Integrating VERITAS Volume Manager (VxVM) with MC/ServiceGuard A.11.09

Manufacturing Part Number: B3936-90048

April 2001

Legal Notices

The information contained in this document is subject to change without notice. *Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose.* Hewlett-Packard shall not be liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

HP ClusterView, HP OpenView, HP OpenView IT/Operations, MC/LockManager, and MC/ServiceGuard are products of Hewlett-Packard Company, and all are protected by copyright. VERITAS and VERITAS Volume Manager are registered trademarks of VERITAS Software Corporation. EMC^2, Symmetrix, and PowerPath are registered trademarks of EMC Corporation

Corporate Offices:

Hewlett-Packard Co.
3000 Hanover St.
Palo Alto, CA 94304

Use, duplication or disclosure by the U.S. Government Department of Defense is subject to restrictions as set forth in paragraph (b)(3)(ii) of the Rights in Technical Data and Software clause in FAR 52.227-7013.

Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19(c)(1,2).

Use of this manual and flexible disc(s), compact disc(s), or tape cartridge(s) supplied for this pack is restricted to this product only. Additional copies of the programs may be made for security and back-up purposes only. Resale of the programs in their present form or with alterations, is expressly prohibited. A copy of the specific warranty terms applicable to your Hewlett-Packard product and replacement parts can be obtained from your local Sales and Service Office.

1 Integrating VERITAS Volume Manager (VxVM) with MC/ServiceGuard A.11.09

This document describes the use of the VERITAS Volume Manager (VxVM) Version 3.1 in clusters configured with MC/ServiceGuard or ServiceGuard OPS Edition A.11.09 as patched with ServiceGuard and ServiceGuard OPS Edition patch PHSS_23511. Topics are as follows:

- Summary of Features Provided by Patch PHSS_23511
- Data Protection
- Volume Managers for Data Storage
- VxVM Planning
- Installing VxVM Software on an Existing Cluster
- Creating a Storage Infrastructure with VxVM
- Creating the Package Control Script
- Customizing the Package Control Script
- Troubleshooting

Please note the following restrictions when using VxVM storage with ServiceGuard:

1. ServiceGuard A.11.09 is supported only with the following version of VERITAS Volume Manager 3.1 for HP-UX (VxVM):
   - VxVM B.03.10.5 or later (on HP-UX 11i)
2. ServiceGuard/ServiceGuard OPS Edition patch PHSS_23511 or later must be installed.
3. Shared activation is not supported with VxVM; therefore, VxVM cannot be used for OPS data in clusters built with ServiceGuard OPS Edition. However, VxVM can be used for non-OPS data in OPS clusters.
4. The EMC PowerPath product is not currently supported with VxVM

disk groups. The online VxVM release notes will be updated once support becomes available. Refer to the latest version of the *VERITAS Volume Manager 3.1 for HP-UX Release Notes* on http://docs.hp.com for additional information.

5. VxVM cannot be used with MetroCluster with Continuous Access XP or MetroCluster with EMC SRDF. VxVM cannot be used in continental clusters that employ physical data replication with Continuous Access XP or with EMC SRDF.
6. The disk monitor provided in the HA Monitors product (B5736DA) is designed for use only with LVM volume groups. Therefore, you cannot configure a package dependency on a monitor for VxVM disk groups.

The following products are described in this document:

- Base VERITAS Volume Manager 3.1 for HP-UX (B7961AA)
- VERITAS Volume Manager 3.1 for HP-UX (B9116AA)
- VERITAS Volume Manager 3.1 FastResync Option for HP-UX (B9118AA)

Before attempting to use VxVM storage with ServiceGuard, please refer to the following:

- *VERITAS Volume Manager 3.1 Administrator's Guide*. This contains a glossary of VERITAS terminology.
- *VERITAS Volume Manager 3.1 Storage Administrator Administrator's Guide*
- *VERITAS Volume Manager 3.1 Release Notes*

These documents are available on the HP-UX documentation web site at the following URL:

`http://docs.hp.com/hpux/11i`

in the section headed "VERITAS Volume Manager." Refer to this web site for the latest information. For information on individual VERITAS commands, refer to the VERITAS man pages.

Summary of Features Provided by Patch PHSS_23511

Support for the use of VERITAS Volume Manager (VxVM) is enabled by patch PHSS_23511, which adds the following to ServiceGuard and ServiceGuard OPS Edition:

- Changes to the package control script. These changes include additional functionality to import a VxVM disk group at package startup time and deport it at package halt time. The script also checks to ensure that the disk group is not imported on another node.
- Changes to the ServiceGuard RC script, `/sbin/init.d/cmcluster`. These changes cause disk groups which do *not* have the autoimport flag set to have their owner ID cleared at boot time.
- Awareness of VxVM disk groups in ServiceGuard configuration commands and the configuration daemon. Error messages will not be printed about VxVM disks from ServiceGuard configuration commands.

VxVM support will be fully documented in the manual *Managing MC/ServiceGuard* at the time of the next ServiceGuard release.

Data Protection

It is required that you provide data protection for your highly available system, using one of two methods:

- Disk Mirroring
- Disk Arrays using RAID Levels and Multiple Data Paths

Disk Mirroring

Disk mirroring is one method for providing data protection. The logical volumes used for ServiceGuard packages should be mirrored. ServiceGuard does not provide protection for data on your disks. This capability is provided for LVM storage with HP's MirrorDisk/UX product, and for VxVM with the VERITAS Volume Manager (B9116AA). When you configure logical volumes using software mirroring, the members of each mirrored set contain exactly the same data. If one disk should fail, the storage manager will automatically keep the data available by accessing the other mirror. One-way or two-way mirroring in LVM (or additional plexes with VxVM) may be used to allow for online backups or even to provide an additional level of high availability.

To protect against FibreChannel or SCSI bus failures, each copy of the data must be accessed by a separate bus; that is, you cannot have all copies of the data on disk drives connected to the same bus.

While it is valuable to mirror your root disks, it is critical for high availability that you mirror your data disks. In the event of damage to a root disk of one SPU, the other SPUs in the cluster will take over control of the applications. However, if a data disk is damaged, and not mirrored, any application which depends on that disk will not be available until the problem with the disk is repaired. Even then, it may not be possible to recover the data on that disk.

Disk Arrays using RAID Levels and Multiple Data Paths

An alternate method of achieving protection for your data is to employ a disk array with hardware RAID levels that provide data redundancy, such as RAID Level 1 or RAID Level 5. The array provides data

redundancy for the disks. This protection needs to be combined with the use of redundant host bus interfaces (SCSI or FibreChannel) between each node and the array. The use of redundant interfaces, configured with LVM's PV Links feature or VxVM's dynamic multipathing (DMP), protects against single points of failure in the I/O channel, and RAID 1 or 5 configuration provides redundancy for the storage media. DMP is available as a separate component of VxVM. DMP for active/active devices requires B9116AA, but DMP for active/passive devices is available for no charge with the base product, B7961AA.

Volume Managers for Data Storage

A volume manager is a tool that lets you create units of disk storage known as storage groups. Storage groups contain logical volumes for use on single systems and in high availability clusters. In ServiceGuard clusters, storage groups are activated by package control scripts.

ServiceGuard allows a choice of volume managers for data storage:

- HP-UX Logical Volume Manager (LVM) and (optionally) MirrorDisk/UX
- VERITAS Volume Manager 3.1 for HP-UX (VxVM)—Base and Add-on Products

This section explains some of the differences between these available volume managers and offers suggestions about appropriate choices for your cluster environment.

NOTE The HP-UX Logical Volume Manager is described in *Managing Systems and Workgroups*. A complete description of VERITAS volume management products is available in the *VERITAS Volume Manager 3.1 Release Notes*.

HP-UX Logical Volume Manager (LVM)

Logical Volume Manager (LVM) is the legacy storage management product on HP-UX. Its use is supported with ServiceGuard A.11.09 on both HP-UX 11.0 and 11i, as well as with prior versions of ServiceGuard and of HP-UX. Included with the operating system, LVM is available on all cluster nodes. It supports the use of MirrorDisk/UX, which is an add-on product that allows disk mirroring with up to two mirrors (for a total of three copies of the data).

Currently, the HP-UX root disk must be configured as an LVM volume group. (Note that the HP-UX root disk is not the same as the VxVM *root disk group*, rootdg, which must be configured in addition to the HP-UX root disk on any node that uses VERITAS Volume Manager products.) The ServiceGuard cluster lock disk also is configured using a disk configured in an LVM volume group.

LVM continues to be supported on HP-UX single systems and on ServiceGuard clusters.

VERITAS Volume Manager (VxVM)

The Base VERITAS Volume Manager (B7961AA) is provided at no additional cost with HP-UX 11i. This includes basic volume manager features, including a Java-based GUI, known as `vmsa`. It is possible to configure cluster storage for ServiceGuard with only the Base VERITAS Volume Manager. However, only a limited set of features is available.

The add-on product, VERITAS Volume Manager (B9116AA), provides a full set of enhanced volume manager capabilities in addition to basic volume management. This includes features such as mirroring, dynamic multipathing for active/active storage devices, and hot relocation. The VERITAS FastResync option (B9118AA) is another add-on product, available with an additional charge, that reduces the time taken to resynchronize a split mirror to the volume. This product requires the purchase of the VERITAS Volume Manager (B9116AA).

VxVM can be used in clusters that:

- require a fast cluster startup time.
- do not require shared storage group activation.
- do not have all nodes cabled to all disks.
- need to use RAID 5 or striped mirroring.

Propagation of Disk Groups in VxVM

With VxVM, a disk group can be created on any node, with the cluster up or not. The user then needs to go to each node and validate that disk group by trying to import it.

Package Startup Time with VxVM

With VxVM, each disk group is imported by the package control script that uses the disk group. This means that cluster startup time is not affected, but individual package startup time might be increased because VxVM imports the disk group at the time the package starts up.

Comparison of Volume Managers

The following table summarizes some of the advantages and disadvantages of the volume managers that are currently available.

Table 1-1 Pros and Cons of Volume Managers with ServiceGuard

Product	Pros	Cons
Logical Volume Manager (LVM)	• Legacy system is robust and familiar to HP-UX users • Existing packages do not need to be changed • Supports up to 16 nodes per cluster • Supports use of PV links for multiple data path • Supports exclusive activation as well as read-only activation from multiple nodes • Can be used to configure a cluster lock disk	• Lacks flexibility and extended features of other volume managers • PV links are active/standby only, with one link active at a time
Base VERITAS Volume Manager—B7961AA (VxVM)	• Software is free with HP-UX 11i and later • Simple basic features provided	• Limited feature set • Supports up to four nodes without mirroring, RAID 5 or DMP for active/active storage devices • May cause a delay at package startup time due to lengthy vxdg import • Cannot be used as a cluster lock device

Table 1-1 Pros and Cons of Volume Managers with ServiceGuard

Product	Pros	Cons
VERITAS Volume Manager—B9116AA (VxVM)	• Disk group configuration can be done from any node • Enhanced set of volume management features, including software mirroring, RAID 0/1, RAID 5, and dynamic multi-pathing for active/active storage devices • Supports up to 16 nodes per cluster	• Does not support activation on multiple nodes in either shared mode or read-only mode • May cause delay at package startup time due to lengthy vxdg import • Enhanced features require purchase of additional license • Cannot be used as a cluster lock device

VxVM Planning

When designing a storage configuration using VxVM disk groups, consider the following:

- You must create a *rootdg* disk group on each cluster node that will be using VERITAS storage. *This is not the same as the HP-UX root disk, and cannot be used as the HP-UX root disk.* The VxVM root disk group can only be imported on the node where it is created.
- VxVM disk groups may be created before cluster configuration if desired.
- High availability applications, services, and data should be placed in separate disk groups from non-high availability applications, services, and data.
- You must not group two different high availability applications, services, or data, whose control needs to be transferred independently, onto the same disk group.
- Your HP-UX root disk must belong to an LVM volume group that is *not* shared among cluster nodes.
- The cluster lock disk can only be configured with an LVM volume group.
- VxVM disk group names should not be entered into the cluster configuration ASCII file. These names are not inserted into the cluster configuration ASCII file by `cmquerycl`.

VxVM Worksheet

The following worksheet will help you organize and record your specific physical disk configuration. Make as many copies as you need. Fill out the worksheet and keep it for future reference. This worksheet only includes disk groups and physical volumes. The Package Configuration worksheet (presented later in this chapter) contains more space for recording information about the logical volumes and filesystems that are part of each disk group.

Figure 1-1 VERITAS Disk Group and Disk Worksheet

```
==========================================================================

   Disk Group Name:      _______dg01_____________________________________

   Disk Name:   _____________________c1t2d0_____________________________

   Disk Name:   _____________________c2t2d0_____________________________

   Disk Name:   _____________________c3t2d0_____________________________

   Disk Name:   _____________________c4t2d0_____________________________

   Disk Name:   _____________________c5t2d0_____________________________

   Disk Name:   _____________________c6t2d0_____________________________

   Disk Name:   _____________________c7t2d0_____________________________

   Disk Group Name:      _______dg02_____________________________________

   Disk Name:   _____________________c1t3d0_____________________________

   Disk Name:   _____________________c2t3d0_____________________________

   Disk Name:   _____________________c3t3d0_____________________________

   Disk Name:   _____________________c4t3d0_____________________________

   Disk Name:   _____________________c5t3d0_____________________________

   Disk Name:   _____________________c6t3d0_____________________________

   Disk Name:   _____________________c7t3d0_____________________________
```

Installing VxVM Software on an Existing Cluster

You use the `swinstall` command to install the VERITAS VxVM 3.1 software on each cluster node. The installation process requires a system re-boot. If you are using the VxVM software with an existing cluster, this means taking each node out of the cluster while the VERITAS Volume Manager software is being installed.

Before installing VxVM software, refer to the section “Preparing to Install the VERITAS Volume Manager on a System with LVM Volume Groups” in the *VERITAS Volume Manager 3.1 Release Notes*.

Use the following procedure for each node, one node at a time:

1. Move any currently running packages to an adoptive node, or if it is not appropriate to move the packages, simply halt them on the node.
2. Remove the node from active cluster membership using the `cmhaltnode` command.
3. Install the ServiceGuard patch that enables the use of VxVM disk groups in clusters. The patch ID is PHSS_23511.
4. Run `swinstall`, and install the VxVM software according to the instructions in the *VERITAS Volume Manager Version 3.1 Release Notes*. At the end of the install, the node reboots.
5. If the node is not configured to re-join the cluster on re-boot, use the `cmrunnode` command to return the node to active cluster membership.
6. Move packages back to the node from the adoptive node or, if the packages were halted in step 1, re-start them on the node.

Creating a Storage Infrastructure with VxVM

This section shows how to configure storage using the command set of the VERITAS Volume Manager (VxVM). Before starting, make sure the directory in which VxVM commands are stored (`/usr/lib/vxvm/bin`) is in your path. Once you have created the root disk group (described next), you can use VxVM commands or the Storage Administrator GUI, `vmsa`, to carry out configuration tasks. If you are using `vmsa`, be sure the Storage Administrator server is running before you launch the GUI. Details are given in the *VERITAS Volume Manager 3.1 Release Notes*. For more information, refer to the *VERITAS Volume Manager 3.1 Storage Administrator Administrator's Guide*. If you are using commands, refer to the VxVM man pages.

Creating a Root Disk Group

If you are about to create disk groups for the first time, you need to initialize the Volume Manager. This is done by creating a disk group on each node known as *rootdg* that contains at least one disk. Use the following command after installing VxVM on each node:

```
# vxinstall
```

This displays a menu-driven program that steps you through the VxVM initialization sequence. From the main menu, choose the "Custom" option, and specify the disk you wish to include in *rootdg*.

IMPORTANT

The *rootdg* in the VxVM Volume Manager is not the same as the HP-UX root disk, and *cannot* be used for the HP-UX root file system (/). Note also that *rootdg* cannot be used for shared storage. However, *rootdg* can be used for other local filesystems (e.g., `/export/home`), so it need not be wasted. Creating a *rootdg* disk group is only necessary the first time you use the Volume Manager.

Initializing Disks for VxVM

You need to initialize the physical disks that will be employed in VxVM disk groups. To initialize a disk, log on to one node in the cluster, then use the `vxdiskadm` program to initialize multiple disks, or use the `vxdisksetup` command to initialize one disk at a time, as in the following example:

```
# vxdisksetup -i c0t3d2
```

Initializing Disks Previously Used by LVM

If a physical disk has been previously used with LVM, you should use the `pvremove` command to delete the LVM header data from all the disks in the volume group. In addition, if the LVM disk was previously used in a cluster, you will have to re-initialize the disk with the `pvcreate -f` command first, to remove the cluster ID from the disk.

NOTE These commands make the disk and its data unusable by LVM, and allow it to be initialized by VxVM. (The commands should only be used if you have previously used the disk with LVM and do not want to save the data on it.)

You can remove LVM header data from the disk as in the following example (note that all data on the disk will be erased):

```
# pvcreate -f /dev/rdsk/c0t3d2
```

```
# pvremove /dev/rdsk/c0t3d2
```

Then, use the `vxdiskadm` program to initialize multiple disks for VxVM, or use the `vxdisksetup` command to initialize one disk at a time, as in the following example:

```
# vxdisksetup -i c0t3d2
```

Creating Disk Groups

Use `vxdiskadm`, or use the `vxdg` command to create disk groups, as in the following example:

```
# vxdg init logdata c0t3d2
```

Verify the configuration with the following command:

```
# vxdg list
NAME          STATE                ID

rootdg         enabled             971995699.1025.node1
logdata        enabled             972078742.1084.node1
```

Creating Volumes

Use the `vxassist` command to create logical volumes. The following is an example:

```
# vxassist -g logdata make log_files 1024m
```

This command creates a 1024 MB volume named *`log_files`* in a disk group named logdata. The volume can be referenced with the block device file `/dev/vx/dsk/logdata/log_files` or the raw (character) device file `/dev/vx/rdsk/logdata/log_files`. Verify the configuration with the following command:

```
# vxprint -g logdata
```

The output of this command is shown in the following example:

```
TY  NAME       ASSOC        KSTATE   LENGTH   PLOFFS   STATE   TUTILO  PUTILO

v   logdata    fsgen        ENABLED  1024000          ACTIVE
pl  logdata-01 system       ENABLED  1024000          ACTIVE
```

Creating File Systems

If your installation uses file systems, create them next. Use the following commands to create a file system for mounting on the logical volume just created:

1. Create the file system on the newly created volume:

   ```
   # newfs -F vxfs /dev/vx/rdsk/logdata/log_files
   ```

2. Create a directory to mount the volume:

   ```
   # mkdir /logs
   ```

3. Mount the volume:

   ```
   # mount /dev/vx/dsk/logdata/log_files /logs
   ```

4. Check to make sure the file system is present, then unmount the file system:

   # **umount /logs**

Deporting Disk Groups

After creating the disk groups that are to be used by ServiceGuard packages, use the following command with each disk group to allow the disk group to be imported by the package control script on several cluster nodes:

vxdg deport <DiskGroupName>

where **<DiskGroupName>** is the name of the disk group that will be activated by the control script.

When all disk groups have been deported, you must issue the following command on all cluster nodes to allow them to access the disk groups:

vxdctl enable

Re-Importing Disk Groups

After deporting disk groups, they are not available for use on the node until they are imported again either by a package control script or with a `vxdg import` command. If you need to import a disk group manually for maintenance or other purposes, you import it, start up all its logical volumes, and mount filesystems as in the following example:

vxdg import dg_01

vxvol -g dg_01 startall

mount /dev/vx/dsk/dg_01/myvol /mountpoint

NOTE Unlike LVM volume groups, VxVM disk groups are *not* entered in the cluster ASCII configuration file.

Clearimport at System Reboot Time

At system reboot time, the cmcluster RC script does a `vxdisk clearimport` on all disks formerly imported by the system, provided

they have the *noautoimport* flag set, and provided they are not currently imported by another running node. The `clearimport` clears the host ID on the disk group, to allow any node that is connected to the disk group to import it when the package moves from one node to another.

Using the clearimport at reboot time allows ServiceGuard to clean up following a node failure, for example, a system crash during a power failure. Disks that were imported at the time of the failure still have the node's ID written on them, and this ID must be cleared before the rebooting node or any other node can import them with a package control script.

Note that the `clearimport` is done for disks previously imported with *noautoimport* set on *any* system that has ServiceGuard installed, whether it is configured in a cluster or not.

Creating the Package Control Script

The package control script contains all the information necessary to run all the services in the package, monitor them during operation, react to a failure, and halt the package when necessary.

Each package must have a separate control script, which must be executable. The control script is placed in the package directory and is given the same name as specified in the RUN_SCRIPT and HALT_SCRIPT parameters in the package ASCII configuration file. The package control script template contains both the run instructions and the halt instructions for the package. You can use a single script for both run and halt operations, or, if you wish, you can create separate scripts.

Use the following procedure to create a control script for the sample package *pkg1*.

First, generate a control script template:

```
# cmmakepkg -s /etc/cmcluster/pkg1/pkg1.sh
```

You may customize the script, as described in the next section.

Customizing the Package Control Script

Check the definitions and declarations at the beginning of the control script using the information in the Package Configuration worksheet. You need to customize as follows:

- Update the `PATH` variable to reflect any required paths needed to start your services.
- If you are using LVM, enter the names of volume groups to be activated using the `VG[]` array parameters, and select the appropriate options for the storage activation command, including options for mounting and unmounting filesystems, if desired. Do *not* use the `VXVM_DG[]` parameters for LVM volume groups.
- If you are using VxVM disk groups, enter the names of VxVM disk groups that will be imported using the `VXVM_DG[]` array parameters. Enter one disk group per array element. Do *not* use the `VG[]` parameter for VxVM disk groups. You do *not* specify an activation command.
- Add the names of logical volumes and file systems that will be mounted on them.
- Define IP subnet and IP address pairs for your package.
- Add service name(s).
- Add service command(s).
- Add a service restart parameter, if desired.

NOTE

Use care in defining service run commands. Each run command is executed by the control script in the following way:

- The `cmrunserv` command executes each run command and then monitors the process id of the process created by the run command.
- When the command started by `cmrunserv` exits, ServiceGuard determines that a failure has occurred and takes appropriate action, which may include transferring the package to an adoptive node.
- If a run command is a shell script that runs some other command and then exits, ServiceGuard will consider this exit as a *failure*.

To avoid problems in the execution of control scripts, ensure that each run command is the name of an actual service and that its process remains alive until the actual service stops.

If you need to define a set of run and halt operations in addition to the defaults, create functions for them in the sections under the heading `CUSTOMER_DEFINED_FUNCTIONS`.

How Control Scripts Manage VxVM Disk Groups

VxVM disk groups are outside the control of the ServiceGuard cluster. The package control script uses standard VxVM commands to import and deport these disk groups. (For details on importing and deporting disk groups, refer to the discussion of the *`import`* and *`deport`* options in the `vxdg` man page.)

The control script imports disk groups using the `vxdg` command with the `-tfC` options. The `-t` option specifies that the disk is imported with the *`noautoimport`* flag, which means that the disk will not be automatically re-imported at boot time. Since disk groups included in the package control script are only imported by ServiceGuard packages, they should not be auto-imported.

The `-f` option allows the disk group to be imported even if one or more disks (a mirror, for example) is not currently available. The `-C` option clears any existing host ID that might be written on the disk from a prior activation by another node in the cluster. If the disk had been in use on another node which has gone down with a TOC, then its host ID may still be written on the disk, and this needs to be cleared so the new node's ID can be written to the disk. Note that the disk groups are not imported clearing the host ID if the host ID is set and matches a node that is not in a failed state. This is to prevent accidental importation of a disk group on multiple nodes which could result in data corruption.

WARNING **Although ServiceGuard uses the -C option within the package control script framework, this option should not normally be used from the command line. The "Troubleshooting" section shows some situations where you might need to use -C from the command line.**

The following example shows the command with the same options that are used by the control script:

```
# vxdg -tfC import dg_01
```

This command takes over ownership of all the disks in disk group dg_01, even though the disk currently has a different host ID written on it. The command writes the current node's host ID on all disks in disk group dg_01 and sets the **noautoimport** flag for the disks. This flag prevents a disk group from being automatically re-imported by a node following a reboot. If a node in the cluster fails, the host ID is still written on each disk in the disk group. However, if the node is part of a ServiceGuard cluster then on reboot the host ID will be cleared by the owning node from all disks which have the noautoimport flag set, even if the disk group is not under ServiceGuard control. This allows all cluster nodes, which have access to the disk group, to be able to import the disks as part of cluster operation.

The control script also uses the `vxvol startall` command to start up the logical volumes in each disk group that is imported.

Package Control Script Template File

The following is an excerpt from the package control script template file. The file contains separate sections for activation of VxVM and LVM storage groups:

```
# **********************************************************************
# *                                                                    *
# *        HIGH AVAILABILITY PACKAGE CONTROL SCRIPT (template)         *
# *                                                                    *
# *       Note: This file MUST be edited before it can be used.        *
# *                                                                    *
# **********************************************************************

# UNCOMMENT the variables as you set them.

# Set PATH to reference the appropriate directories.
PATH=/sbin:/usr/bin:/usr/sbin:/etc:/bin

# VOLUME GROUP ACTIVATION:
# Specify the method of activation for volume groups.
# Leave the default ("VGCHANGE="vgchange -a e") if you want volume
# groups activated in exclusive mode. This assumes the volume groups have
# been initialized with 'vgchange -c y' at the time of creation.
#
# Uncomment the first line (VGCHANGE="vgchange -a e -q n"), and comment
# out the default, if your disks are mirrored on separate physical paths.
#
# Uncomment the second line (VGCHANGE="vgchange -a e -q n -s"), and comment
# out the default, if your disks are mirrored on separate physical paths,
# and you want the mirror resynchronization to occur in parallel with
```

Customizing the Package Control Script

```
# the package startup.
#
# Uncomment the third line (VGCHANGE="vgchange -a y") if you wish to
# use non-exclusive activation mode. Single node cluster configurations
# must use non-exclusive activation.
#
#
# VGCHANGE="vgchange -a e -q n"
# VGCHANGE="vgchange -a e -q n -s"
# VGCHANGE="vgchange -a y"

VGCHANGE="vgchange -a e"# Default

# VOLUME GROUPS
# Specify which volume groups are used by this package. Uncomment VG[0]=""
# and fill in the name of your first volume group. You must begin with
# VG[0], and increment the list in sequence.
#
# For example, if this package uses your volume groups vg01 and vg02, enter:
#         VG[0]=vg01
#         VG[1]=vg02
#
# The volume group activation method is defined above. The filesystems
# associated with these volume groups are specified below.
#
#VG[0]=""

# VxVM DISK GROUPS
# ********************************************************************
# *                                                                  *
# * NOTE: This control script template enables the use of the VxVM     *
# * volume manager in place or in addition to the use of LVM in order  *
# * to manage your mass storage subsystem.                           *
# *                                                                  *
# * Before attempting to install, configure or use the VxVM volume     *
# * manager, the user should first read and understand the SG/VxVM     *
# * Whitepaper that describes the SG VxVM interactions. This           *
# * whitepaper can be found at:                                      *
# *                                                                  *
# * http://docs.hp.com/hpux/ha                                       *
# *                                                                  *
# * and is entitled:                                                 *
# * "Integrating VERITAS Volume Manager (VxVM) with                  *
# *  MC/ServiceGuard A.11.09"                                        *
# *                                                                  *
# * The following should be noted for VxVM use:                      *
# *                                                                  *
# * 1. VxVM is only supported from HP on the HP-UX 11i OS. As a        *
# * result SG or SGOPS with VxVM is only supported on HP-UX 11i.       *
# *                                                                  *
# * 2. VxVM does not provide shared activation.                      *
# *                                                                  *
# * 3. VxVM cannot be used with MetroCluster with Continous Access     *
# * XP or MetroCluster with EMC SRDF.                                *
# *                                                                  *
# * 4. This script should only be used on systems that have patch      *
# * PHSS_23511 or later installed. Other configurations are not        *
# * supported.                                                       *
# ********************************************************************

# Specify which VxVM disk groups are used by this package. Uncomment
# VXVM_DG[0]="" and fill in the name of your first disk group. You must
# begin with VXVM_DG[0], and increment the list in sequence.
```

```
#
# For example, if this package uses your disk groups dg01 and dg02, enter:
#         VXVM_DG[0]="dg01"
#         VXVM_DG[1]="dg02"
##
#VXVM_DG[0]=""
#
# NOTE: A package could have LVM volume groups and VxVM
#       disk groups.
#
# FILESYSTEMS
# Specify the filesystems which are used by this package. Uncomment
# LV[0]=""; FS[0]=""; FS_MOUNT_OPT[0]="" and fill in the name of your first
# logical volume, filesystem and mount option for the file system. You must
# begin with LV[0], FS[0] and FS_MOUNT_OPT[0] and increment the list in
# sequence.
#
# For the LVM example, if this package uses the file systems pkg1a and
# pkg1b, which are mounted on the logical volumes lvol1 and lvol2 with
# read and write options enter:
#          LV[0]="/dev/vg01/lvol1"; FS[0]="/pkg1a"; FS_MOUNT_OPT[0]="-o rw"
#          LV[1]="/dev/vg01/lvol2"; FS[1]="/pkg1b"; FS_MOUNT_OPT[1]="-o rw"
#
#
# For the VxVM example, if this package uses the file systems
# pkg1a and pkg1b, which are mounted on the volumes lvol1 and lvol2
# with read and write options enter:
#      LV[0]="/dev/vx/dsk/dg01/lvol1"; FS[0]="/pkg1a"; FS_MOUNT_OPT[0]="-o rw"
#      LV[1]="/dev/vx/dsk/dg01/lvol2"; FS[1]="/pkg1b"; FS_MOUNT_OPT[1]="-o rw"
#
# The filesystems are defined as triplets of entries specifying the logical
# volume, the mount point and the mount options for the file system. Each
# filesystem will be fsck'd prior to being mounted. The filesystems will be
# mounted in the order specified during package startup and will be unmounted
# in reverse order during package shutdown. Ensure that volume groups
# referenced by the logical volume definitions below are included in
# volume group definitions above.
#
#LV[0]=""; FS[0]=""; FS_MOUNT_OPT[0]=""
#
# FILESYSTEM UNMOUNT COUNT
# Specify the number of unmount attempts for each filesystem during package
# shutdown.  The default is set to 1.
FS_UMOUNT_COUNT=1

# FILESYSTEM MOUNT RETRY COUNT.
# Specify the number of mount retrys for each filesystem.
# The default is 0. During startup, if a mount point is busy
# and FS_MOUNT_RETRY_COUNT is 0, package startup will fail and
# the script will exit with 1.  If a mount point is busy and
# FS_MOUNT_RETRY_COUNT is greater than 0, the script will attempt
# to kill the user responsible for the busy mount point
# and then mount the file system.  It will attempt to kill user and
# retry mount, for the number of times specified in FS_MOUNT_RETRY_COUNT.
# If the mount still fails after this number of attempts, the script
# will exit with 1.
# NOTE: If the FS_MOUNT_RETRY_COUNT > 0, the script will execute
# "fuser -ku" to freeup busy mount point.
FS_MOUNT_RETRY_COUNT=0

# IP ADDRESSES
# Specify the IP and Subnet address pairs which are used by this package.
# Uncomment IP[0]="" and SUBNET[0]="" and fill in the name of your first
```

Customizing the Package Control Script

```
# IP and subnet address. You must begin with IP[0] and SUBNET[0] and
# increment the list in sequence.
#
# For example, if this package uses an IP of 192.10.25.12 and a subnet of
# 192.10.25.0 enter:
#          IP[0]=192.10.25.12
#          SUBNET[0]=192.10.25.0 # (netmask=255.255.255.0)
#
# Hint: Run "netstat -i" to see the available subnets in the Network field.
#
# IP/Subnet address pairs for each IP address you want to add to a subnet
# interface card.  Must be set in pairs, even for IP addresses on the same
# subnet.
#
#IP[0]=""
#SUBNET[0]=""

# SERVICE NAMES AND COMMANDS.
# Specify the service name, command, and restart parameters which are
# used by this package. Uncomment SERVICE_NAME[0]="", SERVICE_CMD[0]="",
# SERVICE_RESTART[0]="" and fill in the name of the first service, command,
# and restart parameters. You must begin with SERVICE_NAME[0], SERVICE_CMD[0],
# and SERVICE_RESTART[0] and increment the list in sequence.
#
# For example:
#          SERVICE_NAME[0]=pkg1a
#          SERVICE_CMD[0]="/usr/bin/X11/xclock -display 192.10.25.54:0"
#          SERVICE_RESTART[0]=""  # Will not restart the service.
#
#          SERVICE_NAME[1]=pkg1b
#          SERVICE_CMD[1]="/usr/bin/X11/xload -display 192.10.25.54:0"
#          SERVICE_RESTART[1]="-r 2"    # Will restart the service twice.
#
#          SERVICE_NAME[2]=pkg1c
#          SERVICE_CMD[2]="/usr/sbin/ping"
#          SERVICE_RESTART[2]="-R" # Will restart the service an infinite
#                                  number of times.
#
# Note: No environmental variables will be passed to the command, this
# includes the PATH variable. Absolute path names are required for the
# service command definition.  Default shell is /usr/bin/sh.
#
#SERVICE_NAME[0]=""
#SERVICE_CMD[0]=""
#SERVICE_RESTART[0]=""

# DEFERRED_RESOURCE_NAME

# Specify the full path name of the 'DEFERRED' resources configured for
# this package. Uncomment DEFERRED_RESOURCE_NAME[0]="" and fill in the
# full path name of the resource.
#
# DEFERRED_RESOURCE_NAME[0]=""

# DTC manager information for each DTC.
# Example: DTC[0]=dtc_20
#DTC_NAME[0]=
```

The above excerpt from the control script shows the assignment of values to a set of variables. The remainder of the script uses these variables to control the package by executing LVM or VxVM commands, HP-UX

commands, and ServiceGuard commands including cmrunserv, cmmodnet, and cmhaltserv. Examine a copy of the control script template to see the flow of logic. Use the following command:

```
# cmmakepkg -s | more
```

The main function appears at the end of the script.

Note that individual variables are optional; you should include only as many as you need for proper package operation. For example, if your package does not need to activate a volume group, omit the VG or VXVM_DG variables; if the package does not use services, omit the corresponding SERVICE_NAME, SERVICE_CMD, and SERVICE_RESTART variables; and so on.

If you have defined an EMS resource in the package configuration file that is labeled as DEFERRED, you need to define a DEFERRED_RESOURCE_NAME in the package control script. Specify only the deferred resources, using the DEFERRED_RESOURCE_NAME parameter:

```
DEFERRED_RESOURCE_NAME[0]="/net/interfaces/lan/status/lan0"
DEFERRED_RESOURCE_NAME[1]="/net/interfaces/lan/status/lan1"
```

After customizing the script, distribute it to each node in the cluster using rcp, ftp, or your favorite method of copying.

Troubleshooting

This section describes some approaches to solving problems that may occur with VxVM disk groups in a cluster environment. For most problems, it is helpful to use the vxdg print command to display the disk groups currently imported on a specific node. Also, you should consult the package control script log files for messages associated with importing and deporting disk groups on particular nodes.

Force Import and Deport After Node Failure

After certain failures, packages configured with VxVM disk groups will fail to start, and the following error will be seen in the package log file:

```
vxdg: Error dg_01 may still be imported on ftsys9
```

```
        ERROR:  Function check_dg failed
```

This can happen if a package is running on a node which then fails before the package control script can deport the disk group. In these cases, the host name of the node that had failed is still written on the disk group header.

When the package starts up on another node in the cluster, a series of messages is printed as in the following example (the hostname of the failed system is ftsys9, and the disk group is dg_01):

```
check_dg: Error dg_01 may still be imported on ftsys9

To correct this situation, logon to ftsys9 and
execute the following command:
vxdg deport dg_01

Once dg_01 has been deported from ftsys9,
this package may be restarted via either cmmodpkg(1M)
or cmrunpkg(1M).

In the event that ftsys9 is either powered off
or unable to boot, then dg_01 must be force
imported.

****************** WARNING**************************

The use of force import can lead to data corruption if
ftsys9 is still running and has dg_01
imported. It is imperative to positively determine that
ftsys9 is not running prior to performing the force
import. See -C option on vxdg(1M).

*****************************************************

To force import dg_01, execute the following
commands on the local system:
    vxdg -tfC import $vg
    vxdg deport $vg
```

Follow the instructions in the message to use the force import option (-C) to allow the current node to import the disk group. Then deport the disk group, after which it can be used again by the package. Example:

```
# vxdg -tfC import dg_01
# vxdg deport dg_01
```

The force import will clear the host name currently written on the disks in the disk group, after which you can deport the disk group without error so it can then be imported by a package running on a different node.

CAUTION This force import procedure should only be used when you are certain the disk is not currently being accessed by another node. If you force import a disk that is already being accessed on another node, data corruption can result.