Windows NT boot loader, Ntldr, is replaced by Windows Boot Manager (Bootmgr.exe) and a set of system-specific boot loaders. In the new configuration, Windows Boot Manager is generic and unaware of the specific requirements for each operating system, and each system-specific boot loader is optimized for the system that it loads.

What Happened to Boot.ini?

On BIOS-based computers that are running only Windows Vista, the Boot.ini text file is gone (not hidden) and any remnants of it on interim beta test builds are ignored. On computers with both earlier versions of Windows and with Windows Vista, the Boot.ini file remains to support the older versions but it does not affect booting in Windows Vista.

On EFI-based computers that are running Windows Vista, boot options are still stored in NVRAM. However, in Windows Vista you use BCDEdit to edit boot options on an EFI-based computer just as you would on a BIOS-based computer, instead of accessing NVRAM directly by using Windows APIs or specialized tools such as NvrBoot.

Windows Vista also includes new BCD classes that are supported by the WMI provider and enable you to edit BCD programmatically. For information about BCD classes, see the end of this tip.

How Do I Edit the Boot Options?

BCDEdit (Bcdedit.exe) is a command-line tool that edits boot options in Windows Vista. It replaces Bootcfg (a boot options editing tool that is included in Windows XP and Windows Server 2003) and NvrBoot (a boot options editing tool for EFI-based computers).

Use BCDEdit to edit the boot configuration in Windows Vista. You can continue to use Bootcfg and NvrBoot and to edit the Boot.ini file in versions of Windows earlier than Windows Vista, even if they are installed on a computer that also has Windows Vista installed.

Debugging Windows Vista - The Boot.ini File Bug (Microsoft Corporation)

Configuring Kernel Debugging in Windows Vista

You can use the following procedure to create a boot entry for debugging in Windows Vista. This procedure uses BCDEdit, which is the boot option editing tool for Windows Vista.

Briefly, the steps are:

1. Open a Command Prompt window with elevated privileges.
2. Edit the global debugging settings, if necessary.
3. Display the global debugging settings.
4. Create a new boot entry by copying an existing boot entry.
5. Add the new boot entry to the boot menu.
6. Enable debugging on the new boot entry.
7. Verify that debugging is set on the new boot entry.
8. Change the default boot entry.
9. Reboot the computer.

The rest of this tip describes each step in more detail.

1. Open a Command Prompt window with elevated privileges.
The Command Prompt window must have elevated privileges, even if you are a member of the Administrator group on the computer.
 - Click the Start button, point to All Programs, and then click Accessories.
 - Right-click Command Prompt, and then click Run Elevated.
2. Edit the global debugging settings, if necessary.
In Windows Vista, the debugging connection type (serial, 1394, or USB) and the attributes of the connection type (debugport or channel) are determined by a global parameter, dbgsettings, that affects all boot entries, although you can set the debugging connection type and attributes individually for particular boot entries.

The default global debugging settings for Windows Vista are:

```
Debugger type:          serial
Debugger port number:   1
Debugger baudrate:     115200
```

To change the global debugging settings, use the following command syntax.

```
bcdedit /dbgsettings {serial [baudrate:value] [debugport:value] |
1394 [channel:value] | USB}
```

For example:

```
bcdedit /dbgsettings 1394 channel:1
bcdedit /dbgsettings serial debugport:2
```

If the command succeeds, BCDEdit displays the following message:

Debugging Windows Vista - The Boot.ini File Bug

(Microsoft Corporation)

The operation completed successfully.

3. Display the global debugging settings.
To verify that a command to change the global debugging setting was successful, use the following command to display all boot settings on the system.

```
bcdedit /enum all
```

In the list that appears, the global debugging settings are the last entry in the list.

The following example shows the settings after the example `bcdedit /dbgsettings 1394 channel:1` command from step 2 earlier in this tip.

...

Kernel Debugger Settings Group

```
-----  
Identifier:           {dbgsettings}  
Type:                 20100000  
Debugger type:        1394  
Debugger 1394 channel: 1
```

...

4. Create a new boot entry by copying an existing boot entry.
At the command prompt, type the following command. This command copies the Microsoft Windows boot entry that was last used to boot Windows, identified as {current}, and creates a new boot entry named DebugEntry. (This name is used throughout the rest of this tip. You can assign any name to the new entry.)

```
bcdedit /copy {current} /d DebugEntry
```

If the command succeeds, BCDEdit displays a message such as:

```
The entry was successfully copied to {68602c25-5097-11da-99de-000802209f1b}.
```

The global universal identifier (GUID) that appears between braces in the preceding message is the identifier of the new boot entry. You use the identifier to represent the entry in all subsequent BCDEdit commands.

If the command fails, be sure that you are running in a Command Prompt window with elevated privileges and that all of the command parameters are spelled correctly, including the braces around {current}.

5. Add the new boot entry to the boot menu.
In Windows Vista, new boot loader entries are not added to the boot menu automatically. If you skip this step, the boot menu might not appear (because there is only one boot loader entry) or the menu appears, but does not list the new boot loader entry.

Debugging Windows Vista - The Boot.ini File Bug (Microsoft Corporation)

You can place the boot loader entries in any order. Separate each identifier (the GUID or a reserved identifier, such as {current}) with a space. To add the DebugEntry boot entry to the boot menu after the {current} entry, use the following command.

```
bcdedit /displayorder {current} {ID}
```

In this command, {ID} is the GUID of the DebugEntry boot loader entry, including the braces ({}), as the following example shows:

```
bcdedit /displayorder {current} {68602c25-5097-11da-99de-000802209f1b}
```

To verify that the display order is correct, use the following command:

```
bcdedit
```

When you type bcdedit without additional parameters, BCDEdit displays the boot manager entry and the boot loader entries in the order that they will appear in the menu.

The Windows Boot Manager entry also includes the boot menu display order, as the following example shows.

```
...  
  
Windows Boot Manager  
-----  
Identifier:           {bootmgr}  
Type:                10100002  
Device:              partition=C:  
Description:         Windows Boot Manager  
Locale:              en-US  
Inherit options:    {emssettings}  
                   {dbgsettings}  
                   {5189b25c-5558-4bf2-bca4-289b11bd29e2}  
Default:             {current}  
Display order:    {current}  
                   {68602c25-5097-11da-99de-000802209f1b}  
Timeout:            30  
  
...
```

6. Enable debugging on the new boot entry.
Use the following procedure to enable debugging on the DebugEntry boot entry that you created in step 4 earlier in this tip. The first step is necessary because you need the GUID of the boot entry for the command that sets debugging.

To list all of the boot loader entries on the computer, type the following command.

```
bcdedit /enum osloader
```

Find the boot entry named DebugEntry and copy the GUID, including the surrounding braces ({}), from the Identifier attribute of the entry.

Debugging Windows Vista - The Boot.ini File Bug (Microsoft Corporation)

```
...  
  
Windows Boot Loader  
-----  
Identifier:           {68602c25-5097-11da-99de-000802209f1b}  
Type:                10200003  
Device:              partition=C:  
Path:                \Windows\system32\winload.exe  
Description:         DebugEntry  
Locale:              en-US  
Inherit options:     {emssettings}  
                    {dbgsettings}  
                    {5189b25c-5558-4bf2-bca4-289b11bd29e2}  
  
Windows device:      partition=C:  
Windows root:        \Windows  
No Execute policy:   OptIn  
  
...
```

To enable debugging on the DebugEntry boot entry, use the following command syntax.

```
bcdedit /debug {ID} ON
```

where {ID} represents the GUID that you copied from the boot entry in step 2.

The following example demonstrates this command with the GUID from the example in step 2.

```
bcdedit /debug {68602c25-5097-11da-99de-000802209f1b} ON
```

If the command succeeds, BCDEdit displays the following message at the command prompt.

```
The operation completed successfully.
```

7. Verify that debugging is set on the new boot entry.
Use the following procedure to verify that debugging is set on the new DebugEntry boot entry.

To list the boot loader entries on the computer, use the following command.

```
bcdedit /enum osloader
```

Find the boot entry named DebugEntry. The result should include a Kernel debugger attribute on the entry with a value of Yes, as the following example shows.

Debugging Windows Vista - The Boot.ini File Bug (Microsoft Corporation)

```
...

Windows Boot Loader
-----
Identifier:           {68602c25-5097-11da-99de-000802209f1b}
Type:                10200003
Device:              partition=C:
Path:                \Windows\system32\winload.exe
Description:      DebugEntry
Locale:              en-US
Inherit options:     {emssettings}
                    {dbgsettings}
                    {5189b25c-5558-4bf2-bca4-289b11bd29e2}

Windows device:      partition=C:
Windows root:        \Windows
No Execute policy:   OptIn
Kernel debugger:   Yes

...
```

8. Change the default boot entry.

The default boot entry is the boot entry that the boot manager selects when the boot menu timeout expires. Unlike earlier versions of Windows, in Windows Vista the order of boot entries in the boot menu (the display order) does not determine which boot loader entry is the default.

By default, the Microsoft Windows boot entry that the system creates is the default entry. Because the default entry is also the boot entry that is used to boot the system most recently, it is identified as {current}.

If you want the new debug-enabled boot loader entry, DebugEntry, to be the default boot entry, use the following command.

```
bcdedit /default {ID}
```

where {ID} is the GUID of the DebugEntry boot entry, as the following example shows.

```
bcdedit /default {68602c25-5097-11da-99de-000802209f1b}
```

If the command is successful, BCDedit displays the following message at the command prompt.

```
The operation completed successfully.
```

To see this change, use the bcdedit command (without parameters) to display the entries that appear in the boot menu.

```
bcdedit
```

This command produces the following output. Notice that all instances of the DebugEntry GUID have been replaced by the reserved identifier {default}.

The identifiers {default} and {current} are reserved identifiers that indicate the default boot entry and the boot entry that was used most recently to boot the system. If a boot entry is both the default and the current entry, its identifier is {current}. If a boot entry is not the default or the current entry, BCDEdit restores its original GUID as its identifier.

Debugging Windows Vista - The Boot.ini File Bug (Microsoft Corporation)

You can identify the boot entry by its name (in the Description attribute), but you must use the identifier, not the description, in BCDEdit commands.

```
Windows Boot Manager
-----
Identifier:           {bootmgr}
Type:                10100002
Device:              partition=C:
Description:      Windows Boot Manager
Locale:              en-US
Inherit options:     {emssettings}
                    {dbgsettings}
                    {5189b25c-5558-4bf2-bca4-289b11bd29e2}

Default:          {default}
Display order:       {current}
                    {default}
Timeout:             30

Windows Boot Loader
-----
Identifier:           {current}
Type:                10200003
Device:              partition=C:
Path:                \Windows\system32\winload.exe
Description:      Microsoft Windows
Locale:              en-US
Inherit options:     {emssettings}
                    {dbgsettings}
                    {5189b25c-5558-4bf2-bca4-289b11bd29e2}

Windows device:      partition=C:
Windows root:        \Windows
No Execute policy:   OptIn

Windows Boot Loader
-----
Identifier:           {default}
Type:                10200003
Device:              partition=C:
Path:                \Windows\system32\winload.exe
Description:      DebugEntry
Locale:              en-US
Inherit options:     {emssettings}
                    {dbgsettings}
                    {5189b25c-5558-4bf2-bca4-289b11bd29e2}

Windows device:      partition=C:
Windows root:        \Windows
No Execute policy:   OptIn
Kernel debugger:     Yes
```

9. Reboot the computer.

When you reboot your computer, the Windows Boot Manager displays the boot menu. You can select the Microsoft Windows boot entry or the DebugEntry. If you do not select either choice, the Windows Boot Manager selects the default, DebugEntry.

Debugging Windows Vista - The Boot.ini File Bug

(Microsoft Corporation)

Windows Boot Manager

Choose the operating system or tool you want to start:
(Use the arrow keys to highlight your choice.)

Microsoft Windows

DebugEntry [debugger enabled]

To specify an advanced option for this choice, press F8.