Installing an Unsigned Driver During Development and Test - Windows drivers

By EliotSeattle

Archived: 2026-04-05 22:47:52 UTC

By default, 64-bit versions of Windows Vista and later versions of Windows will load a kernel-mode driver only if the kernel can verify the driver signature. However, this default behavior can be disabled during early driver development and for non-automated testing. Developers can use one of the following mechanisms to temporarily disable load-time enforcement of a valid driver signature. However, to fully automate testing of a driver that is installed by Plug and Play (PnP), the catalog file of the driver must be signed. Signing the driver is required because Windows Vista and later versions of Windows display a driver signing dialog box for unsigned drivers that require a system administrator to authorize the installation of the driver, potentially preventing any user without the necessary privileges from installing the driver and using the device. This PnP driver installation behavior cannot be disabled on Windows Vista and later versions of Windows.

Windows Vista and later versions of Windows support the F8 Advanced Boot Option -- "Disable Driver Signature Enforcement" -- that disables load-time signature enforcement for a kernel-mode driver only for the current system session. This setting does not persist across system restarts.

Attaching an active kernel debugger to a development or test computer disables load-time signature enforcement for kernel-mode drivers. To use this debugging configuration, attach a debugging computer to a development or test computer, and enable kernel debugging on the development or test computer by running the following command:

```
bcdedit -debug on
```

To use BCDEdit, the user must be a member of the Administrators group on the system and run the command from an elevated command prompt. To open an elevated Command Prompt window, create a desktop shortcut to *Cmd.exe*, select and hold (or right-click) the shortcut, and select **Run as administrator**.

However, there are situations in which a developer might need to have a kernel debugger attached, yet also need to maintain load-time signature enforcement. For example, when a driver stack has an unsigned driver (such as a filter driver) that fails to load it may invalidate the entire stack. Because attaching a debugger allows the unsigned driver to load, the problem appears to vanish as soon as the debugger is attached. Debugging this type of issue may be difficult.

In order to facilitate debugging such issues, the kernel-mode code signing policy supports the following registry value:

```
HKLM\SYSTEM\CurrentControlSet\Control\CI\DebugFlags
```

This registry value is of type REG_DWORD, and can be assigned a value based on a bitwise OR of one or more of the following flags:

0x00000001
This flag value configures the kernel to break into the debugger if a driver is unsigned. The developer or tester can then choose to load the unsigned driver by entering **g** at the debugger prompt.

0x00000010
This flag value configures the kernel to ignore the presence of the debugger and to always block an unsigned driver from loading.

If this registry value does not exist in the registry or has a value that is not based on the flags described previously, the kernel always loads a driver in kernel debugging mode regardless of whether the driver is signed.

Note This registry value does not exist in the registry by default. You must create the value in order to debug the kernel-mode signature verification.
