

# open



USE



IMPROVE



EVANGELIZE

## xVM HVM domU PV Disk Driver Architecture

Edward Pilatowicz <[edward.pilatowicz@sun.com](mailto:edward.pilatowicz@sun.com)>  
Staff Engineer, Solaris Kernel Group

開  
放  
的  
열린  
مفتوح  
libre  
मुक्त  
ಮುಕ್ತ  
livre  
libero  
ముక్త  
开放的  
açık  
open  
nyílt  
•••••  
πικρό  
オープン  
livre  
ανοικτό  
offen  
otevřený  
öppen  
открытый  
வெளிப்படை



# HVM domU PV Disk IO Background 1/2

- HVM PV disk driver support approach is fundamentally different from HVM PV network support. Don't confuse the two.
- HVM PV networking support involves:
  - Hiding emulated hardware networking devices.
  - Exposing the native PV networking devices.
- HVM PV disk support involves:
  - Hiding the native PV disk devices.
  - Interposing on the emulated hardware disk devices and redirecting accesses to the PV disk devices.



## HVM domU PV Disk IO Background 2/2

- How do we interpose on disk accesses when running in an HVM environment?
  - Prepend /platform/i86hvm to the kernel module search path.
  - Deliver all the necessary HVM PV support modules in this new path.
  - Supply “replacement” PV aware disk drivers in this new path. These “replacement” drivers have the same name as the normal disk drivers they are replacing. ie, “sd” and “cmdk”.



# HVM domU Disk IO - without PV drivers

IDE disk path example:

```
/dev/dsk/c0d0?? → /devices/pci@0,0/pci-ide@1,1/ide@0/cmdk@0,0
```

IDE cdrom path example:

```
/dev/dsk/c1t0d0?? → /devices/pci@0,0/pci-ide@1,1/ide@1/sd@0,0
```

Kernel Module Search Path:

```
/platform/i86pc/kernel
```

```
/kernel
```

```
/usr/kernel
```

Modules Loaded:

```
/kernel/drv/{cmdk|sd|ata}
```

```
/kernel/misc/cmlb
```



# HVM domU Disk IO - with PV drivers

IDE disk path example:

```
/dev/dsk/c0d0?? → /devices/pci@0,0/pci-ide@1,1/ide@0/cmdk@0,0
```

IDE cdrom path example:

```
/dev/dsk/clt0d0?? → /devices/pci@0,0/pci-ide@1,1/ide@1/sd@0,0
```

PV disk and cdrom path example:

```
No /dev paths → /devices/xpvd/xdf@????
```

Kernel Module Search Path:

```
/platform/i86hvm/kernel
```

```
/platform/i86pc/kernel
```

```
/kernel
```

```
/usr/kernel
```

Kernel Modules Loaded:

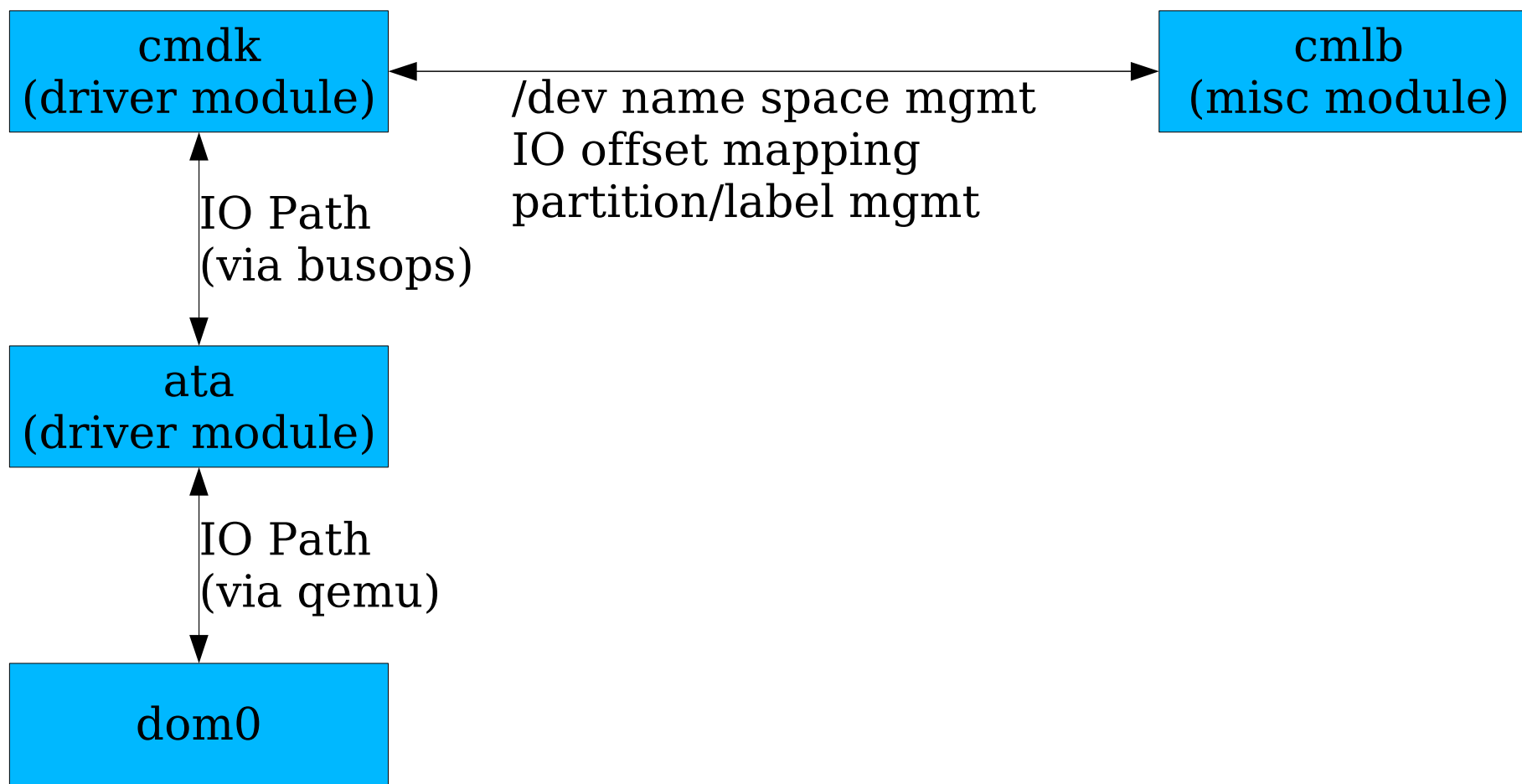
```
/platform/i86hvm/kernel/drv/{cmdk|sd|xdf}
```

```
/platform/i86hvm/kernel/misc/{hvm_cmdk|hvm_sd|hvm_bootstrap}
```

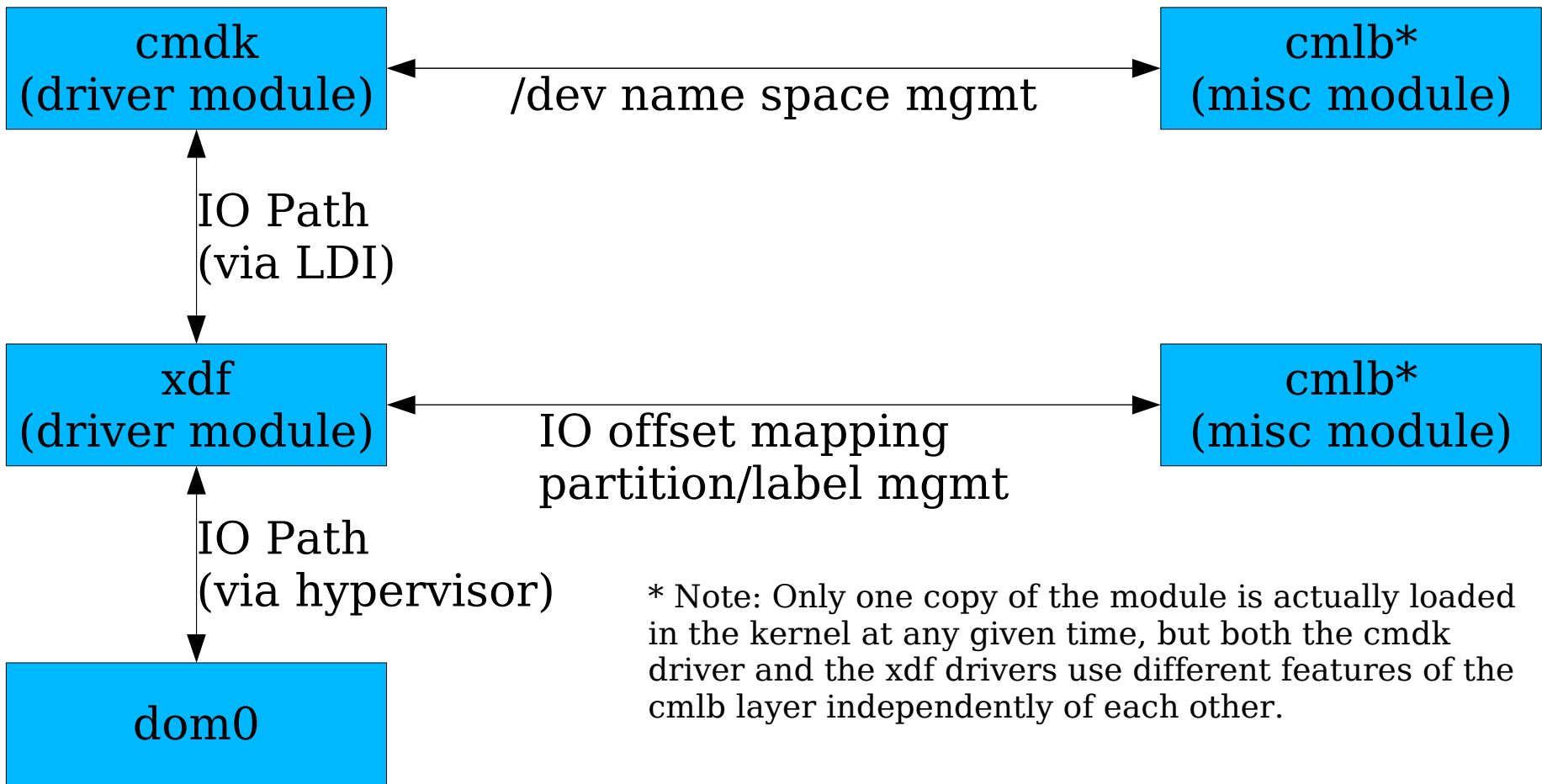
```
/kernel/drv/ata
```

```
/kernel/misc/cmlb
```

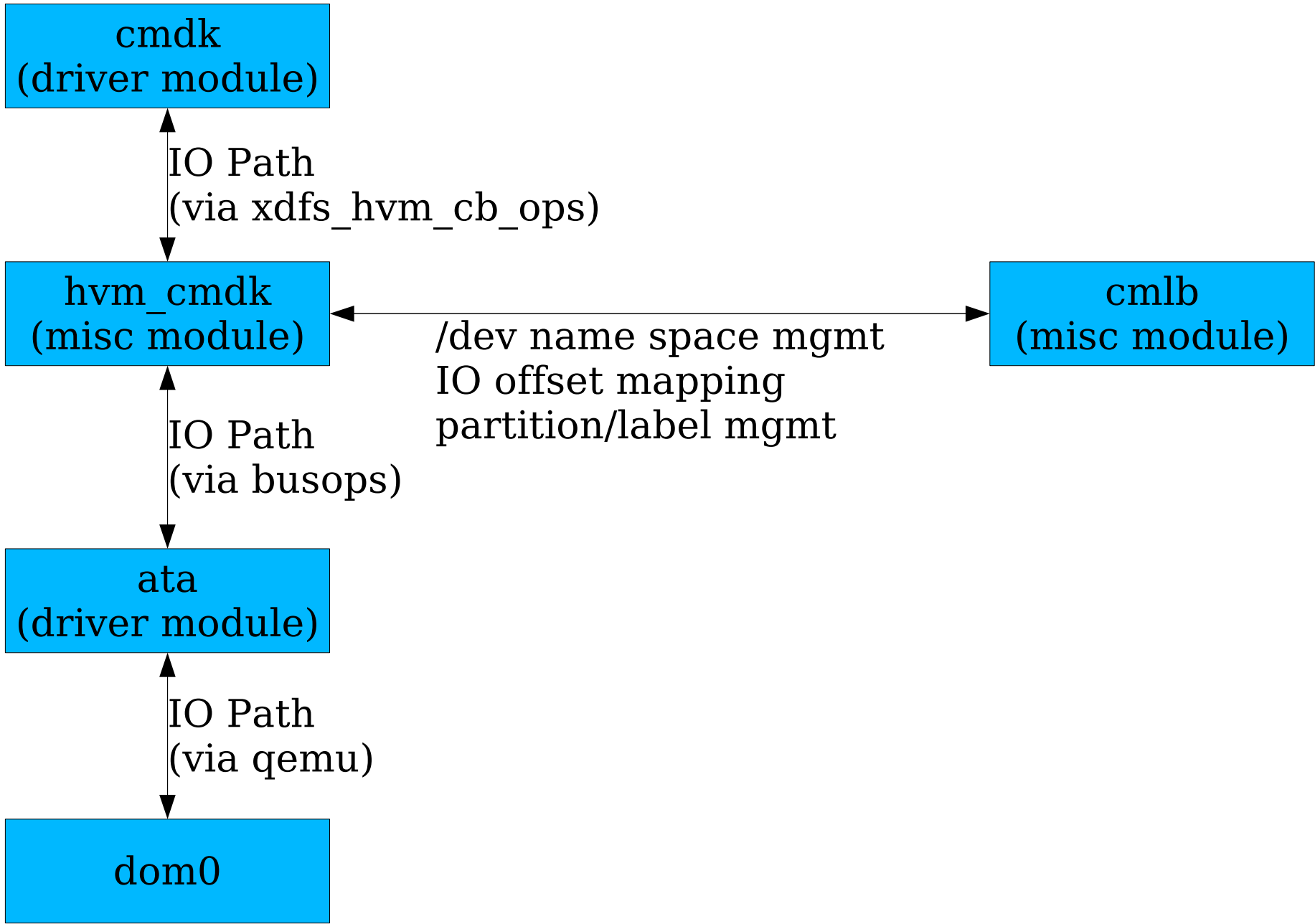
# HVM domU Disk IO - no PV drivers



# HVM domU Disk IO - PV drivers enabled



# HVM domU Disk IO - PV drivers fallback





# Kernel modules to source mappings

/kernel/drv/sd

usr/src/uts/common/io/scsi/targets/sd.c

usr/src/uts/common/io/scsi/targets/sd\_xbuf.c

/kernel/drv/cmdk

usr/src/uts/intel/io/dktp/disk/cmdk.c

/platform/i86hvm/kernel/drv/sd

usr/src/uts/i86pc/i86hvm/io/xdf\_shell.c

usr/src/uts/i86pc/i86hvm/io/pv\_sd.c

/platform/i86hvm/kernel/drv/cmdk

usr/src/uts/i86pc/i86hvm/io/xdf\_shell.c

usr/src/uts/i86pc/i86hvm/io/pv\_cmdk.c

/platform/i86hvm/kernel/misc/hvm\_sd

Same source files as /kernel/drv/sd

/platform/i86hvm/kernel/misc/hvm\_cmdk

Same source files as /kernel/drv/cmdk

/platform/i86hvm/kernel/misc/hvm\_bootstrap

usr/src/uts/i86pc/i86hvm/io/hvm\_bootstrap.c



## Source code notes - 1/4

- `xdf_shell.c`
  - Common code for creating a replacement disk driver that redirects accesses to the xdf driver. (The xdf driver is the native PV disk driver.)
  - One copy of this file is compiled into each replacement disk driver.
  - If PV access to a disk device is not possible, this code falls back to using HVM access. see `xdfs_probe()`.
  - If you want to understand this code, **READ THE COMMENTS AT THE TOP OF THE FILE! :)**



## Source code notes - 2/4

- `pv_cmdk.c` & `pv_sd.c`
  - All non-static interfaces in these files are directly consumed by `xdf_shell.c`. (See the comments in `xdf_shell.h` for these interface definitions.)
  - HVM disk device path to PV disk device path mappings are hardcoded here. If future emulated hardware platforms have different disk device paths, mappings for those new platforms will have to be added here. (see `xdfs_c_h2p_map` for more details.)



## Source code notes - 3/4

- `pv_cmdk.c`
  - IDE PV disk redirection driver specific code.
- `pv_sd.c`
  - SCSI PV disk redirection driver specific code.
  - Currently only support IDE cdrom devices.
  - Doesn't support uSCSI operations.
- `hvm_bootstrap.c`hvmboot_rootconf()`
  - Called before root is mounted to force attachment of all native PV disk devices.



## Source code notes - 4/4

- `XPV_HVM_DRIVER`
  - A Define used for compiling HVM PV specific support code. It's used extensively within xVM specific code. It's also used in common driver code to compile HVM specific versions of code. (It's used in `sd.c` when generating `hvm_sd` and in `cmdk.c` when generating `hvm_cmdk`.)



## PV CD-ROM Eject Support - 1/3

- Native PV domUs have no concept of “ejecting” cdroms. To “change” PV cdrom media you detach the existing PV device and attach a new PV device with the new image. This is the equivalent of using DR (Dynamic Reconfiguration) to do a cdrom media change.
- The DR approach to media changes doesn't work in HVM domains because qemu emulates cdroms as IDE devices, and IDE devices don't support DR.



## PV CD-ROM Eject Support - 2/3

- To support cdrom media changes in HVM domains, Xen has a “block-configure” operation. That's good. But here's the bad news:
  - Block-configure is not supported for PV domains.
  - Linux (the official xen reference implementation) doesn't support HVM PV cdrom access.
  - This means there are no existing interfaces for managing media operations on PV cdrom devices.
- Sigh. So to support PV cdrom devices we have created our own media requests management interfaces.



## PV CD-ROM Eject Support - 3/3

- Media request operations requires coordination between dom0 and domU and are done via the xenbus.
- Media request support is required for PV cdrom access. If a dom0 doesn't support media request operations, the domU will revert to emulated hardware cdrom access.
- Two media operations are currently supported: Eject and Lock
- See the `XB?_MEDIA_*` defines and comments in `xendev.h` for more details.

# open



USE



IMPROVE



EVANGELIZE

## Thank you!

Edward Pilatowicz <[edward.pilatowicz@sun.com](mailto:edward.pilatowicz@sun.com)>  
Staff Engineer, Solaris Kernel Group  
<http://blogs.sun.com/edp>

“open” artwork and icons by chandan:  
<http://blogs.sun.com/chandan>

開  
放  
的  
열린  
مفتوح  
libre  
मुक्त  
ಮುಕ್ತ  
livre  
libero  
ముక్త  
开放的  
açık  
open  
nyílt  
•••••  
πικρ  
オープン  
livre  
ανοικτό  
offen  
otevřený  
öppen  
открытый  
வெளிப்படை