

## AMD64 specific boot options

There are many others (usually documented in driver documentation), but only the AMD64 specific ones are listed here.

### Machine check

Please see Documentation/x86/x86\_64/machinecheck for sysfs runtime tunables.

mce=off

Disable machine check

mce=no\_cmci

Disable CMCI (Corrected Machine Check Interrupt) that Intel processor supports. Usually this disablement is not recommended, but it might be handy if your hardware is misbehaving.

Note that you'll get more problems without CMCI than with due to the shared banks, i.e. you might get duplicated error logs.

mce=dont\_log\_ce

Don't make logs for corrected errors. All events reported as corrected are silently cleared by OS.

This option will be useful if you have no interest in any of corrected errors.

mce=ignore\_ce

Disable features for corrected errors, e.g. polling timer and CMCI. All events reported as corrected are not cleared by OS and remained in its error banks.

Usually this disablement is not recommended, however if there is an agent checking/clearing corrected errors (e.g. BIOS or hardware monitoring applications), conflicting with OS's error handling, and you cannot deactivate the agent, then this option will be a help.

mce=bootlog

Enable logging of machine checks left over from booting. Disabled by default on AMD because some BIOS leave bogus ones. If your BIOS doesn't do that it's a good idea to enable though to make sure you log even machine check events that result in a reboot. On Intel systems it is enabled by default.

mce=nobootlog

Disable boot machine check logging.

mce=tolerancelevel[,monarchtimeout] (number,number)

tolerance levels:

0: always panic on uncorrected errors, log corrected errors

1: panic or SIGBUS on uncorrected errors, log corrected errors

2: SIGBUS or log uncorrected errors, log corrected errors

3: never panic or SIGBUS, log all errors (for testing only)

Default is 1

Can be also set using sysfs which is preferable.

monarchtimeout:

Sets the time in us to wait for other CPUs on machine checks. 0 to disable.

nomce (for compatibility with i386): same as mce=off

Everything else is in sysfs now.

## boot-options.txt

### APICs

`apic`            Use IO-APIC. Default

`noapic`        Don't use the IO-APIC.

`disableapic`   Don't use the local APIC

`nolapic`       Don't use the local APIC (alias for i386 compatibility)

`pirq=...`     See Documentation/x86/i386/IO-APIC.txt

`noapictimer`   Don't set up the APIC timer

`no_timer_check` Don't check the IO-APIC timer. This can work around problems with incorrect timer initialization on some boards.

`apicmaintimer` Run time keeping from the local APIC timer instead of using the PIT/HPET interrupt for this. This is useful when the PIT/HPET interrupts are unreliable.

`noapicmaintimer` Don't do time keeping using the APIC timer.  
Useful when this option was auto selected, but doesn't work.

`apicpmtimer`   Do APIC timer calibration using the pmtimer. Implies `apicmaintimer`. Useful when your PIT timer is totally broken.

### Early Console

syntax: `earlyprintk=vga`  
         `earlyprintk=serial[,ttySn[,baudrate]]`

The early console is useful when the kernel crashes before the normal console is initialized. It is not enabled by default because it has some cosmetic problems.  
Append `,keep` to not disable it when the real console takes over.  
Only `vga` or `serial` at a time, not both.  
Currently only `ttyS0` and `ttyS1` are supported.  
Interaction with the standard serial driver is not very good.  
The VGA output is eventually overwritten by the real console.

### Timing

`notsc`  
Don't use the CPU time stamp counter to read the wall time.  
This can be used to work around timing problems on multiprocessor systems with not properly synchronized CPUs.

`report_lost_ticks`  
Report when timer interrupts are lost because some code turned off interrupts for too long.

`nohpet`

## boot-options.txt

Don't use the HPET timer.

### Idle loop

idle=poll

Don't do power saving in the idle loop using HLT, but poll for rescheduling event. This will make the CPUs eat a lot more power, but may be useful to get slightly better performance in multiprocessor benchmarks. It also makes some profiling using performance counters more accurate. Please note that on systems with MONITOR/MWAIT support (like Intel EM64T CPUs) this option has no performance advantage over the normal idle loop. It may also interact badly with hyperthreading.

### Rebooting

reboot=b[ios] | t[riple] | k[bd] | a[cpu] | e[fi] [, [w]arm | [c]old

bios Use the CPU reboot vector for warm reset

warm Don't set the cold reboot flag

cold Set the cold reboot flag

triple Force a triple fault (init)

kbd Use the keyboard controller. cold reset (default)

acpi Use the ACPI RESET\_REG in the FADT. If ACPI is not configured or the ACPI reset does not work, the reboot path attempts the reset using the keyboard controller.

efi Use efi reset\_system runtime service. If EFI is not configured or the EFI reset does not work, the reboot path attempts the reset using the keyboard controller.

Using warm reset will be much faster especially on big memory systems because the BIOS will not go through the memory check. Disadvantage is that not all hardware will be completely reinitialized on reboot so there may be boot problems on some systems.

reboot=force

Don't stop other CPUs on reboot. This can make reboot more reliable in some cases.

### Non Executable Mappings

noexec=on|off

on Enable (default)

off Disable

### SMP

additional\_cpus=NUM Allow NUM more CPUs for hotplug

(defaults are specified by the BIOS, see

Documentation/x86/x86\_64/cpu-hotplug-spec)

### NUMA

numa=off Only set up a single NUMA node spanning all memory.

numa=noacpi Don't parse the SRAT table for NUMA setup

## boot-options.txt

numa=fake=<size>[MG]

If given as a memory unit, fills all system RAM with nodes of size interleaved over physical nodes.

numa=fake=<N>

If given as an integer, fills all system RAM with N fake nodes interleaved over physical nodes.

## ACPI

acpi=off Don't enable ACPI

acpi=ht Use ACPI boot table parsing, but don't enable ACPI interpreter

acpi=force Force ACPI on (currently not needed)

acpi=strict Disable out of spec ACPI workarounds.

acpi\_sci={edge,level,high,low} Set up ACPI SCI interrupt.

acpi=noirq Don't route interrupts

## PCI

pci=off Don't use PCI

pci=conf1 Use conf1 access.

pci=conf2 Use conf2 access.

pci=rom Assign ROMs.

pci=assign-busses Assign busses

pci=irqmask=MASK Set PCI interrupt mask to MASK

pci=lastbus=NUMBER Scan upto NUMBER busses, no matter what the mptable says.

pci=noacpi Don't use ACPI to set up PCI interrupt routing.

## IOMMU (input/output memory management unit)

Currently four x86-64 PCI-DMA mapping implementations exist:

1. <arch/x86\_64/kernel/pci-nommu.c>: use no hardware/software IOMMU at all (e.g. because you have < 3 GB memory).  
Kernel boot message: "PCI-DMA: Disabling IOMMU"
2. <arch/x86\_64/kernel/pci-gart.c>: AMD GART based hardware IOMMU.  
Kernel boot message: "PCI-DMA: using GART IOMMU"
3. <arch/x86\_64/kernel/pci-swiotlb.c> : Software IOMMU implementation. Used e.g. if there is no hardware IOMMU in the system and it is need because you have >3GB memory or told the kernel to us it (iommu=soft))  
Kernel boot message: "PCI-DMA: Using software bounce buffering for IO (SWIOTLB)"
4. <arch/x86\_64/pci-calgary.c> : IBM Calgary hardware IOMMU. Used in IBM pSeries and xSeries servers. This hardware IOMMU supports DMA address mapping with memory protection, etc.  
Kernel boot message: "PCI-DMA: Using Calgary IOMMU"

## boot-options.txt

```
iommu=[<size>][,noagp][,off][,force][,noforce][,leak[=<nr_of_leak_pages>]]  
      [,memaper[=<order>]][,merge][,forcesac][,fullflush][,nomerge]  
      [,noaperture][,calgary]
```

### General iommu options:

off	Don't initialize and use any kind of IOMMU.
noforce	Don't force hardware IOMMU usage when it is not needed. (default).
force	Force the use of the hardware IOMMU even when it is not actually needed (e.g. because < 3 GB memory).
soft	Use software bounce buffering (SWIOTLB) (default for Intel machines). This can be used to prevent the usage of an available hardware IOMMU.

### iommu options only relevant to the AMD GART hardware IOMMU:

<size>	Set the size of the remapping area in bytes.
allowed	Overwrite iommu off workarounds for specific chipsets.
fullflush	Flush IOMMU on each allocation (default).
nofullflush	Don't use IOMMU fullflush.
leak	Turn on simple iommu leak tracing (only when CONFIG_IOMMU_LEAK is on). Default number of leak pages is 20.
memaper[=<order>]	Allocate an own aperture over RAM with size 32MB<<order>. (default: order=1, i.e. 64MB)
merge	Do scatter-gather (SG) merging. Implies "force" (experimental).
nomerge	Don't do scatter-gather (SG) merging.
noaperture	Ask the IOMMU not to touch the aperture for AGP.
forcesac	Force single-address cycle (SAC) mode for masks <40bits (experimental).
noagp	Don't initialize the AGP driver and use full aperture.
allowdac	Allow double-address cycle (DAC) mode, i.e. DMA >4GB. DAC is used with 32-bit PCI to push a 64-bit address in two cycles. When off all DMA over >4GB is forced through an IOMMU or software bounce buffering.
nodac	Forbid DAC mode, i.e. DMA >4GB.
panic	Always panic when IOMMU overflows.
calgary	Use the Calgary IOMMU if it is available

### iommu options only relevant to the software bounce buffering (SWIOTLB) IOMMU implementation:

swiotlb=<pages>[,force]	
<pages>	Prereserve that many 128K pages for the software IO bounce buffering.
force	Force all IO through the software TLB.

### Settings for the IBM Calgary hardware IOMMU currently found in IBM pSeries and xSeries machines:

```
calgary=[64k,128k,256k,512k,1M,2M,4M,8M]  
calgary=[translate_empty_slots]  
calgary=[disable=<PCI bus number>]  
panic      Always panic when IOMMU overflows
```

64k,...,8M - Set the size of each PCI slot's translation table when using the Calgary IOMMU. This is the size of the translation

#### boot-options.txt

table itself in main memory. The smallest table, 64k, covers an IO space of 32MB; the largest, 8MB table, can cover an IO space of 4GB. Normally the kernel will make the right choice by itself.

`translate_empty_slots` - Enable translation even on slots that have no devices attached to them, in case a device will be hotplugged in the future.

`disable=<PCI bus number>` - Disable translation on a given PHB. For example, the built-in graphics adapter resides on the first bridge (PCI bus number 0); if translation (isolation) is enabled on this bridge, X servers that access the hardware directly from user space might stop working. Use this option if you have devices that are accessed from userspace directly on some PCI host bridge.

#### Debugging

`oops=panic` Always panic on oopses. Default is to just kill the process, but there is a small probability of deadlocking the machine. This will also cause panics on machine check exceptions. Useful together with `panic=30` to trigger a reboot.

`kstack=N` Print N words from the kernel stack in oops dumps.

`pagefaulttrace` Dump all page faults. Only useful for extreme debugging and will create a lot of output.

`call_trace=[old|both|newfallback|new]`  
old: use old inexact backtracer  
new: use new exact dwarf2 unwinder  
both: print entries from both  
newfallback: use new unwinder but fall back to old if it gets stuck (default)

#### Miscellaneous

`nogbpages` Do not use GB pages for kernel direct mappings.

`gbpages` Use GB pages for kernel direct mappings.