# Xen Configuration File Options

Version: 1.0
Author: Stephen Spector (stephen.spector@xen.org) + Community Support on xen-users mailing list
Date: June 16, 2009

To Do:

1) Add common usage settings for standard configurations that are used

**SUMMARY**

This file contains a complete summary of all the configuration options available in open source Xen. I am using the Xen 3.4 source tree so some of these options may not be available in previous versions. The complete list of options is viewable in the python source file **create.py** in *xen/tools/python/xen/xm/*.

The file contains two types of configuration settings: options and variables. The options are listed below with a ** before them and variables are listed below in bold/italics.

**OPTIONS**

**Help**

- ** **help:** Print this help.
  *help* or *h* (default = 0)

- ** **help_config**: Print the available configuration variables (vars) for the "configuration script
  *help_config* (default = 0)

**Misc**

- ** **quiet:**Quiet.
  *quiet* or *q* (default =0)

- ** **path:** Search path for configuration scripts
  *path* (default='.:/etc/xen')

- ** **defconfig**: Use the given Python configuration script
  *defconfig* or *f* (default='xmdefconfig')

- ** **config:** Domain configuration to use (SXP)
  *config* or *F* (default=None)

- ** **dryrun:** Dry run – prints the configuration in SXP but does not create the domain
  *dryrun* or *n* (defaut = 0)

- ** **xmldryrun:** Dry run – prints the configuration in XML but does not create the domain
  *xmldryrun* or x (defaut = 0)

- ** **skipdtd**: Skip DTD checking - skips checks on XML before creating.
  *skipdtd* or *s* (default = 0)

- ** **paused:** Leave the domain paused after it is created
  *paused* or *p* (default=0)

- ** **vncviewer**: Connect to the VNC display after the domain is created
  *vncviewer (*default = 0)

- ** **vncviewer-autopass**: Pass VNC password to viewer via stdin and -autopass
  *vncviewer-autopass (*default = 0)

- ** **console_autoconnect:**Connect to the console after the domain is created.
  *console_autoconnect* or *c* (domain=0)

## VARIABLES

### Kernel + Memory Size

- **kernel**: Path to the kernel image
  *kernel* (Default=' '; Value='FILE')

- **loader:** Path to HVM firmware
  *loader* (Default=' ' ;Value='FILE')

- **features:** Features to enable in guest kernel
  *features* (Default=' ';Value='FEAUTRES')

- **ramdisk:** Path to ramdisk image (optional)
  *ramdisk="/data/guest1/initrd.img"*

- **builder**: Function to use to build the domain
  *builder* (Default='linux'; Value='FUNCTION')

- **memory:** Domain memory in MB
  *memory* (Default=128; Value='MEMORY')

- **maxmem:** Maximum domain memory in MB
  *maxmem* (Default=None; Value='MEMORY')

- **boot:** Default Boot Device
  *boot* (Default='c';Value='a|b|c|d')

- **shadow_memory:** Domain shadow memory in MB
  *shadow_memory* (Default=0; Value='MEMORY)

- **bootloader:** Path to bootloader
  *bootloader* (default=None; Value='File')

- **bootargs:** Arguments to pass to boot loader
  *bootargs* (default=None; Value='Name')

- **bootentry:** DEPRECATED. Entry to boot via boot loader. Use bootargs.
  bootentry (default=None; Value='Name')

- **s3integrity:** Should domain memory integrity be verified during S3? (0=protection is disabled; 1=protection is enabled.
  *s3integrity* (Default=1; Value='TBOOT_MEMORY_PROTECT')

- **machine_address_size**: Maximum machine address size
  *machine_address_size* (Default=None;Value='BITS')

- **suppress_spurious_page_faults**: Do not inject spurious page faults into this guest
  *suppress_spurious_page_faults* (Default=None;Value='yes|no')

**CPU**

- **cpu:** CPU to run the VCPU0 on
  *cpu* (default=None; Value='CPU')

- **cpus:** CPUS to run the domain on
  *cpus* (default=None; Value='CPUS')

- **cpu_cap**: Set the maximum amount of cpu. CAP is a percentage that fixes the maximum amount of cpu
  *cpu_cap* (default=None; Value='CAP')

- **cpu_weight**: Set the cpu time ratio to be allocated to the domain
  *cpu_weight* (default=None; Value='WEIGHT')
  ○

- **vcpus:** # of Virtual CPUS in domain
  vc*pus* (default=1; Value='VCPUS')

- **vcpus_avail:** Bitmask for virtual CPUs to make available immediately
  *vcpus_avail* (default=None; Value='VCPUS')

- **cpuid:** Cpuid Description
  *cpuid* (Default=[];Value="IN[,SIN]:eax=EAX,ebx=EBX,ecx=EXC,edx=EDX")

- **cpuid_check:** Cpuid check Description
  *cpuid_check* (Default=[];Value="IN[,SIN]:eax=EAX,ebx=EBX,ecx=EXC,edx=EDX")

## Networking

- **hostname:** Set the kernel IP hostname
  *hostname* (Default=";Value=”NAME”)

- **ip:** Set the kernel IP interface address.
  *ip* (Default=' ' ; Value='IPADDR' )

- **interface:** Set the kernel IP interface name.
  i*nterface* (Default='”eth0”;Value=”INTF”)

- **dhcp**: Set the kernel dhcp option
  *dhcp* (Default='off'; Values=”off|dhcp”)

- **vif:** Add a network interface with the given MAC address and bridge. The vif is configured by calling the given configuration script. If type is not specified, default is netfront. If mac is not specified a random MAC address is used. If not specified then the network backend chooses it's own MAC address. If bridge is not specified the first bridge found is used. If script is not specified the default script is used. If backend is not specified the default backend driver domain is used. If vifname is not specified the backend virtual interface will have name vifD.N where D is the domain id and N is the interface id. If rate is not specified the default rate is used. If model is not specified the default model is used. If accel is not specified an accelerator plugin module is not used. This option may be repeated to add more than one vif. Specifying vifs will increase the number of interfaces as needed.
  *vif* (Default=[]; Value="type= TYPE, mac=MAC, bridge=BRIDGE, ip=IPADDR, script=
  SCRIPT," + \ "backend=DOM, vifname=NAME, rate=RATE,
  model=MODEL, accel=ACCEL" )

- **vtpm:** Add a TPM interface. On the backend side use the given instance as virtual TPM instance. The given number is merely the preferred instance number. The hotplug script will determine which instance number will actually be assigned to the domain. The associtation between virtual machine and the TPM instance number can be found in /etc/xen/vtpm.db. Use the backend in the given domain. The type parameter can be used to select a specific driver type that the VM can use. To prevent a fully virtualized domain (HVM) from being able to access an emulated device model, you may specify 'paravirtualized' here.
  *vtpm* (Default= [] ; Value= "instance=INSTANCE,backend=DOM,type=TYPE")

- **netmask**: Set the kernel IP netmask
  *netmask* (Default=' ' ; Value='MASK' )

- **gateway**: Set the kernel IP gateway.
  g*ateway* (Default=' ' ; Value=”IPADDR” )

- **nfs_server**: Set the address of the NFS server for NFS root.
  *nfs_server* (Default=None;Value=”IPADDR”)

- **nfs_root:** Set the path of the root NFS directory.
  *nfs_root* (Default=None;Value=”PATH”)

- **device_model:** Path to device model program.
  *device_model* (Default=None;Value='FILE')

- **uuid**: xenstore UUID (universally unique identifier) to use. One will be randomly generated if this option is not set, just like MAC addresses for virtual network interfaces. This must be a unique value across the entire cluster.
  uuid (Default=None;Value='')

- **ioports:** Add a legacy I/O range to a domain, using given params (in hex). For example 'ioports=02f8-02ff'. The option may be repeated to add more than one i/o range
  *ioports* (Default= [] ; Value= 'FROM[-TO]')

**PCI**

- **pci:**Add a PCI device to a domain, using given params (in hex). For example 'pci=c0:02.1'. If VSLOT is supplied the device will be inserted into that virtual slot in the guest, else a free slot is selected. If msitranslate is set, MSI-INTx translation is enabled if possible. Guest that doesn't support MSI will get IO-APIC type IRQs translated from physical MSI, HVM only. Default is 1. The option may be repeated to add more than one pci device. If power_mgmt is set, the guest OS will be able to program the power states D0-D3hot of the device, HVM only. Default=0.
  *pci* (Default=[]; Value=BUS:DEV.FUNC[@VSLOT][,msitranslate=0|1][,power_mgmt=0|1]

- **vscsi:** Add a SCSI device to a domain. The physical device is PDEV, which is exported to the domain as VDEV(X:X:X:X)
  *vscsi* (Default= [];Value= 'PDEV,VDEV[,DOM]')

- **pci_msitranslate**: Global PCI MSI-INTx translation flag (0=disable;1=enable)
  *pci_msitranslate* (Default=1; Value='TRANSLATE')

- **pci_power_mgmt:** Global PCI Power Management flag (0=disable; 1=enable)
  *pci_power_mgmt* (Default=0; Value='POWERMGT')

- **xen_platform_pci**: Is xen_platform_used?
  xen_platform_pci (Default=1; Value='0|1')

- **serial:**Path to serial or pty or vc
  *serial* (Default=' ';Value='FILE')

- **keymap**: Set keyboard layout used
  *keymap* (Default=' '; Value='FILE')

- **usb:** Emulate USB devices
  *usb* (Default=0; Value='no|yes')

- **usbdevice:** Name of USB device

*usbdevice* (Default=' '; Value='NAME')

## HVM

- **viridian:**Expose Viridian interface to x86 HVM guest?
  *viridian* (default=0; Value='VIRIDIAN')

- **pae:** Disable or enable PAE of HVM domain
  *pae* (default=1; Value='PAE')

- **acpi:** Disable or enable ACPI of HVM domain.
  *acpi* (default=1; Value='ACPI')

- **apic:** Disable or enable APIC mode
  apic (default=1; Value='APIC')

## Timers

- **rtc_timeoffset:** Set RTC offset
  *rtc_timeoffset* (default=0; Value='RTC_TIMEOFFSET')

- **timer_mode:** Timer mode (0=delay virtual time when ticks are missed; 1=virtual time is always wallclock time
  *timer_mode* (default=1; Value='TIMER_MODE')

- **localtime:** Is RTC set to localtime?
  Localtime (Default=0; Value='no|yes')

- **vpt_align:** Enable aligning all periodic vpt to reduce timer interrupts
  *vpt_align* (default=1; Value='VPT_ALIGN')

- **vhpt:** Log2 of domain VHPT size for IA64
  *vhpt* (default=0; Value='VHPT')

- **hpet:** Enable virtual high-precision event timer
  *hpet* (default=0; Value='HPET')

- **hap:**Hap status (0=hap is disabled;1=hap is enabled.)
  *hap* (Default=1; Value='HAP')

## Drivers

- **irq:** Add an IRQ (interrupt line) to a domain. For example 'irq=7'. This option may be repeated to add more than one IRQ
  *irq* (Default = []; Value = 'IRQ')

- **blkif:** Make the domain a block device backend.
  *blkif* (Default=0; Value='no|yes')

- **netif:** Make the domain a network interface backend
  *netif* (Default=0; Value='no|yes')

- **tmpif:** Make the domain a TPM interface backend
  *tmpif* (Default=0; Value='no|yes')

- **vfb:**Make the domain a framebuffer backend. Both sdl=1 and vnc=1 can be enabled at the same time.For vnc=1, connect an external vncviewer. The server will listen on ADDR (default 127.0.0.1) on port N+5900. N defaults to the domain id. If vncunused=1, the server will try to find an arbitrary unused port above 5900. vncpasswd overrides the XenD configured default password. For sdl=1, a viewer will be started automatically using the given DISPLAY and XAUTHORITY, which default to the current user's ones. OpenGL will be used by default unless opengl is set to 0. keymap overrides the XendD configured default layout file
  *vfb* (Default=[];Value= "vnc=1,sdl=1, vncunused=1, vncdisplay=N, vnclisten=ADDR,
  display=DISPLAY, xauthority=XAUTHORITY,vncpasswd=PASSWORD,
  opengl=1, keymap=FILE")

## Disk Devices

- **root:** Set the root= parameter on the kernel command line.
  Use a device, e.g. /dev/sda1, or /dev/nfs for NFS root
  *root* (Default =' '; Value='DEVICE')

- **disk**: Add a disk device to a domain. The physical device is DEV, which is exported to the domain as VDEV. The disk is read-only if MODE is 'r', read-write if MODE is 'w'. If DOM is specified it defines the backend driver domain to use for the disk. The option may be repeated to add more than one disk
  *disk* (default=[] ; Value='phy:DEV,VDEV,MODE[,DOM]')

- **access_control:**Add a security label and the security policy reference that defines it.
  The local ssid reference is calculated when starting/resuming the domain. At this time, the policy is checked against the active policy as well. This way, migrating through save/restore is covered and local labels are automatically created correctly on the system where a domain is started / resumed.
  *access_control* (Default= [] ; Value=”policy=POLICY,label=LABEL")

## Behavior

- **on_poweroff:** Behavior when a domain exits with reason 'poweroff
  *on_poweroff* (Default=None; Value='destroy|restart|preserve|rename-restart')

- **on_reboot:** Behavior when a domain exits with reason 'reboot'
  *on_reboot* (Default=None; Value='destroy|restart|preserve|rename-restart')

- **on_crash:** Behavior when a domain exits with reason 'crash'
  *on_crash* (Default=None; Value='destroy|restart|preserve|rename-restart|coredump-destroy|coredump-restart'')

- **on_xend_start:** Action to preform when xend starts
  *on_xend_start* (Default='ignore'; Value='ignore|start')

- **on_xend_stop:** Behaviour when Xend stops:
  - ignore: Domain continues to run;
  - shutdown: Domain is shutdown;
  - suspend: Domain is suspended;
  *on_xend_stop* (Default='ignore'; Value='ignore|shutdown|suspend')

- **target**: Set domain target
  *target* (Default=0; Value='TARGET')

## Graphics and Audio

- **console**: Port to export the domain console on
  *console = /dev/console*

- **nographic**: Should device models use graphics
  *nographic* (Default=0;Value='no|yes')

- **soundhw:** Should device models enable audio device
  *soundhw* (Default=' ';Value='audiodev')

- **sdl:**Should the device model use SDL?
  s*dl* (Default=None;Value='')

- **opengl:** Enable\Disable OpenGL
  *opengl* (Default=None; Value=' ')

- **vnc:** Should the device model use VNC?
  v*nc* (Default=None;Value=' ')

- **vncunused:** Try to find an unused port for the VNC server. Only valid when vnc=1
  *vncdunused* (Default=1;Value=' ')

- **videoram:** Maximum amount of videoram a guest can allocate for frame buffer
  *videoram* (Default=4; Value='MEMORY')

- **vncdisplay:** VNC Display to use
  *vncdisplay* (Default=None;Value=' ')

- **vnclisten:** Address for VNC server to listen on *vnclisten* (Default=None; Value=' ')

- **vncpasswd:** Password for VNC console on HVM domain
  *vncpasswd='xxxxx'* (default=None)

- **vncviewer**: Spawn a vncviewer listening for a vnc server in the domain. The address of the vncviewer is passed to the domain on the kernel command line using 'VNC_SERVER=<host>:<port>'. The port used by vnc is 5500 + DISPLAY. A display value with a free port is chosen if possible.\nOnly valid when vnc=1.\nDEPRECATED
  *vncviewer (*default = None; Value = 'no|yes')

- **vncconsole:** Spawn a vncviewer process for the domain's graphical console. Only valid when vnc=1
  *vncconsole* (default=None; Value = 'no|yes')

- **stdvga**: Use std vga or Cirrhus Logic Graphics
  *stdvga* (Default=0; Value='no|yes')

- **isa:** Simulate an ISA only system
  *isa* (Default=0; Value='no|yes')

- **guest_os_type:** Guest OS type running in HVM
  *guest_os_type* (Default='default'; Value='NAME')

- **extra:** Set extra arguments to append to the kernel command line
  *extra* (Default=' ' ; Value = "ARGS" )

- **fda**: Path to fda
  *fda* (Default=' ';Value=FILE)

- **fdd**: Path to fdb
  *fdb* (Default=' ';Value=FILE)

- **display**: X11 display to use
  *display* (Default=None;Value='DISPLAY')

- **xauthority**: X11 authority to use
  *display* (Default=None;Value='XAUTHORITY')