Managing Shared Storage in a Sun™ Cluster 3.0 Environment With Solaris™ Volume Manager Software

Kristien Hens, Enterprise Services
Peter Dennis, Solaris Sustaining and Engineering

Sun BluePrints™ OnLine - November 2002

http://www.sun.com/blueprints

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95045 U.S.A.
650 960-1300
Part No.: 817-0656-10
Revision 1.0
Edition: November 2002
Managing Shared Storage in a Sun™ Cluster 3.0 Environment With Solaris™ Volume Manager Software

With Sun™ Cluster 3.0 software, you can use two volume managers: VERITAS Volume Manager (VxVM) software, and Sun's Solaris™ Volume Manager software, which was previously called Solstice DiskSuite™ software.

Traditionally, VxVM has been the volume manager of choice for shared storage in enterprise-level configurations. In this Sun BluePrints™ OnLine article, we describe a free and easy-to-use alternative, Solaris Volume Manager software, which is part of the Solaris™ 9 Operating Environment (Solaris 9 OE). This mature product offers similar functionality to VxVM. Moreover, it is tightly integrated into the Sun Cluster 3.0 software framework and, therefore, should be considered to be the volume manager of choice for shared storage in this environment. It should be noted that Solaris Volume Manager software cannot be used to provide volume management for Oracle RAC/OPS clusters.

To support our recommendation to use Solaris Volume Manager software, we present the following topics:

- “Using Solaris Volume Manager Software With Sun Cluster 3.0 Framework” on page 2 explains how Solaris Volume Manager software functions in a Sun Cluster 3.0 environment.
- “Configuring Solaris Volume Manager Software in the Sun Cluster 3.0 Environment” on page 9 provides a run book and a reference implementation for creating disksets and volumes (metadevices)¹ in a Sun Cluster 3.0 framework.
- “Advantages of Using Solaris Volume Manager Software in a Sun Cluster 3.0 Environment” on page 14 summarizes the advantages of using Solaris Volume Manager software for shared storage in a Sun Cluster 3.0 environment.

¹. With the new release of Solaris Volume Manager software, references to “metadevices” are replaced by the term “volumes.”
Note – The recommendations presented in this article are based on the use of the Solaris 9 OE and Sun Cluster 3.0 update 3 software.

Using Solaris Volume Manager Software With Sun Cluster 3.0 Framework

Before we present our reference configuration, we describe some concepts to help you understand how Solaris Volume Manager software functions in a Sun Cluster 3.0 environment. Specifically, we focus on the following topics:

- Sun Cluster software's use of DID (disk ID) devices to provide a unique and consistent device tree on all cluster nodes.
- Solaris Volume Manager software's use of disksets, which enable disks and volumes to be shared among different nodes, and the diskset's representation in the cluster called a device group.
- The use of mediators to circumvent the tight replica quorum (which is different from the cluster quorum) rule of Solaris Volume Manager software, and to allow clusters to operate in the event of specific multiple failures.
- The use of soft partitions and the mdmonitor daemon with Solaris Volume Manager software. While these components are not related to the software's use in a Sun Cluster environment, they should be considered part of any good configuration.

Using DID Names to Ensure Device Path Consistency

With Sun Cluster 3.0 software, it is not necessary to have an identical hardware configuration on all nodes. However, different configurations may lead to different logical Solaris OE names on each node. Consider a cluster where one node has a storage array attached on a host bus adapter (HBA) in the first peripheral component interconnect (PCI) slot. On the other node, the array is attached to an HBA in the second slot. A shared disk on target 30 may end up being referred to as /dev/rdsk/c1t30d0 on the first node and as /dev/rdsk/c2t30d0 on the other node. In this case, the physical Solaris OE device path, and thus the major-minor number combination, are different on each node.
In a non-clustered environment, Solaris Volume Manager software uses the logical Solaris OE names as building blocks for volumes. However, in a clustered environment, the volume definitions are accessible on all the nodes and should, therefore, be consistent; the name and the major/minor numbers should be consistent across all the nodes. Sun Cluster software provides a framework of consistent and unique disk names and major/minor number combinations. Such names are created when you install the cluster and they are referred to as DID names. They can be found in /dev/did/rdsk and /dev/did/dsk and are automatically synchronized on the cluster nodes such that the names and the major/minor numbers are consistent between nodes. Sun Cluster 3.0 software uses the serial numbers on the disks to guarantee that the same name exists for a given disk in the cluster.

Always use DID names when referring to disk drives to create disksets and volumes with Solaris Volume Manager software in a Sun Cluster 3.0 environment.

Using Disksets to Share Disks and Volumes Among Nodes

Using Solaris Volume Manager software in a Sun Cluster environment differs from performing a standalone installation in that it introduces the concept of disksets, and as such we have two separate categories of state database replicas, local and shared.

On all nodes, local state database replicas must be created. These local state database replicas contain configuration information for locally created volumes. For example, volumes that are part of the mirrors on the boot disk. Local state database replicas also contain information about disksets that are created in the cluster: The name of the set, the names of the hosts that can own the set, the disks in it and whether they have a replica on them and, if configured, the mediator hosts. This is a major difference between Solaris Volume Manager software and VxVM, because in VxVM, each diskgroup is self-contained: Each disk within the group contains the group to which it belongs and the host that currently owns the group. If the last disk in a VxVM diskgroup is deleted, the group is deleted by definition.

A diskset is always only imported (owned) on one node. This is called the current primary of that diskset. This means that although more nodes can be attached to the diskset and can potentially import the diskset upon failure of the primary node, only one node can effectively do input/output (I/O) to the volumes in the diskset. The term shared storage is, therefore, a bit misleading: They are not shared in the sense that all nodes access the disks simultaneously, but in the sense that different nodes are potential primaries for the set.

2. In a Sun Cluster 3.0 environment, nodes are recorded by their cluster node ids, not by their actual names.
The creation of disksets involves three steps. First, a diskset receives a name and a primary host. This action creates an entry for the diskset in the local state database of that host. Then, a second host (and, if required, up to six other hosts) can be added. The rpc.metad daemon on the first node contacts the rpc.metad daemon on the second host, instructing it to create an entry for the diskset in the second host’s local state database.

Now, disks can be added to the diskset. Again, the primary hosts rpc.metad daemon will contact the second host so that the local state databases on both nodes contain the same information.

Note you can add disks to any node that can potentially own the diskset and the request is forwarded (proxied) to the primary node. This is done through the rpc.metaclild daemon, which allows you to administer disksets from any cluster node. Neither rpc.metad and rpc.metaclild should be hardened out of a cluster that is using Solaris Volume Manager software because they are both essential to the operation of the Solaris Volume Manager software components.

Disks that you add to a diskset are automatically repartitioned if they do not already have a valid slice 7. A valid slice 7 is a slice that starts at cylinder 0 and is large enough to hold a state replica. No other slice, not even the backup slice 2, should overlap with this slice. If these criteria are not met, the disk is repartitioned to conform to these criteria. Such disks then contain a slice 7, which has the minimum number of cylinders to hold a replica, and a slice 0 that takes up the rest of the space. There is no overlap slice 2 anymore. Slice 7 contains the diskset specific state database replicas. These replicas are automatically created and are equally divided across disks and controllers. A diskset replica contains all the information about the volumes that have been created within the diskset.

Using Device Groups to Manage Disks and Volumes

Sun Cluster 3.0 software provides automatic exporting and importing of Solaris Volume Manager disksets and VxVM diskgroups. To accomplish this, you have to identify the diskset or diskgroup to the cluster. For each device (disk, tape, Solaris Volume Manager diskset, or VxVM diskgroup) that should be managed by the cluster, ensure that there is an entry in the cluster configuration repository.

When a diskset or diskgroup is known to the cluster, it is referred to as a device group. A device group is an entry in the cluster repository that defines extra properties for the diskgroup or diskset. A device group can have the following characteristics:

- A node list that corresponds to the node list defined in the diskset.

3. The default size of a replica is 8192 blocks and the size of the cylinder is dependent on the disk geometry. This size can be determined by using prtvtoc(1M).
A preferred node where the cluster attempts to bring the device group online when the cluster boots. This effectively means that when all cluster nodes are booted at the same time, the diskset is imported by its preferred node.

A failback policy, that if set to true, migrates the disk set to the preferred node if the node is online. If the preferred node joins the cluster later, it will become the owner of the diskset (that is, the diskset will switch from the node that currently owns it to the preferred node).

Sun Cluster software also provides extensive failure fencing mechanisms to avoid data access by unauthorized nodes during device group transitions.

One of the major advantages of using Solaris Volume Manager software in a Sun Cluster 3.0 environment is that the creation and deletion of device groups does not involve extra administration. When you create or delete a diskset with Solaris Volume Manager software commands, the cluster framework is automatically notified that it should create or delete a corresponding entry in the Cluster Configuration repository. You can also manually change the preferred node and failback policy with standard cluster interfaces.

Using Mediators to Manage Replica Quorum Votes

Disksets have their own replicas, which are either added to a disk when the disk is put into the diskset, or when an administrator manually adds them using the `metadb` command. Replicas should be evenly distributed across the storage enclosures that contain the disks, and they should be evenly distributed across the disks on a per-disk-controller basis. In an ideal environment, this distribution means that any one failure in the storage (disk, controller, or storage enclosure) does not affect the operation of Solaris Volume Manager software.

In a physical configuration that has an even number of storage enclosures, the loss of half of the storage enclosures (for example, due to power loss) leaves only 50 percent of the diskset replicas available. While the diskset is owned by a node, this will not create a problem. However, if the diskset is released, on a subsequent import, the replicas will be marked as being stale because the replica quorum of greater than 50 percent will not have been reached. This means that all the data on the diskset will be read-only, and operator intervention will be required. If, at any point, the number of available replicas for either a diskset or the local ones falls below 50 percent, the node will abort itself to maintain data integrity.

To override this feature, you can configure a set to have mediators. Mediators are hosts that can import [take] a diskset, and, when required, they provide an additional vote when a quorum vote is required (for example, on a diskset import [take]). To assist the replica quorum requirement, mediators also have a quorum...
requirement that either greater than 50 percent of them are available, or the available mediators are marked as being up to date, this means the mediator is golden and is marked as such.

Replicas not only contain configuration information of the volumes, they also contain state information about the volumes. That is the current state of a volume, whether it is online and working or not. In addition, some of the Solaris Volume Manager software components provide some sort of state. For example, mirrors use a bitmap to show which side of the mirror has the latest data. This type of information changes as part of Solaris Volume Manager software's operation. Each replica has an associated commit count that counts the number of changes made to it and is incremented when such a change occurs (volume state change, a mirror bitmap change) and it is propagated to the mediators. If, for any reason, 50 percent of the replicas fail and the diskset has to be reimported on a node, the commit count of the available replicas is compared with the commit count in the mediators. If the counts are the same, the diskset is imported and all volumes will be enabled4. At this point, the mediators are marked as golden. If a mediator quorum is lost (for example, if a node fails) the available mediators will still act as an extra vote in the diskset replica requirement.

For example, if there are two hosts (node1 and node2) and two storage enclosures (pack1 and pack2), diskset replicas are distributed evenly between pack1 and pack2 and node1 owns the diskset. If pack1 dies, only 50 percent of diskset replicas are available and the mediators on both hosts are marked as golden. If node1 now dies, node2 can import [take] the diskset because 50 percent of the diskset replicas are available and the mediator on node2 is golden. If mediators were not configured, node2 would not have been able to import [take] the diskset without operator intervention.

Mediators do not, however, protect against simultaneous failures. If both pack1 and node1 fail at the same time, the mediator on node2 will not have been marked as golden and there will not be an extra vote for the diskset replica quorum, making operator intervention necessary. Because the nodes should be on an uninterrupted power supply (UPS), which means the mediators should have enough time to be marked as golden, this type of failure is unlikely.

**Using Soft Partitions as a Basis for File Systems**

After adding a disk to a diskset, you can modify the partition layout, that is break up the default slice 0 and spread the space between the slices (including slice 0). If slice 7 contains a replica, leave it alone to avoid corrupting the replica on it. Because Solaris Volume Manager software supports soft partitioning, we recommend that you leave slice 0 untouched.

---

4. If 50 percent of replicas are available and the diskset is owned, volumes can still be written to, the mediators are only used if the diskset needs to be deported and reimported (maybe on a different node).
Consider a soft partition as a subdivision of a physical Solaris OE slice or as a subdivision of a mirror, redundant array of independent disks (RAID) 5, or striped volume. The number of soft partitions you can create is limited by the size of the underlying device and by the number of possible volumes (nmd, as defined in /kernel/drv/md.conf). The default number of possible volumes is 128. Note that all soft partitions created on a single diskset are part of one diskset and cannot be independently primaried to different nodes.

Soft partitions are composed of a series of extents that can be located at arbitrary locations on the underlying media. They can be extended up to the amount of space on the underlying device.

You can create and use soft partitions in two ways:

- Create them on top of a physical disk slice and use them as building blocks for mirrors or RAID 5 volumes, just as you would use a physical slice.
- Create them on top of a mirror or RAID 5 volume.

In our example, we use the second approach. We consider this the best solution for two reasons:

- Sizing and resizing soft partitions is limited only by the size of the underlying device. If the underlying device is a Solaris OE slice, it is not always possible to increase the size of the soft partition while keeping the file system on the slice intact. However, it is much easier to grow a Solaris Volume Manager software volume, and then grow the soft partition on top of it.
- Creating different soft partitions on top of a large mirrored volume allows you to use the Solaris Volume Manager software namespace more efficiently and consistently. Consider the following example: You create one large mirror (d2) on top of two submirrors (d0 and d1). On top of d2, you create soft partitions d10, d11, d12, and so on. On these soft partitions, you create file systems. If you did it the other way around, you would have to create soft partitions d10, d11, d12, and so on, as well as corresponding soft partitions on the other disks d20, d21, d22, and so on. Then, you would have to create the stripes to use as submirrors on top of the soft partitions and finally create the mirrors. In this scenario, you would have used twice as many soft partitions names and, therefore, more of the Solaris Volume Manager software namespace.

While we recommend the second approach, keep in mind that a disadvantage of this approach is that you will have to perform a complete disk resync if the disk fails, while the first approach would only require a resync of the defined soft partitions.
Using the `mdmonitord` Daemon to Enable Active Volume Monitoring

The `mdmonitord` daemon quickly fails volumes that have faulty disk components. It does this by probing configured volumes, including volumes in disksets that are currently owned by the node where the daemon runs (note that the daemon runs on all the nodes in a cluster). The probe is a simple `open(2)` of the top level volume that causes the Solaris Volume Manager software kernel components to open underlying devices. The probe eventually causes the physical disk device to open. If the disk has failed, the probe will fail all the way back up the chain, and the daemon can take the appropriate action.

If the volume is a mirror, the mirror must be in use for the submirror’s component to be marked as errored. (That is, another application must have it open, for example, if it has a mounted file system on it). If the mirror is not open, the daemon reports an error and performs no other action to prevent unrequired resyncs for unused mirrors.

In the case of a RAID5 device, the device fails right away because there is not as much redundancy as there is in a mirror (a second failure in the RAID5 device makes the device inoperable), and it is better to have the cost of the failure immediately.

The `mdmonitord` daemon has two modes of operation: interrupt mode and periodic probing mode.

- In interrupt mode, the daemon waits for disk failure events. If the daemon detects a failure, it probes the configured volumes as previously described. This is the default behavior.

- In periodic probing mode, you can specify certain time intervals for the daemon to perform probes by giving the daemon the `-t` option, followed by the number of seconds between each probe. The daemon also waits for disk failure events.

The `mdmonitord` daemon is useful if your system contains volumes that are accessed infrequently. Without the daemon, a disk failure can go unnoticed for quite some time, unless you manually check the configuration with the `metastat -i` command. This might not be a problem at first sight, but can be catastrophic if the failed disk is the cluster’s quorum disk, or if an entire storage array has failed. These scenarios are described as follows:

- If a quorum disk fails and, subsequently, a node fails, cluster operation is seriously impaired. We recommend that you put the quorum disk in a diskset and make it part of a submirror so it is monitored by the `mdmonitord` daemon. Depending on the usage of the mirror, you might want to consider configuring the `mdmonitord` daemon to do timed probes. If the mirror is well used, that is, if it has plenty of I/O going to it, you might not want this.
If mediators are configured, they provide extra votes to guarantee replica quorum. Moreover, if, after an array fails, the remaining replicas are updated, the mediators are set to golden. Now, if a node is lost, the golden status on the remaining node allows Solaris Volume Manager software to continue updating the replicas without intervention. Using the mdmonitord daemon to regularly check the status of volumes increases the possibility of replica updates after a storage array fails and, therefore, improves the possibility that mediator status will be set to golden.

Configuring Solaris Volume Manager Software in the Sun Cluster 3.0 Environment

In this section, we present a sample configuration that consists of two nodes, phys-node-1 and phys-node-2. Each node has two internal drives and is connected to two hardware RAID devices. On each disk array, two logical unit numbers (LUNs) are created. Sun Cluster 3.0 software is installed on both nodes, and a quorum device is chosen. The two LUNS on the first array are known in the cluster as /dev/did/dsk/d3 and /dev/did/dsk/d4. The LUNs on the second array are /dev/did/dsk/d5 and /dev/did/dsk/d6. We create a diskset, nfsds, that contains one mirror, with two submirrors. On top of the mirror, we create four soft partitions that can be used to create file systems. Because this setup has two storage arrays, we must configure mediators.
To Configure Solaris Volume Manager Software on All Cluster Nodes

1. Create local replicas.

Because Solaris Volume Manager software is part of the Solaris 9 OE distribution, you do not have to install it separately. However, for the software to operate, it must have at least one local replica. The preferred minimum is three replicas. Use the following commands to create the local state database replicas:

```
# metadb -af -c 2 c1t0d0s7
# metadb -a -c 2 c2t0d0s7
```

Document the commands you use to create the local state database replicas and keep the information offline so you can refer to it if you have to recreate the configuration.

2. Mirror the boot disk.

We highly recommend that you mirror boot disks on all nodes. The procedure to do this is the same on a cluster node as it is on a noncluster node, with the following exceptions:

- After installing the cluster, the device to mount in /etc/vfstab for the global device’s file system changes from a logical Solaris OE name (c#t#d#s#) to a DID name. However, you should use the logical Solaris OE name (c#t#d#s#) as a building block for the first submirror.
- The volume containing the /global/.devices/node@nodeid file system should have a unique name on each cluster node.

For example, if /dev/md/dsk/d91 on node1 is mounted globally under /global/.devices/node@node1, create a volume /dev/md/dsk/d92 on node2 to be mounted under /global/.devices/node@node2. (The device to mount should be unique in the mnttab maintained by the Solaris OE kernel. Because the global device’s file systems on each node are mounted globally and, therefore, appear in the mnttab on both nodes, they should have different devices to mount in /etc/vfstab.)

The sdmsetup script, described in the Sun BluePrints OnLine article “Configuring Boot Disks With Solaris Volume Manager Software,” is available from the Sun BluePrints OnLine Web site at:

http://www.sun.com/solutions/blueprints/online.html

This script is cluster-aware and helps automate the mirroring and cloning of boot disks of cluster nodes.
3. [Optional] Change the interval of the `mdmonitord` daemon on all the nodes. Edit the `/etc/rc2.d/S95svm.sync` script to add a time interval for periodic checking as follows:

```bash
if [ -x $MDMONITOR ]; then
    $MDMONITOR -t 3600
    error=$?
    case $error in
    0) ;;
    *) echo "Could not start $MDMONITOR. Error $error."
    ;;
    esac
fi
```

4. Restart the `mdmonitord` daemon to make the changes effective as follows:

```bash
# /etc/rc2.d/S95svm.sync stop
# /etc/rc2.d/S95svm.sync start
```

▼ To Create Disksets

1. Document the commands you use to create the disksets and to add the disks, and keep the information offline.

2. Define the diskset and the nodes that can master the diskset as follows:

```bash
# metaset -s nfsds -a -h phys-node-1 phys-node-2
```

Only run the `metaset` command on one node. The `metaset` command issues `rpc` calls to the other node to create an entry in the other node's local replicas for this set.

3. Add drives to the diskset using their fully qualified DID names as follows:

```bash
# metaset -s nfsds -a /dev/did/rdsk/d3 /dev/did/rdsk/d4 /dev/did/rdsk/d5 /dev/did/rdsk/d6
```

Again, `rpc` calls are made to the other node to ensure that it also has the necessary information in its local state replicas.
4. When Solaris Volume Manager software creates disksets, it automatically tells the cluster software to create an entry in the cluster database for the diskset. This entry is called a device group, which you can check using the following command:

```bash
# scstat -D
```

5. Configure mediators on each node, as follows:

```bash
# metaset -s nfsds -a -m phys-node-1
# metaset -s nfsds -a -m phys-node-2
```

6. Check the status of the mediators as follows:

```bash
# medstat -s nfsds
```

▼ To Create Volumes and File Systems

1. Create the following volumes.
   - A mirrored volume composed of two striped submirrors. Each stripe is composed of the two slice 0s of the LUN in each storage array.
   - Four soft partitions on top of this mirror.

```bash
# metainit nfsds/d1 1 2 /dev/did/rdsk/d3s0 /dev/did/rdsk/d4s0
# metainit nfsds/d2 1 2 /dev/did/rdsk/d5s0 /dev/did/rdsk/d6s0
# metainit nfsds/d0 -m nfsds/d1
# metattach nfsds/d0 nfsds/d2
# metainit nfsds/d10 -p nfsds/d0 200m
# metainit nfsds/d11 -p nfsds/d0 200m
# metainit nfsds/d12 -p nfsds/d0 200m
# metainit nfsds/d13 -p nfsds/d0 200m
```

2. Create an `/etc/lvm/md.tab` file on both nodes. Keep a copy of this file offline so you can use it to recreate the configuration, if it becomes necessary to do so.
a. On the node that is the current primary of the diskset, supply the following commands:

```
# metastat -s nfsds -p >> /etc/lvm/md.tab
```

b. On the other node, supply the following commands:

```
# metaset -s nfsds -p | tail +2 >> /etc/lvm/md.tab
```

3. Create file systems on the logical partitions and mount them globally.
   a. On one node, create the file system as shown here:

```
# newfs /dev/md/nfsds/rdsk/d10
# newfs /dev/md/nfsds/rdsk/d11
# newfs /dev/md/nfsds/rdsk/d12
# newfs /dev/md/nfsds/rdsk/d13
```

b. On all nodes, create mount points as follows:

```
# mkdir /global/nfsd/fs1 /global/nfsd/fs2 /global/nfsd/fs3 /global/fs4
```

c. On all nodes, put following entries in /etc/vfstab as shown here:

```
/dev/md/nfsds/dsk/d10 /dev/md/nfsds/rdsk/d10 /global/nfsd/fs1 ufs 2 yes global
/dev/md/nfsds/dsk/d11 /dev/md/nfsds/rdsk/d11 /global/nfsd/fs2 ufs 2 yes global
/dev/md/nfsds/dsk/d12 /dev/md/nfsds/rdsk/d12 /global/nfsd/fs3 ufs 2 yes global
/dev/md/nfsds/dsk/d13 /dev/md/nfsds/rdsk/d13 /global/nfsd/fs4 ufs 2 yes global
```
d. On one node, mount the global file systems as follows:

```
# mount /global/nfsd/fs1
# mount /global/nfsd/fs2
# mount /global/nfsd/fs3
# mount /global/nfsd/fs4
```

e. Test the configuration.

i. On one node, run the following commands:

```
# metaset
# scstat -D
```

ii. On all nodes, run the following command:

```
# df -k
```

Advantages of Using Solaris Volume Manager Software in a Sun Cluster 3.0 Environment

When designing a solution using Sun Cluster 3.0 software, you must decide which volume manager to use. In addition to being freely included in the Solaris 9 OE distribution, Solaris Volume Manager software has additional features that make it the volume manager of choice.

The following list is not exhaustive, but it explains why Solaris Volume Manager software deserves its place in enterprise-level configurations.

- The introduction of soft partitioning eliminates the seven-slice limit imposed by the Solaris OE. Solaris Volume Manager software is now equal to other logical volume managers when very large disks are used.
- The `mdmonitord` daemon introduces active volume monitoring.
Solaris Volume Manager software is a true cluster-aware application. It automatically registers disksets with the cluster and synchronizes volumes in the global device namespace, thereby eliminating configuration headaches. This makes it the easiest volume manager to use in combination with Sun Cluster 3.0 software.

The use of soft partitions on top of big mirrored volumes guarantees a clean and consistent volume namespace.

All nodes that can import the diskset are locally known on each cluster node, adding extra security to the configuration. Rogue nodes that have not been authorized by valid nodes will never forcibly take the diskset.

Solaris Volume Manager software, the Solaris 9 OE, and Sun Cluster 3.0 software were developed and are supported by Sun, eliminating possible supportability issues.

Solaris Volume Manager software uses the concepts of a replica quorum and a commit count that is kept in each state database replica to ensure database consistency and to avoid database corruption.

The Solaris Volume Manager software command line interface is extremely intuitive. The time required to become familiar with the product is considerably less than the time required to become familiar with competitive products.

If the need arises, tearing down and rebuilding a Solaris Volume Manager software configuration is very simple and easy to do.

---

About the Authors

Kristien Hens has worked for Sun for three years. She is currently an instructor for Internal Technical Training, teaching mostly Sun Cluster courses. Before joining Sun she worked as a developer in the field of computational linguistics.

Peter Dennis has worked as an engineer within the Solaris Sustaining and Engineering group for four years. Before that, he worked as a support engineer in Sun Services for three years.

---

Ordering Sun Documents

The SunDocs™ program provides more than 250 manuals from Sun Microsystems, Inc. If you live in the United States, Canada, Europe, or Japan, you can purchase documentation sets or individual manuals through this program.
Accessing Sun Documentation Online

The docs.sun.com web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. The URL is http://docs.sun.com/

To reference Sun BluePrints OnLine articles, visit the Sun BluePrints OnLine Web site at: http://www.sun.com/blueprints/online.html