Last reviewed: 06/02/2009

```
                     HP iLO2 NMI Watchdog Driver
             NMI sourcing for iLO2 based ProLiant Servers
                       Documentation and Driver by
               Thomas Mingarelli <thomas.mingarelli@hp.com>
```

The HP iLO2 NMI Watchdog driver is a kernel module that provides basic
watchdog functionality and the added benefit of NMI sourcing. Both the
watchdog functionality and the NMI sourcing capability need to be enabled
by the user. Remember that the two modes are not dependant on one another.
A user can have the NMI sourcing without the watchdog timer and vice-versa.

Watchdog functionality is enabled like any other common watchdog driver. That
is, an application needs to be started that kicks off the watchdog timer. A
basic application exists in the Documentation/watchdog/src directory called
watchdog-test.c. Simply compile the C file and kick it off. If the system
gets into a bad state and hangs, the HP ProLiant iLO 2 timer register will
not be updated in a timely fashion and a hardware system reset (also known as
an Automatic Server Recovery (ASR)) event will occur.

The hpwdt driver also has four (4) module parameters. They are the following:

```
soft_margin - allows the user to set the watchdog timer value
allow_kdump - allows the user to save off a kernel dump image after an NMI
nowayout    - basic watchdog parameter that does not allow the timer to
              be restarted or an impending ASR to be escaped.
priority    - determines whether or not the hpwdt driver is first on the
              die_notify list to handle NMIs or last. The default value
              for this module parameter is 0 or LAST. If the user wants to
              enable NMI sourcing then reload the hpwdt driver with
              priority=1 (and boot with nmi_watchdog=0).
```

NOTE: More information about watchdog drivers in general, including the ioctl
      interface to /dev/watchdog can be found in
      Documentation/watchdog/watchdog-api.txt and Documentation/IPMI.txt.

The priority parameter was introduced due to other kernel software that relied
on handling NMIs (like oprofile). Keeping hpwdt's priority at 0 (or LAST)
enables the users of NMIs for non critical events to be work as expected.

The NMI sourcing capability is disabled by default due to the inability to
distinguish between "NMI Watchdog Ticks" and "HW generated NMI events" in the
Linux kernel. What this means is that the hpwdt nmi handler code is called
each time the NMI signal fires off. This could amount to several thousands of
NMIs in a matter of seconds. If a user sees the Linux kernel's "dazed and
confused" message in the logs or if the system gets into a hung state, then
the hpwdt driver can be reloaded with the "priority" module parameter set
(priority=1).

1. If the kernel has not been booted with nmi_watchdog turned off then
   edit /boot/grub/menu.lst and place the nmi_watchdog=0 at the end of the
   currently booting kernel line.
2. reboot the sever
3. Once the system comes up perform a rmmod hpwdt
4. insmod /lib/modules/`uname -r`/kernel/drivers/char/watchdog/hpwdt.ko

priority=1

Now, the hpwdt can successfully receive and source the NMI and provide a log
message that details the reason for the NMI (as determined by the HP BIOS).

Below is a list of NMIs the HP BIOS understands along with the associated
code (reason):

No source found	00h
Uncorrectable Memory Error	01h
ASR NMI	1Bh
PCI Parity Error	20h
NMI Button Press	27h
SB_BUS_NMI	28h
ILO Doorbell NMI	29h
ILO IOP NMI	2Ah
ILO Watchdog NMI	2Bh
Proc Throt NMI	2Ch
Front Side Bus NMI	2Dh
PCI Express Error	2Fh
DMA controller NMI	30h
Hypertransport/CSI Error	31h

-- Tom Mingarelli
   (thomas.mingarelli@hp.com)