The Next Generation Mass Storage Stack

HP-UX 11i v3

Abstract

This white paper discusses the next generation mass storage subsystem in HP-UX 11i v3. It is intended for system administrators or operators who have experience with HP-UX and manage mass storage devices such as disks and tape drives.

Publication History

- February 2007: Initial Publication
- September 2007: Addition of load balancing policies closest_path and pref_tport, coincident with the September 2007 release of HP-UX 11i v3, Ignite-UX inventory blocking, and corrected VxFS file system size
- April 2008: Addition of weighted_rr load balancing policy, coincident with the March 2008 release of HP-UX 11i v3, `ioinit` command changes
- April 2009: Addition of `mediainit` command and `ioinit` command changes

Background

HP-UX 11i v3 introduces a new representation of mass storage devices called the **agile view**. The central idea of the agile view is that disk devices and tape drives are identified the actual object, not by a hardware path to the object. Paths to a device can change dynamically, and multiple paths to a single device can be transparently treated as a single virtualized path, with I/O distributed across those multiple paths. This representation increases the reliability, adaptability, performance, and scalability of the mass storage stack, all without the need for operator intervention.

This document addresses the following topics:

Features of the Next Generation Mass Storage Stack
Describes new mass storage functionality in HP-UX 11i v3. Major new features include scalability, agile addressing, multi-pathing and load balancing, adaptability, and performance. Each section includes background on previous supported features, reasons for the changes, and a summary of user-visible changes.

Introduction to the Agile View
Describes the new mass storage user interfaces. This is a different take on the mass storage stack features, and includes the new type of device special file (**DSF**) and new hardware addressing model for mass storage.

Changes to I/O and Mass Storage Subsystems
Describes the effects on I/O commands. This includes new options for existing commands, new commands, and changes to tunable parameters.

Changes to Other Subsystems
Describes the effects on HP-UX subsystems. This is a high-level description of the changes to subsystems outside the I/O subsystem, such as LVM.

Migrating to the Agile View
Provides an overview of how to migrate applications, the kernel, and the system from the legacy view to the agile view.

The paper concludes with a roadmap to other resources on the agile view and related changes.

In addition, there are four appendixes:

Summary of Changes
A short summary of the user-visible changes to commands, device file names, and tunables.

Using ioscan
A sample output of the `ioscan` command using several of the new options.

Using scsimgr
The most common examples of setting device tunables.

Interpreting lunpath hardware paths
How to interpret the address elements for the new lunpath hardware paths.

Features of the Next Generation Mass Storage Stack

This section describes the new features of the next generation mass storage stack and the commands that changed due to those features. For benefits associated with these features, refer to Benefits of Migration.

Scalability

The next generation mass storage stack increases the server mass storage capacities in the following areas:

Number of I/O busses
The number of I/O busses on HP-UX 11i v1 and v2 is limited to 256 bus instances. This limit has been removed for HP-UX 11i v3.

Note: Persistent device special files (DSFs), described in Agile Addressing, must be used to access any bus with an instance greater than 255.

Number of Logical Units (LUNs) supported
In previous releases, the supported number of LUNs was based on the number of active LUN paths per server. On HP-UX 11i v3, 16384 LUNs are supported.

LUN Size The I/O system supports LUNs greater than two TB in size.

Number of distinct I/O paths to a LUN
Previously, a LUN could have up to eight physical I/O paths. For HP-UX 11i v3, this increases to 32 I/O paths per LUN, up to a maximum of 65536 LUN paths per server.

File system size File systems can be as large as eight EB (8192 TB). This limit is constrained by any volume manager and file system type size restrictions; for example, LVM supports a maximum volume size of 16 TB, so the maximum file system size under LVM is 16 TB.

Feature	HP-UX 11i v3	HP-UX 11i v2
Number of I/O busses	no limit	256
Number of LUNs supported	16384 (architectural limit of 16 M)	8192 active
LUN size	> 2 TB (subject to driver support)	2 TB
Number of I/O paths to a single LUN	32	8
File system size supported	32 TB at initial release (subject to volume manager support) (architectural limit of 8 EB)	32 TB

In addition, the mass storage stack has been enhanced to take advantage of large multi-CPU server configurations for greater parallelism. Adding more mass storage to a server does not appreciably slow down the boot process or the `ioscan` command.

Impact

The increased limits do not affect the usage of any commands or libraries. Some command output fields can appear wider due to larger numbers.

One significant change is in hardware paths and DSFs. The increased number of I/O busses and LUNs makes encoding path information in the DSF minor number impractical, so the I/O subsystem has introduced a new naming convention and minor number convention for mass storage devices. If you have any existing DSFs, they are completely backward compatible. However, you cannot use existing DSFs to address more than 256 I/O busses or 32768 LUNs. In HP-UX 11i v3, you can use new DSFs that enable the larger I/O configurations, transparent multi-pathing, and agile addressing.

For additional information on scalability and mass storage limits in HP-UX 11i v3, see the *HP-UX 11i v3 Mass Storage I/O Scalability* white paper in For more information.

Agile Addressing

The most visible change to the mass storage stack in HP-UX 11i v3 is the addition of **agile addressing**, also known as **persistent LUN binding**.

Background

Historically, DSFs for mass storage devices had their hardware path, or **lunpath**, encoded in both their name and their minor number. For example, the DSF `/dev/dsk/c3t15d0` is at SCSI controller instance 3, SCSI target 15, and SCSI LUN 0. This **legacy** style of DSF has three significant shortcomings:

- If any mass storage had multiple lunpaths, there were multiple DSFs, one for each lunpath. All disk management products had to be aware of multiple paths, with each product generating its own, often conflicting, solution.
- If the path to a disk changed, the names of its associated DSFs had to change as well, requiring applications and volume groups to be reconfigured. In SAN environments, an HBA port change, switch port change, or controller port change can trigger such hardware path changes.
- The size of the I/O configuration was limited by the format of the DSF minor number and naming convention. With reserved bits for the card instance, target, and LUN, only 256 controllers, 16 targets per controller, and 8 LUNs per target were allowed. Interface drivers that supported the larger addressing model of SCSI-3 devices had to create virtual controllers, virtual targets, and virtual LUNs.

In HP-UX 11i v3, you can create **persistent** DSFs; that is, you can create a single DSF for each unique LUN in the server, no matter how many lunpaths the LUN has or if any of those lunpaths change. With this persistent binding, the DSF name and minor number no longer have hardware path information encoded in them. Instead, they have a simple device instance number such as `/dev/disk/disk3`. Rather than mapping to the LUN hardware path, the persistent DSF maps to the LUN WorldWide Identifier (WWID), a resource on the LUN similar to a serial number. The WWID is unique to a LUN and does not change for that LUN. If any of a LUN lunpaths change, the DSF bound to the LUN does *not* change – hence, the DSF has **agile addressing**.

Impact

HP encourages the use of persistent DSFs; they are created by default on newly installed servers. Existing DSFs continue to work as before, and are retained on an update to HP-UX 11i v3. Legacy DSFs are backward compatible and are not affected by persistent DSFs on the same server. A device can be simultaneously accessed via legacy and persistent DSFs.

All commands are backward compatible, and work with either legacy or persistent DSFs, except as noted in Backward Compatibility. Some commands have new options to manage persistent DSFs and DSF migration:

`insf`	By default, creates both persistent and legacy DSFs for new devices.
`insf -L`	Restores legacy DSFs and legacy configuration information.
`rmsf -L`	Aids in migration by removing all legacy DSFs and legacy configuration information.
`ioscan -m dsf`	Maps persistent DSFs to their equivalent legacy DSFs and vice versa.
`ioscan -N`	Prints persistent DSFs when used with the `-n` option to list DSFs.
`ioinit -rC` *`class`*	Reassigns device instance numbers, starting from 0 (zero), for all existing devices in the given *class*. To create new DSFs for the devices whose instance number has changed, run `insf -e -C` *`class`*.
`ioinit -A`	Performs Critical Resource Analysis (CRA) on all of the devices corresponding to the hardware paths specified in *`infile`*, before assigning the new instance numbers. You must use the `-f` option with this option.
`io_redirect_dsf`	Associates a new disk with an existing set of DSFs. This is used when replacing an internal disk or a disk in a JBOD.

For additional information on the mass storage device naming conventions in HP-UX 11i v3, see the *HP-UX 11i v3 Mass Storage Device Naming* white paper in For more information.

Multi-Pathing and Load Balancing

Background

Agile addressing creates a single DSF for each mass storage device, regardless of the number of hardware paths to the disk. The mass storage stack in HP-UX 11i v3 uses that agility to provide transparent multi-pathing. In other words, if a LUN has multiple lunpaths, I/O requests can be transparently distributed across all available lunpaths to the LUN, using a choice of load balancing algorithms. This eliminates the need for add-on multi-pathing products.

If a lunpath fails, the mass storage stack automatically disables the failed lunpath and I/O continues on all available enabled lunpaths. Any failed or nonresponsive lunpaths are monitored, so that when a failed lunpath is recovered it is automatically and transparently reincorporated into any load balancing. In addition, any new LUNs or new lunpaths are also automatically discovered and added to load balancing.

Impact

If none of your disks are multi-pathed, there is no impact to your server.

Existing legacy DSFs automatically inherit multi-pathing. I/O requests and I/O control operations can be processed along any hardware path to a LUN, even if you use legacy DSFs. If you do not want multi-pathing of legacy DSFs, disable it using the new `scsimgr` command. For more information, see Disabling Multi-Pathing on Legacy DSFs.

For information on the interaction between native multi-pathing and third-party multi-pathing products such as SecurePath and VERITAS DMP, see the impacts section on Third-Party Multi-Pathing Products.

If you are using LVM alternate links to manage multi-pathed LVM disks, see the impacts section on Logical Volume Manager.

Command changes include the following:

`ioscan -m lun`	Displays the multiple paths to a LUN.
`ioscan -P health`	Indicates whether a lunpath is disabled.
`scsimgr`	Enables the selection of different load balancing algorithms, either for a single LUN, a set of LUNs, or all LUNs, by setting the `load_bal_policy` attribute. Also controls whether legacy DSFs allow multi-pathing, by setting the `leg_mpath_enable` attribute.

For additional information on multi-pathing and agile addressing in HP-UX 11i v3, see the *HP-UX 11i v3 Native Multi-Pathing for Mass Storage* white paper in For more information.

Adaptability

The next generation mass storage stack enhances the ability of a server to adapt dynamically to hardware changes, without shutting down the server or reconfiguring software.

Background

Asynchronous changes to a server usually require some action on your part; for example, adding a new disk requires the creation of new DSFs. Over the last several releases, HP-UX has been enhanced to automatically adapt to a changing environment, and HP-UX 11i v3 continues this trend.

A server running HP-UX 11i v3 automatically detects the creation or modification of SCSI LUNs. If new LUNs are added, persistent DSFs are automatically created. If the addressing, size, or I/O block size of an existing LUN changes, the mass storage stack detects this without user intervention.

When such changes occur, the mass storage stack notifies any relevant subsystems. For example, if a LUN expands (**dynamic LUN expansion**), its associated disk driver, volume manager, and file system are notified. The volume manager or file system can then automatically expand a volume group and file system.

The next generation mass storage stack can also remove PCI host bus adapters (HBAs) without shutting down the server. Coupled with existing online addition and replacement features, online deletion enables you to replace a PCI card with a different PCI card, as long as the HBA slot permits it and no system critical devices are affected. You can also change the driver associated with a LUN; if the software drivers do not support rebinding online, the system remembers the changes and defers them until the next server reboot.

The I/O subsystem now maintains dynamic status for mass storage and networking devices. Each device and HBA now has a health property that you can print with the `ioscan` command. The boot and crashdump subsystems monitor the status of multi-pathed devices, and automatically failover to an alternate available path if an existing path goes offline.

You can also track down stale devices, that is, devices that are configured on a server but no longer physically connected, and remove their definitions from the system.

Impact

Command changes include the following:

`iobind`	Binds a specific driver to a hardware path. This new command accepts a hardware path, a driver name, and an optional instance number to assign. The drivers involved must support online rebinding; otherwise, the rebinding can be deferred to next reboot. The `ioscan` command with the existing `-M` and `-H` options also binds a driver to a hardware path, but does not attempt to unbind any existing driver at the specified hardware path.
`ioscan -b`	Explicitly defers a binding operation until the next reboot.
`ioscan -B`	Lists any pending deferred binding operations.
`ioscan -r`	Removes any pending deferred binding operations.
`ioscan -P` *`property`*	Displays a device property. Use the following values for the `property` argument: • `health` (new in HP-UX 11i v3) • `is_inst_replaceable` (new in the HP-UX 11i v3 March 2008 release) • `error_recovery` (new in the HP-UX 11i v3 March 2008 release) • `bus_type` • `cdio` • `is_block` • `is_char` • `is_pseudo` • `b_major` • `c_major` • `minor` • `class` • `driver` • `hw_path` • `id_bytes` • `instance` • `module_name` • `sw_state` • `hw_type` • `description` All these values, except as noted, were displayed in previous releases by using the `-F` option.
`lssf -s`	Displays stale DSFs—devices that have no associated hardware.
`rmsf -x`	Removes stale entries from the I/O configuration as well as their associated DSFs.

Performance

In the performance arena, the mass storage stack takes advantage of multiple hardware paths and multiple CPUs to parallelize many operations:

- Native multi-pathing and load balancing provide better use of I/O channel bandwidth.
- More concurrent I/O operations provide a dramatic reduction of I/O scan times, both at boot time and in response to `ioscan`. Tests show scans completing up to ten times faster.
- Increased maximum I/O request size from 1 MB to 2 MB.
- Improved performance tracking tools, including more statistical information at all levels of the I/O system and enhancements to `sar` to display data about tapes, HBAs, and separate lunpaths.

Impact

The impact of multi-pathing and load balancing is described in Multi-Pathing and Load Balancing.

Several new command options provide performance information:

`scsimgr get_stat`	Displays mass storage statistics, either at a global level or for a particular LUN, target, or controller.
`glance -U`	Displays HBA port-level statistics. The motif version of `glance` (`gpm`) has a new `IO by HBA` report under the `Disk details` menu.
`sar -H`	Displays HBA port-level activity.
`sar -L`	Displays separate activity for each lunpath.
`sar -t`	Displays tape activity.
`sar -R`	Displays disk reads per second and disk writes per second as separate columns.

For additional information on I/O performance in HP-UX 11i v3, see the *HP-UX 11i v3 Mass Storage I/O Performance Improvements* white paper in For more information.

Introduction to the Agile View

This section shows how the agile view appears as the DSF names and hardware paths differ from previous releases. The DSFs, paths, and `ioscan` output for a sample system are displayed with an explanation of how they change for the next generation mass storage stack. After this comparison, additional sections provide more detailed information about the hardware paths and DSF types.

The example configuration has three HBAs: one parallel SCSI HBA and two Fibre Channel HBAs connected to a disk array. There are four disks on the example system, two on parallel SCSI and two in the disk array. One of the Fibre Channel disks is multi-pathed and is connected to both Fibre Channel cards.

Note: The configuration diagrams and `ioscan` listings have been edited for readability; entries for non-mass storage devices have been omitted.

Legacy View

On releases prior to HP-UX 11i v3, the disks on the example configuration included the hardware paths, driver names, and DSFs shown in Figure 1. This environment is referred to as the **legacy view**.

In Figure 1, the HBAs are on the left side. The disks are shown on the right side with their associated DSFs inside. Connections from the HBAs to the disks are shown as a solid line. Above the connection is the hardware path shown by the I/O commands such as `ioscan`.

Following are important concepts displayed in Figure 1:

- Parallel SCSI disks use a SCSI-2 addressing paradigm, which supports up to eight LUNs per target and 16 targets per controller. Thus, addressing beyond the HBA consists of a target port ID and a LUN ID. These are directly incorporated into the hardware path for the disk.
- Fibre Channel disks use SCSI-3 addressing, which supports a larger addressing model in which the number of target paths per controller or LUN paths per target are limited only by device, controller, or transport protocol addressing restrictions. The addressing beyond the HBA contains the worldwide port name and LUN ID. However, in the legacy view, the Fibre Channel hardware path uses the SCSI-2 addressing model by creating virtual controllers, virtual targets, and virtual LUNs.
- The DSF name for each disk contains hardware path information. The multi-pathed disk has two different DSFs, one for each hardware path. There is no indication that both DSFs refer to the same disk.

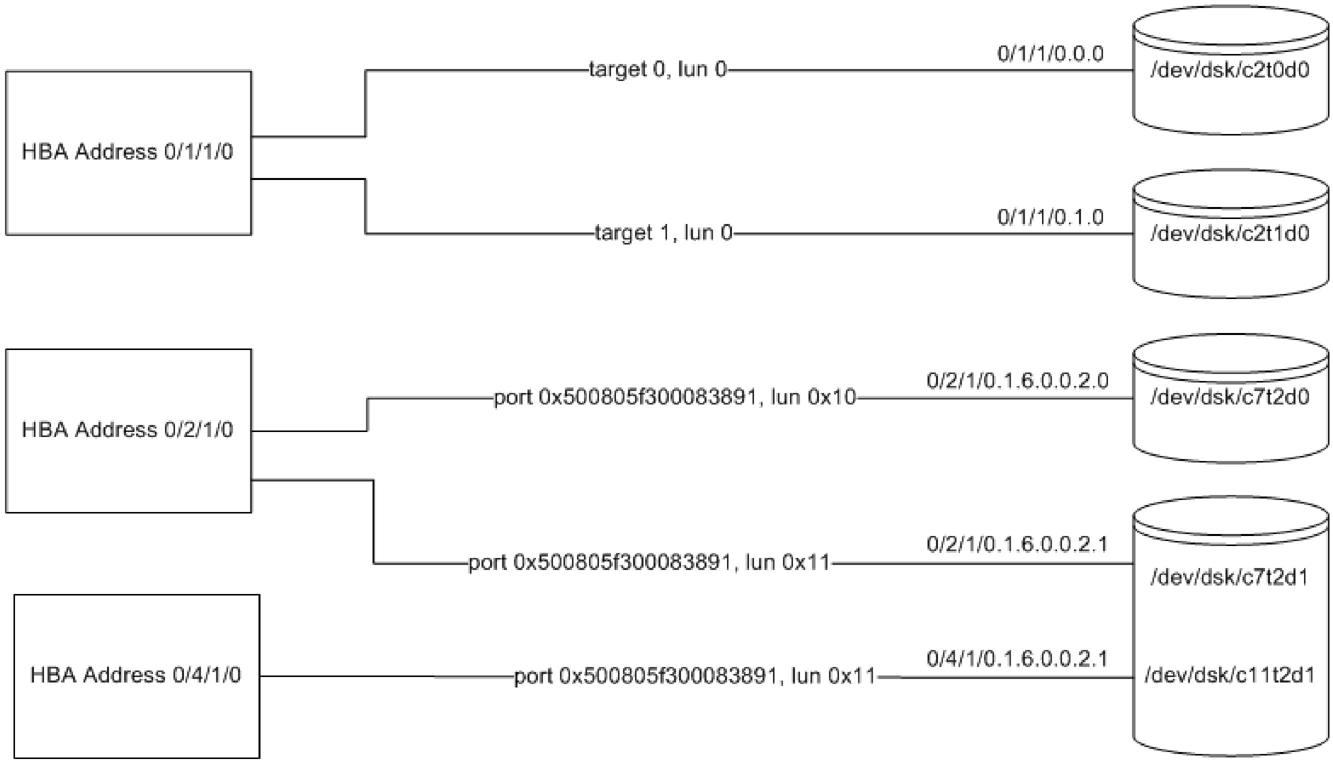

Figure 1: Legacy View

```
# ioscan -fkn
Class     I H/W Path             Driver    S/W State H/W Type  Description
=============================================================================
ext_bus   2 0/1/1/0              mpt       CLAIMED   INTERFACE SCSI Ultra320
target    0 0/1/1/0.0            tgt       CLAIMED   DEVICE
disk      1 0/1/1/0.0.0          sdisk     CLAIMED   DEVICE    HP 36.4GST336753LC
                                 /dev/dsk/c2t0d0     /dev/rdsk/c2t0d0
target    1 0/1/1/0.1            tgt       CLAIMED   DEVICE
disk      0 0/1/1/0.1.0          sdisk     CLAIMED   DEVICE    HP 36.4GST336753LC
                                 /dev/dsk/c2t1d0     /dev/rdsk/c2t1d0
fc        0 0/2/1/0              td        CLAIMED   INTERFACE Fibre Channel Adapter
fcp       0 0/2/1/0.1            fcp       CLAIMED   INTERFACE FCP Domain
ext_bus   7 0/2/1/0.1.6.0.0      fcparray  CLAIMED   INTERFACE FCP Array Interface
target    9 0/2/1/0.1.6.0.0.2    tgt       CLAIMED   DEVICE
disk     22 0/2/1/0.1.6.0.0.2.0 sdisk     CLAIMED   DEVICE    HP MSA VOLUME
                                 /dev/dsk/c7t2d0     /dev/rdsk/c7t2d0
disk     23 0/2/1/0.1.6.0.0.2.1 sdisk     CLAIMED   DEVICE    HP MSA VOLUME
                                 /dev/dsk/c7t2d1     /dev/rdsk/c7t2d1
fc        1 0/4/1/0              td        CLAIMED   INTERFACE Fibre Channel Adapter
fcp       1 0/4/1/0.1            fcp       CLAIMED   INTERFACE FCP Domain
ext_bus  11 0/4/1/0.1.6.0.0      fcparray  CLAIMED   INTERFACE FCP Array Interface
target   17 0/4/1/0.1.6.0.0.2    tgt       CLAIMED   DEVICE
disk     45 0/4/1/0.1.6.0.0.2.1 sdisk     CLAIMED   DEVICE    HP MSA VOLUME
                                 /dev/dsk/c11t2d1    /dev/rdsk/c11t2d1
```

Agile View

The **agile view** includes a new persistent type of disk and tape DSF, and represents hardware pathing to disk and tape devices in ways that support larger configurations and enable transparent multi-pathing.

The naming of DSFs and hardware paths is shown in Figure 2. Like the legacy view diagram, the HBAs are on the left, disks are on the right, with connections in between them.

Following are important concepts displayed in Figure 2:

- The hardware path elements beyond the HBA are now printed in hexadecimal notation.
- The hardware paths follow a more natural addressing model: SCSI-2 for parallel SCSI and SCSI-3 for Fibre Channel. Fibre Channel addressing is much longer than before, but is no longer forced into a SCSI-2 paradigm. The address elements, such as the target port worldwide name, correlate directly to path information displayed by the `fcmsutil` command and other disk array controller programs. If you perform SAN configuration, these hardware paths are familiar to you.
- Each disk now has a virtualized hardware path known as the **LUN hardware path**, which represents the disk itself–*not* the path to the disk. Despite having multiple hardware paths, the disk has only one LUN hardware path. This hardware path starts with a virtual root address of 64000. Addressing beyond that virtual root consists of a virtual bus address and a virtual LUN ID.
- The DSF name for each disk no longer contains path information. The multi-pathed disk has a single **persistent DSF** regardless of the number of physical paths to it.
- The `ioscan` listing includes the following changes:
    - The emulated domain, controller, target, and LUN for Fibre Channel have been removed. Entries in the `ioscan` listing beneath the HBA for both parallel SCSI and Fibre Channel include a simplified target path and lun path, shown as class `tgtpath` and `lunpath`, respectively.
    - Each physical path to a disk is now referred to as a **lunpath**, and its hardware path is called a **lunpath hardware path**. Multi-pathed disks have multiple lunpaths, and the description of the lunpath refers to its assigned disk. Since the DSF refers to the disk rather than its path, lunpaths do not have DSFs associated with them.
    - Each disk has a single `disk` class entry. The hardware path shown by `ioscan` is the LUN hardware path, the description is the same as the legacy view, and the DSFs are the persistent DSFs assigned to the disk.
    - Target paths now have a driver named `estp`. Lunpaths use the `eslpt` driver, and disks use a new `esdisk` disk driver.

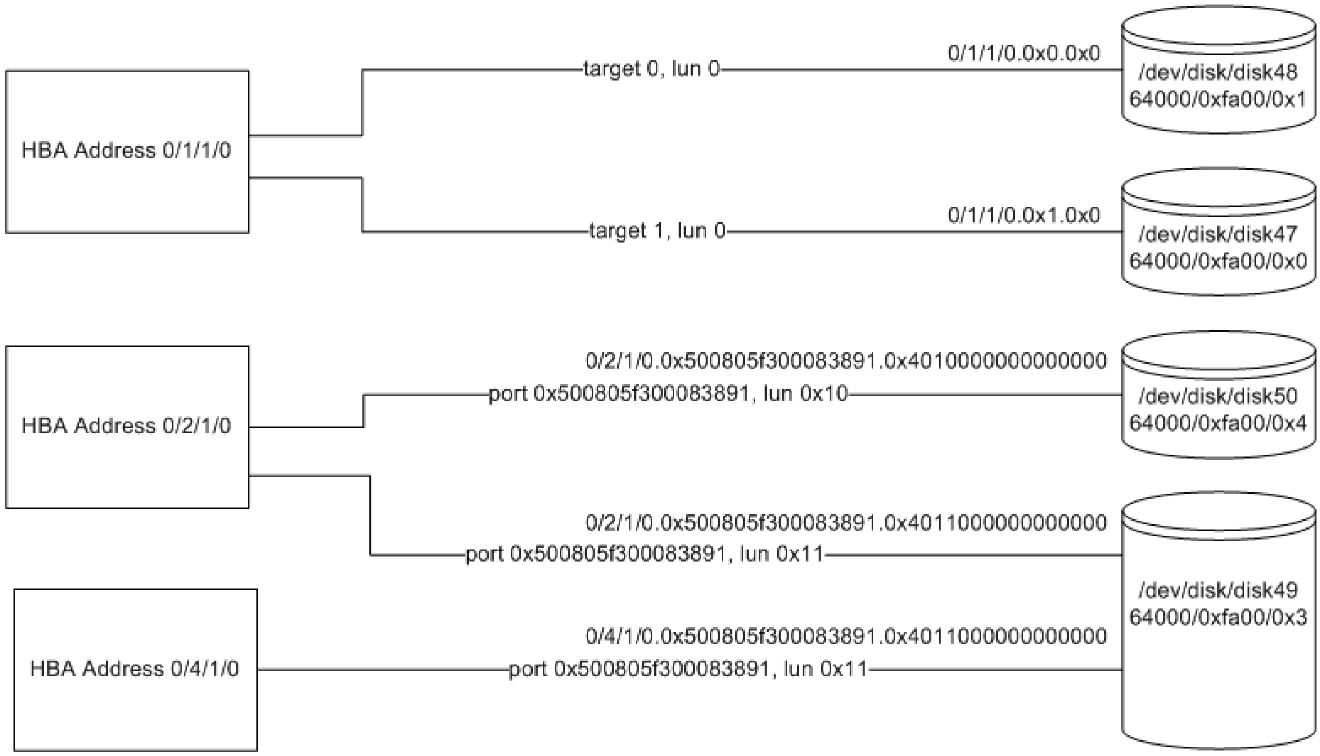

Figure 2: Agile View

```
# ioscan -N -fkn
Class     I H/W Path                   Driver S/W State H/W Type  Description
============================================================================
ext_bus   2 0/1/1/0                    mpt    CLAIMED   INTERFACE SCSI Ultra320
tgtpath   4 0/1/1/0.0x0                estp   CLAIMED   TGT_PATH  parallel_scsi target
lunpath   1 0/1/1/0.0x0.0x0            eslpt  CLAIMED   LUN_PATH  LUN path for disk48
tgtpath   3 0/1/1/0.0x1                estp   CLAIMED   TGT_PATH  parallel_scsi target
lunpath   0 0/1/1/0.0x1.0x0            eslpt  CLAIMED   LUN_PATH  LUN path for disk47
fc        0 0/2/1/0                    td     CLAIMED   INTERFACE Fibre Channel Adapter
tgtpath   6 0/2/1/0.0x500805f300083891 estp   CLAIMED   TGT_PATH  fibre_channel target
lunpath   9 0/2/1/0.0x500805f300083891.0x4010000000000000
                                       eslpt  CLAIMED   LUN_PATH  LUN path for disk50
lunpath   6 0/2/1/0.0x500805f300083891.0x4011000000000000
                                       eslpt  CLAIMED   LUN_PATH  LUN path for disk49
fc        1 0/4/1/0                    td     CLAIMED   INTERFACE Fibre Channel Adapter
tgtpath   8 0/4/1/0.0x500805f300083891 estp   CLAIMED   TGT_PATH  fibre_channel target
lunpath  30 0/4/1/0.0x500805f300083891.0x4011000000000000
                                       eslpt  CLAIMED   LUN_PATH  LUN path for disk49
disk     47 64000/0xfa00/0x0           esdisk CLAIMED   DEVICE    HP 36.4GST336753LC
                                      /dev/disk/disk47      /dev/rdisk/disk47
disk     48 64000/0xfa00/0x1           esdisk CLAIMED   DEVICE    HP 36.4GST336753LC
                                      /dev/disk/disk48      /dev/rdisk/disk48
disk     49 64000/0xfa00/0x3           esdisk CLAIMED   DEVICE    HP MSA VOLUME
                                      /dev/disk/disk49     /dev/rdisk/disk49
disk     50 64000/0xfa00/0x4           esdisk CLAIMED   DEVICE    HP MSA VOLUME
                                      /dev/disk/disk50     /dev/rdisk/disk50
```

Hardware Path Details

This section contains more information about the hardware paths shown in the Legacy View and Agile View.

In HP-UX 11i v3, there are three different types of paths to a device: legacy hardware path, lunpath hardware path, and LUN hardware path. All three are numeric strings of hardware components, with each number typically representing the location of a hardware component on the path to the device.

Legacy hardware path

This is the format used in releases prior to HP-UX 11i v3. It is displayed in the legacy view. It is composed of a series of bus-nexus addresses separated by a slash (/) leading to the HBA. Beyond the HBA, additional address elements are separated by a period (.).

For parallel SCSI devices, the addressing is a simple target and LUN as follows:

0/1/1/0.1.0

For Fibre Channel devices, legacy addressing is emulated with a domain, area, port, virtual bus, virtual target, and virtual LUN as follows:

0/2/1/0.1.6.0.0.2.1

Lunpath hardware path

This format enables the use of more targets and LUNs than are permitted under legacy hardware paths. It is printed in the agile view. Its format is identical to a legacy hardware path up to the HBA, and represents the same path to the LUN. Beyond the HBA, additional elements are printed in hexadecimal notation and separated by a period (.). The leading elements represent a transport-dependent target address. The final element is a LUN address, a 64-bit representation of the LUN identifier reported by the target.

The following lunpath hardware path uses the Fibre Channel transport:

0/2/1/0.0x50001fe1500170ac.0x4017000000000000

The following lunpath hardware path uses parallel SCSI:

0/1/1/0.0xd.0x0

For more specifics on the components of a lunpath hardware path, see Appendix D: Interpreting Lunpath Hardware Paths.

LUN hardware path

This format is a virtualized path that represents all the lunpaths to a single LUN. It is printed in the agile view. Instead of a series of bus-nexus addresses leading to the HBA, the path contains a virtual bus-nexus (called the **virtual root node**) with an address of 64000. Addressing beyond the virtual root node consists of a virtual bus address and a virtual LUN ID, delimited by slash (/) characters. An example of a LUN hardware path is as follows:

64000/0xfa00/0x22

As a virtualized path, the LUN hardware path is only a handle to the LUN and does not represent the LUN physical location. Instead it is based on the LUN WorldWide Identifier (WWID). It remains the same if new physical paths to the device are added, if existing physical paths are removed, or if a physical path changes. This LUN binding persists across reboots, but it is not guaranteed to persist across installations. Reinstalling a system or installing an identically configured system can create a different set of LUN hardware paths.

The following three formats are three different representations of the same LUN. A single LUN can have all of the following addresses.

- 0/2/1/0.1.6.0.0.2.1
  0/4/1/0.1.6.0.0.2.1
- 0/2/1/0.0x500805f300083891.0x4011000000000000
  0/4/1/0.0x500805f300083891.0x4011000000000000
- 64000/0xfa00/0x3

In the sample configuration, the LUN is multi-pathed with two hardware paths. The first two addresses represent the hardware paths in legacy format, the next two addresses represent the paths in lunpath format, and the final address represents the single LUN hardware path.

To view alternate path formats for a specific hardware path, specify it as an argument to `ioscan -H` with the `-m hwpath` option. If you do not specify a hardware path, `ioscan` displays all hardware paths for all mass storage devices. For example, to see all the hardware paths for device at legacy hardware path 0/2/1/0.1.6.0.0.2.0, enter the following command:

```
# ioscan -m hwpath -H 0/2/1/0.1.6.0.0.2.0
Lun H/W Path      Lunpath H/W Path                             Legacy H/W Path
==============================================================================
64000/0xfa00/0x4
                  0/2/1/0.0x500805f300083891.0x4010000000000000 0/2/1/0.1.6.0.0.2.0
```

To see all the hardware paths for the sample configuration, enter the following command:

```
# ioscan -m hwpath
Lun H/W Path      Lunpath H/W Path                             Legacy H/W Path
==============================================================================
64000/0xfa00/0x0
                  0/1/1/0.0x1.0x0                              0/1/1/0.1.0
64000/0xfa00/0x1
                  0/1/1/0.0x0.0x0                              0/1/1/0.0.0
64000/0xfa00/0x3
                  0/2/1/0.0x500805f300083891.0x4011000000000000 0/2/1/0.1.6.0.0.2.1
                  0/4/1/0.0x500805f300083891.0x4011000000000000 0/4/1/0.1.6.0.0.2.1
64000/0xfa00/0x4
                  0/2/1/0.0x500805f300083891.0x4010000000000000 0/2/1/0.1.6.0.0.2.0
```

DSF Details

There are two types of DSFs for mass storage: **legacy** DSFs and **persistent** DSFs. Both can be used to access a given mass storage device independently, and both can coexist on a system.

Legacy DSF

The only type of mass storage DSF available in releases prior to HP-UX 11i v3, so it is associated with the legacy view. It is locked to a particular lunpath, and does not support agile addressing. Each lunpath requires a different DSF, so a multi-pathed LUN has multiple DSFs, one for each lunpath. (Note that on HP-UX 11i v3, legacy DSFs support multi-pathing by default—that is, I/O requests to one legacy DSF may use any lunpath to the device. See Disabling Multi-Pathing on Legacy DSFs for more information.)

A legacy DSF contains hardware path information such as SCSI controller, target, and LUN in the device file name and minor number. The minor number field widths for controller address (8 bits), target address (4 bits), and LUN address (3 bits) limit the system to 255 distinct controllers and 32768 distinct lunpaths. Systems with mass storage devices beyond those limits are unable to address them using legacy DSFs.

The name and minor number also include any driver-specific options. For tape devices, this includes tape density and rewind behavior.

The naming convention for legacy DSFs is described in *mksf(1M)* as follows:

```
/dev/dsk/cXtYdZ
/dev/rdsk/cXtYdZ
/dev/dsk/cXtYdZsP
/dev/rdsk/cXtYdZsP
/dev/rmt/cXtYdZ_options
/dev/rac/cXtYdZ_options
/dev/rscsi/cXtYdZ
```

Where
`X` is the instance number of the HBA
`Y` is the target address
`Z` is the LUN unit number
`P` is the optional partition number

Persistent DSF

Associated with a LUN hardware path and seen in the agile view. Because it is based on the LUN hardware path rather than the lunpath, a persistent DSF transparently supports agile addressing. In other words, a persistent DSF is unchanged if the LUN is moved from one HBA to another, moved from one switch or hub port to another, presented using a different target port to the host, or configured with multiple hardware paths. Like the LUN hardware path, the binding of DSF to a device persists across reboots, but is not guaranteed to persist across installations.

The persistent DSF minor number contains no hardware path information, and its name follows a simplified naming convention: `/dev/`*`subdir`*`/`*`classinstance`*

Where
`subdir` is the subdirectory for the device class, such as `disk`, `tape`, `rdisk`, or `rtape`
`class` is the device class, either `disk` or `tape`
`instance` is the instance number assigned to the device

Each class of device has its own set of instance numbers, so each combination of class and instance number refers to exactly one device. Note: The instance numbers might not be sequentially ordered based on hardware path, as instance numbers are assigned in the order of discovery in HP-UX 11i v3.

DSFs for disks are now under the `/dev/[r]disk` directory, and DSFs for tape devices are under the `/dev/rtape` directory. This avoids any confusion with existing legacy DSFs, which reside in `/dev/[r]dsk` and `/dev/rmt`. The naming convention for persistent DSFs is described in *mksf(1M)* as follows:

```
/dev/disk/diskN
/dev/rdisk/diskN
/dev/disk/diskN_pP
/dev/rdisk/diskN_pP
/dev/rtape/tapeNoptions
/dev/rchgr/autochNoptions
/dev/pt/ptN
```

Where
`N` is the instance number of the disk or tape
`P` is the optional partition number

The mapping from persistent to legacy DSF name is described in the following table:

Persistent DSF Name	**Legacy DSF Name**	**Description**
/dev/disk/disk#	/dev/dsk/c#t#d#	The entire disk (block access)
/dev/rdisk/disk#	/dev/rdsk/c#t#d#	The entire disk (raw access)
/dev/disk/disk#_p#	/dev/dsk/c#t#d#s#	Partition on the disk (block access)
/dev/rdisk/disk#_p#	/dev/rdsk/c#t#d#s#	Partition on the disk (raw access)
/dev/rtape/tape#*options*	/dev/rmt/c#t#d#*options*	Tape device (raw access)
/dev/rchgr/autoch#	/dev/rac/c#t#d#_*options*	Autochanger device (raw access)
/dev/pt/pt*instance*	/dev/rscsi/c#t#d#	Pass-through device (raw access)

Note: A single persistent DSF maps to multiple legacy DSFs if the device is multi-pathed.

Accessing the Agile View

To retain backward compatibility, most commands show the legacy view of mass storage by default. You can select the agile view with command line options or a graphical toggle, as documented for each command. For example, `ioscan` shows the legacy view by default, and switches to the agile view if you use the `-N` option. System Manager Homepage (SMH) provides a toggle to select the preferred display view or to select which type of DSF should be used.

In addition, some commands display a specific view based on input parameters. For example, `lssf` uses the agile view if you use a persistent DSF as a command line argument. Other commands display the type of DSF or hardware path saved from a previous configuration. For example, if some LVM disks were configured in the legacy view and others in the agile view, `vgdisplay` displays the LVM configuration using a mix of both views.

Finally, some commands, such as `setboot`, display information only in the agile view. Since this is not backward compatible, such exceptions are noted in the Backward Compatibility section.

Changes to I/O and Mass Storage Subsystems

This section describes the user-visible changes to the mass storage commands and configuration interfaces.

Changes to Existing Commands

Existing I/O commands are backward compatible; next generation features are controlled by new options to existing commands or by the mode of the command arguments. The changes are described in detail in the command manpages, and are summarized in the following sections.

ioscan(1M)

There are several new options and features for the `ioscan` command in HP-UX 11i v3. If you do not use these new options, the `ioscan` output matches that of previous releases.

Note: For readability, examples in this section are reformatted and abbreviated. See The Agile View and Appendix B: Using ioscan for complete sample outputs.

`-N` Displays the agile view of the I/O configuration. The output is changed as follows:

- Targets are displayed as class `tgtpath` and hardware type `TGT_PATH`.
- Each mass storage device has at least two entries in the `ioscan` output, one for each lunpath and one for the device itself.
  - For each lunpath, `ioscan` displays the class `lunpath`, the lunpath hardware path, and the description `LUN path for` *`device`*, where *`device`* is the device class and instance number, such as `disk30`.
  - For each mass storage device, `ioscan` displays the LUN hardware path instead of the legacy hardware path.
- With the existing `-n` option, instead of displaying legacy DSFs, `ioscan` displays persistent DSFs. These are displayed below the device entry, not below each lunpath.

For example:

```
# ioscan -N -n -f
Class     I H/W Path  Driver    S/W State H/W Type  Description
=============================================================
tgtpath   4 0/1/1/0.0x0
                      estp      CLAIMED   TGT_PATH  parallel_scsi target
lunpath   1 0/1/1/0.0x0.0x0
                      eslpt     CLAIMED   LUN_PATH  LUN path for disk48
tgtpath   6 0/2/1/0.0x500805f300083891
                      estp      CLAIMED   TGT_PATH  fibre_channel target
lunpath   6 0/2/1/0.0x500805f300083891.0x4011000000000000
                      eslpt     CLAIMED   LUN_PATH  LUN path for disk49
tgtpath   8 0/4/1/0.0x500805f300083891
                      estp      CLAIMED   TGT_PATH  fibre_channel target
lunpath 30 0/4/1/0.0x500805f300083891.0x4011000000000000
                      eslpt     CLAIMED   LUN_PATH  LUN path for disk49
disk     48 64000/0xfa00/0x1
                      esdisk    CLAIMED   DEVICE    HP 36.4GST336753LC
                                      /dev/disk/disk48   /dev/rdisk/disk48
disk     49 64000/0xfa00/0x3
                      esdisk    CLAIMED   DEVICE    HP MSA VOLUME
                                      /dev/disk/disk49   /dev/rdisk/disk49
```

`-P` *`property_name`*

Displays the value of the given property for all entries. Use the `-d`, `-C`, `-I` or `-H` options or a device file name to limit the `ioscan` output to particular entries. In HP-UX 11i v3, the following properties are defined:

- `health`
- `bus_type`
- `cdio`
- `is_block`
- `is_char`
- `is_pseudo`
- `b_major`
- `c_major`
- `minor`
- `class`
- `driver`
- `hw_path`
- `id_bytes`
- `instance`
- `module_name`
- `sw_state`
- `hw_type`
- `description`
- `error_recovery`
- `is_inst_replaceable` (requires patches PHKL_37458 and PHCO_37479 for releases prior to March 2008)

These fields are defined in *ioscan(1M)*.

The `health` property is new in HP-UX 11i v3. Note that this property may not be updated until the device is accessed. The following table lists values for `health`:

`online`	Device is online and functional.
`offline`	Device has gone offline and is inaccessible.
`limited`	Device is online but performance is degraded due to links, paths, or connections being offline.
`unusable`	An error condition occurred that requires manual intervention.
`disabled`	Device has been disabled or suspended.
`standby`	Device is functional but not in use.

`-m dsf [`*`dsf_name`*`]`

Displays the mappings between a persistent DSF and its equivalent legacy DSFs. Either type of DSF can be specified. If no DSF name is specified, the mappings are displayed for all valid character DSFs. For example:

```
# ioscan -m dsf /dev/dsk/c19t0d1
Persistent DSF           Legacy DSF(s)
========================================
/dev/disk/disk43         /dev/dsk/c19t0d1

# ioscan -m dsf /dev/disk/disk43
Persistent DSF           Legacy DSF(s)
========================================
/dev/disk/disk43         /dev/dsk/c9t0d1
```

```
/dev/dsk/c11t0d1
```

`-m hwpath [-H` *`hardware_path`*`]`

Displays the mappings between a LUN legacy, lunpath, and LUN hardware paths. Any type of hardware path can be specified. If no hardware path is specified, the mapping is displayed for all entries. For example:

```
# ioscan -m hwpath -H 0/4/1/1.124.2.0.0.0.1
Lun H/W Path    Lunpath H/W Path                     Legacy H/W Path
====================================================================
64000/0xfa00/0x1
                0/4/1/1.0x50001fe15008a738.0x4001000000000000
                                                     0/4/1/1.124.2.0.0.0.1

# ioscan -m hwpath -H 0/4/1/1.0x50001fe15008a738.0x4001000000000000
Lun H/W Path    Lunpath H/W Path                     Legacy H/W Path
====================================================================
64000/0xfa00/0x1
                0/4/1/1.0x50001fe15008a738.0x4001000000000000
                                                     0/4/1/1.124.2.0.0.0.1
```

`-m lun [-H` *`LUN_hardware_path`*`] [`*`dsf_name`*`]`

Displays the mappings between a LUN hardware path and its lunpath hardware paths. Use the `-d`, `-C`, `-I` or `-H` options or a device file name to limit the `ioscan` output to particular entries. For example:

```
# ioscan -m lun /dev/disk/disk50
Class I Lun H/W Path     Driver  S/W State H/W Type  Health Description
=======================================================================
disk 50 64000/0xfa00/0x4 esdisk  CLAIMED   DEVICE    online HP MSA VOLUME
             0/2/1/0.0x500805f300083899.0x4010000000000000
             0/2/1/0.0x500805f300083891.0x4010000000000000
             0/4/1/0.0x500805f300083899.0x4010000000000000
             0/4/1/0.0x500805f300083891.0x4010000000000000
                      /dev/disk/disk50   /dev/rdisk/disk50
```

`-s` Displays stale entries–those devices that have no associated hardware.

`-U` Initiates a hardware scan on all entries that have no associated driver. It probes only nodes that are in the `UNCLAIMED` state.

`-e` Displays the Extensible Firmware Interface (EFI) format on HP Integrity systems and Boot Console Handler (BCH) format on PA-RISC systems. For more information on BCH format, see the impacts section on PA-RISC Boot Subsystem.

`-b, -r, -B`

Initiates, removes, and lists deferred bindings. The existing `-M` and `-H` options force a named driver to bind to a specified hardware path. If that driver does not support online binding or the currently bound driver cannot unbind, then the binding operation is deferred until the next reboot. The `-b` option explicitly requests a deferred binding rather than an immediate one. The `-r` option, when used with the `-H` option, removes a deferred binding. The `-B` option lists all deferred bindings. For examples, see Force-Binding a Driver.

insf(1M)

The `insf` command assigns instance numbers to devices and installs DSFs. When run without options, `insf` creates both legacy and persistent DSFs for all new devices. The `insf` command supports one new option:

-L Creates legacy DSFs and enables the support of legacy DSFs. When used with the -v option, insf -L reports whether the legacy mode is enabled or disabled. For example:

```
# insf -Lv
insf: Legacy mode enabled
```

lssf(1M)

The lssf command displays information about a DSF. For persistent DSFs, the output shows the LUN hardware path. For example:

```
# lssf /dev/dsk/c2t0d0
sdisk card instance 2 SCSI target 0 SCSI LUN 0 section 0 at address 0/1/1/0.0.0
/dev/dsk/c2t0d0
# lssf /dev/disk/disk47
esdisk section 0 at address 64000/0xfa00/0x0 /dev/disk/disk47
```

The lssf command also supports two new options:

-s Displays stale DSFs (DSFs for which the hardware is not accessible). The output can contain both legacy and persistent DSFs. For example:

```
# lssf -s
Stale Block Device Files
------------------------
/dev/dsk/c4t0d0
/dev/dsk/c4t1d0
/dev/dsk/c4t3d0
/dev/disk/disk18
/dev/disk/disk19

Stale Character Device Files
----------------------------
/dev/rdsk/c4t0d0
/dev/rdsk/c4t1d0
/dev/rdsk/c4t3d0
/dev/rdisk/disk18
/dev/rdisk/disk19
```

-c Performs critical resource analysis on a specified DSF. You can use this option during migration to confirm that an application is not using legacy DSFs.

mksf(1M)

The mksf command creates a single DSF. In HP-UX 11i v3, it enables the -H option to use lunpath hardware paths. The mksf command supports one new option:

-P Creates a pass-through persistent DSF for the esdisk, estape, and eschgr drivers.

rmsf(1M)

The rmsf command removes DSFs and device definitions from the system. It supports the following new options:

-u Unbinds a driver from a given hardware path when used with the -H option, as described in Force-Binding a Driver.

`-x` Removes stale entries from the I/O configuration (devices for which there appears to be no hardware) and associated DSFs.

`-L` Removes legacy DSFs and their entries in the I/O configuration. This option is typically used during migration in conjunction with `insf`. The `rmsf -L` command disables legacy addressing support by removing all legacy DSFs and legacy configuration information, while `insf -L` re-enables legacy addressing support by recreating the legacy DSFs and configuration information. For more information, see Migrating to the Agile View.

ioinit(1M)

The `ioinit` command tests and maintains consistency between the kernel I/O data structures and the I/O configuration files. It can also reassign instance numbers for devices. In HP-UX 11i v3, instance numbers are assigned as devices are discovered, so instance numbers for a given device class may not be sequentially ordered based on hardware path. Because persistent DSF names contain the instance number, you might want to reassign instance numbers to remove "holes" in the numbering or create consistent DSF names for devices shared between servers. As of HP-UX 11i v3, `ioinit` provides the ability to reassign instance numbers without a reboot, also known as online instance number reassignment, and the ability to reassign all the instance numbers of a given class, starting from zero, on the next boot.

The existing `-f` *`infile`* option, used to reassign instance numbers, is changed as follows:

- The *`infile`* must contain all the valid devices.
- If none of the drivers of the devices provided support online reassignment, you must reboot the system to complete the instance number reassignment, as in previous releases of HP-UX.
- If all the drivers of the devices provided support online reassignment, the instance number reassignment happens immediately, without rebooting the system.
- HP discourages using an *`infile`* containing a mix of drivers that support online instance number reassignment and drivers that do not. HP recommends running `ioinit` twice, once with all the drivers that support online instance number reassignment and then with all the drivers that do not. But vice versa is not recommended. If you want to use both types of drivers in the same *`infile`*, you must halt the special file daemon `sfd` before running `ioinit`. To do this, edit `/etc/inittab` to comment out the entry related to `sfd` and run `init q`. To complete the instance number reassignment, you must then reboot the system.

  Note: Online instance number reassignment requires support in the device driver. If a driver supports online instance number reassignment, it supports the `is_inst_replaceable` property. This property is available in the HP-UX 11i v3 March 2008 release, or you can enable it by installing PHKL_37458 and PHCO_37479. For more information, see *ioscan(1M)*.

  Note: If the *`infile`* has more than 14 devices whose drivers support online reassignment, you must install patches PHKL_36333 and PHCO_36315 to complete the reassignment online.

The `ioinit` command also supports the following new options:

`-rC` *`class`*

Forces the kernel to reassign instance numbers starting from 0 for all devices belonging to the given *`class`*. If the reassignment succeeds, `ioinit` reboots the system. Existing DSFs are not removed, but their mappings might change; that is, old instance numbers might be reused. To create new DSFs for the devices whose instance number has changed, run `insf -e -C class`. Use this option with caution. You must be sure that the applications using these DSFs are modified to point to the newly created DSFs.

`-v` *`ioconfig_file`*

Checks an I/O configuration file for corruption. The *`ioconfig_file`* argument specifies the file to check.

-A With the `-f` option, performs Critical Resource Analysis (CRA) on all the devices corresponding to the hardware paths specified in *`infile`*. You must specify the `-f` option with this option. Instance numbers are reassigned as specified in *`infile`*, only if CRA of all these devices report SUCCESS. Existing device special files of LUNs whose instance numbers are to be reassigned are deleted and new device special files are created with the new instance numbers after successful instance number reassignment. If the driver associated with at least one given hardware path does not support online instance number reassignment, you must also specify the `-r` option. If the *`infile`* contains a hardware path of a node whose driver does not support online instance number reassignment and `-r` option is not specified, the command aborts the operation. If this option is used, restarting `sfd` is not required.

For example, to perform CRA and reassign instance numbers for the devices specified in the *`infile`*, provided all the drivers associated with the hardware paths in the *`infile`* support online instance number reassignment, enter the following command:

```
# ioinit -f infile -A
```

To perform CRA and reassign instance numbers for the devices specified in the *`infile`*, if few drivers associated with the hardware paths in the *`infile`* do not support online instance number reassignment, enter the following command:

```
# ioinit -f infile -A –r
```

scsictl(1M)

The `scsictl` command controls SCSI targets and LUNs. It supports the following new option:

`-t` *`tgtid`*

Operates on a SCSI target instead of a LUN. The *`tgtid`* argument specifies the target port ID. The list of operations is defined in *scsictl(1M)*.

mediainit(1)

The `mediainit` command initializes mass storage media by formatting the media, writing and reading test patterns to verify media integrity, and sparing any defective blocks found. Formatting destroys the existing user data only in the area being initialized. It does not destroy the user data on the entire disk; you can easily retrieve the original data.

Before you can reuse or redeploy a disk, you must erase the user data completely to ensure data security.

In HP-UX Update 4, `mediainit` provides the disk scrub feature. Disk scrubbing overwrites the entire disk with a single character. Because all addressable locations of the disk are overwritten, retrieval of the original data becomes more difficult and data security is ensured.

The disk scrub feature is supported only for “disk” class devices. The `mediainit` command, by default scrubs the media three times. It enables you to specify the character to use for disk scrubbing. With the `-t` option, you can also select the number of iterations of scrubbing to be done.

The `mediainit` command provides new options `–S`, `-C`, and `–t`, which can be used to scrub the disk devices.

Syntax:

`mediainit –S [–c scrub_character] [-t scrub_count] special_file`

`-S` Scrubs the disk. You must specify the `-c` and `-t` options with the `–S` option. If you do not specify both options, the device is scrubbed three times.

`-c` *scrub_character*
Specifies the character to use to scrub the disk. You can specify any character between 0-9, a-z, or A-Z. You must specify this option with the `-S` option.

`-t` *scrub_count*
Specifies the number of times to scrub the disk. You must specify this option with the `-S` option.

special_file
Specifies the path name of the character (raw) device special file associated with the device to be scrubbed. The `mediainit` command aborts if you do not have either read or write permission to the device special file, or if the device is currently open for any other process. The `mediainit` command opens the device in exclusive mode. Note: Before a device special file is scrubbed, if any file system is on the disk, you must unmount the file system.

Examples:

1. To scrub the disk twice with the character 0 (zero), enter:
   # **mediainit -S -c 0 -t 2** ***character_device_special_file***

2. To scrub the device in the DoD 5220.22-M approved method using the user specified character 2, enter:
   # **mediainit -S -c 2** ***character_device_special_file***

For optimal performance, use the following guidelines:

- The `mediainit` command provides the DoD 5220.22-M approved method of disk scrubbing with the `–S` option. Using this option scrubs the disk three times with three random characters.
- In case of disk arrays, ensure that each individual physical disk drive is scrubbed separately.
- In case of mirroring of disks, you must ensure that both the disks that form a mirror must be scrubbed separately.
- Ensure no other processor application is trying to access the disk while disk scrubbing is in progress.

Note: HP does not guarantee that the data scrubbed using the `mediainit` disk scrub is completely irretrievable. If might still be possible to retrieve the original data using sophisticated data retrieval techniques.

New Commands

There are four new system administration commands to manage mass storage:

- `iobind(1M)`
- `iofind(1M)`
- `io_redirect_dsf(1M)`
- `scsimgr(1M)`

iobind(1M)

The `iobind` command unbinds the driver from an existing LUN and binds a new driver to it. You can use `iobind` to override a default driver selection made by the operating system. In releases prior to HP-UX 11i v3, forcing a particular driver to claim a specified device was done through `driver` statements in the `/stand/system` file. This method is supported in HP-UX 11 v3, but the `iobind` command supersedes it.

The `iobind` command syntax is as follows:

```
# iobind -M driver_name -H hw_path [-I instance]
```

This command unbinds a driver from the device at the specified hardware path. If the unbinding is successful, the named driver is then bound to the device. If the unbinding cannot be done immediately, either because the device is in use or the driver does not support it, `iobind` prints a warning, and the binding is applied on the next reboot. The driver bound explicitly using the above mechanism is retained across reboots. You can specify an instance number to be assigned to the new binding with `-I`.

io_redirect_dsf(1M)

The `io_redirect_dsf` command is used when you replace a mass storage device with a similar device, and you want to use the existing DSF to access the replacement device. This command redirects the DSF to the replacement disk by assigning the instance number of the replaced disk to the new disk. The DSF name of the new disk, which is created using the base name and instance number, is the same as that of the replaced disk. After physically replacing the disk, use the `io_redirect_dsf` command as follows:

```
# io_redirect_dsf -H old_device_LUN_hw_path -N new_device_LUN_hw_path
```

or

```
# io_redirect_dsf -d old_dsf_name -n new_dsf_name
```

The replacement disk must belong to the same class as the original disk for the DSF redirection to succeed. The `io_redirect_dsf` command can only be used to redirect DSFs in the agile view. Redirection of a DSF from a disk to a new disk results in a short application downtime.

Note: Use the `io_redirect_dsf` command only if the replacement disk has a different WWID from the original disk. Replacing an internal disk or a disk in a JBOD changes the WWID because the LUN maps directly to the physical disk. Disk LUNs in arrays are not mapped to physical disks. Replacing a disk in an array does not change the LUN WWID as seen from the host, so `io_redirect_dsf` is not necessary.

iofind(1M)

The `iofind` command is a tool to assist migration from the legacy view to the agile view. It scans ASCII files on a system, locates references to legacy DSFs and hardware paths, and optionally replaces them with their agile equivalents. You can limit the search patterns to specific DSFs or hardware paths. The search can recurse from the root directory, or be limited to a specific set of directories. If no DSF or hardware path is specified, `iofind` uses `ioscan` to get the list of valid DSFs or hardware paths on the system and creates a file containing the mapping for all entries. If you choose to replace the legacy information with its agile equivalent, the original ASCII files are preserved in a backup directory.

For additional information on migrating to the agile view, see the *HP-UX 11i v3 Persistent DSF Migration Guide* in For more information.

scsimgr(1M)

The `scsimgr` command provides a single command line interface to manage and diagnose the mass storage stack. It is designed to work with persistent DSFs, but also works with legacy DSFs for a limited set of operations.

The `scsimgr` command includes the following features:

- Retrieves and clears driver statistics
- Displays status information about SCSI objects
- Retrieves, sets, or saves attributes of SCSI objects
- Disables and enables SCSI objects
- Performs SCSI task management functions such as LUN and target resets
- Performs miscellaneous SCSI commands such as inquiry and self-tests

In addition to gathering statistics and printing device information, you can use `scsimgr` to set attributes, which replace SCSI tunable parameters set at the operating system level. Attributes can be set globally like kernel tunable parameters, or can be restricted to a particular device type, device instance, driver, SCSI target, vendor, or product.

In HP-UX 11i v3, `scsimgr` provides generic management capabilities for the SCSI subsystem and driver-specific management capabilities for disk drivers.

For more information, including a list of commands and keywords for the `scsimgr` command, see the *scsimgr*(1M) man page and Appendix C: Using scsimgr. A more detailed white paper entitled *Scsimgr SCSI Management and Diagnostics Utility* is available on the web; see the For more information section.

Obsolete and Deprecated Features

Infinite I/O Retries

In previous releases, certain types of disk I/O request failures were retried indefinitely by the mass storage stack. Starting with HP-UX 11i v3, this behavior is configurable; the default behavior retries failing I/O requests a finite number of times. If all the retries fail, the I/O request returns a failure notification to the calling application.

Some applications are designed to expect I/O requests to always succeed. HP recommends testing your applications with the new finite retry policy to determine how they behave when an I/O request fails.

To control the retry policy, use the `scsimgr` command to set the `infinite_retries_enable` (which toggles the policy between infinite and finite) and `max_retries` (which defines the number of retries in the finite case) attributes. Changes to these attributes take effect immediately and do not require a reboot.

For example, to restore infinite retries, enter the following command:

```
# scsimgr set_attr –a infinite_retries_enable=true
```

To set the retry policy for a particular device to finite retries, enter the following:

```
# scsimgr set_attr -d esdisk -D device_file –a infinite_retries_enable=false
```

To change the number of retries for all disks, enter the following:

```
# scsimgr set_attr -N /escsi/esdisk -a max_retries=new_value
```

Note: To set the number of retries, the retry policy must be set to finite.

Legacy Device Special Files and Hardware Paths

As of HP-UX 11i v3, the legacy view is deprecated, and all its components—naming conventions, DSFs, and hardware paths—will be obsoleted in future versions of HP-UX.

Tunable Kernel Parameters

The following tunable kernel parameters related to mass storage are obsolete in HP-UX 11i v3:

`scsi_max_qdepth` Controls the maximum number of I/O operations that a LUN can queue up for execution. It has been replaced with the `max_q_depth scsimgr` attribute. To change the queue depth on a LUN, enter the following command:

```
# scsimgr set_attr -D device_file –a max_q_depth=new_value
```

For example, to set the queue depth for disk14 to 32, enter the following:

```
# scsimgr set_attr -D /dev/rdisk/disk14 -a max_q_depth=32
```

To set the queue depth to 8 persistently for all LUNs on the system, enter the following:

```
# scsimgr save_attr –N /escsi/esdisk –a max_q_depth=8
```

`scsi_max_phys` Sets the maximum data size the SCSI subsystem accepts for an I/O request. It has been replaced with the `escsi_maxphys scsimgr` attribute. For example, to set the maximum data size persistently to 64 4KB pages, enter the following command:

```
# scsimgr save_attr –a escsi_maxphys=64
```

`default_disk_ir` Enables or disables the use of a device write cache, also known as immediate reporting. It has been replaced with the `immediate_report` parameter to the `scsictl` command. To disable immediate reporting for a SCSI device on the system, enter the following command:

```
# scsictl –m immediate_report=0 device_file
```

To disable immediate reporting for a set of disks based on vendor id, product id, and/or firmware revision, use the `scsimgr` command to set the `disable_flags` attribute of the `esdisk` driver. This attribute can be set at any settable attribute scope to control a number of functions including write cache enable (WCE). For example, to persistently disable write cache and 16 bytes read/write CDBs on all disks managed by the `esdisk` driver, enter the following command:

```
# scsimgr save_attr –N /escsi/esdisk –a disable_flags=”WCE RW16”
```

For more information about settable attribute scope, see *scsimgr(1M)* and for the `disable_flags` attribute, see *scsimgr_esdisk(7)*.

Changes to Existing Procedures

Most I/O operations in HP-UX 11i v3 are no different from HP-UX 11i v2. To enable the agile view for a command, you may need to specify a command-line option such as `–N` for `ioscan` or toggle a button in a GUI. However, you perform the following three tasks differently in HP-UX 11i v3:

- Replacing a Disk
- Force-Binding a Driver
- Disabling Multi-Pathing on Legacy DSFs

Replacing a Disk

Replacing an internal disk or a disk in a JBOD, even at the same hardware path, changes its WWID because the WWID is tied to the physical disk. Since persistent DSFs are tied to the LUN WWID, the mass storage stack considers the replacement a *new* disk and creates new DSFs for it. If your intent was to *replace* the disk, not *add* a new disk, you must use the `io_redirect_dsf` command to specify to the mass storage stack that the existing DSF applies to the replacement disk. For example:

```
# io_redirect_dsf -d old_dsf_name -n new_dsf_name
```

This command redirects the DSF from the original disk DSF specified with `-n` to the replacement disk DSF specified with `-N`. For example, if you replaced the disk with the DSF name `/dev/disk/disk3`, and the mass storage stack assigned the name `/dev/disk/disk1` to its replacement, enter the following:

```
# io_redirect_dsf -d /dev/disk/disk3 -n /dev/disk/disk1
```

This command redirects the DSF of `disk3` to point to `disk1`. Both disks must belong to the class `disk`.

For a complete example of disk replacement, see the troubleshooting chapter of the *HP-UX System Administrator's Guide: Logical Volume Management*.

Note: Disk LUNs in an array device are not tied to physical disks. Replacing a disk in an array does not change the LUN WWID as seen from the host, so `io_redirect_dsf` is not necessary. Use the `io_redirect_dsf` command only if the replacement disk has a different WWID from the original disk.

Force-Binding a Driver

Forcing a particular driver to claim a specified device, thus overriding any default selection, is no longer done using `driver` statements in the `/stand/system` file.

In HP-UX 11i v3, `driver` statements in `/stand/system` are still used for devices in the legacy view. Under the agile view, use the `iobind` command to force-bind drivers for all devices as follows:

```
# iobind -M driver_name -H device_hw_path [-I instance]
```

This command binds the named driver to the device at the specified path. If the device is already claimed by a different driver that does not support unbinding or if the named driver does not support online binding, the binding is applied at the next reboot. The driver bound explicitly using the command is retained across reboots. Optionally, you can specify an instance number with `-I`.

To explicitly defer the force-binding until the next reboot, use the `-b` option as follows:

```
# ioscan -b -M driver_name -H device_hw_path [-I <instance>]
```

To display any deferred bindings, use the `-B` option as follows:

```
# ioscan -B
```

To remove any deferred bindings, use the `-r` option as follows:

```
# ioscan –r -H device_hw_path
```

Disabling Multi-Pathing on Legacy DSFs

By default, the multi-pathing feature of the next generation mass storage stack enables I/O requests and I/O control operations to be processed along any hardware path to a LUN. Even if legacy DSFs are used for I/O, requests can still be routed through a different hardware path. This maximizes availability and parallelism.

To force legacy DSFs to use backward-compatible multi-pathing behavior, you can use the `scsimgr` command to configure a global device tunable called `leg_mpath_enable`. If you set `leg_mpath_enable`, multi-pathing is enabled on all legacy DSFs, and I/O control operations (`ioctls`) are processed on any available path. This is the default behavior. If `leg_mpath_enable` is not set, no multi-pathing is performed on any legacy DSF, and `ioctls` are processed only on the LUN path corresponding to the legacy DSF. To force the backward-compatible behavior persistently for the entire server, enter the following:

```
# scsimgr save_attr -a leg_mpath_enable=false
```

To limit the behavior to a particular LUN, set the per-LUN device tunable, which overrides the global value as follows:

```
# scsimgr save_attr -D /dev/rdisk/disk4 -a leg_mpath_enable=false
```

Note: Multi-pathing through persistent DSFs is not affected by this tunable.

Changes to Other Subsystems

This section describes how other parts of HP-UX are affected by the next generation mass storage stack. Most subsystems that deal with I/O or mass storage have been updated to use the agile view. They allow the use of both persistent and legacy device special files, and the use of legacy, lunpath, and LUN hardware paths. Some subsystems only support one of the two views, and some commands have new options related to the mass storage stack; these are summarized below.

Crashdump

The dump subsystem supports both the legacy view and the agile view. If a dump device is configured using a legacy DSF, `crashconf` converts it to an equivalent persistent DSF and selects an available lunpath hardware path for the dump device.

The dump subsystem is aware of multi-pathed devices and supports automatic dump device path failover. If a configured path goes offline, the dump subsystem automatically selects an alternate available hardware path and reconfigures the dump device.

When a LUN dynamically expands or contracts, the dump subsystem automatically updates its internal data structures to use the new device size.

Using `lvlnboot`, `vxvmboot`, and the `/stand/system` file to configure dump devices is deprecated and will be obsoleted in a future release.

For additional information on the dump subsystem in HP-UX 11i v3, see the *HP-UX 11i v3 Crash Dump Improvements* white paper in For more information.

fcmsutil(1M)

The replace_dsk option to the fcmsutil command is no longer available. It is replaced with a combination of the replace_wwid and replace_leg_dsf options to the scsimgr command. The replace_wwid option validates the change of a LUN associated with a lunpath. The replace_leg_dsf option validates the change of a LUN associated with a legacy DSF.

For example, to validate the LUN association of a lunpath with hardware path 0/1/1/0.1.1, enter the following command:

```
# scsimgr replace_wwid -H 0/1/1/0.1.1
```

To validate the replacement of the legacy LUN associated with /dev/rdsk/c0t0d0, enter the following command:

```
# scsimgr replace_leg_dsf -D /dev/rdsk/c0t0d0
```

HP System Management Homepage

HP System Management Homepage (HP SMH) implements a toggle in its screens dealing with mass storage, such as screens for peripherals and file systems. This toggle enables you to view and configure information using the legacy or agile view, as shown below:

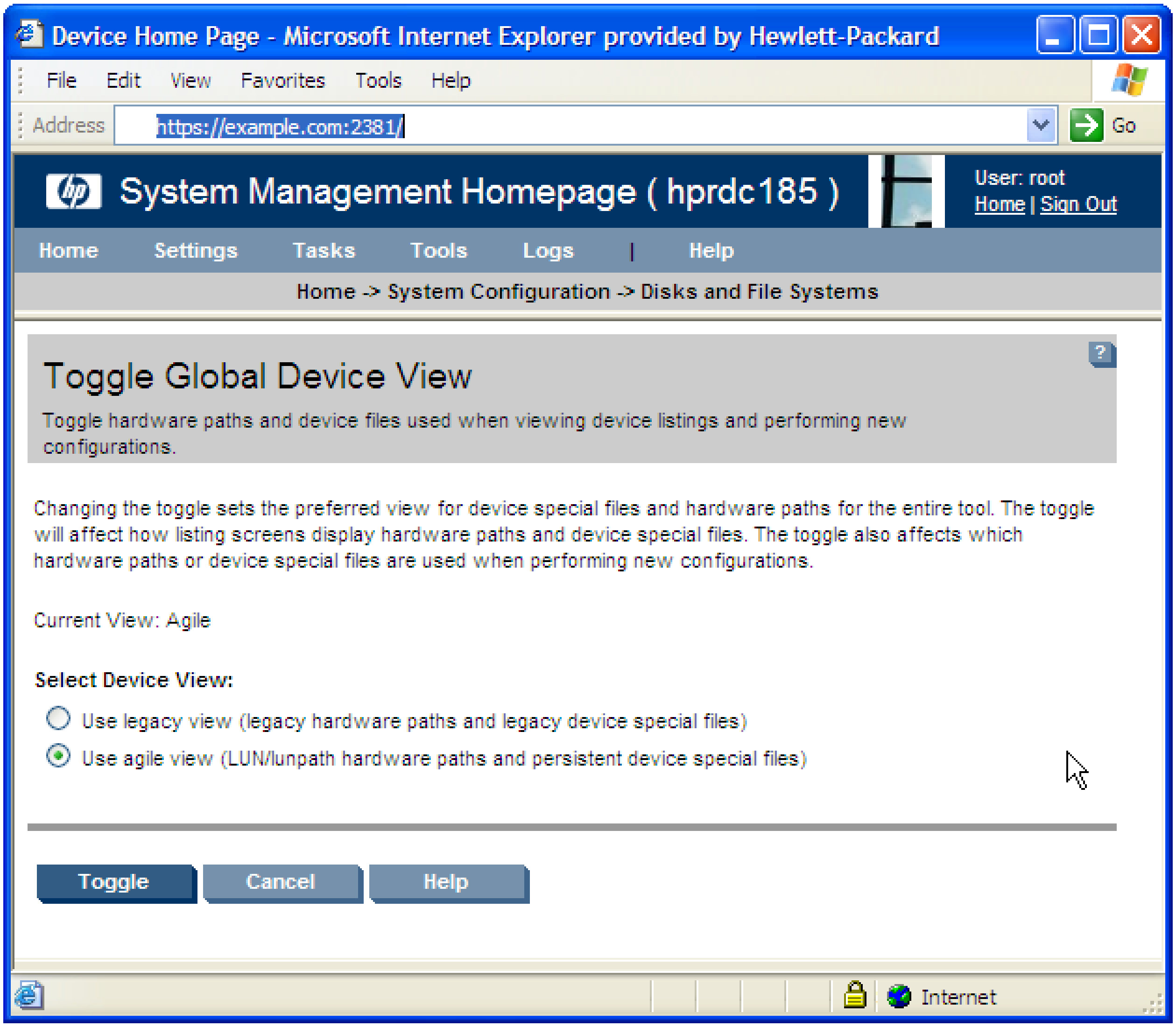

Figure 3: SMH Toggle Global Device View Screen

The setting of this toggle persists between HP SMH sessions by creating a small file when the user selects the agile view for a system. This file is used by various HP SMH applications including *pdweb* and *fsweb*.

For additional information on HP SMH, see the *HP System Management Homepage* white paper in For more information.

Ignite-UX

The Ignite-UX product operates in the agile view to take advantage of the mass storage stack multi-pathing and failover features.

Ignite-UX displays lunpath hardware paths for the location of the target install media and network devices in its Root Disk selection screen, as follows:

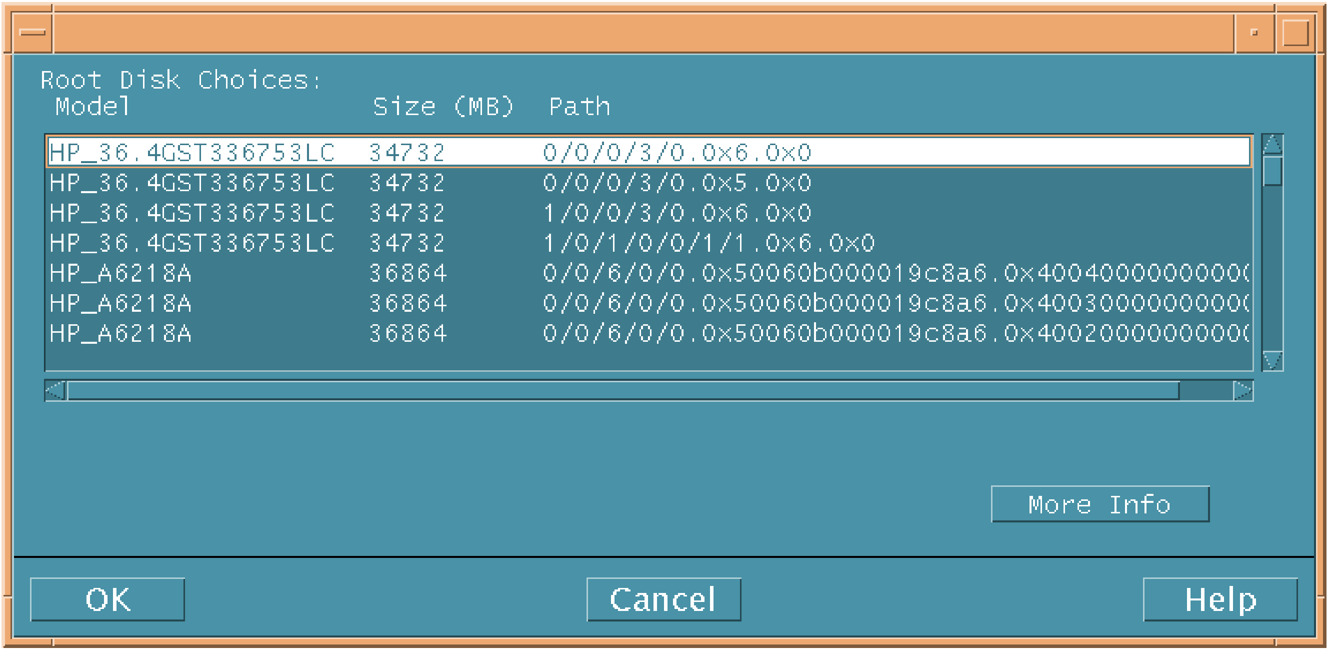

Figure 4: Ignite-UX Root Disk Selection Screen

Note: Starting with Ignite-UX version C.7.2.x, the Ignite-UX variables *inventory_block_path* and *inventory_block_protocols* enable you to hide disks from the I/O configuration process, based respectively on the disk hardware path or the I/O protocol used to access them. This inventory blocking reduces the number of disks displayed in the selection screens while improving the I/O inventory performance. For more information, see the "I/O Configuration" section of *instl_adm(4)*.

After selecting a disk on the Root Disk selection screen, you can access the More Info screen, which displays additional information about a chosen root disk, including the legacy hardware path, WWID, and the associated persistent and legacy DSFs. This enables you to map the agile view to the legacy naming convention from previous releases, as follows:

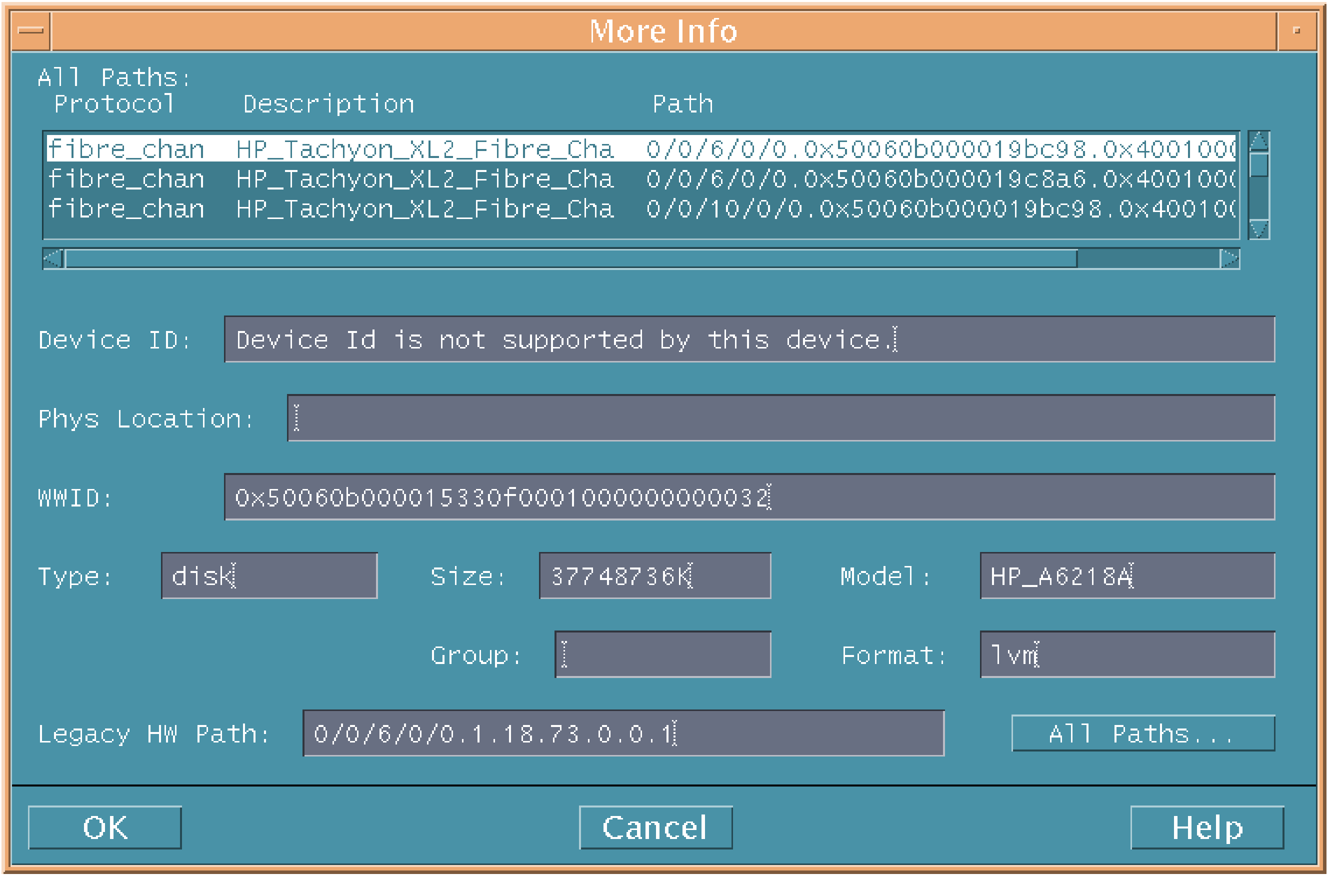

Figure 5: Ignite-UX More Info Screen

During an installation, Ignite-UX configures the root, dump, and swap devices with their agile addressing representation. This means that after installing, the commands that display the boot, dump, and swap devices show them using the agile form.

In previous releases, Ignite-UX associated the same DSFs to the same I/O components for any identically configured systems. This is not true in HP-UX 11i v3, given the parallelism of the I/O system. There is no guarantee that installing or reinstalling a system will generate the same set of DSFs on the system, although a recovery retains the original set of DSFs.

Logical Volume Manager

Logical Volume Manager (LVM) supports both the legacy and agile views. In particular, LVM supports the use of both legacy and persistent DSFs within the same volume group. New options to the `vgscan` and `vgimport` commands, described below, affect how LVM creates LVM configuration information.

By default, `vgscan` recovers LVM configuration information (the `/etc/lvmtab` file) using kernel information for activated volume groups or using legacy DSFs for volume groups that have not been activated since the last boot. If you specify the new `-N` option, then `vgscan` uses persistent DSFs. If you specify the new `-B` option, then `vgscan` populates the `/etc/lvmtab` file using both legacy and persistent DSFs.

By default, when importing a volume group in shared mode, `vgimport` populates the `/etc/lvmtab` file using legacy DSFs. If you specify the new `-N` option together with `-s`, then `vgimport` uses persistent DSFs.

LVM configuration is stored in `/etc/lvmtab` in the same way in which it was configured. For example, if a physical volume is added to a volume group using its legacy DSF, the entry is stored in legacy format. Since Ignite-UX creates the root disk with agile addressing, after a cold install the `lvlnboot` command displays the root disk using its agile form. If the system was updated rather than installed, `lvlnboot` displays the root disk in its legacy form.

A new LVM utility, `vgdsf`, converts existing volume group configuration information from legacy to persistent DSFs.

You can manage alternate links to a multi-pathed device outside of LVM using the next generation mass storage stack. By default, the mass storage stack balances the I/O load across all available paths to a disk. However, you can use the `scsimgr` command to emulate LVM PVLink functionality and to handle LUN failure and load balancing. For more information, see Disabling Multi-Pathing on Legacy DSFs.

Note: LVM continues to support alternate links to a device, but their use is no longer required or recommended.

LVM also supports dynamic LUN expansion. If you increase the size of a LUN, use the `vgmodify` command to incorporate the additional space into the volume group. LVM does not support dynamic LUN contraction or block size changes. If either of these events occurs, LVM marks the LUN as unavailable.

Even though the I/O subsystem supports disks larger than 2 TB, LVM only accesses the first 2 TB of any disk.

For more information about LVM in HP-UX 11i v3, see the white papers in the For more information section.

Offline Diagnostic Environment

The Offline Diagnostic Environment (ODE) has a switch to display either the legacy or agile view. On PA-RISC servers, ODE displays lunpath hardware paths in BCH format (described in PA-RISC Boot Subsystem) in both decimal and hexadecimal. On Integrity servers, ODE displays lunpath hardware paths in hexadecimal only.

PA-RISC Boot Subsystem

On PA-RISC systems, the boot console handler (BCH) locates boot devices and launches preboot applications. The boot loader (`hpux`) loads and launches the operating system. Both the BCH and boot loader are limited to 32-bit address elements, which impacts the mass storage stack as follows:

- If the boot path was set using the agile view, the BCH `PATH` command displays the lunpath hardware path in a different format from the `setboot` or `ioscan -N` command. If the boot path includes any 64-bit elements, the BCH displays them as two 32-bit numbers. For example, if the primary boot path is 0/2/1/0.0x500508b300903331.0x4002000000000000, the BCH displays the following:

  ```
  Primary boot path: 0/2/1/0.1342507187.9450289.1073872896
                     0/02/01/0.500508b3.903331.40020000(hex)
  ```

  The leading 0x is omitted, the path elements are split into 32-bit parts, and the last 8 zeroes are omitted.

  Older systems may print the primary boot path in decimal only. You must manually convert the address to recognize the hexadecimal path elements.

- When using the BCH to enter a lunpath hardware path in the `BOOT` or `PATH` commands, you must use the 32-bit format. For example:

  ```
  Main Menu: Enter command > PA PRI 0/2/1/0.0x500508b3.0x903331.0x40020000.0x0
  ```

- The BCH `SEArch` command prints legacy hardware paths only. To display lunpath hardware paths in BCH format, use the `ioscan` command with the `-e` option.

- When using the boot loader, you must use the 32-bit format. For example, to boot from a device with a lunpath hardware path of 1/2/0.0x51ef000012003400.0x4000000000000000, enter the following:

```
ISL> hpux boot (1/2/0.0x51ef0000.0x12003400.0x40000000.0x0;0)/stand/vmunix
```

To display hardware paths in BCH format while the operating system is running, use the `ioscan` command with the `-e` option. The BCH hardware paths are printed in both hexadecimal and decimal. For example:

```
# ioscan -N -k -e -H 0/4/1/1.0x50001fe15008a73c.0x4002000000000000
H/W Path                                       Class   Description
==================================================================
0/4/1/1.0x50001fe15008a73c.0x4002000000000000 lunpath LUN path for disk44
    0/4/1/1.0x50001fe1.0x5008a73c.0x40020000,
0/4/1/1.1342185441.1342744380.1073872896
```

setboot(1M)

The `setboot` command displays and modifies boot variables in system stable storage (also known as nonvolatile RAM or NVRAM). It displays information in the agile view only, as lunpath hardware paths and persistent DSFs:

```
# setboot
Primary bootpath : 0/1/1/0.0x0.0x0 (/dev/rdisk/disk47)
HA Alternate bootpath : 0/1/1/0.0x1.0x0 (/dev/rdisk/disk48)
Alternate bootpath : 0/1/1/0.0x1.0x0 (/dev/rdisk/disk48)
```

When setting boot paths, `setboot` accepts a legacy hardware path, a lunpath hardware path, or a persistent DSF. If you specify a persistent DSF, `setboot` selects an available lunpath hardware path to write in system stable storage. For example, the following three commands are equivalent:

```
# setboot -a /dev/rdisk/disk47
Alternate boot path set to 0/1/1/0.0x0.0x0 (/dev/rdisk/disk47)
# setboot -a 0/1/1/0.0x1.0x0
# setboot -a 0/1/1/0.1.0
```

The `setboot` command is aware of multi-pathed devices and supports automatic boot path failover. If the hardware path written into stable storage goes offline, `setboot` retrieves an alternate available hardware path to the LUN and writes it into system stable storage.

For additional information on the `setboot` command in HP-UX 11i v3, see the *Setboot(1M) in HP-UX 11i v3* white paper in For more information.

Support Tools Manager

Support Tools Manager (STM) includes a new menu that displays the agile view.

Third-Party Multi-Pathing Products

Native multi-pathing affects the following third-party multi-pathing products:

SecurePath	SecurePath is not supported on HP-UX 11i v3. For information on migrating from SecurePath to native multi-pathing, see the *HP-UX 11i v2 to 11i v3 Mass Storage Stack Update Guide* in For More Information.

PowerPath	PowerPath does not manage multi-pathing on HP-UX 11i v3. Native multi-pathing is used for EMC devices. For information on migrating from PowerPath to native multi-pathing, see the *HP-UX 11i v2 to 11i v3 Mass Storage Stack Update Guide* in For More Information.
Dynamic Multipathing	VERITAS Dynamic Multipathing (DMP) does not support the agile view. See the *VERITAS Volume Manager 4.1 Administrator's Guide* for information about the coexistence of DMP with native multi-pathing in HP-UX 11i v3.

VERITAS File System

In HP-UX 11i v3, the maximum tested file system size for VERITAS File System (VxFS) 4.1 is 32 TB, subject to size restrictions imposed by any underlying volume manager. For example, LVM supports a maximum volume size of 16 TB, so the maximum size of a VxFS 4.1 file system under LVM is 16 TB. For additional information on supported file system sizes, see the white paper *Supported File and File System Sizes for HFS and JFS*, available at the HP documentation website http://docs.hp.com.

VERITAS Volume Manager

VERITAS Volume Manager (VxVM) 4.1 is supported on HP-UX 11i v3 but it does not support the agile view. Only legacy DSFs are supported.

Virtual Partitions

Virtual Partitions (vPars) A.05.01 does not support the use of lunpath hardware paths or lun hardware paths on the vPars command line. When using the vPars commands to specify hardware paths, use the legacy hardware path format. However, after a virtual partition is created, you can use the agile view on that partition.

Migrating to the Agile View

This section describes how to migrate from the legacy view to the agile view. It includes the benefits of migration, compatibility and coexistence of the two views, and the migration process.

Benefits of Migration

The next generation mass storage stack provides numerous advantages over previous releases. They include the following:

Ease of Use

- Agile addressing, so SAN reconfiguration does not affect DSF binding
- Automatic detection and configuration of new LUNs
- Transparent native multi-pathing of LUNs, using a single DSF and hardware path for all lunpaths
- Integration of multi-pathing with boot and dump subsystems
- Full backward compatibility in legacy mode
- Tools to aid migration to agile view

Manageability

- New `scsimgr` command for managing and troubleshooting mass storage
- Reduction in number of DSFs
- Integration with system management GUIs (SMH and SIM)
- Improved performance tracking tools

Scalability

- Increased architectural and tested limits
- Increased parallelism to take advantage of large multi-CPU server configurations

Performance	• Improved I/O bandwidth through native multi-pathing with choice of load balancing algorithms • Faster I/O scan time through parallelized scanning of mass storage devices • Faster dump time by writing concurrently to multiple dump devices

Backward Compatibility

HP-UX 11i v3 provides full backward compatibility when using legacy mode, except in very limited cases. Legacy DSFs remain on the system and continue to work as before, unless you explicitly disable legacy mode. All commands execute in legacy mode by default, meaning that their behavior is unchanged from previous releases if you use legacy options and input parameters. All legacy ioctls and APIs are supported in legacy mode on HP-UX 11i v3.

The exceptions to full backward compatibility are:

- The `setboot` command displays information in the agile view only, as described in Changes to Other Subsystems: setboot(1M).
- Cold-installing HP-UX 11i v3 configures the boot, swap, and dump devices to use persistent DSFs, as described in Installing versus Updating.
- By default, native multi-pathing is enabled for legacy DSFs. I/O requests directed to a legacy DSF may be silently routed through a different lunpath. This change affects LVM alternate link behavior as well as third-party multi-pathing products. You can restore the legacy behavior using the `scsimgr` command, as described in Disabling Multi-Pathing on Legacy DSFs.
- Replacing an internal disk or a disk in a JBOD requires additional steps to notify the mass storage stack, as described in Changes to Existing Procedures: Replacing a Disk.

Note: Legacy mode is deprecated in HP-UX 11i v3. Refer to the HP-UX 11i Version 3 Release Notes for a complete list of deprecated features.

Coexistence

The agile view of mass storage supersedes the existing legacy view. However, in HP-UX 11i v3 the two views can exist in parallel. The behavior of existing legacy DSFs is not affected by any persistent DSFs on the same server. All mass storage commands are backward compatible and function with either legacy or persistent DSFs, except as described in Backward Compatibility. Most applications that support both legacy and persistent DSFs support a mix of legacy and persistent DSFs.

Installing versus Updating

If you cold-install HP-UX 11i v3, both legacy and persistent DSFs are automatically created. By default, the installation process configures system devices like the boot, root, swap, and dump devices to use persistent DSFs. Configuration files such as `/etc/fstab` and `/etc/lvmtab` contain references to persistent DSFs.

If you update from HP-UX 11i v2 to 11i v3, existing legacy DSFs are retained and persistent DSFs are created. Configuration files are not updated, so system devices continue to use the existing legacy DSFs.

The following table summarizes the differences between installing and updating to HP-UX 11i v3:

Cold install HP-UX 11i v3	**Update to HP-UX 11i v3**
Creates legacy DSFs	Maintains existing legacy DSFs
Creates persistent DSFs	Creates persistent DSFs
Creates system device configuration with persistent	Maintains existing system device configuration with

DSFs. For example:
`# lvlnboot -v`
`Boot Definitions for Volume Group vg00:`
`Physical Volumes in Root Volume Group:`
`     /dev/disk/disk48_p2 -- Boot Disk`
`Boot: lvol1  on:  /dev/disk/disk48_p2`
`Root: lvol3  on:  /dev/disk/disk48_p2`
`Swap: lvol2  on:  /dev/disk/disk48_p2`
`Dump: lvol2  on:  /dev/disk/disk48_p2, 0`
	legacy DSFs. For example:
	`# lvlnboot -v`
	`Boot Definitions for Volume Group vg00:`
	`Physical Volumes in Root Volume Group:`
	`     /dev/dsk/c0t0d0s2 -- Boot Disk`
	`Boot: lvol1  on:  /dev/dsk/c0t0d0s2`
	`Root: lvol3  on:  /dev/dsk/c0t0d0s2`
	`Swap: lvol2  on:  /dev/dsk/c0t0d0s2`
	`Dump: lvol2  on:  /dev/dsk/c0t0d0s2, 0`

Sets default view to legacy	Sets default view to legacy
Enables multi-pathing on legacy and persistent DSFs	Enables multi-pathing on legacy and persistent DSFs
Sets I/O retry policy to infinite retries	Sets I/O retry policy to infinite retries

Table 1: Differences Between Installing and Updating to HP-UX 11i v3

Migrating an Existing System

The migration process from the legacy view to the agile view has a dedicated white paper entitled *HP-UX 11i v3 Persistent DSF Migration Guide*. It is described in the For More Information section.

Note: Some systems can only be partially migrated away from the use of legacy DSFs. Some applications, such as ISVs, do not yet support the agile view. Check the list of restrictions in the *HP-UX 11i v3 Persistent DSF Migration Guide*.

Migrating Kernel Software

If you are an independent software or hardware vendor (ISV/IHV) who delivers kernel components, your migration to the next generation mass storage environment might require changes to your software. For example, interface drivers must now register their controller and target devices with the SCSI stack. Class drivers must register with the SCSI stack and may claim LUNs based on their peripheral data type, vendor identifier, and product identifier. Lock management, I/O prioritization, error recovery, and asynchronous event notification have also been modified.

For more information on migrating to HP-UX 11i v3, see the following documents on the HP websites:

- Driver Development Kit
  http://www.hp.com/go/hpux_ddk
- Software Transition Kit
  http://devresource.hp.com/STK/

Conclusion

The next generation mass storage stack represents a paradigm shift in the management of disks and tapes. Agile addressing and native multi-pathing coupled with increased adaptability, scalability, and performance make the I/O subsystem in HP-UX 11i v3 easier to manage. New and expanded commands, as well as integration of other software components with the mass storage stack, present a simplified and consistent interface to mass storage. Finally, the ability to select either the legacy or the agile view retains backward compatibility with previous HP-UX releases.

Glossary

Agile Addressing

The ability to address a LUN with the same device special file regardless of the physical location of the LUN or the number of paths leading to it. The DSF for a LUN remains the same even if the LUN is moved from one HBA to another, moved from one switch or hub port to another, presented using a different target port to the host, or configured with multiple hardware paths. Also referred to as **persistent LUN binding**.

Agile View

The representation of LUNs using lunpath hardware paths, LUN hardware paths, and persistent DSFs, introduced in HP-UX 11i v3.

DSF

Device Special File. A file associated with an I/O device. DSFs are read and written the same as ordinary files, but requests to read or write result in activation of the associated device.

Hardware Path

A series of numbers representing the physical or virtualized location of a device. The path is a sequence of I/O addresses that share a hierarchical relationship. The address elements may not correspond to physical hardware addresses, and may represent only a "handle" to a device rather than a physical path to it.

HBA

Host Bus Adapter. A physical I/O interface that provides I/O processing and connectivity between a server and a storage device.

Legacy DSF

A DSF with the hardware path information such as SCSI bus, target, and LUN embedded in the file minor name and file name, such as `/dev/dsk/c#t#d#`.

Legacy Hardware Path

The representation of a hardware path as it exists in releases prior to HP-UX 11i v3. It is composed of a series of bus-nexus addresses separated by slashes (/) leading to the HBA. After the HBA, additional address elements (such as domain, area, port, target, and LUN) are separated by periods (.). The string 0/2/1/0.1.4.0.0.2.7 is an example of a legacy hardware path.

Legacy View

The representation of legacy hardware paths and legacy DSFs as in releases prior to HP-UX 11i v3.

LUN

A SCSI logical unit. This refers to an end storage device such as a disk, tape, floppy, or CD. This is the logical unit itself and does not represent the path to the logical unit.

LUN Hardware Path

A hardware path for a SCSI LUN that virtualizes all paths to the LUN. The first path element is 64000, followed by a virtual bus instance and a logical unit number. Multi-pathed LUNs have a single LUN hardware path. The string 64000/0xfa00.0x22 is an example of a LUN hardware path.

Lunpath

The physical hardware path leading to a SCSI logical unit. A SCSI LUN can have more than one lunpath.

Lunpath Hardware Path

The representation of a hardware path for a mass storage device. It is identical in format to a legacy hardware path up to the HBA. After the HBA, additional addressing is represented in hexadecimal format. The string 0/2/1/0.0x50001fe1500170ac.0x4017000000000000 is an example of a lunpath hardware path.

Multi-pathing

The detection, correlation, and coordinated usage of multiple hardware paths leading to the same LUN.

Pass-Through DSF

A DSF that enables direct access to SCSI devices. The `esctl` pass-through driver enables support of devices which are not normally supported by the `esdisk`, `estape`, and `eschgr` drivers.

Persistent DSF

A DSF conforming to the naming model introduced in HP-UX 11i v3 to support agile addressing. The device file name contains an instance number, such as `/dev/disk/disk#`, and the minor number has no hardware path information.

Persistent LUN Binding

See Agile Addressing.

WWID

Worldwide Identifier. A unique identifier for a SCSI device obtained from EVPD INQUIRY page 0x83 of `id_type` 1, 2, 3, 7, 8 and an association of 0.

Appendix A: Summary of Changes

This appendix summarizes the changes to the mass storage stack in HP-UX 11i v3. For details on command changes, see the command manpages.

New Commands

Command	Usage
`scsimgr(1M)`	Manages and troubleshoots mass storage. Syntax: `scsimgr [-fptv]` *`command`* `[`*`identifier`*`]` `[`*`keyword`*`]`… `[`*`argument`*`]`… `scsimgr -h [-d` *`driver`*`]` `[`*`command`*`]` `-h` Displays general help or command-specific help information `-f` Forces execution of commands `-p` Displays parsable output `-t` Displays terse output `-v` Displays verbose output The *`command`* can be one of the following: `get_stat` Retrieves statistics `clear_stat` Clears statistics `get_info` Retrieves status information `lun_map` Lists LUN paths of a LUN `get_attr` Gets values and description of attributes `set_attr` Changes current values of settable attributes `save_attr` Saves values of settable attributes `ddr_add` Adds settable attribute scopes for drivers `ddr_del` Removes settable attribute scopes for drivers `ddr_list` Lists settable attribute scopes for drivers `ddr_name` Creates a settable attribute scope covering a SCSI object `disable` Disables a SCSI object `enable` Enables a SCSI object `replace_wwid` Validates binding between a new LUN and LUN paths `replace_leg_dsf` Changes binding of a legacy device file to a LUN `lun_reset` Resets a LUN `warm_bdr` Performs a warm reset on a target device `cold_bdr` Performs a cold reset on a target device `set_devid` Sets device identifier for a LUN `get_devid` Retrieves device identifier for a LUN `sync_cache` Requests synchronization of a block device cache `erase` Erases blocks of an optical block device `inquiry` Performs SCSI command inquiry The valid values for *`argument`* and *`keyword`* are dependent on the value of *`command`* The *`identifier`* can be one of the following: `-d` *`driver`* `-D` *`dsf`* `-H` *`hw_path`* `-C` *`class`* `-I` *`instance`* `-N` *`attribute_scope`*

iofind(1M)	Finds references to legacy view in system files, and optionally updates them to agile view. Syntax: iofind [-h] [[-n|-H] [-f *file*] [-d *directory* [-i *filelist*]] [-R [-p] [-F]]] -h Help -n Searches for legacy DSF names -H Searches for legacy hardware paths -f *file* Searches for DSFs or hardware paths contained in the named file -d *directory* Limits recursive search to the named directory -i *filelist* Limits search to the named ASCII files -R Replaces legacy DSFs with persistent ones -p Previews changes -F Forces changes without user input
iobind(1M)	Changes driver bound to a LUN. Syntax: iobind -H *hw_path* -M *driver* [-I *instance*] [-f] -H *hw_path* LUN hardware path -M *driver* Driver name -I *instance* Instance number -f Force
io_redirect_dsf(1M)	Redirects a DSF to a new device. Syntax: io_redirect_dsf -H *old_hw_path* -N *new_hw_path* -H *old_hw_path* LUN hardware path of replaced disk -N *new_hw_path* LUN hardware path of replacement disk Alternate Syntax: io_redirect_dsf -d *old_dsf_name* -n *new_dsf_name* -d *old_dsf_name* Persistent DSF of replaced disk -n *new_dsf_name* Persistent DSF of replacement disk

Existing Commands with New Options

Command	**Usage**
ioscan(1M)	Scans an I/O system. -N Displays the agile view (shows LUN and lunpath hardware paths and persistent DSFs instead of the legacy representations) -P *property_name* Displays property *property_name* -m dsf [*dsfname*] Displays mapping of persistent DSF to legacy DSFs -m hwpath [-H *hwpath*] Displays mapping of legacy hardware paths to their lunpath hardware paths and LUN hardware paths -m lun Displays mapping of a LUN hardware path to its lunpath hardware paths -e Displays EFI device paths (on Integrity Servers) or BCH device paths (on PA Servers) -s Displays stale I/O nodes -b Initiates deferred binding -r Reverses a deferred binding -B Lists deferred bindings -U Scans all unclaimed I/O nodes

insf(1M)	Installs special (device) files.	
	-L	Enables legacy addressing by restoring legacy DSFs (if legacy view was disabled with rmsf -L)
	-Lv	Displays whether the legacy mode is enabled or disabled
lssf(1M)	Lists special files.	
	-s	Lists stale special files
	-c *dsf*	Performs a critical resource analysis on a DSF
mksf(1M)	Makes a DSF.	
	-P	Creates a pass-through persistent DSF for the esdisk, estape, and eschgr drivers
rmsf(1M)	Removes a DSF.	
	-u -H *hwpath*	Unbinds driver at a specified LUN hardware path
	-x	Removes stale entries from the I/O configuration file
	-L	Disables legacy addressing mode by removing legacy DSFs and legacy I/O configuration information
ioinit(1M)	Maintains consistency between kernel I/O data structures and I/O configuration files.	
	-rC *class*	Reassigns device instance numbers, starting from 0, for a specified device class
	-v *ioconfig_file*	Checks an I/O configuration file for corruption
	-A	Performs Critical Resource Analysis (CRA) on all the devices corresponding to the hardware paths specified in *infile*. You must use this with the -f option.
mediainit(1M)	Formats and scrubs a mass storage device.	
	-S	Scrubs the specified disk three times.
	-c *scrub_character*	Specifies the character to use for disk scrubbing. You must use this with the -S option.
	-t *scrub_count*	Specifies the number of passes for a disk scrub. You must use this with the -S option.
	special file	Specifies the character device special file of the disk to be scrubbed.

Deprecated and Obsolete Tunable Kernel Parameters

Obsolete Tunable	**Replacement**	**Description**
scsi_max_qdepth	max_q_depth scsimgr attribute	Maximum number of I/O operations that a target queues up for execution.
scsi_max_phys	escsi_maxphys scsimgr attribute	Maximum data size the SCSI subsystem accepts for an I/O request.
default_disk_ir	immediate_report scsictl mode parameter or esdisk disable_flags parameter	Enables or disables the use of a device write cache in the SCSI subsystem.

New Device Tunables

Tunable	**Usage**
leg_mpath_enable	Enables multi-pathing through legacy DSFs.
infinite_retries_enable	Forces the I/O system to infinitely retry certain I/O request failures.

`max_retries`	Configures the maximum number of times to retry a failing I/O request, subject to finite retry policy.

New Mass Storage DSF Naming Conventions

Persistent DSF name	**Legacy DSF name**	**Description**
/dev/disk/disk#	/dev/dsk/c#t#d#	The entire disk (block access)
/dev/rdisk/disk#	/dev/rdsk/c#t#d#	The entire disk (raw access)
/dev/disk/disk#_p#	/dev/dsk/c#t#d#s#	Partition on the disk (block access)
/dev/rdisk/disk#_p#	/dev/rdsk/c#t#d#s#	Partition on the disk (raw access)
/dev/rtape/tape#*options*	/dev/rmt/c#t#d#*options*	Tape device (raw access)
/dev/rchgr/autoch#	/dev/rac/c#t#d#_*options*	Autochanger device (raw access)
/dev/pt/pt*instance*	/dev/rscsi/c#t#d#	Pass-through device

Appendix B: Using *ioscan*

Note: The output from the following examples are trimmed for readability.

To display the agile view, use `ioscan` with the `-N` option. For example:

```
# ioscan -N -fn
Class     I H/W Path       Driver   S/W State H/W Type  Description
===============================================================================
root      0                root       CLAIMED BUS_NEXUS
ioa       0 0              sba        CLAIMED BUS_NEXUS System Bus Adapter (1229)
ba        0 0/0            lba        CLAIMED BUS_NEXUS Local PCI-X Bus Adapter
(122e)
OO        0 0/0/1/0        UsbOhci    CLAIMED INTERFACE USB OHCI Interface
OO        1 0/0/1/1        UsbOhci    CLAIMED INTERFACE USB OHCI Interface
OO        2 0/0/1/2        UsbEhci    CLAIMED INTERFACE USB EHCI Interface
sideba    0 0/0/2/0        side_multi CLAIMED INTERFACE CMD IDE controller
ext_bus   0 0/0/2/0.0      side       CLAIMED INTERFACE IDE Primary Channel
                             /dev/side0
tgtpath   0 0/0/2/0.0.0x0  estp       CLAIMED TGT_PATH  ide target served by side
driver
lunpath  50 0/0/2/0.0.0x0.0x0 eslpt   CLAIMED LUN_PATH  LUN path for disk60
ext_bus   1 0/0/2/0.1      side       CLAIMED INTERFACE IDE Secondary Channel
lan       0 0/0/3/0        intl100    CLAIMED INTERFACE Intel PCI Pro 10/100Tx
Server Adapter
ba        1 0/1            lba        CLAIMED BUS_NEXUS Local PCI-X Bus Adapter
(122e)
ext_bus   2 0/1/1/0        mpt        CLAIMED INTERFACE SCSI Ultra320
                             /dev/mpt2
tgtpath   4 0/1/1/0.0x0    estp       CLAIMED TGT_PATH  parallel_scsi target
served by mpt driver
lunpath   1 0/1/1/0.0x0.0x0 eslpt     CLAIMED LUN_PATH  LUN path for disk48
tgtpath   3 0/1/1/0.0x1    estp       CLAIMED TGT_PATH  parallel_scsi target
served by mpt driver
lunpath   0 0/1/1/0.0x1.0x0 eslpt     CLAIMED LUN_PATH  LUN path for disk47
tgtpath   1 0/1/1/0.0x7    estp       CLAIMED TGT_PATH  parallel_scsi target
served by mpt driver
ext_bus   3 0/1/1/1        mpt        CLAIMED INTERFACE SCSI Ultra320
                             /dev/mpt3
tgtpath   2 0/1/1/1.0x7    estp       CLAIMED TGT_PATH  parallel_scsi target
served by mpt driver
lan       1 0/1/2/0        igelan     CLAIMED INTERFACE HP PCI 1000Base-T Core
ba        2 0/2            lba        CLAIMED BUS_NEXUS Local PCI-X Bus Adapter
(122e)
slot      0 0/2/1          pci_slot   CLAIMED SLOT      PCI Slot
fc        0 0/2/1/0        td         CLAIMED INTERFACE HP Tachyon XL2 Fibre
Channel Mass Storage Adapter
                             /dev/td0
tgtpath   6 0/2/1/0.0x500805f300083891
                           estp       NO_HW   TGT_PATH  fibre_channel target
served by td driver
lunpath   3 0/2/1/0.0x500805f300083891.0x0
                           eslpt      NO_HW   LUN_PATH  LUN path for ctl4
lunpath  25 0/2/1/0.0x500805f300083891.0x4007000000000000
                           eslpt      NO_HW   LUN_PATH  LUN path for disk59
lunpath  24 0/2/1/0.0x500805f300083891.0x4008000000000000
                           eslpt      NO_HW   LUN_PATH  LUN path for disk58
lunpath  23 0/2/1/0.0x500805f300083891.0x4009000000000000
                           eslpt      NO_HW   LUN_PATH  LUN path for disk57
lunpath   9 0/2/1/0.0x500805f300083891.0x4010000000000000
                           eslpt      NO_HW   LUN_PATH  LUN path for disk50
```

```
lunpath   6 0/2/1/0.0x500805f300083891.0x4011000000000000
                           eslpt      NO_HW   LUN_PATH  LUN path for disk49
tgtpath   5 0/2/1/0.0x500805f300083899
                           estp       CLAIMED TGT_PATH  fibre_channel target
served by td driver
lunpath   2 0/2/1/0.0x500805f300083899.0x0
                           eslpt      NO_HW   LUN_PATH  LUN path for ctl4
lunpath  22 0/2/1/0.0x500805f300083899.0x4007000000000000
                           eslpt      NO_HW   LUN_PATH  LUN path for disk59
lunpath  21 0/2/1/0.0x500805f300083899.0x4008000000000000
                           eslpt      NO_HW   LUN_PATH  LUN path for disk58
lunpath  19 0/2/1/0.0x500805f300083899.0x4009000000000000
                           eslpt      NO_HW   LUN_PATH  LUN path for disk57
lunpath   4 0/2/1/0.0x500805f300083899.0x4011000000000000
                           eslpt      NO_HW   LUN_PATH  LUN path for disk49
ba        3 0/3            lba        CLAIMED BUS_NEXUS Local PCI-X Bus Adapter
(122e)
slot      1 0/3/1          pci_slot   CLAIMED SLOT      PCI Slot
lan       2 0/3/1/0        igelan     CLAIMED INTERFACE HP A6825-60101 PCI
1000Base-T Adapter
ba        4 0/4            lba        CLAIMED BUS_NEXUS Local PCI-X Bus Adapter
(122e)
slot      2 0/4/1          pci_slot   CLAIMED SLOT      PCI Slot
fc        1 0/4/1/0        td         CLAIMED INTERFACE HP Tachyon XL2 Fibre
Channel Mass Storage Adapter
                             /dev/td1
tgtpath   8 0/4/1/0.0x500805f300083891
                           estp       NO_HW   TGT_PATH  fibre_channel target
served by td driver
lunpath  27 0/4/1/0.0x500805f300083891.0x0
                           eslpt      NO_HW   LUN_PATH  LUN path for ctl4
lunpath  49 0/4/1/0.0x500805f300083891.0x4007000000000000
                           eslpt      NO_HW   LUN_PATH  LUN path for disk59
lunpath  48 0/4/1/0.0x500805f300083891.0x4008000000000000
                           eslpt      NO_HW   LUN_PATH  LUN path for disk58
lunpath  46 0/4/1/0.0x500805f300083891.0x4009000000000000
                           eslpt      NO_HW   LUN_PATH  LUN path for disk57
lunpath  33 0/4/1/0.0x500805f300083891.0x4010000000000000
                           eslpt      NO_HW   LUN_PATH  LUN path for disk50
lunpath  30 0/4/1/0.0x500805f300083891.0x4011000000000000
                           eslpt      NO_HW   LUN_PATH  LUN path for disk49
tgtpath   7 0/4/1/0.0x500805f300083899
                           estp       CLAIMED TGT_PATH  fibre_channel target
served by td driver
lunpath  26 0/4/1/0.0x500805f300083899.0x0
                           eslpt      NO_HW   LUN_PATH  LUN path for ctl4
lunpath  47 0/4/1/0.0x500805f300083899.0x4007000000000000
                           eslpt      NO_HW   LUN_PATH  LUN path for disk59
lunpath  45 0/4/1/0.0x500805f300083899.0x4008000000000000
                           eslpt      NO_HW   LUN_PATH  LUN path for disk58
lunpath  43 0/4/1/0.0x500805f300083899.0x4009000000000000
                           eslpt      NO_HW   LUN_PATH  LUN path for disk57
lunpath  28 0/4/1/0.0x500805f300083899.0x4010000000000000
                           eslpt      NO_HW   LUN_PATH  LUN path for disk50
lunpath  29 0/4/1/0.0x500805f300083899.0x4011000000000000
                           eslpt      NO_HW   LUN_PATH  LUN path for disk49
ba        5 0/5            lba        CLAIMED BUS_NEXUS Local PCI-X Bus Adapter
(122e)
slot      3 0/5/1          pci_slot   CLAIMED SLOT      PCI Slot
ba        6 0/6            lba        CLAIMED BUS_NEXUS Local PCI-X Bus Adapter
(122e)
tty       0 0/6/1/0        asio0      CLAIMED INTERFACE PCI SimpleComm (103c1290)
```

```
                             /dev/cul0p0     /dev/mux0       /dev/ttyd0p0
                             /dev/diag/mux0  /dev/tty0p0
tty       1 0/6/1/1        asio0      CLAIMED INTERFACE PCI Serial (103c1048)
                             /dev/MPdiag1    /dev/mux1       /dev/tty1p4
                             /dev/cul1p0     /dev/tty1p0     /dev/ttyd1p0
                             /dev/diag/mux1  /dev/tty1p2
graphics  0 0/6/2/0        gvid_core  CLAIMED INTERFACE PCI Display (10025159)
processor 0 120            processor  CLAIMED PROCESSOR Processor
processor 1 121            processor  CLAIMED PROCESSOR Processor
ba        7 250            pdh        CLAIMED BUS_NEXUS Core I/O Adapter
ipmi      0 250/0          ipmi       CLAIMED INTERFACE IPMI Controller
                             /dev/ipmi
tty       2 250/1          asio0      CLAIMED INTERFACE Built-in RS232C
                             /dev/CCITT      /dev/diag/mux2  /dev/ttyd2p0
                             /dev/c2p0_lp    /dev/mux2
                             /dev/cua2p0     /dev/tty2p0
tty       3 250/2          asio0      CLAIMED INTERFACE Built-in RS232C
                             /dev/diag/mux3  /dev/mux3       /dev/tty3p0
acpi_node 0 250/3          acpi_node  CLAIMED INTERFACE Acpi Hardware
esvroot   0 64000/0xfa00   esvroot    CLAIMED VIRTBUS   Escsi virtual root
disk     47 64000/0xfa00/0x0 esdisk   CLAIMED DEVICE    HP 36.4GST336753LC
                             /dev/disk/disk47      /dev/rdisk/disk47
                             /dev/disk/disk47_p1   /dev/rdisk/disk47_p1
                             /dev/disk/disk47_p2   /dev/rdisk/disk47_p2
                             /dev/disk/disk47_p3   /dev/rdisk/disk47_p3
disk     48 64000/0xfa00/0x1 esdisk    CLAIMED DEVICE    HP 36.4GST336753LC
                             /dev/disk/disk48      /dev/rdisk/disk48
                             /dev/disk/disk48_p1   /dev/rdisk/disk48_p1
                             /dev/disk/disk48_p2   /dev/rdisk/disk48_p2
                             /dev/disk/disk48_p3   /dev/rdisk/disk48_p3
ctl       4 64000/0xfa00/0x2 esctl    NO_HW   DEVICE    HP      MSA CONTROLLER
                             /dev/pt/pt4      /dev/pt/pt_ctl4
disk     49 64000/0xfa00/0x3 esdisk   NO_HW   DEVICE    HP      MSA VOLUME
                             /dev/disk/disk49   /dev/rdisk/disk49
disk     50 64000/0xfa00/0x4 esdisk   NO_HW   DEVICE    HP      MSA VOLUME
                             /dev/disk/disk50   /dev/rdisk/disk50
disk     57 64000/0xfa00/0xb esdisk   NO_HW   DEVICE    HP      MSA VOLUME
                             /dev/disk/disk57   /dev/rdisk/disk57
disk     58 64000/0xfa00/0xc esdisk   NO_HW   DEVICE    HP      MSA VOLUME
                             /dev/disk/disk58   /dev/rdisk/disk58
disk     59 64000/0xfa00/0xd esdisk   NO_HW   DEVICE    HP      MSA VOLUME
                             /dev/disk/disk59   /dev/rdisk/disk59
disk     60 64000/0xfa00/0xe esdisk   CLAIMED DEVICE    TEAC    DV-28E-B
                             /dev/disk/disk60   /dev/rdisk/disk60
```

To show the mapping of LUN hardware paths to lunpath hardware paths, use `ioscan` with the `-m lun` option. For example:

```
# ioscan -m lun
Class I Lun H/W Path     Driver  S/W State H/W Type  Health   Description
======================================================================
disk 47 64000/0xfa00/0x0 esdisk  CLAIMED   DEVICE    online   HP 36.4GST336753LC
             0/1/1/0.0x1.0x0
                      /dev/disk/disk47      /dev/rdisk/disk47
                      /dev/disk/disk47_p1   /dev/rdisk/disk47_p1
                      /dev/disk/disk47_p2   /dev/rdisk/disk47_p2
                      /dev/disk/disk47_p3   /dev/rdisk/disk47_p3
disk 48 64000/0xfa00/0x1 esdisk  CLAIMED   DEVICE    online   HP 36.4GST336753LC
             0/1/1/0.0x0.0x0
                      /dev/disk/disk48      /dev/rdisk/disk48
                      /dev/disk/disk48_p1   /dev/rdisk/disk48_p1
                      /dev/disk/disk48_p2   /dev/rdisk/disk48_p2
                      /dev/disk/disk48_p3   /dev/rdisk/disk48_p3
ctl   4 64000/0xfa00/0x2 esctl   NO_HW     DEVICE    online   HP   MSA CONTROLLER
             0/2/1/0.0x500805f300083899.0x0
             0/2/1/0.0x500805f300083891.0x0
             0/4/1/0.0x500805f300083899.0x0
             0/4/1/0.0x500805f300083891.0x0
                      /dev/pt/pt4      /dev/pt/pt_ctl4
disk 49 64000/0xfa00/0x3 esdisk  NO_HW     DEVICE    offline  HP   MSA VOLUME
             0/2/1/0.0x500805f300083899.0x4011000000000000
             0/2/1/0.0x500805f300083891.0x4011000000000000
             0/4/1/0.0x500805f300083899.0x4011000000000000
             0/4/1/0.0x500805f300083891.0x4011000000000000
                      /dev/disk/disk49   /dev/rdisk/disk49
disk 50 64000/0xfa00/0x4 esdisk  NO_HW     DEVICE    offline  HP   MSA VOLUME
             0/2/1/0.0x500805f300083899.0x4010000000000000
             0/2/1/0.0x500805f300083891.0x4010000000000000
             0/4/1/0.0x500805f300083899.0x4010000000000000
             0/4/1/0.0x500805f300083891.0x4010000000000000
                      /dev/disk/disk50   /dev/rdisk/disk50
disk 57 64000/0xfa00/0xb esdisk  NO_HW     DEVICE    offline  HP   MSA VOLUME
             0/2/1/0.0x500805f300083899.0x4009000000000000
             0/2/1/0.0x500805f300083891.0x4009000000000000
             0/4/1/0.0x500805f300083899.0x4009000000000000
             0/4/1/0.0x500805f300083891.0x4009000000000000
                      /dev/disk/disk57   /dev/rdisk/disk57
disk 58 64000/0xfa00/0xc esdisk  NO_HW     DEVICE    offline  HP   MSA VOLUME
             0/2/1/0.0x500805f300083899.0x4008000000000000
             0/2/1/0.0x500805f300083891.0x4008000000000000
             0/4/1/0.0x500805f300083891.0x4008000000000000
             0/4/1/0.0x500805f300083899.0x4008000000000000
                      /dev/disk/disk58   /dev/rdisk/disk58
disk 59 64000/0xfa00/0xd esdisk  NO_HW     DEVICE    offline  HP   MSA VOLUME
             0/2/1/0.0x500805f300083899.0x4007000000000000
             0/2/1/0.0x500805f300083891.0x4007000000000000
             0/4/1/0.0x500805f300083899.0x4007000000000000
             0/4/1/0.0x500805f300083891.0x4007000000000000
                      /dev/disk/disk59   /dev/rdisk/disk59
disk 60 64000/0xfa00/0xe esdisk  CLAIMED   DEVICE    online   TEAC DV-28E-B
             0/0/2/0.0.0x0.0x0
                      /dev/disk/disk60   /dev/rdisk/disk60
```

To show the mapping of a persistent DSF to its equivalent legacy paths, and vice versa, use `ioscan` with the `-m dsf` option. For example:

```
# ioscan -m dsf /dev/disk/disk43
Persistent DSF           Legacy DSF(s)
========================================
/dev/disk/disk43         /dev/dsk/c9t0d1
                         /dev/dsk/c11t0d1
                         /dev/dsk/c7t0d1
                         /dev/dsk/c5t0d1
                         /dev/dsk/c13t0d1
                         /dev/dsk/c17t0d1
                         /dev/dsk/c15t0d1
                         /dev/dsk/c19t0d1

# ioscan -m dsf /dev/dsk/c19t0d1
Persistent DSF           Legacy DSF(s)
========================================
/dev/disk/disk43         /dev/dsk/c19t0d1
```

To show the mapping of a LUN hardware path to its equivalent lunpath and legacy hardware paths, and vice versa, use `ioscan` with the `-m hwpath` option. For example:

```
# ioscan -m hwpath -H 64000/0xfa00/0x1
Lun H/W Path   Lunpath H/W Path                              Legacy H/W Path
====================================================================
64000/0xfa00/0x1
               0/4/1/1.0x50001fe15008a738.0x4001000000000000 0/4/1/1.124.2.0.0.0.1
               0/4/1/1.0x50001fe15008a739.0x4001000000000000 0/4/1/1.124.3.0.0.0.1
               0/4/1/1.0x50001fe15008a73d.0x4001000000000000 0/4/1/1.124.5.0.0.0.1
               0/4/1/1.0x50001fe15008a73c.0x4001000000000000 0/4/1/1.124.4.0.0.0.1
               0/4/1/0.0x50001fe15008a73c.0x4001000000000000 0/4/1/0.124.4.0.0.0.1
               0/4/1/0.0x50001fe15008a73d.0x4001000000000000 0/4/1/0.124.5.0.0.0.1

# ioscan -m hwpath -H 0/4/1/1.124.2.0.0.0.1
Lun H/W Path   Lunpath H/W Path                              Legacy H/W Path
====================================================================
64000/0xfa00/0x1
               0/4/1/1.0x50001fe15008a738.0x4001000000000000 0/4/1/1.124.2.0.0.0.1

# ioscan -m hwpath -H 0/4/1/1.0x50001fe15008a738.0x4001000000000000
Lun H/W Path   Lunpath H/W Path                              Legacy H/W Path
====================================================================
64000/0xfa00/0x1
               0/4/1/1.0x50001fe15008a738.0x4001000000000000 0/4/1/1.124.2.0.0.0.1
```

To show the `health` property of all disks, use `ioscan` with the `-P` option. For example:

```
# ioscan -P health -C disk
Class      I  H/W Path          health
=====================================
disk      43  64000/0xfa00/0x1  online
disk      44  64000/0xfa00/0x2  online
disk      45  64000/0xfa00/0x3  online
disk      46  64000/0xfa00/0x4  online
disk      47  64000/0xfa00/0x5  online
disk      48  64000/0xfa00/0x6  online
disk      49  64000/0xfa00/0x7  online
disk      50  64000/0xfa00/0x8  online
```

Appendix C: Using *scsimgr*

This appendix presents the syntax of the `scsimgr` command, a partial list of available operations, and some use cases. It is not intended to be a complete description of the `scsimgr` command. For information on the `scsimgr` command capabilities, see *scsimgr(1M)* and the white paper described in For More Information.

Introduction to *scsimgr*

The `scsimgr` command manages and troubleshoots SCSI objects and the mass storage subsystem. The command syntax is as follows:

`scsimgr [-fpv] *command* [-d *driver*] [*identifier*] [*keyword*]... [*argument*]...`

`-f` Forces the execution of commands without user input. Without `-f`, `scsimgr` prompts for confirmation before executing a disruptive or destructive operation.

`-p` Displays parsable output. The `scsimgr` output is condensed to one line with output fields separated by a colon (:).

`-v` Displays verbose output.

`command`
: Selects the operation to perform.

`-d` *`driver`*
: Specifies the name of the driver for driver-specific commands.

`identifier`
: Specifies the SCSI object on which the `command` applies. The *`identifier`* can be one of the following:

`-D` *`dsf`* The character DSF for a LUN or SCSI controller. For a SCSI controller, *`dsf`* must be a persistent DSF.

`-H` *`hw_path`* The hardware path of a LUN or SCSI controller. This can be either a lunpath hardware path, a LUN hardware path, or a target path.

`-C` *`class`* `-I` *`instance`*
: The device class and instance number of a LUN, LUN path, target path, or SCSI controller.

`-N` *`scope`* A scope of objects. This selects a subset of mass storage objects on the system and applies only to the commands affecting attributes (`get_attr`, `set_attr`, `ddr_add`, and `ddr_del`). The *`scope`* argument has the following form:

`/escsi/`*`driver`*`[/`*`pdt`*`[/`*`vid`*`[/`*`pid`*`[/`*`rev`*`]]]]`

Where

`driver` is the name of a driver, such as `esdisk`.
`pdt` is the peripheral device type in hexadecimal, as returned by a SCSI inquiry.
`vid` is the vendor identifier as returned by a SCSI inquiry, such as `"HP      "`.
`pid` is the product identifier as returned by a SCSI inquiry, such as `"SDLT600  "`.
`rev` is the product revision as returned by a SCSI inquiry, such as `"HP06"`.

This construct allows attributes to be set or retrieved for a set of objects. For example, a *scope* of "/escsi/esdisk" refers to all disks on a system and a *scope* of "/escsi/esdisk/0x0/HP   " refers to all HP disks on a system.

Keyword Provides additional scoping information, such as selecting all lunpaths for a LUN.

Argument Provides additional information specific to the command. For example, when using the get_attr command, *argument* specifies which attribute to print.

Note: This is not the complete command syntax. For the complete set of commands, keywords, and arguments, see *scsimgr(1M)*.

Retrieving Status of SCSI Objects

To retrieve general status of a SCSI object, use the scsimgr get_info command.

To retrieve the status of a controller, use the target as the identifier as follows:

```
# scsimgr get_info -H 0/2/1/0.0x500805f300083899

        STATUS INFORMATION FOR TARGET PATH : 0/2/1/0.0x500805f300083899

Generic Status Information

SCSI services internal state                      = IDLE
Port id                                           = 0x10700
Protocol                                          = fibre_channel
Protocol revision                                 = 4.3
Port name                                         = 0x500805f300083899
Node name                                         = 0x500805f300083890
LUN paths registered (active/inactive)            = 12
```

To retrieve the status of a lunpath, use the lunpath hardware path as the identifier as follows:

```
# scsimgr get_info -H 0/2/1/0.0x500805f300083899.0x0

        STATUS INFORMATION FOR LUN PATH : 0/2/1/0.0x500805f300083899.0x0

Generic Status Information

SCSI services internal state                      = UNOPEN
Open close state                                  = ACTIVE
Protocol                                          = fibre_channel
EVPD page 0x83 description code                   = 1
EVPD page 0x83 description association            = 0
EVPD page 0x83 description type                   = 3
World Wide Identifier (WWID)                      =
Outstanding I/Os                                  = 0
Maximum I/O timeout in seconds                    = 30
Maximum I/O size allowed                          = 2097152
Maximum number of active I/Os allowed             = 1
Current active I/Os                               = 0
Maximum queue depth                               = 1
Queue full delay count                            = 0
```

To retrieve the status of a LUN, use the LUN hardware path or DSF as the identifier as follows:

```
# scsimgr get_info -H 64000/0xfa00/0x3
```

```
        STATUS INFORMATION FOR LUN : 64000/0xfa00/0x3

Generic Status Information

SCSI services internal state                       = UNOPEN
Device type                                        = Direct_Access
EVPD page 0x83 description code                    = 1
EVPD page 0x83 description association             = 0
EVPD page 0x83 description type                    = 3
World Wide Identifier (WWID)                       = 0x600805f30008389000000000bd6700
af
Serial number                                      = P56350D9IOT0B0
Vendor id                                          = HP
Product id                                         = MSA VOLUME
Product revision                                   = 6.32
Other properties                                   =
SPC protocol revision                              = 2
Open count (includes chr/blk/pass-thru/class)      = 0
Raw open count (includes class/pass-thru)          = 0
Pass-thru opens                                    = 0
LUN path count                                     = 4
Active LUN paths                                   = 4
Standby LUN paths                                  = 0
Failed LUN paths                                   = 0
Maximum I/O size allowed                           = 2097152
Preferred I/O size                                 = 2097152
Outstanding I/Os                                   = 0
I/O load balance policy                            = round_robin
Path fail threshold time period                    = 0
Transient time period                              = 60
Tracing buffer size                                = 1024
LUN Path used when policy is path_lockdown         = NA

Driver esdisk Status Information :

Capacity in number of blocks                           = 18876375
Block size in bytes                                    = 512
Number of active IOs                                   = 0
Special properties                                     =
Maximum number of IO retries                           = 45
IO transfer timeout in secs                            = 30
FORMAT command timeout in secs                         = 86400
START UNIT command timeout in secs                     = 60
Timeout in secs before starting failing IO             = 30
IO infinite retries                                    = false
```

Retrieving Target or LUN Statistics

To display statistics collected for a SCSI object, use the `scsimgr get_stat` command.

To display statistics for a SCSI controller, use the target as the identifier as follows:

`# scsimgr get_stat -H 0/2/1/0.0x500805f300083899`

```
        SCSI STATISTICS FOR TARGET PATH : 0/2/1/0.0x500805f300083899

Generic Statistics:

CB_SCAN_ALL events received                          =
Target Probe events received                         = 2
Probe failures due to LUN 0 probe failures           = 0
Probe failures due to REPORT LUNS failures           = 0
```

```
LUN path probe failures                           = 0
Target path offline events from I/F driver        = 0
Target path online events from I/F driver         = 0
Port id change events from I/F driver             = 0
Target Warm Reset events                          = 0
Target Cold Reset events                          = 0
Target Warm reset failures                        = 0
Target Cold Reset failures                        = 0
Invalid port id changes                           = 0
Total I/Os processed                              = 533
Last time cleared                                 = N/A

I/F Common Statistics:

Offline events                                     = 0
WARM/COLD target resets                            = 0
Time of last WARM/COLD target reset                = N/A
Bytes read                                         = 30486
Bytes written                                      = 1200
Outstanding I/Os                                   = 0
```

To display statistics for a LUN, specify the LUN hardware path or DSF as follows:

```
# scsimgr get_stat -D /dev/rdisk/disk49

        STATISTICS FOR LUN :/dev/rdisk/disk49

Generic Statistics:

Overall attempted opens                           = 10
Overall successful  opens                         = 10
Attempted Pass-thru opens                         = 0
Successful  Pass-thru opens                       = 0
Overall closes                                    = 10
Pass-thru closes                                  = 0
Offlines                                          = 0
Onlines                                           = 0
LUN path initializations                          = 4
Class driver open failures                        = 0
Pass-thru driver open failures                    = 0
Open failures due to invalid major number         = 0
Open failures due to LUN being suspended          = 0
Open failures due to partial open of the LUN      = 0
Exclusive mode open failures                      = 0
Resume failures due to LUN not being suspended    = 0
Maximum allowed IO size changes                   = 0
Disabled or Unsupported task management requests = 0
Task management failures due to 0 LUN path count = 0
Invalid task management requests                  = 0
I/Os flushed                                      = 0
Last time cleared                                 = N/A


I/O transfer Statistics:

Bytes read                                        = 19890
Bytes written                                     = 10
Total I/Os processed                              = 392
I/O failures                                      = 0
Retried I/Os                                      = 1
Retried I/O failures                              = 0
I/O failures due to invalid IO size               = 0
```

```
I/Os flushed                                             = 0
Check condition status                                   = 41
Busy status                                              = 0
Queue full status                                        = 0
Reservation conflicts                                    = 0
Invalid Request status                                   = 0
Select Timeout status                                    = 0
Incomplete status                                        = 0
No Resource status                                       = 0
Target Path Offline status                               = 0
IO Timeout status                                        = 0
IO Aborted status                                        = 0
Reset Occurred status                                    = 0
Unrecognized CDB status                                  = 0
Bad Sense data status                                    = 0
Deferred errors                                          = 0
Recovered errors                                         = 0
NOT READY sense status                                   = 0
Medium errors                                            = 0
Hardware errors                                          = 0
Illegal request sense status                             = 40
Unit Attentions                                          = 1
Data protect sense status                                = 0
Blank checks                                             = 0
Vendor specific sense status                             = 0
Copy aborted sense status                                = 0
Command aborted sense status                             = 0
Volume overflow                                          = 0
Asynchronous IO read failures                            = 0
Asynchronous IO write failures                           = 0

Driver esdisk Statistics :

PR requests                                               = 0
Activation requests received                              = 0
Abort requests received                                   = 0
Disable requests received                                 = 0
Enable requests received                                  = 0
LUN path addition requests                                = 4
LUN path deletion requests                                = 0
Persistent Registration failures                          = 0
LUN path offlines                                         = 0
All LUN paths offlines                                    = 0
LUN path back online                                      = 0
Capacity increases                                        = 0
Capacity reductions                                       = 0
Block size changes                                        = 0
IO failures due to misalignment or boundary               = 0
Unexpected media changes                                  = 0
Last time  cleared                                        = N/A
```

Mapping Multi-Pathed LUNs

To retrieve information about all the lunpaths for a LUN, use the `scsimgr lun_map` command as follows:

```
# scsimgr lun_map -D /dev/rdisk/disk49

        LUN PATH INFORMATION FOR LUN : /dev/rdisk/disk49

Total number of LUN paths     = 4
World Wide Identifier(WWID)     = 0x600805f3000838900000000bd6700af
```

```
LUN path : lunpath4
Class                          = lunpath
Instance                       = 4
Hardware path                  = 0/2/1/0.0x500805f300083899.0x4010000000000000
SCSI transport protocol        = fibre_channel
State                          = UNOPEN
Last Open or Close state       = ACTIVE

LUN path : lunpath7
Class                          = lunpath
Instance                       = 7
Hardware path                  = 0/2/1/0.0x500805f300083891.0x4010000000000000
SCSI transport protocol        = fibre_channel
State                          = UNOPEN
Last Open or Close state       = ACTIVE

LUN path : lunpath28
Class                          = lunpath
Instance                       = 28
Hardware path                  = 0/4/1/0.0x500805f300083899.0x4010000000000000
SCSI transport protocol        = fibre_channel
State                          = UNOPEN
Last Open or Close state       = ACTIVE

LUN path : lunpath34
Class                          = lunpath
Instance                       = 34
Hardware path                  = 0/4/1/0.0x500805f300083891.0x4010000000000000
SCSI transport protocol        = fibre_channel
State                          = UNOPEN
Last Open or Close state       = ACTIVE
```

Obtaining WWID, Serial Number, and LUN id of a LUN

The WWID, serial number, and LUN id of a LUN are attributes. To display the serial number and WorldWide ID of a LUN, use `scsimgr get_attr` as follows:

```
# scsimgr get_attr -D /dev/rdisk/disk49 -a wwid –a serial_number

        SCSI ATTRIBUTES FOR LUN : /dev/rdisk/disk49

name = wwid
current = 0x600805f3000838900000000bd6700af
default =
saved =

name = serial_number
current = P56350D9IOS0KR
default =
saved =
```

To display the information in a format that can be parsed by a script, use the `–p` option as follows:

```
# scsimgr -p get_attr -D /dev/rdisk/disk49 -a wwid -a serial_number
0x600805f3000838900000000bd6700af:P56350D9IOS0KR
```

The LUN id is associated with the lunpath. To display the LUN id, use `scsimgr get_attr` with the lunpath hardware path as the identifier, as follows:

```
# scsimgr get_attr –H 0/2/1/0.0x500805f300083899.0x4011000000000000 -a lunid
```

```
    SCSI ATTRIBUTES FOR LUN PATH : 0/2/1/0.0x500805f300083899.0x4011000000000000

name = lunid
current =0x4011000000000000 (LUN # 17, Flat Space Addressing)
default =
saved =
```

Selecting Load Balancing Algorithms

By setting the `load_bal_policy` attribute, you can select one of the following load balancing algorithms:

`least_cmd_load`	Directs I/O requests through the hardware path with the least outstanding I/O requests.
`round_robin`	Cycles I/O requests through the available hardware paths in round robin fashion.
`cl_round_robin`	Cycles I/O requests through the available hardware paths in round robin fashion, within the locality of CPU on which the I/O was initiated. This is known as cell-aware round robin.
`weighted_rr`	Cycles I/O requests through the available hardware paths in round robin fashion, subject to a user-assigned weight for each path. A hardware path is used until the number of I/O requests matches its "weight", and then another path is chosen.
`preferred_path`	Directs I/O requests to the specified hardware path, if available.
`pref_tport`	Directs I/O requests to the specified target port, if available.
`closest_path`	Directs I/O requests through the hardware path with the lowest memory access latency, based on its affinity with the CPU processing the I/O operation.
`path_lockdown`	Directs I/O requests through a single hardware path. The chosen hardware path is the one that is least loaded when the device is opened. This algorithm is the only one supported for serial devices such as tape drives, changers, and SCSI controller devices, and does not apply to block devices such as disks.

To retrieve the load balancing policy for a LUN, enter the following command:

```
# scsimgr get_attr -D /dev/rdisk/disk49 -a load_bal_policy

        SCSI ATTRIBUTES FOR LUN : /dev/rdisk/disk49

name = load_bal_policy
current = least_cmd_load
default = least_cmd_load
saved =
```

To choose the `least_cmd_load` policy for all disks, set the scope for all SCSI disks using the following command:

```
# scsimgr set_attr -N /esci/esdisk -a load_bal_policy=least_cmd_load
Value of attribute load_bal_policy set successfully
```

To choose the `round_robin` policy for all HP disks, set the scope for all SCSI disks with a vendor identifier of **"HP   "** using the following command:

```
# scsimgr set_attr -N "/escsi/esdisk/0x0/HP      " -a load_bal_policy=round_robin
Value of attribute load_bal_policy set successfully
```

To choose the `round_robin` policy for a single LUN, use the following command:

```
# scsimgr set_attr -D /dev/rdisk/disk49 -a load_bal_policy=round_robin
Value of attribute load_bal_policy set successfully
```

Assigning Aliases to LUNs

There are two methods of assigning a user-friendly name to a LUN.

The first method is to set the `alias` attribute for a LUN. To assign the name `fast_disk` to a LUN, use the following command:

```
# scsimgr set_attr –H 64000/0xfa00/0x3 -a alias=fast_disk
Value of attribute alias set successfully
```

To retrieve the `alias` attribute, use either of the following commands:

```
# scsimgr -p get_attr -H 64000/0xfa00/0x3 -a alias
fast_disk
# scsimgr -p get_attr -D /dev/rdisk/disk49 -a alias
fast_disk
```

The second method is to set the device identifier on the LUN. The device identifier is stored *on the device*, so it remains available if the disk is moved to a different system or connected to multiple systems.

To set the device identifier, use the `scsimgr set_devid` command as follows:

```
# scsimgr set_devid -D /dev/rdisk/disk49 my_disk
Do you really want to set device id? (y/n)? y
scsimgr: Device Identifier successfully set
```

To retrieve the device identifier, use the `scsimgr get_devid` command with either the DSF or LUN hardware path as follows:

```
# scsimgr get_devid -D /dev/rdisk/disk49

 Device Identifier for /dev/rdisk/disk49  = my_disk
# scsimgr get_devid -H 64000/0xfa00/0x3

 Device Identifier for /dev/rdisk/disk49  = my_disk
```

Appendix D: Interpreting Lunpath Hardware Paths

This appendix describes how to interpret address elements in lunpath hardware paths. It also shows how to map between the legacy hardware path and lunpath hardware path formats.

Components of Lunpath Hardware Paths

The lunpath hardware path has three parts: the HBA path, the target address, and the LUN address.

- The **HBA path** is composed of a series of bus-nexus addresses separated by a slash (/). It is identical to the HBA portion of a legacy hardware path.
- The **target address** is transport-specific. If the SCSI transport supports port naming, the target address is a port name. Otherwise, the target address is a port identifier. When printed by I/O commands like `ioscan`, the target address is separated from the HBA path by a period (.) and is printed in hexadecimal notation.
- The **LUN address** is a SCSI 64-bit LUN identifier, built from the address method and the LUN number. When printed by I/O commands, it is separated from the target address by a period (.) and is printed in hexadecimal notation.

The following table shows four examples:

Lunpath Hardware path	HBA path	Target address	LUN address
0/1/1/0.0x1.0x0	0/1/1/0	0x1	0x0
2/0/1.0x5.0x0	2/0/1	0x5	0x0
0/5/1/0.0x50060e8004276e12.0x4000000000000000	0/5/1/0	0x50060e8004276e12	0x4000000000000000
0/4/1/0.0x500805f300083899.0x4010000000000000	0/4/1/0	0x500805f300083899	0x4010000000000000

Parallel SCSI

Because parallel SCSI does not support port naming, the **target address** for a parallel SCSI device is the target port identifier, a value between 0x0 and 0xf.

The **LUN address** is the LUN number, a value between 0x0 and 0x7.

The **legacy hardware path** can be easily derived from the lunpath hardware path. The only difference is that the target and LUN addresses in the legacy hardware path are displayed in decimal instead of hexadecimal.

The following table shows four example lunpath hardware paths and their equivalent legacy hardware paths for parallel SCSI devices:

Lunpath Hardware Path	Legacy Hardware Path
0/0/2/0.0x0.0x0	0/0/2/0.0.0
0/1/1/0.0x1.0x0	0/1/1/0.1.0
0/1/1/0.0x7.0x0	0/1/1/0.7.0
0/0/2/1.0xf.0x0	0/0/2/1.15.0

Fibre Channel

The **target address** for a Fibre Channel device is the target port worldwide port name (WWPN). The `fcmsutil` command with the `get remote` option displays the WWPN as the Target Port World Wide Name. In this example, the WWPN is 0x500805f300083899:

```
# fcmsutil /dev/td0 get remote all

                     Target N_Port_id is = 0x010700
                            Target state = DVS_READY
                           Symbolic Name =
                               Port Type = N_PORT
   SLER Capable (supports Retry & TRID) = NO
              Target Port World Wide Name = 0x500805f300083899
              Target Node World Wide Name = 0x500805f300083890
…
```

The **LUN address** for a Fibre Channel device is a 64-bit LUNid. The LUNid is composed of two bits representing the addressing method, fourteen bits representing the LUN number of the device, and 48 reserved bits, as shown in the table:

Addressing Method (2 bits)	**LUN Number** (14 bits)	**Reserved** (48 bits)
00 – Peripheral Device Addressing 01 – Volume Set Addressing (Flat Space Addressing) 10 – Logical Unit Addressing	0x00-0x3fff	0x00

Because the LUN address is printed in hexadecimal, the first hexadecimal digit contains both the addressing method and the start of the LUN number. For example, the first 16 bits of the LUN address 0x400f000000000000 is binary 0100000000001111. The leading 01 is the addressing method (Flat Space Addressing) and the remaining bits represent the LUN number (15). Thus, the LUN address 0x400f000000000000 represents LUN number 15 using the Flat Space Addressing method. You can easily perform this decoding by using the `scsimgr` command to display the LUN `lunid` attribute. For example:

```
# scsimgr get_attr -H 0/2/1/0.0x500805f300083899.0x400f000000000000 -a lunid

      SCSI ATTRIBUTES FOR LUN PATH : 0/2/1/0.0x500805f300083899.0x400f000000000000

name = lunid
current =0x400f000000000000 (LUN # 15, Flat Space Addressing)
default =
saved =
```

The **legacy hardware path** format for Fibre Channel disks encodes the target port identifier and LUN id as virtual hardware elements. It has the following format:

HBA_path.domain.area.port.controller.target.lun

The *domain* represents the Fibre Channel switch, the *area* is the specific port on the Fibre Channel switch to which the target is plugged in. The *domain*, *area*, and *port* values are extracted from the target port identifier as shown in the table:

Domain (8 bits)	**Area** (8 bits)	**Port** (8 bits)
0-255	0-255	0-255

The *controller*, *target*, and *lun* values are extracted from the LUN id as shown in the table:

Controller (7 bits)	**Target** (4 bits)	**LUN** (3 bits)
0-63	0-15	0-7

To convert a lunpath hardware path to a legacy hardware path, use the `scsimgr get_info` command on the target hardware path to display the port_id, which contains the *domain*, *area*, and *port* values. Extract the *controller*, *target*, and *lun* values from the LUN id.

The following example uses lunpath hardware path 0/2/1/0.0x500805f300083899.0x4011000000000000.

```
# scsimgr get_info -H 0/2/1/0.0x500805f300083899

        STATUS INFORMATION FOR TARGET PATH : 0/2/1/0.0x500805f300083899

Generic Status Information

SCSI services internal state                    = IDLE
Port id                                         = 0x10700
Protocol                                        = fibre_channel
Protocol revision                               = 4.3
Port name                                       = 0x500805f300083899
Node name                                       = 0x500805f300083890
LUN paths registered (active/inactive)          = 12
```

A port_id of 0x10700 yields a *domain* value of 1, an *area* value of 7, and a *port* value of 0.

The LUN address of 0x4011000000000000, which represents LUN id 0x11, yields a *controller* value of 0, a *target* value of 2 and a *lun* value of 1.

The resultant legacy hardware path is 0/2/1/0.1.7.0.0.2.1.

To confirm this mapping, use the `ioscan -m hwpath` command as follows:

```
# ioscan -m hwpath -H 0/2/1/0.0x500805f300083899.0x4011000000000000
Lun H/W Path    Lunpath H/W Path                              Legacy H/W Path
============================================================================
64000/0xfa00/0x4
                0/2/1/0.0x500805f300083899.0x4011000000000000 0/2/1/0.1.7.0.0.2.1
```

Serial Attached SCSI

Serial attached SCSI (SAS) uses an addressing model similar to Fibre Channel.

The **target address** is the SAS Address, which is printed by `sasmgr` with the `get_info` command. You can display all the target addresses on a specified HBA by using the `-N` option and the `lun` qualifier. For example:

```
# sasmgr get_info -N -D /dev/sasd0 -q lun=all
LUN dsf                Hardware Path                 SAS Address
------------------------------------------------------------------
/dev/rdisk/disk14    0/2/1/0.0x500000e010f12f42.0x0 0x500000e010f12f42
/dev/rdisk/disk15    0/2/1/0.0x5000c5000032b739.0x0 0x5000c5000032b739
/dev/rdisk/disk16    0/2/1/0.0x5000c5000032f04d.0x0 0x5000c5000032f04d
/dev/rdisk/disk17    0/2/1/0.0x5000c5000032fe89.0x0 0x5000c5000032fe89
```

The **LUN address** for an SAS device, as with Fibre Channel, is a 64-bit LUNid composed of two bits representing the addressing method, fourteen bits representing the LUN number of the device, and 48 reserved bits. The decoding is available by using the `scsimgr` command to display the LUN `lunid` attribute. For example, using the first address above:

```
# scsimgr get_attr -H 0/2/1/0.0x500000e010f12f42.0x0 -a lunid

      SCSI ATTRIBUTES FOR LUN PATH : 0/2/1/0.0x500000e010f12f42.0x0
```

```
name = lunid
current = 0x0 (LUN # 0, Peripheral Addressing)
default =
saved =
```

The **legacy hardware path** format for SAS disks has the following format:

HBA_path.target.virtual_bus.virtual_target.lun

You can use the `sasmgr` command to correlate SAS addresses with legacy hardware paths, by using the `lun` qualifier *without* the `–N` option. For example:

```
# sasmgr get_info -D /dev/sasd0 -q lun=all
LUN dsf             Hardware Path                 SAS Address
------------------------------------------------------------------
/dev/rdsk/c0t0d0    0/2/1/0.0.0.0.0               0x5000c5000032b739
/dev/rdsk/c0t1d0    0/2/1/0.0.0.1.0               0x5000c5000032f04d
/dev/rdsk/c0t2d0    0/2/1/0.0.0.2.0               0x5000c5000032fe89
/dev/rdsk/c0t3d0    0/2/1/0.0.0.3.0               0x500000e010f12f42
```

You can also use the `ioscan –m hwpath` command to convert a SAS lunpath hardware path to a legacy hardware path, as follows:

```
# ioscan -m hwpath -H 0/2/1/0.0x500000e010f12f42.0x0
Lun H/W Path      Lunpath H/W Path                Legacy H/W Path
====================================================================
64000/0xfa00/0x0
                  0/2/1/0.0x500000e010f12f42.0x0  0/2/1/0.0.0.3.0
```

For more information

To learn more about the agile view and mass storage on HP-UX, see the following documents on the HP documentation website:
http://docs.hp.com/

For information about specific features of the new mass storage stack:

- *HP-UX 11i v3 Mass Storage Device Naming*
- *HP-UX 11i v3 Native Multi-Pathing for Mass Storage*
- *HP-UX 11i v3 Mass Storage I/O Scalability*

For information about system administration:

- *Scsimgr SCSI Management and Diagnostics Utility*
- *SCSI Commands used by HP-UX 11i v3*
- *HP System Management Homepage*
- *Troubleshooting Fibre Channel SANs in HP-UX 11i v3*
- *HP-UX 11i v3 Crash Dump Improvements*
- *setboot(1M)* in HP-UX 11i v3

For information about Logical Volume Manager:

- *LVM New Features in HP-UX 11i v3*

For information about migrating to HP-UX 11i v3 and agile addressing:

- *HP-UX 11i v2 to 11i v3 Mass Storage Stack Update Guide*
- *HP-UX 11i v3 Persistent DSF Migration Guide*
- *LVM Migration from Legacy to Agile naming model HP-UX 11i v3*

For information about performance:

- *HP-UX 11i v3 Mass Storage I/O Performance Improvements*

For information about developing I/O software:

- *Writing Active/Passive Switch Plug-Ins for HP-UX 11i v3*
- *Using the HP-UX libIO*

These documents are available in the *Network and Systems Management* section of http://docs.hp.com/, under *Storage Area Management* (http://docs.hp.com/en/netsys.html#Storage%20Area%20Management).

HP welcomes your input. Use our technical documentation feedback website to give us comments about this white paper, or suggestions for mass storage or related documentation at: http://docs.hp.com/en/feedback.html.

April 2009