Setting up HP SIM 5.x on an HP-UX Serviceguard Cluster

Abstract

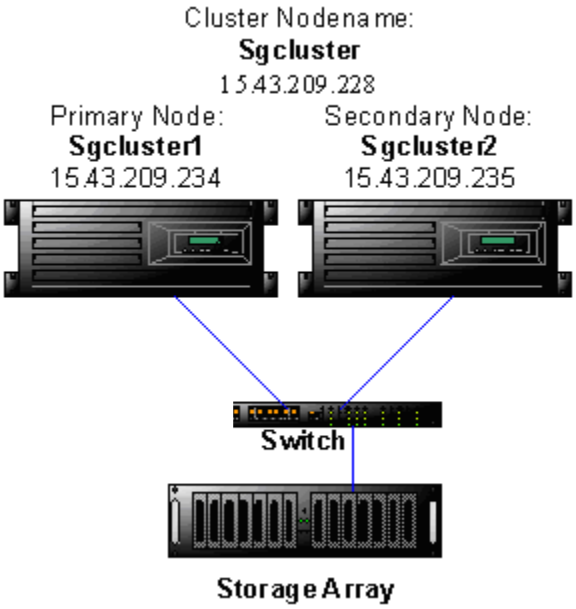
This white paper is intended to assist IT professionals installing HP Systems Insight Manager 5.x (HP SIM) on HP Serviceguard Clusters for HP-UX. The following scenario is explained.

- Setting up the common share volume (Primary and Secondary Node)
- Setting up Serviceguard
- Setting up the Application
- Setting up the Package and Services
- Testing the Failover

Introduction

HP Serviceguard is a specialized high availability clustering software for protecting mission-critical applications from a wide variety of hardware and software failures. Installing HP Systems Insight Manager on a high availability computer system will allow application services to continue in spite of a hardware or software failure. Highly available systems protect users from software failures as well as from failure of a system processing unit (SPU), disk, or local area network (LAN) component. In the event that one component fails, the redundant component takes over, allowing HP SIM to continue monitoring your environment. This paper assumes HP Serviceguard is installed and running. Figure 1 below displays the cluster configuration used for this paper with each node assigned an IP Address.

Figure 1 Basic cluster configuration with Storage Array



Getting started

Target audience

It is assumed that readers of this document have an excellent understanding of the following concepts and services:

- HP-UX 11v1, HP-UX 11v2
- TCP/IP and Clustering concepts
- HP Serviceguard
- PostgreSQL

Settings up the common shared volume

Note: Each disk size is approximately 2GB with 586 physical elements size of 4MB.

On the primary node (sgcluster1), create the shared volume as follow:

1. Enter `mkdir /dev/vg01`.
2. Enter `mknod /dev/vg01/group c 64 0x010000`.
3. Enter `pvcreate –f /dev/rdsk/c4t0d1`.
4. Enter `vgcreate /dev/vg01 /dev/dsk/c4t0d1`.
5. Enter `lvcreate –1 586 –n hpsimsg /dev/vg01`.
6. Enter `newfs –F vxfs /dev/vg01/rhpsimsg`.

   Note: creating new file system on new logical volume.

7. Enter `mkdir /hpsimsg.`

Create new mount point:

1. Enter `mount –F vxfs /dev/vg01/hpsimsg /hpsimsg`.
2. Enter vgexport –v –p –s –m /etc/lvmconf/vg01.mapfile /dev/vg01.

   Copy the LVM map file for the volume group to the secondary or standby machine to the location `/etc/lvmconf/vg01.mapfile`. To perform the following command, the `.rhosts` file must be configured on both of the cluster nodes.

   ```
   rcp /etc/lvmconf/vg01.mapfile sgcluster2:/etc/lvmconf/vg01.mapfile
   ```

3. Modify `/etc/lvmrc` and set *AUTO_VG_ACTIVATE* to 0.
4. Enter # `unmount/hpsimsg.`

On the secondary or standby node (sgcluster2), do the following:

1. Enter `mkdir /dev/vg01`.
2. Enter `mknod /dev/vg01/group c 64 0x010000.`
3. Enter `vgimport –v –s –m /etc/lvmconf/vg01.mapfile /dev/vg01`.
4. Enter `vgchange –a y vg01`.
5. Enter `vgcfgbackup /dev/vg01`.
6. Enter `mkdir /hpsimsg`.
7. Ensure that logical volume, `hpsimsg`, exist by mounting it by entering `mount –F vxfs /dev/vg01/hpsimsg /hpsimsg.`
8. Enter `umount /hpsimsg`.

9. Enter `vgchange –a n vg01`.
10. Modify `/etc/lvmrc` and set *AUTO_VG_ACTIVATE* to 0

Setting up Serviceguard

Set up Serviceguard on the primary node by executing the following commands.

1. Enter `cd /etc/cmcluster.`
2. Create the cluster configuration file:

   ```
   cmquerycl –v –C cmclconfig.ascii –n sgcluster1 –n sgcluster2
   ```

3. Edit the cmclconfig.ascii and make the following changes:

   ```
   CLUSTER_NAME  sgcluster
   HEARTBEAT_INTERVAL  3000000
   NODE_TIMEOUT  6000000
   AUTO_START_TIMEOUT  800000000
   ```

4. Ensure the configuration file contains no error.

   ```
   cmcheckconf –v –C cmclconfig.ascii
   ```

5. Deploy the configuration file:

   ```
   cmapplyconf –v –C cmclconfig.ascii
   ```

6. Bring up the cluster and verify that it work:

   ```
   cmruncl –v
   cmviewcl –v
   ```

7. Stop the cluster:

   ```
   cmhaltcl –v
   ```

Edit the `/etc/rc.config.d/cmcluster` file on both nodes and set *AUTOSTART_CMCLD* to 1

Setting up the application

On the primary node, install HP SIM 5.x, and then initialize HP SIM by executing the command:

```
mxinitconfig –a
```

After mxinitconfig completes, launch a supported browser to interact with HP SIM. Step through the first time wizard to configure the product and to run the initial discovery.

Next shutdown HP SIM by executing the following commands:

```
mxstop
```

/sbin/init.d/hpsmdb stop

Note: Edit the `/etc/passwd` file on each server: Primary and Secondary. Find the entry for `hpsmdb`. Note the current user ID. In this example, it is 104. Change the user ID to a value that is not used on either node. Our example will use 109.

1. Edit `/etc/passwd file`:

   vi /etc/passwd

2. Change the line beginning with `hpsmdb` to reflect the new user ID number:

On the secondary node, cut and paste the whole line if it is not present.

Original: `hpsmdb:*:104:20::/home/hpsmdb:/sbin/sh`

Change to: `hpsmdb:*:109:20::/home/hpsmdb:/sbin/sh`

3. Change the user ownership of `hpsmdb` to reflect the new userID number. This can be done by executing the following command:

```
find /var/opt/hpsmdb –user 104 –exec chown 109 {} \;
```

Configure the product to run in the Serviceguard environment by following the steps below.

1. Run `swverify HPSIM-HP-UX` and note any errors. There should not be any. It should come back with "Verification succeeded."
2. Active the volume group:

```
vgchange –c n vg01
vgchange –a y vg01
```

3. Enter `mount –F vxfs /dev/vg01/hpsimsg /hpsimsg`.
4. Enter `cd /hpsimsg`.

Steps 5 - 6, will create the subdirectory within /hpsimsg

5. Enter `mkdir opt`.
6. Enter `mkdir –p etc/opt`.
7. Enter `mkdir –p var/opt`.
8. Enter `cd /opt`.

Step 9 - 14 copies the file archive and duplicate the directory tree from `/opt to /hpsimsg/opt`, next it will move the original file.

9. Enter `find mx | cpio –pdmuxv /hpsimsg/opt`.
10. Enter `find hpsmdb | cpio –pdmuxv /hpsimsg/opt`.
11. Enter `find hpwebadmin | cpio –pdmuxv /hpsimsg/opt`.
12. Enter `mv mx mx.sg`.
13. Enter `mv hpsmdb hpsmdb.sg`.
14. Enter `mv hpwebadmin hpwebadmin.sg`.

Steps 15 - 17 creates symbolic link to the original path for `mx`, `hpsmdb` and `hpwebadmin`.

Note: To verify the symbolic type *ll*, an arrow will point the original path.

15. Enter `ln –s /hpsimsg/opt/mx /opt/mx`.
16. Enter `ln –s /hpsimsg/opt/hpsmdb /opt/hpsmdb`.
17. Enter `ln –s /hpsimsg/opt/hpwebadmin /opt/hpwebadmin`.
18. Enter `cd /etc/opt`.

Steps 19 - 21 copies the file archive and duplicate the directory tree from `/etc/opt` to `/hpsimsg/etc/opt`, then move the file to `mx.sg`, and then creates a symbolic link to the original path.

19. Enter `find mx | cpio –pdmuxv /hpsimsg/etc/opt`.
20. Enter `mv mx mx.sg`.
21. Enter `ln –s /hpsimsg/etc/opt/mx /etc/opt/mx`.
22. Enter `cd /var/opt`.

Steps 23 – 28 copies the file archive and duplicate the directory tree from `/var/opt` to `/hpsimsg/var/opt` then creates a symbolic link to the original path.

23. Enter `find mx | cpio –pdmuxv /hpsimsg/var/opt`.
24. Enter `find hpsmdb | cpio –pdmuxv /hpsimsg/var/opt`.
25. Enter `mv mx mx.sg`.
26. Enter `mv hpsmdb hpsmdb.sg`.
27. Enter `ln –s /hpsimsg/var/opt/mx /var/opt/mx`.
28. Enter `ln –s /hpsimsg/var/opt/hpsmdb /var/opt/hpsmdb`.
29. Run `swverify` again on HPSIM-HP-UX. You should get the same results as for step #1.

Note: If `swverify` fails, run the command *swjob -a log nutmeg-0044 @ <hostname>:/* . Ensure all the symbolic links are connected.

30. Enter `cd /sbin/init.d.`
31. Enter `mv hpsmdb hpsmdb.sg`.
32. Enter `mv hpsim hpsim.sg.`
33. Enter `rcp hpsmdb.sg sgcluster2:/sbin/init.d`.
34. Enter `rcp hpsim.sg sglcuster2:/sbin/init.d`.
35. Ensure that HP SIM still run properly by executing the following commands:

```
/sbin/init.d/hpsmdb.sg start
```

```
mxstart
```

36. Shutdown the product by executing the following commands:

```
mxstop
/sbin/init.d/hpsmdb.sg stop
```

37. Enter `umount /hpsimsg`.
38. Enter `vgchange –a n vg01`.

On the secondary node (sgcluster2) perform the following steps:

39. Edit the `/etc/PATH` file and add the following to the path:

```
:/opt/mx/bin
```

40. Edit the `/etc/MANPATH` file and add the following to the path:

```
:/opt/hpsmdb/pgsql/man:/opt/mx/share/man/%L:/opt/mx/share/man
```

41. Log out and log back in to source the new path information.
42. Enter `cd /var/opt.`
43. Enter `mv mx mx.sg`.
44. Enter `cd /opt`.
45. Enter `mv hpwebadmin hpwebadmin.sg` if it exist..
46. Enter `ln –s /hpsimsg/opt/mx /opt/mx`.
47. Enter `ln –s /hpsimsg/etc/opt/mx /etc/opt/mx`.
48. Enter `ln –s /hpsimsg/var/opt/mx /var/opt/mx.`
49. Enter `ln –s /hpsimsg/opt/hpwebadmin /opt/hpwebadmin`.
50. Enter `ln –s /hpsimsg/opt/hpsmdb /opt/hpsmdb`.
51. Enter `ln –s /hpsimsg/var/opt/hpsmdb /var/opt/hpsmdb`.

Note: Be sure all the symbolic links are connected by running the long list command from each `/hpsimsg` directory to verify that the links are present. You will note an arrow connecting to the original path.

Step 52 creates the subdirectory `hpsim` in `/etc/cmcluster`.

52. Enter `mkdir /etc/cmcluster/hpsim`.

Verify that HP SIM is running properly on the secondary node.

53. Enter `vgchange –c n vg01`.
54. Enter `vgchange –a y vg01`.
55. Enter `mount /dev/vg01/hpsimsg /hpsimsg`.
56. Start the database by running:

    ```
    /sbin/init.d/hpsmdb.sg start
    ```

57. Start HP SIM by running:

    ```
    /opt/mx/bin/mxstart
    ```

58. Bring up a browser to interactive with HP SIM and verify that it runs properly from the secondary node.

    http://*servername*:280

59. If everything is running okay, stop the product by running:

    ```
    mxstop
    ```

60. Stop the database by running:

    ```
    /sbin/init.d/hpsmdb.sg stop
    ```

61. Enter `umount /hpsimsg`.
62. Enter `vgchange –a n vg01`.

On the primary node, finish configuring Serviceguard for the volume group by executing the following steps:

63. Start the cluster

    ```
    cmruncl –v
    ```

64. Set volume group as cluster aware

    ```
    vgchange –c y vg01
    ```

65. Halt the cluster

    ```
    cmhaltcl –v
    ```

Setting up the packages and service

Working from the primary node, perform the following steps to setup the package and service for Serviceguard.

1. Enter `cd /etc/cmcluster/hpsim.`
2. Create the file `hahpsim.sh`.
3. Enter `vi hahpsim.sh.`
4. Copy content as they appear in Appendix A hahpsim.sh into the file.

5. Set the permission for the file so that only root has access to `hahpsim.sh` by using the command:
   `chmod 700 hahpsim.sh`
6. Copy the file to the secondary node.
7. Enter `rcp hahpsim.sh sgnode2:/etc/cmcluster/hpsim/hahpsim.sh`.
8. Create the package configuration file:
   `cmmakepkg –v –p hpsim.conf`
9. Edit the hpsim.conf file and make the following changes:
   PACKAGE_NAME hahpsim
   NODE_NAME sgcluster1
   NODE_NAME sgcluster2
   RUN_SCRIPT /etc/cmcluster/hpsim/hpsim.cntl
   HALT_SCRIPT /etc/cmcluster/hpsim/hpsim.cntl
   SERVICE_NAME hahpsim_monitor
10. Create the package control file:
    `cmmakepkg –v –s hpsim.cntl`
11. Edit the `hpsim.cntl` file and make the following changes

    Note: The user will have to supply their own IP Address and Subnet:
    VOLUME GROUPS section add:
    VG[0]="vg01"

    FILESYSTEMS section add:
    LV[0]="/dev/vg01/hpsimsg"
    FS[0]="/hpsimsg"
    FS_MOUNT_OPT[0]="-o rw"
    FS_UMOUNT_OPT[0]=""
    FS_FSCK_OPT[0]=""
    FS_TYPE[0]="vxfs"

    FS_UMOUNT_COUNT=2
    IP ADDRESSES section add:
    IP[0]="15.43.209.228"
    SUBNET[0]="15.43.208.0"

    SERVICE NAMES AND COMMANDS section add:
    SERVICE_NAME[0]="hahpsim_monitor"
    SERVICE_CMD[0]="/etc/cmcluster/hpsim/hahpsim.sh monitor"
    SERVICE_RESTART[0]=""

    In the *customer_defined_run_cmds* function replace ":" with:
    `/etc/cmcluster/hpsim/hahpsim.sh start`

    In the *customer_defined_halt_cmds* function replace ":" with:
    `/etc/cmcluster/hpsim/hahpsim.sh stop`
12. Copy the control script to the other node:
    `rcp hpsim.cntl sgcluster2:/etc/cmcluster/hpsim`

13. Verify the configuration file:

```
cmcheckconf –v –P hpsim.conf
```

14. Compile the configuration file:

```
cmapplyconf –v –P hpsim.conf
```

Testing the failover

Start the cluster with the application running on the primary node by typing the following command:

cmruncl –v

cmviewcl –v

Launch a browser to interact with HP SIM:

http://sgcluster:280

Test the failover to the secondary node by stopping the database:

/sbin/init.d/hpsmdb.sg stop

HP SIM should stop running on the primary node and start up on the secondary node. Verify that the browser can still interact with HP SIM.

http://sgcluster:280/\

To fail back to the primary from the secondary node, execute the following commands:

cmmodpkg –e –n sgcluster1 hahpsim

/sbin/init.d/hpsmdb stop

After the fail back occurred, re-enable the secondary node for failover again:

cmmodpkg –e –n sgcluster2 hahpsim

Note: Use the cmviewcl –v command to view information about the high availability cluster.

Appendix A hahpsim.sh

```
#! /sbin/sh

# Copyright (c) 2006 Hewlett-Packard Development Company, L.P.

# High Availability HP System Insight Manager (SIM) script to
# start, stop, and monitor the hpsim product

# Usage: hahpsim.sh <action>
#        where action is start, stop, or monitor

MONITOR_INTERVAL=5


case $1 in

start)

  # start the database and then hpsim
  /sbin/init.d/hpsmdb.sg start
  sleep 15
  /opt/mx/lbin/mxdomainmgr
  /opt/mx/lbin/mxdtf

;;

stop)

  # stop hpsim and then the database
  /opt/mx/bin/mxstop
  sleep 45
  /sbin/init.d/hpsmdb.sg stop
  sleep 30

;;

monitor)

  # monitor the database and hpsim and make sure they are still running
  # if product is not running then exit with error
  echo $(date) Starting High Availability HP SIM Monitoring
  while true
  do
    # check to see if postmaster (database) is still running
    ps -ef | grep -v grep |grep hpsmdb |grep -q postmaster
    if [[ $? -ne 0 ]]
    then
      # postmasster (database) is not running
      echo $(date) Database is not running, exiting monitor
      exit 1
    fi

    # now check to see if hpsim is still running
    # this require checking for mxdominmgr and mxdtf
    ps -ef | grep -v grep | grep -q mxdomainmgr
    if [[ $? -ne 0 ]]
    then
      # mxdomainmgr is not running
      echo $(date) mxdomainmgr is not running, exiting monitor
      exit 2
    fi
```

```
    ps -ef |grep -v grep | grep -q mxdtf
    if [[ $? -ne 0 ]]
    then
      # mxdtf is not running
      echo $(date) mxdtf is not running, exiting monitor
      exit 3
    fi

    sleep $MONITOR_INTERVAL
  done

;;

*)

  # display usage and exit

  echo "  Usage: hahpsim.sh < start | stop | monitor >"
  exit 1

;;

esac
```

Appendix B hpsim.conf

```
# **********************************************************************
# ****** HIGH AVAILABILITY PACKAGE CONFIGURATION FILE (template) ******
# **********************************************************************
# ******* Note: This file MUST be edited before it can be used. *******
# * For complete details about package parameters and how to set them, *
# * consult the Serviceguard Extension for RAC manuals.
# **********************************************************************

# Enter a name for this package. This name will be used to identify the
# package when viewing or manipulating it.  It must be different from
# the other configured package names.

PACKAGE_NAME                    hahpsim

# Enter the package type for this package. PACKAGE_TYPE indicates
# whether this package is to run as a FAILOVER, MULTI_NODE, or
# SYSTEM_MULTI_NODE package.
#
#    FAILOVER     package runs on one node at a time and if a failure
#                 occurs it can switch to an alternate node.
#
#    MULTI_NODE   package runs on multiple nodes at the same time and
#                 can be independently started and halted on
#                 individual nodes. Failures of package components such
#                 as services, EMS resources  or subnets, will cause
#                 the package to be halted only on the node on which the
#                 failure occurred. Relocatable IP addresses cannot be
#                 assigned to MULTI_NODE packages.
#
#    SYSTEM_MULTI_NODE
#                 package runs on all cluster nodes at the same time.
#                 It can not be started and halted on individual nodes.
#                 Both NODE_FAIL_FAST_ENABLED and AUTO_RUN must be set
#                 to YES for this type of package. All SERVICES must
#                 have SERVICE_FAIL_FAST_ENABLED set to YES.
#
# NOTE: Packages which have a PACKAGE_TYPE of MULTI_NODE and
#       SYSTEM_MULTI_NODE are not failover packages and are only
#       supported for use by applications provided by Hewlett-Packard.
#
#       Since MULTI_NODE and SYSTEM_MULTI_NODE packages can run on more
#       than one node at a time and do not failover in the event of a
#       package failure, the following parameters cannot be
#       specified when configuring packages of these types:
#
#          FAILOVER_POLICY
#          FAILBACK_POLICY
#
#       Since an IP address can not be assigned to more than node at a
#       time, relocatable IP addresses can not be assigned in the
#       package control script for MULTI_NODE packages. If volume
#       groups are used in a MULTI_NODE package, they must be
#       activated in a shared mode and data integrity is left to the
#       application. Shared access requires a shared volume manager.
#
# Examples : PACKAGE_TYPE   FAILOVER (default)
#            PACKAGE_TYPE   MULTI_NODE
#            PACKAGE_TYPE   SYSTEM_MULTI_NODE
#
#
```

```
PACKAGE_TYPE                    FAILOVER


# Enter the failover policy for this package. This policy will be used
# to select an adoptive node whenever the package needs to be started.
# The default policy unless otherwise specified is CONFIGURED_NODE.
# This policy will select nodes in priority order from the list of
# NODE_NAME entries specified below.
#
# The alternative policy is MIN_PACKAGE_NODE. This policy will select
# the node, from the list of NODE_NAME entries below, which is
# running the least number of packages at the time this package needs
# to start.

FAILOVER_POLICY                 CONFIGURED_NODE


# Enter the failback policy for this package. This policy will be used
# to determine what action to take when a package is not running on
# its primary node and its primary node is capable of running the
# package. The default policy unless otherwise specified is MANUAL.
# The MANUAL policy means no attempt will be made to move the package
# back to its primary node when it is running on an adoptive node.
#
# The alternative policy is AUTOMATIC. This policy will attempt to
# move the package back to its primary node whenever the primary node
# is capable of running the package.

FAILBACK_POLICY                 MANUAL


# Enter the names of the nodes configured for this package.  Repeat
# this line as necessary for additional adoptive nodes.
#
# NOTE:   The order is relevant.
#         Put the second Adoptive Node after the first one.
#
# Example : NODE_NAME  original_node
#           NODE_NAME  adoptive_node
#
# If all nodes in the cluster are to be specified and order is not
# important, "NODE_NAME *" may be specified.
#
# Example : NODE_NAME  *

NODE_NAME               sgcluster1
NODE_NAME               sgcluster2


# Enter the value for AUTO_RUN. Possible values are YES and NO.
# The default for AUTO_RUN is YES. When the cluster is started the
# package will be automatically started. In the event of a failure the
# package will be started on an adoptive node. Adjust as necessary.
#
# AUTO_RUN replaces obsolete PKG_SWITCHING_ENABLED.

AUTO_RUN                YES


# Enter the value for LOCAL_LAN_FAILOVER_ALLOWED.
# Possible values are YES and NO.
# The default for LOCAL_LAN_FAILOVER_ALLOWED is YES.  In the event of a
```

```
# failure, this permits the cluster software to switch LANs locally
# (transfer to a standby LAN card).  Adjust as necessary.
#
# LOCAL_LAN_FAILOVER_ALLOWED replaces obsolete NET_SWITCHING_ENABLED.

LOCAL_LAN_FAILOVER_ALLOWED    YES


# Enter the value for NODE_FAIL_FAST_ENABLED.
# Possible values are YES and NO.
# The default for NODE_FAIL_FAST_ENABLED is NO.  If set to YES,
# in the event of a failure, the cluster software will halt the node
# on which the package is running. All SYSTEM_MULTI_NODE packages must
have
# NODE_FAIL_FAST_ENABLED set to YES.  Adjust as necessary.

NODE_FAIL_FAST_ENABLED        NO


# Enter the complete path for the run and halt scripts.  In most cases
# the run script and halt script specified here will be the same script,
# the package control script generated by the cmmakepkg command.  This
# control script handles the run(ning) and halt(ing) of the package.
#
# Enter the timeout, specified in seconds, for the run and halt scripts.
# If the script has not completed by the specified timeout value,
# it will be terminated.  The default for each script timeout is
# NO_TIMEOUT.  Adjust the timeouts as necessary to permit full
# execution of each script.
#
# Note: The HALT_SCRIPT_TIMEOUT should be greater than the sum of
# all SERVICE_HALT_TIMEOUT values specified for all services.
#
# The file where the output of the scripts is logged can be specified
# via the SCRIPT_LOG_FILE parameter. If not set, script output is sent
# to a file named by appending '.log' to the script path.
#
#SCRIPT_LOG_FILE

RUN_SCRIPT              /etc/cmcluster/hpsim/hpsim.cntl
RUN_SCRIPT_TIMEOUT            NO_TIMEOUT
HALT_SCRIPT             /etc/cmcluster/hpsim/hpsim.cntl
HALT_SCRIPT_TIMEOUT           NO_TIMEOUT


# Enter the names of the storage groups configured for this package.
# Repeat this line as necessary for additional storage groups.
#
# Storage groups are only used with CVM disk groups. Neither
# VxVM disk groups or LVM volume groups should be listed here.
# By specifying a CVM disk group with the STORAGE_GROUP keyword
# this package will not run until the CVM system multi node package is
# running and thus the CVM shared disk groups are ready for
# activation.
#
# NOTE: Should only be used by applications provided by
#       Hewlett-Packard.
#
# Example : STORAGE_GROUP  dg01
#           STORAGE_GROUP  dg02
#           STORAGE_GROUP  dg03
#           STORAGE_GROUP  dg04
#
```

```
# Enter the names of the dependency condition for this package.
# Dependencies are used to describe the relationship between packages
# To define a dependency, all three attributes are required.
#
# DEPENDENCY_NAME must have a unique identifier for the dependency.
#
# DEPENDENCY_CONDITION
#          This is an expression describing what must be true for
#          the dependency to be satisfied.
#
#          The syntax is: <package name> = UP , where <package name>
#          is the name of a multi-node or system multi-node package.
#
# DEPENDENCY_LOCATION
#           This describes where the condition must be satisfied.
#           The only possible value for this attribute is SAME_NODE
#
# NOTE:
# Dependencies can only be defined for packages within a CFS cluster.
These are
# automatically setup in the SYSTEM-MULTI-NODE and MULTI-NODE packages
created for
# disk groups and mount points. Customers configure dependencies for
FAILOVER type
# packages only; and the dependency would be on a MULTI-NODE mount point
(MP) package.
#
# Example :
#           DEPENDENCY_NAME         SG-CFS-MP-1
#           DEPENDENCY_CONDITION    SG-CFS-MP-1=UP
#           DEPENDENCY_LOCATION     SAME_NODE
#
#DEPENDENCY_NAME
#DEPENDENCY_CONDITION
#DEPENDENCY_LOCATION SAME_NODE
#


# Enter the SERVICE_NAME, the SERVICE_FAIL_FAST_ENABLED and the
# SERVICE_HALT_TIMEOUT values for this package.  Repeat these
# three lines as necessary for additional service names.  All
# service names MUST correspond to the SERVICE_NAME[] entries in
# the package control script.
#
# The value for SERVICE_FAIL_FAST_ENABLED can be either YES or
# NO.  If set to YES, in the event of a service failure, the
# cluster software will halt the node on which the service is
# running.  If SERVICE_FAIL_FAST_ENABLED is not specified, the
# default will be NO.
#
# SERVICE_HALT_TIMEOUT is represented as a number of seconds.
# This timeout is used to determine the length of time (in
# seconds) the cluster software will wait for the service to
# halt before a SIGKILL signal is sent to force the termination
# of the service.  In the event of a service halt, the cluster
# software will first send a SIGTERM signal to terminate the
# service.  If the service does not halt, after waiting for the
# specified SERVICE_HALT_TIMEOUT, the cluster software will send
# out the SIGKILL signal to the service to force its termination.
# This timeout value should be large enough to allow all cleanup
# processes associated with the service to complete.  If the
```

```
# SERVICE_HALT_TIMEOUT is not specified, a zero timeout will be
# assumed, meaning the cluster software will not wait at all
# before sending the SIGKILL signal to halt the service.
#
# Example: SERVICE_NAME                   DB_SERVICE
#          SERVICE_FAIL_FAST_ENABLED      NO
#          SERVICE_HALT_TIMEOUT           300
#
# To configure a service, uncomment the following lines and
# fill in the values for all of the keywords.
#
#SERVICE_NAME                    <service name>
#SERVICE_FAIL_FAST_ENABLED       <YES/NO>
#SERVICE_HALT_TIMEOUT            <number of seconds>

SERVICE_NAME                    hahpsim_monitor

# Enter the network subnet name that is to be monitored for this package.
# Repeat this line as necessary for additional subnet names.  If any of
# the subnets defined goes down, the package will be switched to another
# node that is configured for this package and has all the defined
subnets
# available.
# The subnet names could be IPv4 or IPv6. The network subnet
# names that are to be monitored for this package could be a mix
# of IPv4 or IPv6 subnet names

#SUBNET


# The keywords RESOURCE_NAME, RESOURCE_POLLING_INTERVAL,
# RESOURCE_START, and RESOURCE_UP_VALUE are used to specify Package
# Resource Dependencies.  To define a package Resource Dependency, a
# RESOURCE_NAME line with a fully qualified resource path name, and
# one or more RESOURCE_UP_VALUE lines are required.  The
# RESOURCE_POLLING_INTERVAL and the RESOURCE_START are optional.
#
# The RESOURCE_POLLING_INTERVAL indicates how often, in seconds, the
# resource is to be monitored.  It will be defaulted to 60 seconds if
# RESOURCE_POLLING_INTERVAL is not specified.
#
# The RESOURCE_START option can be set to either AUTOMATIC or DEFERRED.
# The default setting for RESOURCE_START is AUTOMATIC.  If AUTOMATIC
# is specified, Serviceguard will start up resource monitoring for
# these AUTOMATIC resources automatically when the node starts up.
# If DEFERRED is selected, Serviceguard will not attempt to start
# resource monitoring for these resources during node start up.  User
# should specify all the DEFERRED resources in the package run script
# so that these DEFERRED resources will be started up from the package
# run script during package run time.
#
# RESOURCE_UP_VALUE requires an operator and a value.  This defines
# the resource 'UP' condition.  The operators are =, !=, >, <, >=,
# and <=, depending on the type of value.  Values can be string or
# numeric.  If the type is string, then only = and != are valid
# operators.  If the string contains whitespace, it must be enclosed
# in quotes.  String values are case sensitive.  For example,
#
#                                       Resource is up when its value is
#                                       --------------------------------
#       RESOURCE_UP_VALUE       = UP                    "UP"
#       RESOURCE_UP_VALUE       != DOWN                 Any value except
"DOWN"
```

```
#       RESOURCE_UP_VALUE       = "On Course"         "On Course"
#
# If the type is numeric, then it can specify a threshold, or a range to
# define a resource up condition.  If it is a threshold, then any
operator
# may be used.  If a range is to be specified, then only > or >= may be
used
# for the first operator, and only < or <= may be used for the second
operator.
# For example,
#                                       Resource is up when its value is
#                                       --------------------------------
#       RESOURCE_UP_VALUE     = 5               5
(threshold)
#       RESOURCE_UP_VALUE     > 5.1             greater than 5.1
(threshold)
#       RESOURCE_UP_VALUE     > -5 and < 10     between -5 and 10
(range)
#
# Note that "and" is required between the lower limit and upper limit
# when specifying a range.  The upper limit must be greater than the
lower
# limit.  If RESOURCE_UP_VALUE is repeated within a RESOURCE_NAME block,
then
# they are inclusively OR'd together.  Package Resource Dependencies may
be
# defined by repeating the entire RESOURCE_NAME block.
#
# Example : RESOURCE_NAME               /net/interfaces/lan/status/lan0
#           RESOURCE_POLLING_INTERVAL   120
#           RESOURCE_START              AUTOMATIC
#           RESOURCE_UP_VALUE           = RUNNING
#           RESOURCE_UP_VALUE           = ONLINE
#
#           Means that the value of resource
/net/interfaces/lan/status/lan0
#           will be checked every 120 seconds, and is considered to
#           be 'up' when its value is "RUNNING" or "ONLINE".
#
# Uncomment the following lines to specify Package Resource Dependencies.
#
#RESOURCE_NAME              <Full_path_name>
#RESOURCE_POLLING_INTERVAL  <numeric_seconds>
#RESOURCE_START             <AUTOMATIC/DEFERRED>
#RESOURCE_UP_VALUE          <op> <string_or_numeric> [and <op> <numeric>]

# Access Control Policy Parameters.
#
# Three entries set the access control policy for the package:
# First line must be USER_NAME, second USER_HOST, and third USER_ROLE.
# Enter a value after each.
#
# 1. USER_NAME can either be ANY_USER, or a maximum of
#    8 login names from the /etc/passwd file on user host.
# 2. USER_HOST is where the user can issue Serviceguard commands.
#    If using Serviceguard Manager, it is the COM server.
#    Choose one of these three values: ANY_SERVICEGUARD_NODE, or
#    (any) CLUSTER_MEMBER_NODE, or a specific node. For node,
#    use the official hostname from domain name server, and not
#    an IP addresses or fully qualified name.
# 3. USER_ROLE must be PACKAGE_ADMIN. This role grants permission
#    to MONITOR, plus for administrative commands for the package.
#
```

```
# These policies do not effect root users. Access Policies here
# should not conflict with policies defined in the cluster configuration
file.
#
# Example: to configure a role for user john from node noir to
# administer the package, enter:
# USER_NAME  john
# USER_HOST  noir
# USER_ROLE  PACKAGE_ADMIN
```

Appendix C hpsim.cntl

```
# @(#) A.11.17.00 Date: 09/23/05  $
# ************************************************************************
# *                                                                      *
# *        HIGH AVAILABILITY PACKAGE CONTROL SCRIPT (template)           *
# *                                                                      *
# *       Note: This file MUST be edited before it can be used.          *
# *                                                                      *
# ************************************************************************

# The environment variables PACKAGE, NODE, SG_PACKAGE,
# SG_NODE and SG_SCRIPT_LOG_FILE are set by Serviceguard
# at the time the control script is executed.
# Do not set these environment variables yourself!
# The package may fail to start or halt if the values for
# these environment variables are altered.

# NOTE: Starting from 11.17, all environment variables set by
# Serviceguard implicitly at the time the control script is
# executed will contain the prefix "SG_". Do not set any variable
# with the defined prefix, or the control script may not
# function as it should.

. ${SGCONFFILE:=/etc/cmcluster.conf}

# UNCOMMENT the variables as you set them.

# Set PATH to reference the appropriate directories.
PATH=$SGSBIN:/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 ocurr in parallel with
# 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

# CVM DISK GROUP ACTIVATION:
# Specify the method of activation for CVM disk groups.
# Leave the default
# (CVM_ACTIVATION_CMD="vxdg -g \$DiskGroup set
activation=exclusivewrite")
# if you want disk groups activated in the exclusive write mode.
```

```
#
# Uncomment the first line
# (CVM_ACTIVATION_CMD="vxdg -g \$DiskGroup set activation=readonly"),
# and comment out the default, if you want disk groups activated in
# the readonly mode.
#
# Uncomment the second line
# (CVM_ACTIVATION_CMD="vxdg -g \$DiskGroup set activation=sharedread"),
# and comment out the default, if you want disk groups activated in the
# shared read mode.
#
# Uncomment the third line
# (CVM_ACTIVATION_CMD="vxdg -g \$DiskGroup set activation=sharedwrite"),
# and comment out the default, if you want disk groups activated in the
# shared write mode.
#
# CVM_ACTIVATION_CMD="vxdg -g \$DiskGroup set activation=readonly"
# CVM_ACTIVATION_CMD="vxdg -g \$DiskGroup set activation=sharedread"
# CVM_ACTIVATION_CMD="vxdg -g \$DiskGroup set activation=sharedwrite"
CVM_ACTIVATION_CMD="vxdg -g \$DiskGroup set activation=exclusivewrite"

# 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]="vg01"

# CVM DISK GROUPS
# Specify which cvm disk groups are used by this package. Uncomment
# CVM_DG[0]="" and fill in the name of your first disk group. You must
# begin with CVM_DG[0], and increment the list in sequence.
#
# For example, if this package uses your disk groups dg01 and dg02,
enter:
#         CVM_DG[0]=dg01
#         CVM_DG[1]=dg02
#
# The cvm disk group activation method is defined above. The filesystems
# associated with these volume groups are specified below in the CVM_*
# variables.
#
#CVM_DG[0]=""

# NOTE: Do not use CVM and VxVM disk group parameters to reference
# devices used by CFS (cluster file system).  CFS resources are
# controlled by the Disk Group and Mount Multi-node packages.
#
# VxVM DISK GROUPS
# 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
#
# The cvm disk group activation method is defined above.
#
#VXVM_DG[0]=""

#
# NOTE: A package could have LVM volume groups, CVM disk groups and VxVM
#       disk groups.
#
# NOTE: When VxVM is initialized it will store the hostname of the
#       local node in its volboot file in a variable called 'hostid'.
#       The Serviceguard package control scripts use both the values of
#       the hostname(1m) command and the VxVM hostid. As a result
#       the VxVM hostid should always match the value of the
#       hostname(1m) command.
#
#       If you modify the local host name after VxVM has been
#       initialized and such that hostname(1m) does not equal uname -n,
#       you need to use the vxdctl(1m) command to set the VxVM hostid
#       field to the value of hostname(1m). Failure to do so will
#       result in the package failing to start.

# VOLUME GROUP AND DISK GROUP DEACTIVATION RETRY COUNT
# Specify the number of deactivation retries for each disk group and
volume
# group at package shutdown. The default is 0.
DEACTIVATION_RETRY_COUNT=0


# RAW DEVICES
# If you are using raw devices for your application, this parameter
allows
# you to specify if you want to kill the processes that are accessing the
# raw devices at package halt time. If raw devices are still being
accessed
# at package halt time, volume group or disk group deactivation can fail,
# causing the package halt to also fail. This problem usually happens
when
# the application does not shut down properly.
# Note that if you are using Oracle's Cluster Ready Service, killing this
# service could cause the node to reboot.
# The legal values are "YES" and "NO". The default value is "NO".
# The value that is set for this parameter affects all raw devices
associated
# with the LVM volume groups and CVM disk groups defined in the package.
KILL_PROCESSES_ACCESSING_RAW_DEVICES="NO"

# FILESYSTEMS
# Filesystems are defined as entries specifying the logical volume, the
# mount point, the mount, umount and fsck options and type of 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.
#
# Specify the filesystems which are used by this package. Uncomment
```

```
# LV[0]=""; FS[0]=""; FS_MOUNT_OPT[0]=""; FS_UMOUNT_OPT[0]="";
FS_FSCK_OPT[0]=""
# FS_TYPE[0]="" and fill in the name of your first logical volume,
# filesystem, mount, umount and fsck options and filesystem type
# for the file system. You must begin with LV[0], FS[0],
# FS_MOUNT_OPT[0], FS_UMOUNT_OPT[0], FS_FSCK_OPT[0], FS_TYPE[0]
# and increment the list in sequence.
#
# Note: The FS_TYPE parameter lets you specify the type of filesystem to
be
# mounted. Specifying a particular FS_TYPE will improve package failover
time.
# The FSCK_OPT and FS_UMOUNT_OPT parameters can be used to include the
# -s option with the fsck and umount commands to improve performance for
# environments that use a large number of filesystems. (An example of a
# large environment is given below following the decription of the
# CONCURRENT_MOUNT_AND_UMOUNT_OPERATIONS parameter.)
#
# Example: If a package uses two JFS filesystems, pkg01a and pkg01b,
# which are mounted on LVM logical volumes lvol1 and lvol2 for read and
# write operation, you would enter the following:
#      LV[0]=/dev/vg01/lvol1; FS[0]=/pkg01a; FS_MOUNT_OPT[0]="-o rw";
#      FS_UMOUNT_OPT[0]=""; FS_FSCK_OPT[0]=""; FS_TYPE[0]="vxfs"
#
#      LV[1]=/dev/vg01/lvol2; FS[1]=/pkg01b; FS_MOUNT_OPT[1]="-o rw"
#      FS_UMOUNT_OPT[1]=""; FS_FSCK_OPT[1]=""; FS_TYPE[1]="vxfs"
#
LV[0]="/dev/vg01/hpsimsg"
FS[0]="/hpsimsg"
FS_MOUNT_OPT[0]="-o rw"
FS_UMOUNT_OPT[0]=""
FS_FSCK_OPT[0]=""
FS_TYPE[0]="vxfs"

#
# VOLUME RECOVERY
#
# When mirrored VxVM volumes are started during the package control
# bring up, if recovery is required the default behavior is for
# the package control script to wait until recovery has been
# completed.
#
# To allow mirror resynchronization to ocurr in parallel with
# the package startup, uncomment the line
# VXVOL="vxvol -g \$DiskGroup -o bg startall" and comment out the
default.
#
# VXVOL="vxvol -g \$DiskGroup -o bg startall"
VXVOL="vxvol -g \$DiskGroup startall"      # Default

# FILESYSTEM UNMOUNT COUNT
# Specify the number of unmount attempts for each filesystem during
package
# shutdown.  The default is set to 1.
FS_UMOUNT_COUNT=2

# 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

#
# Configuring the concurrent operations below can be used to improve the
# performance for starting up or halting a package.  The maximum value
for
# each concurrent operation parameter is 1024.  Set these values
carefully.
# The performance could actually decrease if the values are set too high
# for the system resources available on your cluster nodes.  Some
examples
# of system resources that can affect the optimum number of concurrent
# operations are: number of CPUs, amount of available memory, the kernel
# configuration for nfile and nproc. In some cases, if you set the number
# of concurrent operations too high, the package may not be able to start
# or to halt.  For example, if you set CONCURRENT_VGCHANGE_OPERATIONS=5
# and the node where the package is started has only one processor, then
# running concurrent volume group activations will not be beneficial.
# It is suggested that the number of concurrent operations be tuned
# carefully, increasing the values a little at a time and observing the
# effect on the performance, and the values should never be set to a
value
# where the performance levels off or declines.  Additionally, the values
# used should take into account the node with the least resources in the
# cluster, and how many other packages may be running on the node.
# For instance, if you tune the concurrent operations for a package so
# that it provides optimum performance for the package on a node while
# no other packages are running on that node, the package performance
# may be significantly reduced, or may even fail when other packages are
# already running on that node.
#
# CONCURRENT VGCHANGE OPERATIONS
# Specify the number of concurrent volume group activations or
# deactivations to allow during package startup or shutdown.
# Setting this value to an appropriate number may improve the performance
# while activating or deactivating a large number of volume groups in the
# package. If the specified value is less than 1, the script defaults it
# to 1 and proceeds with a warning message in the package control script
# logfile.
CONCURRENT_VGCHANGE_OPERATIONS=1

# CONCURRENT FSCK OPERATIONS
# Specify the number of concurrent fsck to allow during package startup.
# Setting this value to an appropriate number may improve the performance
# while checking a large number of file systems in the package. If the
# specified value is less than 1, the script defaults it to 1 and
proceeds
# with a warning message in the package control script logfile.
CONCURRENT_FSCK_OPERATIONS=1

# CONCURRENT MOUNT AND UMOUNT OPERATIONS
# Specify the number of concurrent mounts and umounts to allow during
# package startup or shutdown.
# Setting this value to an appropriate number may improve the performance
# while mounting or un-mounting a large number of file systems in the
package.
# If the specified value is less than 1, the script defaults it to 1 and
```

```
# proceeds with a warning message in the package control script logfile.
CONCURRENT_MOUNT_AND_UMOUNT_OPERATIONS=1

# Example:  If a package uses 50 JFS filesystems, pkg01aa through
pkg01bx,
# which are mounted on the 50 logical volumes lvol1..lvol50 for read and
write
# operation, you may enter the following:
#
#      CONCURRENT_FSCK_OPERATIONS=50
#      CONCURRENT_MOUNT_AND_UMOUNT_OPERATIONS=50
#
#      LV[0]=/dev/vg01/lvol1; FS[0]=/pkg01aa; FS_MOUNT_OPT[0]="-o rw";
#      FS_UMOUNT_OPT[0]="-s"; FS_FSCK_OPT[0]="-s"; FS_TYPE[0]="vxfs"
#
#      LV[1]=/dev/vg01/lvol2; FS[1]=/pkg01ab; FS_MOUNT_OPT[1]="-o rw"
#      FS_UMOUNT_OPT[1]="-s"; FS_FSCK_OPT[1]="-s"; FS_TYPE[0]="vxfs"
#         :           :          :
#         :           :          :
#         :           :          :
#      LV[49]=/dev/vg01/lvol50; FS[49]=/pkg01bx; FS_MOUNT_OPT[49]="-o rw"
#      FS_UMOUNT_OPT[49]="-s"; FS_FSCK_OPT[49]="-s"; FS_TYPE[0]="vxfs"
#
# IP ADDRESSES
# Specify the IP and Subnet address pairs which are used by this package.
# You could specify IPv4 or IPv6 IP and subnet address pairs.
# Uncomment IP[0]="" and SUBNET[0]="" and fill in the name of your first
# 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.
#
# For example, if this package uses an IPv6 IP of 2001::1/64
# The address prefix identifies the subnet as 2001::/64 which is an
available
# subnet.
# enter:
#          IP[0]=2001::1
#          SUBNET[0]=2001::/64
#          (netmask=ffff:ffff:ffff:ffff::)
# Alternatively the IPv6 IP/Subnet pair can be specified without the
prefix
# for the IPv6 subnet.
#          IP[0]=2001::1
#          SUBNET[0]=2001::
#          (netmask=ffff:ffff:ffff:ffff::)
#
# Hint: Run "netstat -i" to see the available IPv6 subnets by looking
# at the address prefixes
# 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]="15.43.209.228"
SUBNET[0]="15.43.208.0"
```

```
# 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]="hahpsim_monitor"
SERVICE_CMD[0]="/etc/cmcluster/hpsim/hahpsim.sh monitor"
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]=

# HA_NFS_SCRIPT_EXTENSION
# If the package uses HA NFS, this variable can be used to alter the
# name of the HA NFS script.  If not set, the name of this script is
# assumed to be "ha_nfs.sh".  If set, the "sh" portion of the default
# script name is replaced by the value of this variable.  So if
# HA_NFS_SCRIPT_EXTENSION is set to "package1.sh", for example, the name
# of the HA NFS script becomes "ha_nfs.package1.sh".  In any case,
# the HA NFS script must be placed in the same directory as the package
# control script.  This allows multiple packages to be run out of the
# same directory, as needed by SGeSAP.
#HA_NFS_SCRIPT_EXTENSION=""


# START OF CUSTOMER DEFINED FUNCTIONS

# This function is a place holder for customer define functions.
# You should define all actions you want to happen here, before the
service is
# started.  You can create as many functions as you need.
```

```
function customer_defined_run_cmds
{
# ADD customer defined run commands.
# : # do nothing instruction, because a function must contain some
command.

/etc/cmcluster/hpsim/hahpsim.sh start

      test_return 51
}

# This function is a place holder for customer define functions.
# You should define all actions you want to happen here, after the
service is
# halted.

function customer_defined_halt_cmds
{
# ADD customer defined halt commands.
# : # do nothing instruction, because a function must contain some
command.

/etc/cmcluster/hpsim/hahpsim.sh stop

      test_return 52
}

# END OF CUSTOMER DEFINED FUNCTIONS


# START OF RUN FUNCTIONS

.
.
.
The remaining content of this file was not changed.
```

For more information

Refer to the following for more information regarding the topics referenced in this paper.

HP Systems Insight Manager 5.x

Overview and features

http://h18013.www1.hp.com/products/servers/management/hpsim/index.html

HP Systems Insight Manager quick specs

http://h18013.www1.hp.com/products/servers/management/hpsim/quickspecs.html

HP SIM Command Line Interface Guide

http://h10018.www1.hp.com/wwsolutions/misc/hpsim-helpfiles/cliguide5.pdf

HP SIM Command Information Library

http://h18013.www1.hp.com/products/servers/management/hpsim/infolibrary.html

HP Serviceguard

Managing Serviceguard (12th edition)

http://lpdocs.cup.hp.com/LP/Documents/B3936-90100.pdf

448492-002, 01/2009