IBM DB2 ESE v.8.2
Implementation and Validation
With Hewlett-Packard's Serviceguard
High Availability Software

IBM DB2 Home Page

The HP Partner Technology Access Center (PTAC) has successfully integrated IBM's DB2 ESE v.8.2 multiple partition 64 bit and single partition 64 bit with Serviceguard A.11.15. The implementation and validation of DB2 ESE v.8.2 followed the HP PTAC High Availability Implementation and Validation Process detailed in:

HP's Partner Technology Access Center High Availability Implementation and Validation Services Process and Methodology

HA Validation Summary
Serviceguard Version: A.11.15
OS Version: HP-UX 11i v2.0 (11.23)
Systems used: rx4640, 4 CPU, 16G RAM
Application: IBM DB2 ESE 8.2 64-bit. Single Partition and Multiple Partition configurations

Implementation and Validation Process Participants:

Hewlett-Packard: Guy Patrick (HP Partner Technology Access Center)
IBM Software: Steve Raspudic (IBM DB2 High Availability Team)
IBM Software: Krzysztof` J. Obrebski (IBM DM Ranger Team)

validated highly-available by

Table of Contents

Executive Overview

Hewlett-Packard's (HP) relationship with the IBM Software Group spans more than a decade, with mutual nationwide Fortune 1000 customers asking for IBM software where reliability, high availability and scalability of HP UNIX enterprise hardware has been and will continue to be critical to their success.

IBM DB2 Information Management products have been available on HP-UX platforms since 1994. To expand the alliance, in December 2003 general availability was announced for DB2 running on the HP-UX 11i v2 operating system with HP Integrity Servers based on Intel® Itanium 2 processors. With this announcement, DB2 continues to build on its momentum in the HP-UX marketplace.

Together HP and IBM offer a world class combination of market leaders in the corporate Business Intelligence (BI) marketplace. IBM DB2 Information Management products today are optimized for HP-UX platforms and are continuously re-engineered with advanced technology, which dramatically improves the efficiency of database development and production environments. IBM DB2 Information Management has expanded support for HP's powerful architectural solutions, including PA-RISC, NT/Linux, and Intel® Itanium 2 platforms, utilizing to the fullest the world's leading technology in large data warehousing and transactional business operations.

Although the HP/DB2 alliance has been successful, the two companies have no intention of resting on their laurels. HP and IBM are working together to ensure that the future of e-business, including the burgeoning market for Internet applications and devices, is one in which IBM and HP will play a growing and deepening role. The two companies have resources dedicated to bringing customers the most integrated, comprehensive and focused solutions in the marketplace.

Executive Relationships: Top executives from both companies conduct quarterly reviews together to ensure optimal integration in both technology and strategic focus.

Joint Technology Development: The alliance sets the stage for HP and IBM DB2 engineering teams to accomplish significance solution improvements and develop optimum performance of DB2 products with HP middleware and hardware.

Marketing: HP and IBM have a close global marketing alliance with dedicated account managers, sales and marketing focusing on how to better serve customer needs.

Service and Support: HP Consulting and IBM Global Services provide specialized assessment and implementation services for joint customers.
Both HP and IBM DB2 are dominant players in the enterprise business intelligence market place. Together, HP and DB2 make up an increasingly requested off-mainframe business intelligence solution.

Why choose IBM Solutions for HP servers?
In today's competitive landscape of price wars and new innovation entrants to the marketplace, more people are selecting IBM DB2 Information Management and HP solutions.

As a foundation, both HP and IBM place ultimate emphasis - and organizational resources - on the customer experience. HP and IBM have been recognized separately for offering superior products and performance, and together are committed to extending this leadership role. Significant ongoing investments in engineering and tuning, in addition to joint technical planning, are a foundation of our alliance. DB2 Information Management products are optimized on a wide range of HP UNIX and Intel-based servers. Upon that foundation, DB2's industry leading scalability, functionality, reliability and pricing make it an ideal choice for customers seeking to run DB2 on HP-UX platforms or in a heterogeneous environment.

High Availability with HP Serviceguard

HP Serviceguard is designed to protect mission-critical applications from a wide variety of hardware and software failures. Serviceguard provides support for up to 16 nodes, which are organized into a high availability (HA) enterprise cluster that delivers highly available application services (up to 150 per cluster) to LAN-attached clients. Serviceguard is currently available for HP-UX (PA-RISC– and Itanium™ Processor Family–based systems) and Linux environments.

HP Serviceguard monitors the health of each node and quickly responds to failures of the following components to minimize downtime:

- system processors
- system memory
- local area network (LAN) media and adapters
- application processors

Since high availability and data integrity are primary design goals, Serviceguard clusters cannot have a single point of failure—data disks are mirrored and multiple LANs are used. Why consider HP Serviceguard or your clustering and high availability solutions?

- proven high availability solution with the largest UNIX® customer base
- robust cluster architecture
- many advanced features and functionalities
- single solution for planned and unplanned downtime
- tightly integrated with hundreds of applications and other HP high availability products
- key component for HP's disaster-tolerant solutions and integrated solutions unique level of protection

Key features and benefits

data protection

- multiple cluster arbitration mechanisms prevent failed nodes from jeopardizing the integrity of application data

availability

- quick automatic detection and recovery time maximizes application availability and minimizes operator error
- the ability to survive multiple node failures provides a unique level of protection
- fast failback shortens startup time for applications in the primary active node
- rolling upgrades ensure application availability during hardware and software maintenance
- integration with HP Workload Manager ensures service level objectives (SLOs) are maintained during planned and unplanned downtime
- the Enterprise Cluster Master toolkit provides quick and easy deployment of applications

flexibility

- multiple cluster configurations—active-active, active standby, and rotating standby—offer flexibility
- foundation for a suite of high availability and disaster tolerant solutions offers choice for your many mission critical needs

manageability

- an intuitive graphical user interface, Serviceguard Manager, reduces the total cost of managing multiple clusters (HP-UX and Linux) from a single console
- support for virtual and hard partitioning addresses the increasing demand for systems consolidation

ROI

- multiple operating system support addresses the need for cost-effectiveness of mission-critical environments
- integration with HP Pay per Use offers a cost-effective disaster-tolerant solution that is unique to HP

http://www.hp.com/go/ha

IBM's DB2 ESE v8.2 in a Highly Available Environment

IBM's DB2 ESE 8.2 is tested and validated with Hewlett Packard's Serviceguard high availability software, and will be referred to as DB2 from here on. DB2 can be monitored and controlled by Serviceguard to provide for a highly available product for use in a mission critical environment. The level of monitoring is highly configurable and subject to the implementation plans of the customer.

The Serviceguard solution for DB2 consists of one package, (db2instance)_(partition), i.e. db2inst1_0. The db2inst1_0 package contains a single service, db2inst1_0_service. The db2inst1_0 package can be configured to provide flexibility in the configuration and characteristics of the high availability solution. It is important that the DB2 instance and package naming conventions match. In addition, the Serviceguard scripts expect the DB2 user to match the DB2 instance. So the package db2inst1_0 would consist of the DB2 instance, db2inst1, and the partition 0. There would also need to be a user of the name db2inst1 that was the owner of this partition. All partitions for the DB2 instance would also be owned by the user db2inst1.

In a representative DB2 architecture, there is a single instance of DB2 running on a single server. For this reason, the Serviceguard restart value for the DB2 package is 0. This means DB2 will be moved to the adoptive host and restarted if any of the resources configured by Serviceguard fail; a power failure, a NIC failure, or a network failure. For more information on responses to failures, reference "*HP Managing Serviceguard Manual"*.

In the validation of DB2 and Serviceguard solution, the "client" used for testing was an HP-UX Workstation running an internal IBM test tool. Since the Serviceguard solution utilizes a relocatable IP address for the db2inst1_0 package, the client does not need to re-configure its session connectivity settings if a fail-over scenario occurs. The client simply continues to re-attempt communications if the host goes down. Once the package is running on the fail-over node, the DB2 session is re-established. Depending on the type of failure, a client in a real-world scenario may be unable to log in for the duration of the fail-over, or the session may time-out and may need to be restarted. This behavior is completely dependent on the client application(s).

This document details a set of tests that show that DB2 is compatible with Serviceguard. The tests show that if a node running the DB2 server fails, DB2 can be up and running on an adoptive node in as little as 30 seconds, without human intervention. This time will vary depending on the amount of activity occurring on the DB2 server, the size of log files, time between checkpoints, the client application using DB2, and other application specific conditions.

A Serviceguard package is dependent on having automated application startup and shutdown scripts. HP Partner Technology Access Center engineers, teaming with IBM engineers, have developed and tested package configuration and control scripts which are suitable for use (after system-specific modification) in a DB2 production environment.

Installation Outline for DB2 and Serviceguard

These steps outline the installation process and configuration changes required for DB2 in a High Availability environment and summarize the full installation as documented. This outline is not intended to replace the detailed *DB2 ESE Installation Guide*.

rx4640a setup

NOTE: rx4640a was used as the primary node for the db2inst1_0 package. This contained the single service, idb2inst1_0_service that monitored the DB2 processes.

1. Create a volume group to put the shared logical volume on, such as /dev/db2inst1_0.
2. Create a logical volume on the new volume group, with mount point, such as /dev/db2inst1_0/db2inst1_0 mounted under /db/db2inst1/NODE0000.
3. Mount the shared logical volume.
4. Install DB2 on the shared disk mounted under /db/db2inst1/NODE0000 as detailed in the DB2 ESE installation guide.
5. Issue a "vgexport –m db2inst1_0.map –s –p –v db2inst1_0"
6. Unmount the shared directories.
7. Issue a "vgchange –a n db2inst1_0"
8. FTP the db2inst1_0.map file to the adoptive node.
9. Telnet into the adoptive node.
10. Create the /dev/db2inst1_0 directory "mkdir /dev/db2inst1_0".
11. Cd into the new directory "cd /dev/db2inst1_0"
12. Create a new group file that matches the major and minor number on the primary node "mknod group c 64 0x010000". Verify the major/minor device info on the primary node with "ls –l /dev/db2inst1_0"
13. Issue a "vgimport –m db2inst1_0.map –s –v db2inst1_0"
14. Activate the VG with "vgchange –a y db2inst1_0"
15. Mount /db/db2inst1/NODE0000 to confirm the import was successful.
16. Repeat steps 1 through 15 for any additional partitions. i.e. db2inst1_2, db2inst1_3, and db2inst1_4.
17. Follow the DB2 installation addendum for Serviceguard as described in Appendix 1.
18. Once the shared filesystem can be mounted on both systems, setup and configure the Serviceguard scripts for DB2.

Serviceguard Configuration

Configuring the Cluster

All of the Serviceguard scripts developed during the validation process have been provided in this document. The following section describes the creation and use of these scripts.

Create the ASCII cluster template file

```
cmquerycl -v -C /etc/cmcluster/cluster.ascii -n rx4640a -n rx4640b
```

Modify the template (cluster.ascii) to reflect the environment and to verify the cluster configuration.

```
cmcheckconf -v -C /etc/cmcluster/cluster.ascii
```

Create the cluster by applying the configuration file. This will create the binary file "cmclconfig" and automatically distribute it among the nodes defined in the cluster.

```
cmapplyconf -v -C /etc/cmcluster/cluster.ascii
```

Start the cluster and check the cluster status. Test the cluster halt also.

```
cmruncl -v -n rx4640a -n rx4640b
cmviewcl -v
cmhaltcl -f -v
cmruncl -n rx4640a -n rx4640b
```

Configuring a Serviceguard Package (On a single node)

Create the db2inst1_0 package configuration file and tailor to the test environment. Do not include the second node at this stage.

```
cd /etc/cmcluster
mkdir db2inst1_0
 cd db2inst1_0
cmmakepkg -p db2inst1_0.conf         # Edit db2inst1_0.conf
```

Create the db2inst1_0 package control script and tailor to the test environment. Do not include application startup/shutdown, service monitoring, or relocatable IP address at this stage.

```
cmmakepkg -s db2inst1_0.cntl         # Edit db2inst1_0.cntl
```

For reference, the db2inst1.cntl and db2inst1.conf scripts used for testing are supplied in the appendices below.

Shut down cluster, verify and distribute the binary configuration files

```
cmhaltcl -f -v
cmapplyconf -v -C /etc/cmcluster/cluster.ascii -P \
                  /etc/cmcluster/db2inst1_0/db2inst1_0.conf
```

Test cluster and package startup. Shut down DB2 if running, and unmount all logical volumes on /dev/db2inst1_0 and deactivate the volume group first.

```
cmruncl                              # Start cluster and package
cmviewcl -v                          # Check that package has started
```

Edit db2inst1.cntl and assign the dynamic IP address of the db2inst1_0 package.

```
cmhaltpkg db2inst1_0
vi db2inst1_0.cntl                   # Edit to add package IP
cmrunpkg -v db2inst1_0               # Start DB2 Package
cmviewcl -v                          # Check package has started and clients
```

Enable switching to a local standby LAN card.

```
vi db2inst1_0.conf            # Net switching enabled = YES
cmapplyconf -v -C /etc/cmcluster/cluster.ascii -P db2inst1_0/db2inst1_0.conf
cmhaltcl -f -v
cmruncl -v
```

Configuring a Serviceguard Package (Adding a second node)

Enable db2inst1_0 to switch to a second node by editing the package control file

```
vi db2inst1_0.conf                    # add NODE_NAME rx4640b
cmapplyconf -v -C /etc/cmcluster/cluster.ascii -P db2inst1_0/db2inst1_0.conf
cmhaltcl -f -v
cmruncl -v
```

Test package switch to rx4640a and back to rx4640b

```
cmhaltpkg db2inst1_0
cmrunpkg -n rx4640b db2inst1_0              # Run package on rx4640b and
                                            # run DB2 and check application
cmmodpkg -e db2inst1_0                      # Enable package switching
cmhaltpkg db2inst1_0
cmrunpkg -n rx4640a db2inst1_0              # Run package on rx4640a and test
                                            # DB2 runs here
cmmodpkg -e db2inst1_0
```

At this point, the db2inst1_0 package is running with DB2 using the system hostname instead of the package IP. See Appendix 1 for exact details for utilizing the package IP instead of the host IP.

Repeat above steps for multiple partitions such as, db2inst1_1, db2inst1_2, etc. Ensure all additional partitions are created on separate disks and volume groups so they can be members of separate packages.

Configuring DB2 in a Serviceguard environment

Once DB2 is installed and configured in the Serviceguard cluster, the DB2 package scripts can be configured.

In testing the Serviceguard integration with IBM engineers, the db2inst1_0.cntl file was configured such that the db2inst1_0_service service has zero restarts and will fail over to the adoptive node in the case of a software or hardware failure. This number can be changed to suit the needs of each install. It is recommended that the restart value be left at 0. DB2 is a robust product and if there is a failure, the probability of a successful restart is low. To ensure a stable DB2 operating environment, it is suggested that Serviceguard be allowed to move the db2inst1_0 package to an adoptive node in the case of any failure.

The DB2 monitor script has been modified from previous versions of DB2. A new utility, db2gcf, is used to monitor the complete health of DB2 ESE. Not only are processes monitored, the IPC's used by DB2 are also monitored.

Test Suite

Output of cmviewcl –v with the db2inst1_0 package running on rx4640a.

```
CLUSTER      STATUS
cluster1     up

  NODE         STATUS       STATE
  rx4640a      up           running

    Network_Parameters:
    INTERFACE    STATUS       PATH         NAME
    PRIMARY      up           0/0/0/0      lan0
    STANDBY      up           0/10/0/0     lan1

    PACKAGE      STATUS       STATE        AUTO_RUN     NODE
    inst32s   up           running      enabled      rx4640a

      Policy_Parameters:
      POLICY_NAME     CONFIGURED_VALUE
      Failover        configured_node
      Failback        manual

      Script_Parameters:
      ITEM       STATUS   MAX_RESTARTS  RESTARTS   NAME
      Service    up                 0         0   inst32s_service

      Node_Switching_Parameters:
      NODE_TYPE    STATUS       SWITCHING    NAME
      Primary      up           enabled      rx4640a      (current)
      Alternate    up           enabled      rp5470d

  NODE         STATUS       STATE
  rp5470d      up           running

    Network_Parameters:
    INTERFACE    STATUS       PATH         NAME
    PRIMARY      up           0/0/0/0      lan0
    STANDBY      up           0/12/0/0     lan4
```

Test results summary

For our testing, we created two DB2 instances, db2inst1 and db2inst2. The instance, db2inst1, was configured with two partitions, 0 and 1. The instance, db2inst2, was a single partition, 0. For all of our testing, these three partitions were used to test both the multiple partition and single partition configurations of DB2. The testing data was broken out into single and multiple partition tests. In both cases, DB2 in a multiple of single partition scenario behaved the same for a single failure. In the event of multiple failures, the multiple partition DB2 configuration was more likely to experience problems. The reason for this is the reliance on HA NFS for the home directory and DB2 partition hierarchy requirements with multiple partition instances.

For the following tests, the client used was an HP-UX client running an internal IBM test utility. The test utility provides for a configurable time-out in case a transaction hangs. Without a time-out mechanism, the client will hang during an HA fail-over, requiring manual intervention. By using the intelligent client provided by IBM, we are able to demonstrate the restart capabilities of the DB2 client and server pieces.

Multiple Partition Tests (Tests 1 – 11)

Test 1

Description: db2inst1_0 package UP on rx4640a, db2inst1_1 package UP on rx4640b, and db2home package UP on rx4640a. The test utility is generating transactions.
Shut db2inst1_0 package down. Test client responses.

```
cmhaltpkg db2inst1_0
```

Results: Existing DB2 connections are closed, and new ones start up, waiting for DB2 to restart.

Test 2

Description: db2inst1_0 package DOWN, db2inst1_1 package UP on rx4640b, and db2home package UP on rx4640a. The test utility is generating transactions. Start down package on its primary node. Test DB2 client response.

```
cmrunpkg db2inst1_0
```

Results: db2inst1_0 package started.
DB2 clients waiting in previous test connected and re-submit transactions, picking up from last failure. ***It is up to the client program to send retries.***

Test 3

Description: db2inst1_0 package UP on rx4640a, db2inst1_1 package UP on rx4640b, and db2home package UP on rx4640a. The test utility is generating transactions.
Shut db2inst1_1 package down. Test client responses.

```
cmhaltpkg db2inst1_1
```

Results: Existing DB2 connections are closed, and new ones start up, waiting for DB2 to restart.

Test 4

Description: db2inst1_0 package up on rx4640a, db2inst1_1 package DOWN, and db2home package UP on rx4640a. The test utility is generating transactions. Start down package on its primary node. Test DB2 client response.

```
cmrunpkg db2inst1_1
```

Results: db2inst1_1 package started.
DB2 clients waiting in previous test connected and re-submit transactions, picking up from last failure. ***It is up to the client program to send retries.***

Test 5

Description: db2inst1_0 package UP on rx4640a, db2inst1_1 package UP on rx4640b, and db2home package UP on rx4640a. The test utility is generating transactions.
Shut db2home package down. Test client responses.

```
cmhaltpkg db2home
```

Results: Any open connections are unaffected, but no new connections are accepted.

Test 6

Description: db2inst1_0 package up on rx4640a, db2inst1_1 package up on rx4640b, and db2home package DOWN. The test utility is generating transactions. Start down package on its primary node. Test DB2 client response.

```
cmrunpkg db2home
```

Results: db2home package started. Existing open connections are still unaffected, and new connections are accepted and complete successfully.

NOTE: NFS was not designed to be a highly available product and is unpredictable in a high availability cluster. In some cases, it may be more prudent to place the DB2 home directory on a separate, dedicated NFS server that is not a part of the DB2 cluster. In the event of a NFS package fail-over, a subsequent failure of a DB2 package cannot be guaranteed to be successful. In the future, a clustered file system will be implemented with HP-UX to eliminate the unreliable NFS aspect of a DB2 multi-partition HA configuration. The NFS home directory is only required with multi-partition DB2 and does not affect single-partition DB2.

Symptoms of a NFS HA failure may be long time-outs in the recovery of the NFS filesystem stale mount. In our testing, it took as long as 5 minutes for the stale NFS mounts to recover after a failure. The problem seemed to only be on the node that was previously the NFS package host where a DB2 multi-partition package was running. A solution would be to NOT have the NFS package run on any node that would also host a DB2 multi-partition package associated with the NFS home directory.

Test 7

Description: db2inst1_0 package UP on rx4640a, db2inst1_1 package UP on rx4640b, and db2home package UP on rx4640a. The test utility is generating transactions. Power off server rx4640a. Verify db2inst1_0 and db2home package switch process.

Results: Power off rx4640a. Cluster manager recognizes loss of rx4640a heartbeat. Cluster reforms with one healthy member, rx4640b. Cluster frees leftover holds on shared volume group. The db2inst1_0 and db2home packages start correctly on rx4640b. All DB2 processes restart automatically on rx4640b. When rx4640a comes up, it does not rejoin the cluster, but must be manually added to the cluster since it has failed.

The DB2 test utility detects that the current transaction has hung; it times out and retries transaction until successful.

Test 8

Description: db2inst1_0 package UP on rx4640a, db2inst1_1 package UP on rx4640b, and db2home package UP on rx4640a. The test utility is generating transactions. Power off server rx4640b. Verify db2inst1_1 package switch process.

Results: Power off rx4640b. Cluster manager recognizes loss of rx4640b heartbeat. Cluster reforms with one healthy member, rx4640a. Cluster frees leftover holds on shared volume group. The db2inst1_1 package starts correctly on rx4640a. All DB2 processes restart automatically on rx4640a. When rx4640b comes up, it does not rejoin the cluster, but must be manually added to the cluster since it has failed.
The DB2 test utility detects that the current transaction has hung; it times out and retries transaction until successful.

Test 9

Description: db2inst1_0 package UP on rx4640a, db2inst1_1 package UP on rx4640b, and db2home package UP on rx4640a. The test utility is generating transactions. Test restart capability of Serviceguard monitor service.

The monitor process DB2.sh starts and monitors the DB2 processes via db2gcf. If the monitor process fails, a signal is sent to Serviceguard and causes a fail-over to the adoptive node of the db2inst1_0 or db2inst1_1 packages.

Results: Killed DB2.sh monitor process on rx4640a and the db2inst1_0 package fails over to the adoptive node. When the package starts on the adoptive node, new connections are accepted and continue successfully.

Test 10

Description: db2inst1_0 package UP on rx4640a, db2inst1_1 package UP on rx4640b, and db2home package UP on rx4640a. The test utility is generating transactions. Test restart capability of Serviceguard monitor service.
The monitor process DB2.sh starts and monitors the DB2 processes via db2gcf. If the monitor process fails, a signal is sent to Serviceguard and causes a fail-over to the adoptive node of the db2inst1_0 or db2inst1_1 packages.

Results: Killed DB2.sh monitor process on rx4640b and the db2inst1_1 package fails over to the adoptive node. When the package starts on the adoptive node, new connections are accepted and continue successfully.

Test 11

Description: Test monitoring and exiting capability of monitor service.

The monitor method used by DB2.sh incorporates the IBM utility, db2gcf. This utility is supplied as part of the standard DB2 install. Since db2gcf monitors the health of DB2, the monitor utility checks the return code to determine of the DB2 instance has failed. A non-zero return code from db2gcf indicates a failure, and will cause a fail-over to the adoptive node of the db2inst1_0 package.

To test the monitor capabilities of DB2.sh, and ensure the db2gcf utility can properly identify an inoperable DB2 instance, we will be killing the unmonitored DB2 processes: see list below.

Kill -9 <db2sysc pid>

Results: The IBM monitor utility used by DB2.sh, db2gcf, detected a failure of DB2 and returned a non-zero return code. The monitor script then returned an exit code 1 to Serviceguard. Serviceguard realized that the service had exited with an error code, and performed a monitor process restart. The db2inst1_0 package has no restarts, so the package was failed over to the adoptive node, rx4640b.

Kill -9 <db2bp pid>

Result: Only local connection associated with the db2bp process is terminated. All other DB2 connections are not affected. DB2 continues to run on current node.

Kill -9 <db2ckpwd pid>

Results: The IBM monitor utility used by DB2.sh, db2gcf, detected a failure of DB2 and returned a non-zero return code. The monitor script then returned an exit code 1 to Serviceguard. Serviceguard realized that the service had exited with an error code, and performed a monitor process restart. The db2inst1_0 package has no restarts, so the package was failed over to the adoptive node, rx4640b.

Kill -9 <db2wdog pid>

Results: The IBM monitor utility used by DB2.sh, db2gcf, detected a failure of DB2 and returned a non-zero return code. The monitor script then returned an exit code 1 to Serviceguard. Serviceguard realized that the service had exited with an error code, and performed a monitor process restart. The db2inst1_0 package has no restarts, so the package was failed over to the adoptive node, rx4640b.

Kill -9 <db2panic pid>

Results: The IBM monitor utility used by DB2.sh, db2gcf, detected a failure of DB2 and returned a non-zero return code. The monitor script then returned an exit code 1 to Serviceguard. Serviceguard realized that the service had exited with an error code, and performed a monitor process restart. The db2inst1_0 package has no restarts, so the package was failed over to the adoptive node, rx4640b.

Kill -9 <db2agent pid>

Results: The IBM monitor utility used by DB2.sh, db2gcf, detected a failure of DB2 and returned a non-zero return code. The monitor script then returned an exit code 1 to Serviceguard. Serviceguard realized that the service had exited with an error code, and performed a monitor process restart. The db2inst1_0 package has no restarts, so the package was failed over to the adoptive node, rx4640b.

kill -9 <db2gds pid>

Results: The IBM monitor utility used by DB2.sh, db2gcf, detected a failure of DB2 and returned a non-zero return code. The monitor script then returned an exit code 1 to Serviceguard. Serviceguard realized that the service had exited with an error code, and performed a monitor process restart. The db2inst1_0 package has no restarts, so the package was failed over to the adoptive node, rx4640b.

Kill -9 <db2tcpcm pid>

Results: The IBM monitor utility used by DB2.sh, db2gcf, detected a failure of DB2 and returned a non-zero return code. The monitor script then returned an exit code 1 to Serviceguard. Serviceguard realized that the service had exited with an error code, and performed a monitor process restart. The db2inst1_0 package has no restarts, so the package was failed over to the adoptive node, rx4640b.

Kill -9 <db2ipccm>

Results: The IBM monitor utility used by DB2.sh, db2gcf, detected a failure of DB2 and returned a non-zero return code. The monitor script then returned an exit code 1 to Serviceguard. Serviceguard realized that the service had exited with an error code, and performed a monitor process restart. The db2inst1_0 package has no restarts, so the package was failed over to the adoptive node, rx4640b.

Kill -9 <db2srvlst>

Results: The IBM monitor utility used by DB2.sh, db2gcf, detected a failure of DB2 and returned a non-zero return code. The monitor script then returned an exit code 1 to Serviceguard. Serviceguard realized that the service had exited with an error code, and performed a monitor process restart. The db2inst1_0 package has no restarts, so the package was failed over to the adoptive node, rx4640b.

Kill-9 <db2resync>

Results: The IBM monitor utility used by DB2.sh, db2gcf, detected a failure of DB2 and returned a non-zero return code. The monitor script then returned an exit code 1 to Serviceguard. Serviceguard realized that the service had exited with an error code, and performed a monitor process restart. The db2inst1_0 package has no restarts, so the package was failed over to the adoptive node, rx4640b.

Kill -9 <db2pclnr>

Results: The IBM monitor utility used by DB2.sh, db2gcf, detected a failure of DB2 and returned a non-zero return code. The monitor script then returned an exit code 1 to Serviceguard. Serviceguard realized that the service had exited with an error code, and performed a monitor process restart. The db2inst1_0 package has no restarts, so the package was failed over to the adoptive node, rx4640b.

Kill -9 <db2loggw>

Results: The IBM monitor utility used by DB2.sh, db2gcf, detected a failure of DB2 and returned a non-zero return code. The monitor script then returned an exit code 1 to Serviceguard. Serviceguard realized that the service had exited with an error code, and performed a monitor process restart. The db2inst1_0 package has no restarts, so the package was failed over to the adoptive node, rx4640b.

Kill -9 <db2dlock>

Results: The IBM monitor utility used by DB2.sh, db2gcf, detected a failure of DB2 and returned a non-zero return code. The monitor script then returned an exit code 1 to Serviceguard. Serviceguard realized that the service had exited with an error code, and performed a monitor process restart. The db2inst1_0 package has no restarts, so the package was failed over to the adoptive node, rx4640b.

Kill -9 <db2pfchr>

Results: The IBM monitor utility used by DB2.sh, db2gcf, detected a failure of DB2 and returned a non-zero return code. The monitor script then returned an exit code 1 to Serviceguard. Serviceguard realized that the service had exited with an error code, and performed a monitor process restart. The db2inst1_0 package has no restarts, so the package was failed over to the adoptive node, rx4640b.

Kill -9 <db2fcmdm>

Results: The IBM monitor utility used by DB2.sh, db2gcf, detected a failure of DB2 and returned a non-zero return code. The monitor script then returned an exit code 1 to Serviceguard. Serviceguard realized that the service had exited with an error code, and performed a monitor process restart. The db2inst1_0 package has no restarts, so the package was failed over to the adoptive node, rx4640b.

Kill -9 <db2fmcd>

Results: This process re-spawns itself and the DB2 monitor utility, db2gcf, does not detect a failure. The database continues to run, there are no disconnects of the existing clients, and new connections are accepted. There is no effect in killing this process.

Single Partition Tests (Tests 1-5)

Test 1

Description: db2inst1_0 package UP on rx4640a. The test utility is generating transactions.
Shut db2inst1_0 package down. Test client responses.

```
cmhaltpkg db2inst1_0
```

Results: Existing DB2 connections are closed, and new ones start up, waiting for DB2 to restart.

Test 2

Description: db2inst1_0 package DOWN. The test utility is generating transactions. Start down package on its primary node. Test DB2 client response.

```
cmrunpkg db2inst1_0
```

Results: db2inst1_0 package started.
DB2 clients waiting in previous test connected and re-submit transactions, picking up from last failure. ***It is up to the client program to send retries.***

Test 3

Description: db2inst1_0 package UP on rx4640a. The test utility is generating transactions. Start down package on its primary node. The test utility is generating transactions. Power off server rx4640a. Verify db2inst1_0 package switch process.

Results: Power off rx4640a. Cluster manager recognizes loss of rx4640a heartbeat. Cluster reforms with one healthy member, rx4640b. Cluster frees leftover holds on shared volume group. The db2inst1_0 package start correctly on rx4640b. All DB2 processes restart automatically on rx4640b. When rx4640a comes up, it does not rejoin the cluster, but must be manually added to the cluster since it has failed.

The DB2 test utility detects that the current transaction has hung; it times out and retries transaction until successful.

Test 4

Description: db2inst1_0 package UP on rx4640a. The test utility is generating transactions. Test restart capability of Serviceguard monitor service.

The monitor process DB2.sh starts and monitors the DB2 processes via db2gcf. If the monitor process fails, a signal is sent to Serviceguard and causes a fail-over to the adoptive node of the db2inst1_0 package.

Results: Killed DB2.sh monitor process on rx4640a and the db2inst1_0 package fails over to the adoptive node. When the package starts on the adoptive node, new connections are accepted and continue successfully.

Test 5

Description: Test monitoring and exiting capability of monitor service.

The monitor method used by DB2.sh incorporates the IBM utility, db2gcf. This utility is supplied as part of the standard DB2 install. Since db2gcf monitors the health of DB2, the monitor utility checks the return code to determine of the DB2 instance has failed. A non-zero return code from db2gcf indicates a failure, and will cause a fail-over to the adoptive node of the db2inst1_0 package.

To test the monitor capabilities of DB2.sh, and ensure the db2gcf utility can properly identify an inoperable DB2 instance, we will be killing the unmonitored DB2 processes: See list below.

Kill -9 <db2sysc pid>

Results: The IBM monitor utility used by DB2.sh, db2gcf, detected a failure of DB2 and returned a non-zero return code. The monitor script then returned an exit code 1 to Serviceguard. Serviceguard realized that the service had exited with an error code, and performed a monitor process restart. The db2inst1_0 package has no restarts, so the package was failed over to the adoptive node, rx4640b.

Kill -9 <db2bp pid>

Result: Only local connection associated with the db2bp process is terminated. All other DB2 connections are not affected. DB2 continues to run on current node.

Kill -9 <db2ckpwd pid>

Results: The IBM monitor utility used by DB2.sh, db2gcf, detected a failure of DB2 and returned a non-zero return code. The monitor script then returned an exit code 1 to Serviceguard. Serviceguard realized that the service had exited with an error code, and performed a monitor process restart. The db2inst1_0 package has no restarts, so the package was failed over to the adoptive node, rx4640b.

Kill -9 <db2wdog pid>

Results: The IBM monitor utility used by DB2.sh, db2gcf, detected a failure of DB2 and returned a non-zero return code. The monitor script then returned an exit code 1 to Serviceguard. Serviceguard realized that the service had exited with an error code, and performed a monitor process restart. The db2inst1_0 package has no restarts, so the package was failed over to the adoptive node, rx4640b.

Kill -9 <db2agent pid>

Results: The IBM monitor utility used by DB2.sh, db2gcf, detected a failure of DB2 and returned a non-zero return code. The monitor script then returned an exit code 1 to Serviceguard. Serviceguard realized that the service had exited with an error code, and performed a monitor process restart. The db2inst1_0 package has no restarts, so the package was failed over to the adoptive node, rx4640b.

kill -9 <db2gds pid>

Results: The IBM monitor utility used by DB2.sh, db2gcf, detected a failure of DB2 and returned a non-zero return code. The monitor script then returned an exit code 1 to Serviceguard. Serviceguard realized that the service had exited with an error code, and performed a monitor process restart. The db2inst1_0 package has no restarts, so the package was failed over to the adoptive node, rx4640b.

Kill -9 <db2tcpcm pid>

Results: The IBM monitor utility used by DB2.sh, db2gcf, detected a failure of DB2 and returned a non-zero return code. The monitor script then returned an exit code 1 to Serviceguard. Serviceguard realized

that the service had exited with an error code, and performed a monitor process restart. The db2inst1_0 package has no restarts, so the package was failed over to the adoptive node, rx4640b.

Kill -9 <db2ipccm>

Results: The IBM monitor utility used by DB2.sh, db2gcf, detected a failure of DB2 and returned a non-zero return code. The monitor script then returned an exit code 1 to Serviceguard. Serviceguard realized that the service had exited with an error code, and performed a monitor process restart. The db2inst1_0 package has no restarts, so the package was failed over to the adoptive node, rx4640b.

Kill -9 <db2srvlst>

Results: The IBM monitor utility used by DB2.sh, db2gcf, detected a failure of DB2 and returned a non-zero return code. The monitor script then returned an exit code 1 to Serviceguard. Serviceguard realized that the service had exited with an error code, and performed a monitor process restart. The db2inst1_0 package has no restarts, so the package was failed over to the adoptive node, rx4640b.

Kill-9 <db2resync>

Results: The IBM monitor utility used by DB2.sh, db2gcf, detected a failure of DB2 and returned a non-zero return code. The monitor script then returned an exit code 1 to Serviceguard. Serviceguard realized that the service had exited with an error code, and performed a monitor process restart. The db2inst1_0 package has no restarts, so the package was failed over to the adoptive node, rx4640b.

Kill -9 <db2pclnr>

Results: The IBM monitor utility used by DB2.sh, db2gcf, detected a failure of DB2 and returned a non-zero return code. The monitor script then returned an exit code 1 to Serviceguard. Serviceguard realized that the service had exited with an error code, and performed a monitor process restart. The db2inst1_0 package has no restarts, so the package was failed over to the adoptive node, rx4640b.

Kill -9 <db2loggw>

Results: The IBM monitor utility used by DB2.sh, db2gcf, detected a failure of DB2 and returned a non-zero return code. The monitor script then returned an exit code 1 to Serviceguard. Serviceguard realized that the service had exited with an error code, and performed a monitor process restart. The db2inst1_0 package has no restarts, so the package was failed over to the adoptive node, rx4640b.

Kill -9 <db2dlock>

Results: The IBM monitor utility used by DB2.sh, db2gcf, detected a failure of DB2 and returned a non-zero return code. The monitor script then returned an exit code 1 to Serviceguard. Serviceguard realized that the service had exited with an error code, and performed a monitor process restart. The db2inst1_0 package has no restarts, so the package was failed over to the adoptive node, rx4640b.

Kill -9 <db2pfchr>

Results: The IBM monitor utility used by DB2.sh, db2gcf, detected a failure of DB2 and returned a non-zero return code. The monitor script then returned an exit code 1 to Serviceguard. Serviceguard realized that the service had exited with an error code, and performed a monitor process restart. The db2inst1_0 package has no restarts, so the package was failed over to the adoptive node, rx4640b.

Kill -9 <db2fcmdm>

Results: The IBM monitor utility used by DB2.sh, db2gcf, detected a failure of DB2 and returned a non-zero return code. The monitor script then returned an exit code 1 to Serviceguard. Serviceguard realized

that the service had exited with an error code, and performed a monitor process restart. The db2inst1_0 package has no restarts, so the package was failed over to the adoptive node, rx4640b.

Kill -9 <db2fmcd>

Results: This process re-spawns itself and the DB2 monitor utility, db2gcf, does not detect a failure. The database continues to run, there are no disconnects of the existing clients, and new connections are accepted. There is no effect in killing this process.

Appendix 1: DB2 installation addendum for a Serviceguard environment

This section describes the implementation of DB2 ESE (v.8.2) for HP-UX in a highly available environment with the use of Serviceguard. The steps described are identical for DB2 V6.1 and V7.x as well as for all supported levels of Serviceguard.

The DB2 executable image

As root, use the `db2setup` to install the db2 executable and configure a db2 instance on the primary machine. Ensure that the instance is valid. For example, login as the instance owner id, and attempt a `db2start` and `db2stop`. These should both complete normally.

Use `db2setup` in an identical manner on the hot standby machine. Ensure that the instance owner id properties are identical between the two machines (id, group membership, group id, instance home directory for example).

Ensure that the service levels are identical between the two machines. Note that on each machine, the location of the installation of the DB2 executable images */opt*, will be local to each machine. One consequence of this is that service packs must be installed as per usual on the primary machine, and this identical sequence must be repeated on the hot standby machine.

Note that this step can be executed either before or after Serviceguard is installed. For high availability, the volume groups upon which the executable image filesystem resides and the instance home directory resides must be mirrored.

The DB2 instance

Set up the database manager configuration parameters as desired. Perform the identical setup on the secondary machine. Note that should the database manager configuration be changed on one of the nodes, the changes will not be automatically propagated to the other node. The same update commands will need to be run in order to ensure that the two database manager configurations are identical. Ensure that for any of the parameters where a filesystem is the supplied argument, the parameter is valid in the context of both the primary and the hot-standby nodes.

Note that the home filesystem of the instance owner will be local to each machine. So, any changes or additions made to the home filesystem (either directly or indirectly) will need to be propagated to the hot standby. For example, the default location for the log user exit executable (which is built and installed by the end-user) is *sqllib/bin* or *sqllib/adm*. This file should be present and identical on both machines.

Note that this step can be executed either before or after Serviceguard is installed. Note that any instance name can be used; the examples that follow will consistently use *inst32s*.

The DB2 database

On the primary machine, create the database and all tablespaces necessary. Ensure that the database and tablespace containers are created on drives which are:

- mirrored or fault tolerant with appropriate RAID level (RAID-5 for example)
- dual-ported (that is, drives physically cabled to both physical nodes)

Note that if any database data does not reside on a shared disk, it will not be visible to the secondary system at failover time.

Both raw disks and filesystems are supported with DB2 ESE; with raw disks possessing the benefit of potentially less failover time due to the absence of a filesystem consistency check which may be performed upon filesystem failover.

For example, assuming that /database is a filesystem mount point for a such a shared filesystem,

```
db2 create database dbname on /database
```

will create a database named *dbname* on the */database* filesystem. Ensure that subsequent tablespace containers are similarly created in shared filesystems or volume groups and that database transaction logs are not moved to non-shared disks.

Ensure that a database connection can be successfully established to *dbname*.

There are a large number of database configuration variables: set them up as you normally would, the AUTORESTART parameter should be set to on. Otherwise, the database connection requests after failover will not succeed.

To ensure that the instance on the second machine is aware of the location of this machine at failover time, issue the following command:

```
db2 catalog database test on /database
```

Careful testing is vital to the successful implementation of any HA solution. At this point, prior to the installation or configuration of any HA specific features, we can ensure that the database is visible on both nodes.

-> manually issue vgchange commands and see whether the database is visible between the two machines

Serviceguard package commands

It is assumed that at this point, the Serviceguard cluster is configured correctly.

The next step is to create individual packages for the instances which you wish to make highly available.

On each machine, issue the following command

```
mkdir /etc/cmcluster/db2inst1_0
```

Within the */etc/cmcluster/db2inst1_0/* directory, install the `DB2.sh` script. Ensure that it is marked as executable.

On the first machine, test the functioning of the `DB2.sh` script in conjunction with the instance in question.

```
/etc/cmcluster/db2inst1_0/DB2.sh start db2inst1 0
```

This command should bring up the database manager successfully.

```
/etc/cmcluster/db2inst1_0/DB2.sh monitor db2inst1 0
                                ^       ^        ^
Action to take ----------------|       |        |
Instance number ------------------------|        |
Partition number ------------------------------|
```

This command should display a list of monitored DB2 pids. Allow the monitor to run for a few iterations; open another window and attempt a kill -9 on any of the monitored pids. The abnormal termination condition should be detected and the monitor should exit.

Now, bring up the database manager again with the command

```
/etc/cmcluster/db2inst1_0/DB2.sh start db2inst1 0
```

Check that the DB2.sh script can successfully bring down the DB2 instance

```
/etc/cmcluster/db2inst1_0/DB2.sh halt db2inst1 0
```

Perform the above sequence of commands on both nodes to ensure the basic functionality of the script.

Creating DB2 packages for Serviceguard

Each instance which will be monitored requires its own package. The convention is to name the package identically to the instance name.

Prior to the package creation step, the cluster must be set up for use with Serviceguard. Please follow the instructions given in the Serviceguard documentation in order to have a working cluster.

From one of the nodes:

```
mkdir /etc/cmcluster/db2inst1_0
cmmakepkg -p /etc/cmcluster/db2inst1_0/db2inst1_0.conf
```

Make the following additions/changes to the *etc/cmcluster/db2inst1_0/db2inst1_0.conf* file:

```
PACKAGE_NAME db2inst1_0
NODE_NAME hostname_of_primary_node
NODE_NAME hostname_of_standby_node
RUN_SCRIPT /etc/cmcluster/db2inst1_0/db2inst1_0.cntl
HALT_SCRIPT /etc/cmcluster/db2inst1_0/db2inst1_0.cntl
SERVICE_NAME      db2inst1_0_service
SERVICE_FAIL_FAST_ENABLED NO
SERVICE_HALT_TIMEOUT 300

cmmakepkg -s /etc/cmcluster/db2inst1_0/db2inst1_0.conf
```

Make the following changes to the */etc/cmcluster/db2inst1_0db2inst1_0.conf* file (this assumes that the volume group is named DB2, and the filesystem mount point is /DB2).

```
VG[0]=DB2
LV[0]=/dev/DB2/DB2; FS[0]=/DB2; FS_MOUNT_OPT="-o rw"
```

Repeat the above two lines (increment the array subscripts) for each volume group/filesystem to be used to store DB2 data.

The IP address specified should be the same one used by remote TCP/IP clients to catalog this node. For example,

```
IP[0]="192.6.173.3"
SUBNET[0]="192.6.173.0"
```

The convention for the service name is to append *_service* to the instance name.

```
SERVICE_NAME[0]=db2inst1_0_service
SERVICE_CMD[0]="/etc/cmcluster/db2inst1_0/DB2.sh monitor db2inst1 0"
```

Add the following just before the *customer_defined_run_cmds* function:

```
DB2_INSTANCE=db2inst1 # Add your specific instance name
DB2_PARTITION=0 # Add your specific partition number
```

Nothing needs to be added to the *customer_defined_halt_cmds* function.

This will complete the steps required at the server. At this point, it is suggested to perform testing via cluster management commands and via disruptions of physical service.

Client Configuration

If the database is accessed remotely via TCP/IP, then a *catalog node* command must be issued in order to inform the clients of the location of the server.

On the client side issue the following command, you would issue the following command for instance.

```
db2 catalog tcpip node nodename remote 156.152.98.175
db2 catalog database test at node nodename
```

This completes the integration of DB2 with Serviceguard. It is suggested that testing be performed again to ensure that the client application behavior is well understood under various failover scenarios.

Appendix 2: “cluster1” cluster configuration file (cluster.ascii)

```
# **********************************************************************
# ********* HIGH AVAILABILITY CLUSTER CONFIGURATION FILE ***************
# ***** For complete details about cluster parameters and how to *******
# ***** set them, consult the Serviceguard manual. **********************
# **********************************************************************

# Enter a name for this cluster.  This name will be used to identify the
# cluster when viewing or manipulating it.

CLUSTER_NAME             cluster1


# Cluster Lock Parameters
#
# The cluster lock is used as a tie-breaker for situations
# in which a running cluster fails, and then two equal-sized
# sub-clusters are both trying to form a new cluster.  The
# cluster lock may be configured using either a lock disk
# or a quorum server.
#
# You can use either the quorum server or the lock disk as
# a cluster lock but not both in the same cluster.
#
# Consider the following when configuring a cluster.
# For a two-node cluster, you must use a cluster lock.  For
# a cluster of three or four nodes, a cluster lock is strongly
# recommended.  For a cluster of more than four nodes, a
# cluster lock is recommended.  If you decide to configure
# a lock for a cluster of more than four nodes, it must be
# a quorum server.

# Lock Disk Parameters.  Use the FIRST_CLUSTER_LOCK_VG and
# FIRST_CLUSTER_LOCK_PV parameters to define a lock disk.
# The FIRST_CLUSTER_LOCK_VG is the LVM volume group that
# holds the cluster lock. This volume group should not be
# used by any other cluster as a cluster lock device.

# Quorum Server Parameters. Use the QS_HOST, QS_POLLING_INTERVAL,
# and QS_TIMEOUT_EXTENSION parameters to define a quorum server.
# The QS_HOST is the host name or IP address of the system
# that is running the quorum server process.  The
# QS_POLLING_INTERVAL (microseconds) is the interval at which
# Serviceguard checks to make sure the quorum server is running.
# The optional QS_TIMEOUT_EXTENSION (microseconds) is used to increase
# the time interval after which the quorum server is marked DOWN.
#
# The default quorum server timeout is calculated from the
# Serviceguard cluster parameters, including NODE_TIMEOUT and
# HEARTBEAT_INTERVAL.  If you are experiencing quorum server
# timeouts, you can adjust these parameters, or you can include
# the QS_TIMEOUT_EXTENSION parameter.
#
# For example, to configure a quorum server running on node
# "qshost" with 120 seconds for the QS_POLLING_INTERVAL and to
# add 2 seconds to the system assigned value for the quorum server
# timeout, enter:
#
# QS_HOST qshost
# QS_POLLING_INTERVAL 120000000
# QS_TIMEOUT_EXTENSION 2000000



# Definition of nodes in the cluster.
# Repeat node definitions as necessary for additional nodes.
# NODE_NAME is the specified nodename in the cluster.
# It must match the hostname and both cannot contain full domain name.
# Each NETWORK_INTERFACE, if configured with IPv4 address,
# must have ONLY one IPv4 address entry with it which could
# be either HEARTBEAT_IP or STATIONARY_IP.
# Each NETWORK_INTERFACE, if configured with IPv6 address(es)
# can have multiple IPv6 address entries(up to a maximum of 2,
# only one IPv6 address entry belonging to site-local scope
# and only one belonging to global scope) which must be all
```

```
# STATIONARY_IP. They cannot be HEARTBEAT_IP.

FIRST_CLUSTER_LOCK_VG     /dev/db2inst1_1

NODE_NAME                 rx4640a
  NETWORK_INTERFACE       lan0
    STATIONARY_IP192.6.173.2
  NETWORK_INTERFACE       lan1
  NETWORK_INTERFACE       lan2
    HEARTBEAT_IP 192.168.1.1
FIRST_CLUSTER_LOCK_PV     /dev/dsk/c7t12d0
# List of serial device file names
# For example:
# SERIAL_DEVICE_FILE      /dev/tty0p0

# Possible standby Network Interfaces for lan0: lan1.

NODE_NAME                 rx4640b
  NETWORK_INTERFACE       lan0
    STATIONARY_IP192.6.173.7
  NETWORK_INTERFACE       lan1
  NETWORK_INTERFACE       lan2
    HEARTBEAT_IP 192.168.1.2
FIRST_CLUSTER_LOCK_PV     /dev/dsk/c7t12d0
# List of serial device file names
# For example:
# SERIAL_DEVICE_FILE      /dev/tty0p0

# Possible standby Network Interfaces for lan0: lan1.

#NODE_NAME                rx4640c
#  NETWORK_INTERFACE      lan0
#    HEARTBEAT_IP192.6.173.12
#  NETWORK_INTERFACE      lan1
# List of serial device file names
# For example:
# SERIAL_DEVICE_FILE      /dev/tty0p0

# Possible standby Network Interfaces for lan0: lan1.

#NODE_NAME                rx4640d
#  NETWORK_INTERFACE      lan0
#    HEARTBEAT_IP192.6.173.17
#  NETWORK_INTERFACE      lan1
# List of serial device file names
# For example:
# SERIAL_DEVICE_FILE      /dev/tty0p0

# Possible standby Network Interfaces for lan0: lan1.


# Cluster Timing Parameters (microseconds).

# The NODE_TIMEOUT parameter defaults to 2000000 (2 seconds).
# This default setting yields the fastest cluster reformations.
# However, the use of the default value increases the potential
# for spurious reformations due to momentary system hangs or
# network load spikes.
# For a significant portion of installations, a setting of
# 5000000 to 8000000 (5 to 8 seconds) is more appropriate.
# The maximum value recommended for NODE_TIMEOUT is 30000000
# (30 seconds).

HEARTBEAT_INTERVAL                1000000
NODE_TIMEOUT              2000000


# Configuration/Reconfiguration Timing Parameters (microseconds).

AUTO_START_TIMEOUT        600000000
NETWORK_POLLING_INTERVAL  2000000

# Package Configuration Parameters.
# Enter the maximum number of packages which will be configured in the cluster.
# You can not add packages beyond this limit.
# This parameter is required.
MAX_CONFIGURED_PACKAGES           9
```

```
# List of cluster aware LVM Volume Groups. These volume groups will
# be used by package applications via the vgchange -a e command.
# Neither CVM or VxVM Disk Groups should be used here.
# For example:
# VOLUME_GROUP            /dev/vgdatabase
# VOLUME_GROUP            /dev/vg02

VOLUME_GROUP              /dev/db2inst1_1
VOLUME_GROUP              /dev/db2inst2_0
VOLUME_GROUP              /dev/db2inst1_0
VOLUME_GROUP              /dev/db2inst1
```

Appendix 3: "db2inst1_0" package configuration file (db2inst1_0.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 MC/Serviceguard Serviceguard OPS Edition 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                      db2inst1_0

# Enter the package type for this package. PACKAGE_TYPE indicates
# whether this package is to run as a FAILOVER 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.
#
#    SYSTEM_MULTI_NODE
#                 package runs on multiple 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 SYSTEM_MULTI_NODE are
#       not failover packages and should only be used for applications
#       provided by Hewlett-Packard.
#
#       Since SYSTEM_MULTI_NODE packages run on multiple nodes at
#       one time, following parameters are ignored:
#
#          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 multiple node packages. If
#       volume groups are assigned to multiple node packages they must
#       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   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 cluster is to be specified and order is not
# important, "NODE_NAME *" may be specified.
#
# Example : NODE_NAME  *

NODE_NAME                rx4640a
NODE_NAME                rx4640b


# 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 specified for all services.

RUN_SCRIPT                      /etc/cmcluster/db2inst1_0/db2inst1_0.cntl
RUN_SCRIPT_TIMEOUT              NO_TIMEOUT
HALT_SCRIPT                     /etc/cmcluster/db2inst1_0/db2inst1_0.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 VxVM-CVM-pkg 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 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 names used by
# cmrunserv and cmhaltserv commands in the run and halt scripts.
#
# 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 in the 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                    db2inst1_0_service
SERVICE_FAIL_FAST_ENABLED       no
SERVICE_HALT_TIMEOUT            30


# 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.

#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>]
```

Appendix 4: “db2inst1_0” package control script (db2inst1_0.cntl)

```
#"(#) A.11.15.00   $Date: 07/16/03 $"
# **********************************************************************
# *                                                                    *
# *        HIGH AVAILABILITY PACKAGE CONTROL SCRIPT (template)         *
# *                                                                    *
# *       Note: This file MUST be edited before it can be used.        *
# *                                                                    *
# **********************************************************************

# The PACKAGE and NODE environment variables are set by
# Serviceguard at the time the control script is executed.
# Do not set these environment variables yourself!

# these environment variables are altered.

. ${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]="db2inst1_0"

# 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]=""

# 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 MC 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.

# 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/db2inst1_0/db2inst1_0"; FS[0]="/db/db2inst1/NODE0000"; FS_MOUNT_OPT[0]="";
FS_UMOUNT_OPT[0]=""; FS_FSCK_OPT[0]=""
#FS_TYPE[0]=""
#
# 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=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

# 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 DISK GROUP OPERATIONS
# Specify the number of concurrent VxVM DG imports or deports to allow
# during package startup or shutdown.
# Setting this value to an appropriate number may improve the performance
# while importing or deporting a large number of disk 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_DISKGROUP_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_DISKGROUP_OPERATIONS=50
#      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]="192.6.173.4"
SUBNET[0]="192.6.173.0"

# 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]="db2inst1_0_service"
SERVICE_CMD[0]="/etc/cmcluster/db2inst1_0/DB2.sh monitor db2inst1 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]=



# 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.
#
#
# Set the following before running!
#
DB2_INSTANCE=db2inst1 # Add your specific instance name here
DB2_PARTITION=0 # Add your specific Partition name here

function customer_defined_run_cmds
{
# ADD customer defined run commands.
if [ -x /etc/cmcluster/${DB2_INSTANCE}_${DB2_PARTITION}/nfs_xmnt ]
#
# Must be multi-partition if nfs_xmnt exists.
# IMPORTANT: Do not put a nfs_xmnt file in the package directory
#  if this is a sungle partition DB.
#
then
  echo "Multiple Partition Mode."
  /etc/cmcluster/${DB2_INSTANCE}_${DB2_PARTITION}/nfs_xmnt start
  counter=30
  while [ ${counter} -gt 0 ]
  do
    if [[ -x /home/${DB2_INSTANCE}/sqllib/bin/db2gcf ]]
    then
      /etc/cmcluster/${DB2_INSTANCE}_${DB2_PARTITION}/DB2.sh start ${DB2_INSTANCE} ${DB2_PARTITION}
      counter=$((${counter}-30))
    else
    print "Failed to start DB2 ... \n"
    print "Waiting for DB2 user home directory to be available.\n"
    sleep 1
    counter=$((${counter}-1))
   fi
  done
else
    echo "Single Partition Mode."
    /etc/cmcluster/${DB2_INSTANCE}_${DB2_PARTITION}/DB2.sh start ${DB2_INSTANCE}
 ${DB2_PARTITION}
fi
}

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

```
function customer_defined_halt_cmds
{
# ADD customer defined halt commands.
/etc/cmcluster/${DB2_INSTANCE}_${DB2_PARTITION}/DB2.sh halt ${DB2_INSTANCE}${DB2
_PARTITION}
}

# END OF CUSTOMER DEFINED FUNCTIONS
```

NOTE: For brevity, the remainder of this file has been truncated, as it was not changed from the standard Serviceguard template.

Appendix 5: “db2inst1_1” package configuration file (db2inst1_1.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 MC/Serviceguard Serviceguard OPS Edition 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                    db2inst1_1

# Enter the package type for this package. PACKAGE_TYPE indicates
# whether this package is to run as a FAILOVER 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.
#
#    SYSTEM_MULTI_NODE
#                 package runs on multiple 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 SYSTEM_MULTI_NODE are
#       not failover packages and should only be used for applications
#       provided by Hewlett-Packard.
#
#       Since SYSTEM_MULTI_NODE packages run on multiple nodes at
#       one time, following parameters are ignored:
#
#          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 multiple node packages. If
#       volume groups are assigned to multiple node packages they must
#       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   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 cluster is to be specified and order is not
# important, "NODE_NAME *" may be specified.
#
# Example : NODE_NAME  *

NODE_NAME                 rx4640b
NODE_NAME                 rx4640a


# 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 specified for all services.

RUN_SCRIPT                        /etc/cmcluster/db2inst1_1/db2inst1_1.cntl
RUN_SCRIPT_TIMEOUT                NO_TIMEOUT
HALT_SCRIPT                       /etc/cmcluster/db2inst1_1/db2inst1_1.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 VxVM-CVM-pkg 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 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 names used by
# cmrunserv and cmhaltserv commands in the run and halt scripts.
#
# 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 in the 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                    db2inst1_1_service
SERVICE_FAIL_FAST_ENABLED       no
SERVICE_HALT_TIMEOUT            30


# 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.

#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>]
```

Appendix 6: “db2inst1_1” package control script (db2inst1_1.cntl)

```
#"(#) A.11.15.00   $Date: 07/16/03 $"
# **********************************************************************
# *                                                                    *
# *        HIGH AVAILABILITY PACKAGE CONTROL SCRIPT (template)         *
# *                                                                    *
# *       Note: This file MUST be edited before it can be used.        *
# *                                                                    *
# **********************************************************************

# The PACKAGE and NODE environment variables 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.

. ${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]="db2inst1_1"

# 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]=""

# 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 MC 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.

# 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/db2inst1_1/db2inst2_1"; FS[0]="/db/db2inst1/NODE0001"; FS_MOUNT_OPT[0]="";
FS_UMOUNT_OPT[0]=""; FS_FSCK_OPT[0]=""
#FS_TYPE[0]=""
#
# 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=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

# 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 DISK GROUP OPERATIONS
# Specify the number of concurrent VxVM DG imports or deports to allow
# during package startup or shutdown.
# Setting this value to an appropriate number may improve the performance
# while importing or deporting a large number of disk 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_DISKGROUP_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_DISKGROUP_OPERATIONS=50
#      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]="192.6.173.5"
SUBNET[0]="192.6.173.0"

# 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]="db2inst1_1_service"
SERVICE_CMD[0]="/etc/cmcluster/db2inst1_1/DB2.sh monitor db2inst1 1"
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]=



# 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.
#
#
# Set the following before running!
#
DB2_INSTANCE=db2inst1 # Add your specific instance name here
DB2_PARTITION=1 # Add your specific Partition name here

function customer_defined_run_cmds
{
# ADD customer defined run commands.
if [ -x /etc/cmcluster/${DB2_INSTANCE}_${DB2_PARTITION}/nfs_xmnt ]
#
# Must be multi-partition if nfs_xmnt exists.
# IMPORTANT: Do not put a nfs_xmnt file in the package directory
#  if this is a sungle partition DB.
#
then
  echo "Multiple Partition Mode."
  /etc/cmcluster/${DB2_INSTANCE}_${DB2_PARTITION}/nfs_xmnt start
  counter=30
  while [ ${counter} -gt 0 ]
  do
    if [[ -x /home/${DB2_INSTANCE}/sqllib/bin/db2gcf ]]
    then
      /etc/cmcluster/${DB2_INSTANCE}_${DB2_PARTITION}/DB2.sh start ${DB2_INSTANCE} ${DB2_PARTITION}
      counter=$((${counter}-30))
    else
    print "Failed to start DB2 ... \n"
    print "Waiting for DB2 user home directory to be available.\n"
    sleep 1
    counter=$((${counter}-1))
   fi
  done
else
    echo "Single Partition Mode."
    /etc/cmcluster/${DB2_INSTANCE}_${DB2_PARTITION}/DB2.sh start ${DB2_INSTANCE}
 ${DB2_PARTITION}
fi
}

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

```
function customer_defined_halt_cmds
{
# ADD customer defined halt commands.
/etc/cmcluster/${DB2_INSTANCE}_${DB2_PARTITION}/DB2.sh halt ${DB2_INSTANCE}${DB2
_PARTITION}
}


# END OF CUSTOMER DEFINED FUNCTIONS
```

NOTE: For brevity, the remainder of this file has been truncated, as it was not changed from the standard Serviceguard template.

Appendix 7: "db2inst2_0" package configuration file (db2inst2_0.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 MC/Serviceguard Serviceguard OPS Edition 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                    db2inst2_0

# Enter the package type for this package. PACKAGE_TYPE indicates
# whether this package is to run as a FAILOVER 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.
#
#    SYSTEM_MULTI_NODE
#                 package runs on multiple 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 SYSTEM_MULTI_NODE are
#       not failover packages and should only be used for applications
#       provided by Hewlett-Packard.
#
#       Since SYSTEM_MULTI_NODE packages run on multiple nodes at
#       one time, following parameters are ignored:
#
#          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 multiple node packages. If
#       volume groups are assigned to multiple node packages they must
#       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   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 cluster is to be specified and order is not
# important, "NODE_NAME *" may be specified.
#
# Example : NODE_NAME  *

NODE_NAME                rx4640a
NODE_NAME                rx4640b


# 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 specified for all services.

RUN_SCRIPT                      /etc/cmcluster/db2inst2_0/db2inst2_0.cntl
RUN_SCRIPT_TIMEOUT              NO_TIMEOUT
HALT_SCRIPT                     /etc/cmcluster/db2inst2_0/db2inst2_0.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 VxVM-CVM-pkg 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 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 names used by
# cmrunserv and cmhaltserv commands in the run and halt scripts.
#
# 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 in the 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                    db2inst2_0_service
SERVICE_FAIL_FAST_ENABLED       no
SERVICE_HALT_TIMEOUT            30


# 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.

#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>]
```

Appendix 8: “db2inst2_0” package control script (db2inst2_0.cntl)

```
#"(#) A.11.15.00   $Date: 07/16/03 $"
# **********************************************************************
# *                                                                    *
# *        HIGH AVAILABILITY PACKAGE CONTROL SCRIPT (template)         *
# *                                                                    *
# *       Note: This file MUST be edited before it can be used.        *
# *                                                                    *
# **********************************************************************
# The PACKAGE and NODE environment variables 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.

. ${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]="db2inst2_0"

# 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]=""

# 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 MC 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.

# 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/db2inst2_0/db2inst2_0"; FS[0]="/db/db2inst2/NODE0000"; FS_MOUNT_OPT[0]="";
FS_UMOUNT_OPT[0]=""; FS_FSCK_OPT[0]=""
LV[0]="/dev/db2inst2_0/db2inst2_0"; FS[0]="/db2inst2"; FS_MOUNT_OPT[0]=""; FS_UMOUNT_OPT[0]="";
FS_FSCK_OPT[0]=""
#FS_TYPE[0]=""
#
# 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=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

# 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 DISK GROUP OPERATIONS
# Specify the number of concurrent VxVM DG imports or deports to allow
# during package startup or shutdown.
# Setting this value to an appropriate number may improve the performance
# while importing or deporting a large number of disk 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_DISKGROUP_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_DISKGROUP_OPERATIONS=50
#      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]="192.6.173.6"
SUBNET[0]="192.6.173.0"

# 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]="db2inst2_0_service"
SERVICE_CMD[0]="/etc/cmcluster/db2inst2_0/DB2.sh monitor db2inst2 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]=



# 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.

DB2_INSTANCE=db2inst2
DB2_PARTITION=0
function customer_defined_run_cmds
{
# ADD customer defined run commands.
if [ -x /etc/cmcluster/${DB2_INSTANCE}_${DB2_PARTITION}/nfs_xmnt ]
#
# Must be multi-partition if nfs_xmnt exists.
# IMPORTANT: Do not put a nfs_xmnt file in the package directory
#  if this is a sungle partition DB.
#
then
  echo "Multiple Partition Mode."
  /etc/cmcluster/${DB2_INSTANCE}_${DB2_PARTITION}/nfs_xmnt start
  counter=30
  while [ ${counter} -gt 0 ]
  do
    if [[ -x /home/${DB2_INSTANCE}/sqllib/bin/db2gcf ]]
    then
      /etc/cmcluster/${DB2_INSTANCE}_${DB2_PARTITION}/DB2.sh start ${DB2_INSTANCE} ${DB2_PARTITION}
      counter=$((${counter}-30))
    else
    print "Failed to start DB2 ... \n"
    print "Waiting for DB2 user home directory to be available.\n"
    sleep 1
    counter=$((${counter}-1))
   fi
  done
else
    echo "Single Partition Mode."
    /etc/cmcluster/${DB2_INSTANCE}_${DB2_PARTITION}/DB2.sh start ${DB2_INSTANCE} ${DB2_PARTITION}
fi
}

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

function customer_defined_halt_cmds
{
# ADD customer defined halt commands.
/etc/cmcluster/${DB2_INSTANCE}_${DB2_PARTITION}/DB2.sh halt ${DB2_INSTANCE}${DB2_PARTITION}
```

```
}

# END OF CUSTOMER DEFINED FUNCTIONS
```

NOTE: For brevity, the remainder of this file has been truncated, as it was not changed from the standard Serviceguard template.

Appendix 9: DB2 monitor script (DB2.sh) (Also functions as startup/shutdown and monitor script)

```
#!/usr/bin/sh
# set -x # for debugg purposes only
#VERSION="@(#) A.10.10             $Revision: 1.10 $ $Date: 97/10/02 12:27:02 $"
# ******************************************************************************
# **** Startup, Shutdown and Monitor Scripts for DB2 (Template) ***
# ******************************************************************************
# ******* Note: This file MUST be edited before it can be used. ********
# ******************************************************************************
#
# The following variables need to be set:
#
#
# MONITOR_INTERVAL:       How often (in seconds) to monitor DB2's state.
#
#
# TIME_OUT:               The amount of time in seconds you want to wait for the
#                         DB2 abort to complete before resorting to killing
#                         the DB2 processes.  The TIME_OUT variable is to
#                         protect against the worst case of a hung database
#                         that would prevent the halt script from completing
#                         and the standby node from starting up the data base.
#                         The value of TIME_OUT must be less than the time out
#                         variable set in the package configuration file.
#
# Examples:
# MONITOR_INTERVAL=10
# TIME_OUT=250
#

MONITOR_INTERVAL=5
TIME_OUT=25
#
# Only valid values for DB_TYPE are "SINGLE_PARTITION" or "MULTIPLE_PARTITION"
DB_TYPE="SINGLE_PARTITION"
#DB_TYPE="MULTIPLE_PARTITION"

#
# This version of the DB2.sh script no longer monitors individual
# DB2 processes.  An IBM DB2 utility, db2gcf, is now used to monitor
# the health of DB2.
#
#
HOST=`hostname`
DATE=`date`
PATH=/sbin:/usr/sbin:/usr/bin
ACTION=${1}
DB2INSTANCE=${2}
DB2NODENUM=${3}
PACKAGE_NAME="${DB2INSTANCE}_${DB2NODENUM}"


userhome=~${DB2INSTANCE?}
eval userhome=$userhome
INSTHOME=${userhome}

################################################################################
# Function: DB2_run_cmds
#
# This function is used to start the DB2 instance.
################################################################################

function DB2_run_cmds
{
#   If using MC/Serviceguard A.10.10 without any patches, you must uncomment
#   the following line; otherwise, this script will be killed before the startup
```

```
#   can be completed.

#   trap "" SIGTERM
#
    export DB2INSTANCE
    if [[ "${DB_TYPE}" = "SINGLE_PARTITION" ]]
    then
      print"\nCleaning up IPC's since we are in SINGLE_PARTITION mode. \n"
      print"\nIf in MULTIPLE_PARTITION mode, tis will cause problems. \n"
      print"\nGo into DB2.sh and correct problem.\n"
      $INSTHOME/sqllib/bin/db2gcf -k -t 30 -i $DB2INSTANCE -p ${DB2NODENUM}
    fi
    $INSTHOME/sqllib/bin/db2gcf -u -t 300 -i $DB2INSTANCE -p ${DB2NODENUM}  -L
    $INSTHOME/sqllib/bin/db2gcf -u -t 300 -i $DB2INSTANCE -p ${DB2NODENUM} -L
    temp_rc=$?

    if [[ ${temp_rc} != 0 ]]
    then
        print "\tERROR code " ${temp_rc} " occurred: Failed to start DB2 server ${DB2INSTANCE}"
    else
         print "\tStarted DB2 instance ${DB2INSTANCE} ${DB2NODENUM} on node  ${HOST}"
        $INSTHOME/sqllib/bin/db2gcf -s -t 300 -i $DB2INSTANCE -p ${DB2NODENUM} -L
        rc=$?
         print "\tdb2gcf -s reports rc of $rc\n"
    fi
}

################################################################################
# Function: DB2_halt_cmds
#
# The command below is this scripts calling itself with the kill option to
# kill any process that will not die normally after waiting for the TIME_OUT
# period.
################################################################################

function DB2_halt_cmds
{
#   Just in case things are hung, start a process that will wait for the timeout
#   period, then kill any remaining processes.  We'll need to monitor this
#   process (set -m), so we can terminate it later if it is not needed.

    set -m
   ${0} kill ${DB2INSTANCE} ${DB2NODENUM} &

#   If the server is up and running, then we shut it down.
    export DB2INSTANCE
    $INSTHOME/sqllib/bin/db2gcf -d -t 20 -i $DB2INSTANCE -p ${DB2NODENUM}
    temp_rc=$?

    if [[ ${temp_rc} != 0 ]]
    then
        print "ERROR: Failed to halt DB2 server ${DB2INSTANCE} cleanly: error ${temp_rc}"
    else
         print "Halted DB2 server ${DB2INSTANCE}."
    fi

#   Make sure all processes have gone away before saying shutdown is complete.
#   This stops the other node from starting up the package before it has been
#   stopped and the file system has been unmounted.

    typeset -i c
    typeset -i num_procs=${#MONITOR_PROCESSES[@]}

    while true
    do
         for i in ${MONITOR_PROCESSES[@]}
         do
            ps -fu ${DB2INSTANCE} | grep ${i} | grep "${DB2NODENUM}[ ]*$" | grep -v grep > /dev/null
            if [[ $? != 0 ]]
            then
                print "\n *** ${i} process has stopped. ***\n"
                c=0
                while (( c < $num_procs ))
                do
                    if [[ ${MONITOR_PROCESSES[$c]} = $i ]]
                    then
                        unset MONITOR_PROCESSES[$c]
                        c=$num_procs
```

```
                    fi
                    (( c = c + 1 ))
                done
            fi
        done

        if [[ ${MONITOR_PROCESSES[@]} = "" ]]
        then
#           Looks like shutdown was successful, so get rid of the script to
#           kill any hung processes, which we started earlier.  Check to see if
#           the script is still running.  If jobs returns that the script is
#           done, then we don't need to kill it.

            job=$(jobs | grep -v Done)
            if [[ ${job} != "" ]]
            then
                print "Killing the kill_hung_processes script\n"
                kill %1
            fi
            exit 0
        fi

        sleep 5
    done
}

################################################################################
# Function: terminate
#
# Called when a SIGTERM is trapped.
################################################################################

function terminate
{
     print "This monitor script has been signaled to terminate\n"
     exit
}

################################################################################
# Function: kill_hung_processes
#
# In case of a database hang we spawn a background process that will issue
# a SIGKILL to each of the monitored processes.  This allows us to guarantee
# a time limit for clean halt attempts.
################################################################################

function kill_hung_processes
{
#   Wait for the TIME_OUT period before sending a kill signal.  The TIME_OUT
#   value should be less than the MC/Serviceguard package time out.


    sleep ${TIME_OUT}

    for i in ${MONITOR_PROCESSES[@]}
    do
        id=`ps -fu "${DB2INSTANCE}" | grep "${DB2NODENUM}[ ]*$" | awk '/'${i}'/ { print $2 }'`
        if [[ ${id} != "" ]]
        then
            echo "Killing PID ${id}"
             kill -9 ${id}
             if [[ $? != 0 ]]
             then
                 print "\n *** ${0} kill_hung_processes function did NOT find process ***\n *** ${i}
running. ***\n"
                exit 0
             else
                 print "\n *** ${0} kill_hung_processes function did find process ***\n *** ${i}
running.  Sent SIGKILL. ***\n"
             fi
        else
             print "\n ***  kill_hung_processes function did NOT find ANY processes running. ***\n"
            exit 0
        fi
    done
}

################################################################################
```

```
# Function: monitor_processes
#############################################################################

function monitor_processes
{

 while true
 do
    $INSTHOME/sqllib/bin/db2gcf -s -i $DB2INSTANCE -p $DB2NODENUM 2> /dev/null > /dev/null
    rc=$?
    if [ $rc != 0 ]; then
        print "\n\n ERROR:Output of db2gcf:\n"
        $INSTHOME/sqllib/bin/db2gcf -s -i $DB2INSTANCE -p $DB2NODENUM
        print "\n\n"
        ps -ef
        print "\n *** DB2 has failed.  Aborting DB2. ***"
        set -m
        print "calling ${0} fault ${DB2INSTANCE?} ${DB2NODENUM?}"
        nohup ${0} fault ${DB2INSTANCE?} ${DB2NODENUM?} &    # The script calls itself with the fault
option.
        set +m
           sleep 999999
    fi
    sleep ${MONITOR_INTERVAL}
 done

}

#############################################################################
# Function: halt_package
#
# Because there is no move command in MC/Serviceguard so the package has to be
# halted first, then disabled from running on the host, then enabled
# to run in the cluster.
#############################################################################

function halt_package
{
    cmhaltpkg ${PACKAGE_NAME}
    cmmodpkg -d -n ${HOST} ${PACKAGE_NAME}
    sleep 1
    cmmodpkg -e ${PACKAGE_NAME}
}

#############################################################################
# FUNCTION STARTUP SECTION.
#
# Test to see if we are being called to run the application, or halt the
# application.
#############################################################################


trap terminate SIGTERM

if [[ -z ${ACTION} || -z ${DB2INSTANCE} ]] then
         print "\n\n *** USAGE: ${0} ACTION DB2INSTANCE [DB2NODENUM] ***\n\n"
         exit
fi

print "\n *** ${HOST}: ${0} ${ACTION} ${DB2INSTANCE} ${DB2NODENUM} at ${DATE} ***\n"

case ${ACTION} in

    fault)
        halt_package
    ;;

    kill)
        kill_hung_processes
    ;;

    monitor)
        monitor_processes
    ;;

    start)
        DB2_run_cmds
    ;;
```

```
    halt)
        DB2_halt_cmds
    ;;
esac
```

Appendix 10: Output of bdf on rx4640a

```
Filesystem          kbytes    used   avail %used Mounted on
/dev/vg00/lvol3     524288  193184  328584   37% /
/dev/vg00/lvol1     524288  118792  402368   23% /stand
/dev/vg00/lvol8    8912896 8793088  119808   99% /var
/dev/vg00/lvol7    8388608 2112456 6227144   25% /usr
/dev/vg00/lvol6    8388608   46600 8276872    1% /tmp
/dev/vg00/lvol5    8388608 3319936 5029144   40% /opt
/dev/vg00/lvol4    8388608 1376224 6957664   17% /home
/dev/db2inst1_0/db2inst1_0
                   71675904   80561 67120638    0% /db/db2inst1/NODE0000
db2inst1home:/db2inst1
                   143351808 1214208 133254048    1% /home/db2inst1
/dev/db2inst2_0/db2inst2_0
                   71675904 1354906 65925938    2% /db/db2inst2/NODE0000
```

Appendix 11: Contents of /etc/fstab on rx4640a

```
# System /etc/fstab file.  Static information about the file systems
# See fstab(4) and sam(1M) for further details on configuring devices.
/dev/vg00/lvol3 / vxfs delaylog 0 1
/dev/vg00/lvol1 /stand vxfs tranflush 0 1
/dev/vg00/lvol4 /home vxfs delaylog 0 2
/dev/vg00/lvol5 /opt vxfs delaylog 0 2
/dev/vg00/lvol6 /tmp vxfs delaylog 0 2
/dev/vg00/lvol7 /usr vxfs delaylog 0 2
/dev/vg00/lvol8 /var vxfs delaylog 0 2
```

Appendix 12: Contents of /etc/hosts on rx4640a

```
## Configured using SAM by root on Wed Apr 21 10:42:46 2004
# @(#)B11.23_LRhosts $Revision: 1.9.214.1 $ $Date: 96/10/08 13:20:01 $
#
# The form for each entry is:
# <internet address>      <official hostname> <aliases>
#
# For example:
# 192.1.2.34     hpfcrm   loghost
#
# See the hosts(4) manual page for more information.
# Note: The entries cannot be preceded by a space.
#       The format described in this file is the correct format.
#       The original Berkeley manual page contains an error in
#       the format description.
#

192.6.173.2      rx4640a
192.168.1.1      rx4640a
192.6.173.7      rx4640b
192.6.173.12     rx4640c
192.6.173.17     rx4640d
192.6.173.3      db2inst1home
192.6.173.4      db2inst1_0
192.6.173.5      db2inst1_1
192.6.173.6      db2inst2
127.0.0.1        localhost        loopback
```

Appendix 13: Output of ioscan on rx4640a

```
Class       I  H/W Path       Driver     S/W State   H/W Type     Description
=============================================================================
root        0                  root       CLAIMED     BUS_NEXUS
ioa         0  0               sba        CLAIMED     BUS_NEXUS    System Bus Adapter (1229)
ba          1  0/0             lba        CLAIMED     BUS_NEXUS    Local PCI-X Bus Adapter (122e)
tty         0  0/0/1/0         asio0      CLAIMED     INTERFACE    PCI SimpleComm (103c1290)
tty         1  0/0/1/1         asio0      CLAIMED     INTERFACE    PCI Serial (103c1048)
usb         0  0/0/2/0         hcd        CLAIMED     INTERFACE    NEC USB Interface
usbhub      0  0/0/2/0.1       hub        CLAIMED     DEVICE       USB Root Hub
usbhid      0  0/0/2/0.1.2     hid        CLAIMED     DEVICE       USB HID Kbd(0) Mouse(1)
usb         1  0/0/2/1         hcd        CLAIMED     INTERFACE    NEC USB Interface
usbhub      1  0/0/2/1.1       hub        CLAIMED     DEVICE       USB Root Hub
usb         2  0/0/2/2         ehci       CLAIMED     INTERFACE    NEC USB Interface
sideba      0  0/0/3/0         side_multi CLAIMED     INTERFACE    CMD IDE controller
ext_bus     0  0/0/3/0.0       side       CLAIMED     INTERFACE    IDE Primary Channel
target      0  0/0/3/0.0.0     tgt        CLAIMED     DEVICE
disk        0  0/0/3/0.0.0.0   sdisk      CLAIMED     DEVICE       TEAC    DW-224E
target      1  0/0/3/0.0.7     tgt        CLAIMED     DEVICE
ctl         0  0/0/3/0.0.7.0   sctl       CLAIMED     DEVICE       Initiator
ext_bus     1  0/0/3/0.1       side       CLAIMED     INTERFACE    IDE Secondary Channel
target      2  0/0/3/0.1.7     tgt        CLAIMED     DEVICE
ctl         1  0/0/3/0.1.7.0   sctl       CLAIMED     DEVICE       Initiator
graphics    0  0/0/4/0         gvid_core  CLAIMED     INTERFACE    PCI Display (10025159)
ba          2  0/1             lba        CLAIMED     BUS_NEXUS    Local PCI-X Bus Adapter (122e)
ext_bus     2  0/1/1/0         c8xx       CLAIMED     INTERFACE    SCSI C1010 Ultra160 Wide LVD A6829-
60101
target      3  0/1/1/0.0       tgt        CLAIMED     DEVICE
disk        1  0/1/1/0.0.0     sdisk      CLAIMED     DEVICE       HP 73.4GMAS3735NC
target      4  0/1/1/0.4       tgt        CLAIMED     DEVICE
ctl         2  0/1/1/0.4.0     sctl       CLAIMED     DEVICE       HP      SAF-TE
target      5  0/1/1/0.6       tgt        CLAIMED     DEVICE
ctl         3  0/1/1/0.6.0     sctl       CLAIMED     DEVICE       Initiator
ext_bus     3  0/1/1/1         c8xx       CLAIMED     INTERFACE    SCSI C1010 Ultra160 Wide LVD A6829-
60101
target      6  0/1/1/1.6       tgt        CLAIMED     DEVICE
ctl         4  0/1/1/1.6.0     sctl       CLAIMED     DEVICE       Initiator
lan         0  0/1/2/0         igelan     CLAIMED     INTERFACE    HP A6825-60101 PCI 1000Base-T
Adapter
ba          3  0/2             lba        CLAIMED     BUS_NEXUS    Local PCI-X Bus Adapter (122e)
ext_bus     4  0/2/1/0         c8xx       CLAIMED     INTERFACE    SCSI C1010 Ultra160 Wide LVD A6829-
60101
target      7  0/2/1/0.6       tgt        CLAIMED     DEVICE
ctl         5  0/2/1/0.6.0     sctl       CLAIMED     DEVICE       Initiator
ext_bus     5  0/2/1/1         c8xx       CLAIMED     INTERFACE    SCSI C1010 Ultra160 Wide LVD A6829-
60101
target      8  0/2/1/1.1       tgt        CLAIMED     DEVICE
tape        0  0/2/1/1.1.0     stape      CLAIMED     DEVICE       HP      C5683A
target      9  0/2/1/1.6       tgt        CLAIMED     DEVICE
ctl         6  0/2/1/1.6.0     sctl       CLAIMED     DEVICE       Initiator
ba          4  0/3             lba        CLAIMED     BUS_NEXUS    Local PCI-X Bus Adapter (122e)
fc          0  0/3/1/0         td         CLAIMED     INTERFACE    HP Tachyon XL2 Fibre Channel Mass
Storage Adapter
lan         1  0/3/2/0         igelan     CLAIMED     INTERFACE    HP A6825-60101 PCI 1000Base-T
Adapter
ba          5  0/4             lba        CLAIMED     BUS_NEXUS    Local PCI-X Bus Adapter (122e)
fc          1  0/4/1/0         td         CLAIMED     INTERFACE    HP Tachyon XL2 Fibre Channel Mass
Storage Adapter
lan         2  0/4/2/0         igelan     CLAIMED     INTERFACE    HP A6825-60101 PCI 1000Base-T
Adapter
ba          6  0/5             lba        CLAIMED     BUS_NEXUS    Local PCI-X Bus Adapter (122e)
ext_bus     6  0/5/1/0         c8xx       CLAIMED     INTERFACE    SCSI C1010 Ultra160 Wide LVD A6829-
60101
target     10  0/5/1/0.6       tgt        CLAIMED     DEVICE
ctl         7  0/5/1/0.6.0     sctl       CLAIMED     DEVICE       Initiator
ext_bus     7  0/5/1/1         c8xx       CLAIMED     INTERFACE    SCSI C1010 Ultra160 Wide LVD A6829-
60101
target     11  0/5/1/1.6       tgt        CLAIMED     DEVICE
ctl         8  0/5/1/1.6.0     sctl       CLAIMED     DEVICE       Initiator
target     12  0/5/1/1.8       tgt        CLAIMED     DEVICE
disk        2  0/5/1/1.8.0     sdisk      CLAIMED     DEVICE       HP 146 GMAP3147NC
target     13  0/5/1/1.9       tgt        CLAIMED     DEVICE
disk        3  0/5/1/1.9.0     sdisk      CLAIMED     DEVICE       HP 146 GST3146807LC
target     14  0/5/1/1.10      tgt        CLAIMED     DEVICE
disk       10  0/5/1/1.10.0    sdisk      CLAIMED     DEVICE       HP 73.4GST373453LC
```

```
target     15  0/5/1/1.11     tgt        CLAIMED     DEVICE
disk       11  0/5/1/1.11.0   sdisk      CLAIMED     DEVICE       HP 73.4GST373453LC
target     16  0/5/1/1.12     tgt        CLAIMED     DEVICE
disk       12  0/5/1/1.12.0   sdisk      CLAIMED     DEVICE       HP 73.4GST373453LC
target     17  0/5/1/1.15     tgt        CLAIMED     DEVICE
ctl         9  0/5/1/1.15.0   sctl       CLAIMED     DEVICE       HP      A6491A
processor   0  120            processor  CLAIMED     PROCESSOR    Processor
processor   1  121            processor  CLAIMED     PROCESSOR    Processor
processor   2  122            processor  CLAIMED     PROCESSOR    Processor
processor   3  123            processor  CLAIMED     PROCESSOR    Processor
ba          7  250            pdh        CLAIMED     BUS_NEXUS    Core I/O Adapter
ipmi        0  250/0          ipmi       CLAIMED     INTERFACE    IPMI Controller
acpi_node   0  250/1          acpi_node  CLAIMED     INTERFACE    Acpi Hardware
ba          0  255/255        swspBus    CLAIMED     VIRTBUS
```

Appendix 14: Output of lanscan on rx4640a

```
Hardware Station        Crd Hdw   Net-Interface  NM  MAC       HP-DLPI DLPI
Path     Address        In# State NamePPA        ID  Type      Support Mjr#
0/1/2/0  0x00306EF400DC 0   UP    lan0 snap0     1   ETHER     Yes     119
0/3/2/0  0x00306EF400A1 1   UP    lan1 snap1     2   ETHER     Yes     119
0/4/2/0  0x00306EF400ED 2   UP    lan2 snap2     3   ETHER     Yes     119
```

Appendix 15: Output of mount on rx4640a

```
/ on /dev/vg00/lvol3 ioerror=nodisable,log,dev=40000003 on Wed Jul  7 10:43:13 2004
/stand on /dev/vg00/lvol1 ioerror=mwdisable,log,tranflush,dev=40000001 on Wed Jul  7 10:43:14 2004
/var on /dev/vg00/lvol8 ioerror=mwdisable,delaylog,dev=40000008 on Wed Jul  7 10:43:27 2004
/usr on /dev/vg00/lvol7 ioerror=mwdisable,delaylog,dev=40000007 on Wed Jul  7 10:43:27 2004
/tmp on /dev/vg00/lvol6 ioerror=mwdisable,delaylog,dev=40000006 on Wed Jul  7 10:43:27 2004
/opt on /dev/vg00/lvol5 ioerror=mwdisable,delaylog,dev=40000005 on Wed Jul  7 10:43:27 2004
/home on /dev/vg00/lvol4 ioerror=mwdisable,delaylog,dev=40000004 on Wed Jul  7 10:43:28 2004
/net on -hosts ignore,indirect,nosuid,soft,nobrowse,dev=44000000 on Wed Jul  7 10:43:55 2004
/db/db2inst1/NODE0000 on /dev/db2inst1_0/db2inst1_0 ioerror=mwdisable,delaylog,dev=40070001 on Wed Jul
7 10:45:29 2004
/home/db2inst1 on db2inst1home:/db2inst1 rsize=32768,wsize=32768,NFSv3,dev=44000001 on Wed Jul  7
10:45:29 2004
/db/db2inst2/NODE0000 on /dev/db2inst2_0/db2inst2_0 ioerror=mwdisable,delaylog,dev=40080001 on Wed Jul
7 10:46:12 2004
```

Appendix 16: Contents of /etc/rc.config.d /netconf on rx4640a

```
# netconf:  configuration values for core networking subsystems
#
# @(#)B11.23_LR $Revision: 1.6.119.6 $ $Date: 97/09/10 15:56:01 $
#
# HOSTNAME:           Name of your system for uname -S and hostname
#
# OPERATING_SYSTEM:   Name of operating system returned by uname -s
#                     ---- DO NOT CHANGE THIS VALUE ----
#
# LOOPBACK_ADDRESS:   Loopback address
#                     ---- DO NOT CHANGE THIS VALUE ----
#
# IMPORTANT:  for 9.x-to-10.0 transition, do not put blank lines between
# the next set of statements

HOSTNAME="rx4640a"
OPERATING_SYSTEM=HP-UX
LOOPBACK_ADDRESS=127.0.0.1

# Internet configuration parameters.  See ifconfig(1m), autopush(1m)
#
# INTERFACE_NAME:     Network interface name (see lanscan(1m))
#
# IP_ADDRESS:         Hostname (in /etc/hosts) or IP address in decimal-dot
#                     notation (e.g., 192.1.2.3)
#
# SUBNET_MASK:        Subnetwork mask in decimal-dot notation, if different
#                     from default
#
# BROADCAST_ADDRESS:  Broadcast address in decimal-dot notation, if
#                     different from default
#
# INTERFACE_STATE:    Desired interface state at boot time.
#                     either up or down, default is up.
#
# DHCP_ENABLE         Determines whether or not DHCP client functionality
#                     will be enabled on the network interface (see
#                     auto_parms(1M), dhcpclient(1M)). DHCP clients get
#                     their IP address assignments from DHCP servers.
#                     1 enables DHCP client functionality; 0 disables it.
#
# For each additional network interfaces, add a set of variable assignments
# like the ones below, changing the index to "[1]", "[2]" et cetera.
#
# IMPORTANT:  for 9.x-to-10.0 transition, do not put blank lines between
# the next set of statements
```

```
INTERFACE_NAME[0]="lan0"
IP_ADDRESS[0]="192.6.173.2"
SUBNET_MASK[0]="255.255.255.128"
BROADCAST_ADDRESS[0]=""
INTERFACE_STATE[0]=""
DHCP_ENABLE[0]=0

# Internet routing configuration.  See route(1m), routing(7)
#
# ROUTE_DESTINATION:  Destination hostname (in /etc/hosts) or host or network
#                     IP address in decimal-dot notation, preceded by the word
#                     "host" or "net"; or simply the word "default".
#
# ROUTE_MASK:         Subnetwork mask in decimal-dot notation, or C language
#                     hexadecimal notation.  This is an optional field.
#                     A IP address, subnet mask pair uniquely identifies
#                     a subnet to be reached. If a subnet mask is not given,
#                     then the system will assign the longest subnet mask
#                     of the configured network interfaces to this route.
#                     If there is no matching subnet mask, then the system
#                     will assign the default network mask as the route's
#                     subnet mask.
#
# ROUTE_GATEWAY:      Gateway hostname (in /etc/hosts) or IP address in
#                     decimal-dot notation.  If local interface, must use the
#                     same form as used for IP_ADDRESS above (hostname or
#                     decimal-dot notation). If loopback interface, i.e.,
#                      127.0.0.1, the ROUTE_COUNT must be set to zero.
#
# ROUTE_COUNT:        An integer that indicates whether the gateway is a
#                     remote interface (one) or the local interface (zero)
#                      or loopback interface (e.g., 127.*).
#
# ROUTE_ARGS:         Route command arguments and options.  This variable
#                     may contain a combination of the following arguments:
#                     "-f", "-n" and "-p pmtu".
#
# For each additional route, add a set of variable assignments like the ones
# below, changing the index to "[1]", "[2]" et cetera.
#
# IMPORTANT:  for 9.x-to-10.0 transition, do not put blank lines between
# the next set of statements

ROUTE_DESTINATION[0]="default"
ROUTE_MASK[0]=""
ROUTE_GATEWAY[0]="192.6.173.1"
ROUTE_COUNT[0]="1"
ROUTE_ARGS[0]=""

# Dynamic routing daemon configuration.  See gated(1m)
#
# GATED:        Set to 1 to start gated daemon.
# GATED_ARGS:   Arguments to the gated daemon.

GATED=0
GATED_ARGS=""

#
# Router Discover Protocol daemon configuration.  See rdpd(1m)
#
# RDPD:         Set to 1 to start rdpd daemon
#

RDPD=0

#
# Reverse ARP daemon configuration.  See rarpd(1m)
#
# RARP:         Set to 1 to start rarpd daemon
#

RARP=0

IP_ADDRESS[2]=192.168.1.1
SUBNET_MASK[2]=255.255.255.0
INTERFACE_NAME[2]=lan2
BROADCAST_ADDRESS[2]=192.168.1.255
INTERFACE_STATE[2]=up
```

Appendix 17: Output of <netstat –r> on rx4640a

```
IPv4 Routing tables:
Dest/Netmask                    Gateway           Flags   Refs Interface  Pmtu
localhost/255.255.255.255       localhost         UH        0  lo0        4136
rx4640a/255.255.255.255         rx4640a           UH        0  lan2       4136
rx4640a/255.255.255.255         rx4640a           UH        0  lan0       4136
db2inst2/255.255.255.255        db2inst2          UH        0  lan0:2     4136
db2inst1_0/255.255.255.255      db2inst1_0        UH        0  lan0:1     4136
192.6.173.0/255.255.255.128     rx4640a           U         4  lan0       1500
192.6.173.0/255.255.255.128     db2inst2          U         4  lan0:2     1500
192.6.173.0/255.255.255.128     db2inst1_0        U         4  lan0:1     1500
192.168.1.0/255.255.255.0       rx4640a           U         2  lan2       1500
loopback/255.0.0.0              localhost         U         0  lo0           0
default/*                       192.6.173.1       UG        0  lan0          0

IPv6 Routing tables:
Destination/Prefix          Gateway                 Flags Refs Interface Pmtu
::1/128                     ::1                     UH      0  lo0       4136
```

Appendix 18: Contents of /etc/passwd on rx4640a

```
root:*:0:3::/:/bin/ksh
daemon:*:1:5::/:/sbin/sh
bin:*:2:2::/usr/bin:/sbin/sh
sys:*:3:3::/:
adm:*:4:4::/var/adm:/sbin/sh
uucp:*:5:3::/var/spool/uucppublic:/usr/lbin/uucp/uucico
lp:*:9:7::/var/spool/lp:/sbin/sh
nuucp:*:11:11::/var/spool/uucppublic:/usr/lbin/uucp/uucico
hpdb:*:27:1:ALLBASE:/:/sbin/sh
nobody:*:-2:-2::/:
www:*:30:1::/:
smbnull:*:101:101:DO NOT USE OR DELETE - needed by Samba:/home/smbnull:/sbin/sh
mysql:*:102:102::/home/mysql:/sbin/sh
db2inst1:*:103:200::/home/db2inst1:/bin/ksh
db2inst2:*:105:200::/home/db2inst2:/bin/ksh
```

Appendix 19: Contents of /etc/services on rx4640a

```
# @(#)B.11.23_ic71services $Revision: 1.32.214.7 $ $Date: 97/09/10 14:50:42 $
#
# This file associates official service names and aliases with
# the port number and protocol the services use.
#
# Some of the services represented below are not supported on HP-UX.
# They are provided solely as a reference.
#
# The form for each entry is:
# <official service name>  <port number/protocol name>  <aliases>
#
# See the services(4) manual page for more information.
# Note: The entries cannot be preceded by a blank space.
#
tcpmux          1/tcp                # TCP port multiplexer (RFC 1078)
echo            7/tcp                # Echo
echo            7/udp                #
discard         9/tcp  sink null     # Discard
discard         9/udp  sink null     #
systat         11/tcp  users         # Active Users
daytime        13/tcp                # Daytime
daytime        13/udp                #
qotd           17/tcp  quote         # Quote of the Day
chargen        19/tcp  ttytst source # Character Generator
chargen        19/udp  ttytst source #
ftp-data       20/tcp                # File Transfer Protocol (Data)
ftp            21/tcp                # File Transfer Protocol (Control)
telnet         23/tcp                # Virtual Terminal Protocol
smtp           25/tcp                # Simple Mail Transfer Protocol
time           37/tcp  timeserver    # Time
time           37/udp  timeserver    #
rlp            39/udp  resource      # Resource Location Protocol
whois          43/tcp  nicname       # Who Is
domain         53/tcp  nameserver    # Domain Name Service
```

```
domain        53/udp  nameserver     #
bootps        67/udp                 # Bootstrap Protocol Server
bootpc        68/udp                 # Bootstrap Protocol Client
tftp          69/udp                 # Trivial File Transfer Protocol
rje           77/tcp  netrjs         # private RJE Service
finger        79/tcp                 # Finger
http          80/tcp  www            # World Wide Web HTTP
http          80/udp  www            # World Wide Web HTTP
link          87/tcp  ttylink        # private terminal link
supdup        95/tcp                 #
hostnames    101/tcp  hostname       # NIC Host Name Server
tsap         102/tcp iso_tsap iso-tsap # ISO TSAP (part of ISODE)
pop          109/tcp postoffice pop2 # Post Office Protocol - Version 2
pop3         110/tcp  pop-3          # Post Office Protocol - Version 3
portmap      111/tcp  sunrpc         # SUN Remote Procedure Call
portmap      111/udp  sunrpc         #
auth         113/tcp  authentication # Authentication Service
sftp         115/tcp                 # Simple File Transfer Protocol
uucp-path    117/tcp                 # UUCP Path Service
nntp         119/tcp  readnews untp  # Network News Transfer Protocol
ntp          123/udp                 # Network Time Protocol
netbios_ns   137/tcp                 # NetBIOS Name Service
netbios_ns   137/udp                 #
netbios_dgm  138/tcp                 # NetBIOS Datagram Service
netbios_dgm  138/udp                 #
netbios_ssn  139/tcp                 # NetBIOS Session Service
netbios_ssn  139/udp                 #
bftp         152/tcp                 # Background File Transfer Protocol
snmp         161/udp  snmpd          # Simple Network Management Protocol Agent
snmp-trap    162/udp  trapd          # Simple Network Management Protocol Traps
xdmcp        177/tcp                 # X Display Manager Control Protocol
xdmcp        177/udp                 # X Display Manager Control Protocol
bgp          179/tcp                 # Border Gateway Protocol
# PV performance tool services entries
pvserver     382/tcp    # PV server
pvalarm      383/tcp    # PV alarm management
svrloc       427/tcp    # Server Location
svrloc       427/udp    # Server Location
# Ports for IPSec
isakmp       500/tcp  isakmp         # IPSec Key Management (ISAKMP)
isakmp       500/udp  isakmp         # IPSec Key Management (ISAKMP)
#
# UNIX services
#
biff         512/udp  comsat         # mail notification
exec         512/tcp                 # remote execution, passwd required
login        513/tcp                 # remote login
who          513/udp  whod           # remote who and uptime
shell        514/tcp  cmd            # remote command, no passwd used
syslog       514/udp                 # remote system logging
printer      515/tcp  spooler        # remote print spooling
talk         517/udp                 # conversation
ntalk        518/udp                 # new talk, conversation
route        520/udp  router routed  # routing information protocol
efs          520/tcp                 # Extended file name server
timed        525/udp  timeserver     # remote clock synchronization
tempo        526/tcp  newdate        #
courier      530/tcp  rpc            #
conference   531/tcp  chat           #
netnews      532/tcp  readnews       #
netwall      533/udp                 # Emergency broadcasting
uucp         540/tcp  uucpd          # uucp daemon
remotefs     556/tcp  rfs_server rfs # Brunhoff remote filesystem
ingreslock  1524/tcp                 #
#
# Other HP-UX services
#
lansrm       570/udp                 # SRM/UX Server
DAServer     987/tcp                 # SQL distributed access
instl_boots 1067/udp                 # installation bootstrap protocol server
instl_bootc 1068/udp                 # installation bootstrap protocol client
nfsd-keepalive 1110/udp              # Client status info
nfsd-status    1110/tcp              # Cluster status info
msql        1111/tcp                 # Mini SQL database server
rlb         1260/tcp                 # remote loopback diagnostic
clvm-cfg    1476/tcp                 # HA LVM configuration
diagmond    1508/tcp                 # Diagnostic System Manager
nft         1536/tcp                 # NS network file transfer
```

```
sna-cs      1553/tcp                # SNAplus client/server
sna-cs      1553/udp                # SNAplus client/server
ncpm-pm     1591/udp                # NCPM Policy Manager
ncpm-hip    1683/udp                # NCPM Host Information Provider
cvmon       1686/udp                # Clusterview cvmon-cvmap communication
registrar   1712/tcp                # resource monitoring service
registrar   1712/udp                # resource monitoring service
ncpm-ft     1744/udp                # NCPM File Transfer
psmond      1788/tcp                # Predictive Monitor
psmond      1788/udp                # Hardware Predictive Monitor
pmlockd     1889/tcp                # SynerVision locking daemon
pmlockd     1889/udp                #
nfsd        2049/udp                # NFS remote file system
nfsd        2049/tcp                # NFS remote file system
netdist     2106/tcp                # update(1m) network distribution service
cvmmon      2300/tcp                # ClusterView Management cluster support
hpidsadmin  2984/tcp              # HP-UX Host Intrusion Detection System admin
hpidsadmin  2984/udp              # HP-UX Host Intrusion Detection System admin
hpidsagent  2985/tcp              # HP-UX Host Intrusion Detection System agent
hpidsagent  2985/udp              # HP-UX Host Intrusion Detection System agent
hp-clic     3384/tcp                #Cluster Management Services
hp-clic     3384/udp                #Hardware Management
rfa         4672/tcp                # NS remote file access
veesm       4789/tcp                # HP VEE service manager
hacl-hb     5300/tcp                # High Availability (HA) Cluster heartbeat
hacl-gs     5301/tcp                # HA Cluster General Services
hacl-cfg    5302/tcp                # HA Cluster TCP configuration
hacl-cfg    5302/udp                # HA Cluster UDP configuration
hacl-probe  5303/tcp                # HA Cluster TCP probe
hacl-probe  5303/udp                # HA Cluster UDP probe
hacl-local  5304/tcp                # HA Cluster Commands
hacl-test   5305/tcp                # HA Cluster Test
hacl-dlm    5408/tcp                # HA Cluster distributed lock manager
omni        5555/tcp                # HP OpenView OmniBack
lanmgrx.osB 5696/tcp                # LAN Manager/X for B.00.00 OfficeShare
hcserver    5710/tcp                # HP Cooperative Services
wbem-http   5988/tcp                # Web-Based Enterprise Management HTTP
wbem-http   5988/udp                # Web-Based Enterprise Management HTTP
wbem-https  5989/tcp                # Web-Based Enterprise Management HTTPS
wbem-https  5989/udp                # Web-Based Enterprise Management HTTPS
grmd        5999/tcp                # graphics resource manager
spc         6111/tcp                # sub-process control
desmevt     6868/tcp                # DE/ Services Monitor, Event Service
pdclientd   6874/tcp                # Palladium print client daemon
pdeventd    6875/tcp                # Palladium print event daemon
iasqlsvr    7489/tcp                # Information Access
recserv     7815/tcp                # SharedX Receiver Service
p7_c33upd   8545/tcp    #TSD acceSS7 configuration update RPC server
p7_c33      8546/tcp    #TSD acceSS7 configuration RPC server
p7_c32      8547/tcp    #TSD acceSS7 communications status RPC server
p7_c35      8548/tcp    #TSD acceSS7 communications configuration RPC server
p7_g06      8549/tcp    #TSD acceSS7 application version registration RPC server
p7_e30      8550/tcp    #TSD acceSS7 event manager RPC server
comms_normal     8551/tcp    # acceSS7 normal priority messages
comms_high       8552/tcp    # acceSS7 high priority messages
c34_main         8553/udp    # acceSS7 Inter-Server messages
ftp-ftam    8868/tcp                # FTP->FTAM Gateway
mcsemon     9999/tcp                # MC/System Environment monitor
console     10000/tcp               # MC/System Environment console multiplexor
actcp       31766/tcp               # ACT Call Processing Server
SrpSiteDaemon    6178/tcp    # acceSS7 Statistics Remote Site query daemon
SrpCentralDaemon 6179/tcp    # acceSS7 Statistics Central Server query daemon
erdb_svr        35100/tcp    # acceSS7 Statistics Central Database
erdb_bck        35101/tcp    # acceSS7 Statistics Database Backup
hp-sco          19410/tcp    # HP SCO port number
hp-sco          19410/udp    # HP SCO port number
hp-sca          19411/tcp    # HP SCA port number
hp-sca          19411/udp    # HP SCA port number

#
# Kerberos (Project Athena/MIT) services
#
kerberos5    88/udp   kdc             # Kerberos 5 kdc
klogin       543/tcp                # Kerberos rlogin -kfall
kshell       544/tcp  krcmd         # Kerberos remote shell -kfall
ekshell      545/tcp  krcmd         # Kerberos encrypted remote shell -kfall
kerberos     750/udp  kdc           # Kerberos (server) udp -kfall
kerberos     750/tcp  kdc           # Kerberos (server) tcp -kfall
```

```
kerberos_master 751/tcp kadmin       # Kerberos kadmin
krbupdate    760/tcp  kreg           # Kerberos registration -kfall
kpasswd      761/tcp  kpwd           # Kerberos "passwd" -kfall
eklogin     2105/tcp                 # Kerberos encrypted rlogin -kfall
# The X10_LI server for each display listens on ports 5800 + display number.
# The X10_MI server for each display listens on ports 5900 + display number.
# The X11 server for each display listens on ports 6000 + display number.
# The X11 font server listens on port 7000.
# Do NOT associate other services with these ports.
# Refer to the X documentation for details.

hpoms-ci-lstn   5403/tcp  #SAP spooler support
hpoms-dps-lstn  5404/tcp  #SAP spooler support
samd            3275/tcp        # sam daemon

dtspc           6112/tcp        #subprocess control

swat      901/tcp# SAMBA Web-based Admin Tool
#
DB2_db2inst1     60000/tcp
DB2_db2inst1_1   60001/tcp
DB2_db2inst1_2   60002/tcp
DB2_db2inst1_END 60003/tcp
#
xdb2inst1       60004/tcp
#
DB2_db2inst2    60010/tcp
DB2_db2inst2_1  60011/tcp
DB2_db2inst2_2  60012/tcp
DB2_db2inst2_END        60013/tcp
#
xdb2inst2       55555/tcp
```

Appendix 20: Contents of /stand/system on rx4640a

```
*
* Created on Wed Jul  7 10:44:37 2004
*
version 1
configuration nextboot "imported from /stand/system" [40ec0c45]
*
* Module entries
*
module stape static [3F56E2F0]
module swsp static 30103.106.106
module hsx static 30103.106.106
module swspBus static 30103.106.106
module prm best [3F56E2F0]
module mpt best [3F4A8371]
module vols best [3F41B706]
module vol best [3F41B706]
module vxdmp best [3F41B577]
module vxvm best [3F41B706]
module lv best [3F559170]
module lvm best [3F559170]
module vxportal best [3F559170]
module vxfs best [3F559170]
module pfil auto 0.1.0
module igelan best [3F454271]
module iether best [3F4542A0]
module gelan best [3F454178]
module fddi4 best [3F4122D1]
module td best [3F533FD9]
module cifs best [3F465E27]
module pckt best [3F559170]
module ptm best [3F559170]
module pts best [3F559170]
module ptem best [3F559170]
module ldterm best [3F559170]
module ffs best [3F559170]
module pipemod best [3F559170]
module pipedev best [3F559170]
module tirdwr best [3F559170]
module timod best [3F559170]
module sc best [3F559170]
module echo best [3F559170]
module sad best [3F559170]
module strlog best [3F559170]
module clone best [3F559170]
module hpstreams best [3F559170]
module cachefsc best [3F559170]
module autofsc best [3F559170]
module rpcmod best [3F559170]
module nfsm best [3F559170]
module nfs_client best [3F559170]
module nfs_server best [3F559170]
module nfs_core best [3F559170]
module nms best [3F559170]
module netdiag1 best [3F56E2F0]
module token_arp best [3F559170]
module dlpi best [3F559170]
module intl100 best [3F559170]
module btlan best [3F559170]
module tels best [3F559170]
module telm best [3F559170]
module tun best [3F559170]
module uipc best [3F56E2F0]
module inet best [3F559170]
module rng loaded 0.1.0
module cdfs best 0.1.0
module dev_config best [3F56E2F0]
module dmem best [3F56E2F0]
module diag2 best [3F56E2F0]
module c8xx best [3F56E2F0]
module pdh best [3F56E2F0]
module lion_psm best [3F56E2F0]
module ia64_psm best [3F56E2F0]
module wxb_hp best [3F56E2F0]
module sac best [3F56E2F0]
```

```
module acpi_node best [3F56E2F0]
module LCentIf best [3F56E2F0]
module ipmi best [3F56E2F0]
module pty1 best [3F56E2F0]
module pty0 best [3F56E2F0]
module azusa_psm best [3F56E2F0]
module gvid_core best [3F56E2F0]
module sctl best [3F56E2F0]
module sdisk best [3F56E2F0]
module tgt best [3F56E2F0]
module side best [3F56E2F0]
module side_multi best [3F56E2F0]
module ehci best [3F56E2F0]
module hid best [3F56E2F0]
module hub best [3F56E2F0]
module hcd best [3F56E2F0]
module asio0 best [3F56E2F0]
module lba best [3F56E2F0]
module sba best [3F56E2F0]
module root best [3F56E2F0]
module gvid_info loaded 0.1.0
module gvid_him_fgl auto 0.1.0
module gvid_him_rad auto 0.1.0
module drmfgl auto 0.1.0
module drmfglrx auto 0.1.0
*
* Swap entries
*
*
* Dump entries
*
dump lvol
*
* Driver binding entries
*
*
* Tunables entries
*
tunable msgseg 32767
tunable maxuprc 1500
tunable nstrpty 60
tunable nflocks 8192
tunable ninode 5734
tunable msgmnb 65535
tunable msgtql 3277
tunable msgmap 3279
tunable semmns 8193
tunable semmni 4096
tunable semmnu 4096
tunable shmmax 15447262003
tunable shmmni 4096
tunable msgmax 65535
```

Appendix 21: Output of swlist on rx4640a

```
# Initializing...
# Contacting target "rx4640a"...
#
# Target:  rx4640a:/
#

  ACXX                    C.05.50        HP aC++
  Accounting              B.11.23        Accounting
  Asian-Core              B.11.23        Asian Core
  Asian-PRINTER           B.11.23        Asian Printer Support
  Asian-TERM              B.11.23        Asian Terminal Support
  Asian-UTILITY           B.11.23        Asian Utility
  AudioSubsystem B.11.23        HP-UX Audio Subsystem
  Auxiliary-Opt           B.11.23        Auxiliary Optimizer for HP Languages.
  Bastille                B.02.01.02     HP-UX Security Hardening Tool
  C-ANSI-C                C.05.50        HP C/ANSI C Compiler
  C-Dev-Tools             B.11.23        C Language Development Tools
  CDE                     B.11.23        HP-UX CDE User Interface
  CIFS-Client             A.01.09        CIFS Client
  CIFS-Development        A.01.09.04     HP CIFS Server Source Code Files
  CIFS-Server             A.01.09.04     HP CIFS Server (Samba) File and Print Services
  CM-Provider-MOFB.02.02.00     CM Provider and MOF
  COMPLIBS                B.11.23        Compiler Support Librarys
  CPQswsp                 A.3.0C.00F.00F HP StorageWorks Secure Path Device Driver and utilities for
HSx class of Disk Arrays
  Caliper                 B.11.23        HP IA-64 Performance Measurement Tools
  Cluster-MonitorA.11.15.00     HP Cluster Monitor
  Cluster-OM              B.02.02.00     HP Cluster API
  Contrib-Tools           B.11.23.01.21  Contributed Tools
  DB2V81CAE               8.1.0.64       Administration Client for HP-UX
  DB2V81CC                8.1.0.64       Control Center Help (HTML)
  DB2V81CONN              8.1.0.64       Connect Enterprise Edition for HP-UX
  DB2V81JHLP              8.1.0.64       Java Help (HTML)
  DB2V81MSG               8.1.0.64       Product Messages
  DB2V81SDK               8.1.0.64       Application Development Tools for HP-UX
  DB2V81WGRP              8.1.0.64       UDB Workgroup Server Edition for HP-UX
  DB2V81XENT              8.1.0.64       UDB Enterprise Server Edition for HP-UX
  DCE-Core                B.11.23        HP DCE Core Client Software
  DCE-CoreTools           B.11.23        DCE Application development Support
  DiskQuota               B.11.23        DiskQuota
  EMS-Config              A.04.00.01     EMS Config
  EMS-Core                A.04.00.01     EMS Core Product
  EMS-DBMon               A.04.00.01     EMS Database Monitor
  EMS-DskMon              A.04.00.01     EMS Disk Resource Monitor
  EMS-MIBMon              A.04.00.01     EMS MIB Resource Monitor Product
  FC-TACHYON-TL           B.11.23.01     PCI FibreChannel;Supptd HW=A6795A,A5158A
  FCMassStorage           B.11.23        Fibre Channel Mass Storage
  FDDI-PCI                B.11.23.00     HP PCI FDDI Driver
  Fonts                   B.11.23        Fonts to enable layered technologies to Display and Print
  GE-DRV          B.11.23.01     HP PCI Gigabit Ethernet Driver
  GLib                    1.2.10.4       A library of handy utility functions
  GSS-API                 B.11.23        GSS-API Version 1.0
  GTK+                    1.2.10.4       The GIMP Toolkit set of widgets for X
  Gettext                 0.10.39.4      GNU gettext NLS library
  Glance          C.03.71.23     HP GlancePlus/UX
  IETHER-DRV              B.11.23.01     HP PCI 100BT/Gigabit Ethernet Driver
  IGELAN-DRV              B.11.23.01     HP PCI Gigabit Ethernet Driver
  IPF-HP          A.03.05.06     HP IPFilter 3.5alpha5
  Ignite-UX               B.5.0.35       HP-UX System Installation Services
  ImagingSubsystem        B.11.23        HP-UX Image Viewer Subsystem
  IntegratedLoginB.11.23        Integrated Login Product
  International           B.11.23        International
  InternetSrvcs           B.11.23        General network applications and daemons
  JFS                     B.11.23        The Base VxFS File System
  Jdk13                   1.3.1.09.07    Java2 1.3 SDK
  Jdk14                   1.4.1.03.01    Java2 1.4 SDK
  Jpi13                   1.3.1.09.07    Java2 1.3 Netscape Plugin
  Jpi14                   1.4.1.03.01    Java2 1.4 Netscape Plugin
  Jre13                   1.3.1.09.07    Java2 1.3 RTE 1.3
  Jre14                   1.4.1.03.01    Java2 1.4 RTE
  Judy-lib                B.11.11.04.15  Judy Library - development and runtime libraries for handling
dynamic arrays
  KRB5-Client             B.11.23        Kerberos V5 Client Version 1.0
```

```
KWDB                    B.11.23        KWDB
KernDevKit              B.11.23        HP-UX Kernel Developer Kit
KernelConfig            B.11.23        Kernel Configuration
Keyshell                B.11.23        Keyshell
LVM                     B.11.23        LVM
Libiconv                1.6.1.3        Libiconv - Character set conversion library.
MSDOS-Utils             B.11.23        MSDOS-Utils
MailUtilities           B.11.23        User mail agents and related tools
MeasureWare             C.03.71.23     MeasureWare Software/UX
MeasurementInt C.03.71.23     HP-UX Measurement Interface for 11.23
Mozilla                 1.2.1.03.00    Mozilla 1.2
MozillaSrc              1.2.1.03.00    Mozilla 1.2 Source Distribution
NFS                     B.11.23        ONC/NFS; Network-File System,Information Services,Utilities
NParProvider            B.01.00        nPartition Provider
Netscape                B.11.22.06.60  Netscape 7.0
Networking              B.11.23        HP-UX_Lanlink_Product
NonHP-Terminfo B.11.23        Non HP terminfo files
OE                      B.11.23        HP-UX OE control script product
OOANSIC                 B.11.23        HP Object Oriented ANSI C infrastructure
OPS-Provider-MOF        B.02.02.00     OPS Provider and MOF
OS-Core                 B.11.23        Core Operating System, plus Software Terms & Conditions
OVSNMPAgent             B.11.23        HP-UX_SNMP_Agent_Product
OnlineJFS               B.11.23        Online features of the VxFS File System
PAM-Kerberos            A.01.10        PAM-Kerberos Version 1.10
PAM-NTLM                A.01.09        HP NTLM Pluggable Authentication Module
PFIL-HP                 A.01.11.06     HP IPFilter PFIL Interface
PRM-Sw-Krn              C.01.01        Process Resource Manager PRM-Sw-Krn product
PRM-Sw-Lib              C.02.01.01     Process Resource Manager PRM-Sw-Lib product
Package-ManagerA.11.15.00     HP Package-Manager
PartitionManager        B.11.23.01.00  Partition Manager for HP-UX
PeriphDevWeb            B.11.23.00     HP-UX Peripheral Device Tool
Perl5-32                D.5.8.0.A      The 32 Bit Perl Programming Language with Extensions
Perl5-64                D.5.8.0.A      The 64 Bit Perl Programming Language with Extensions
PrinterMgmt             B.11.23        PrinterMgmt
Proc-Resrc-Mgr C.02.01.01     Process Resource Manager Proc-Resrc-Mgr product
ProgSupport             B.11.23        ProgSupport
SG-NFS-Tool             A.11.23.01     MC/Serviceguard NFS Script Templates
SG-Oracle-Tool B.01.10        MC/Serviceguard Oracle Script Templates
SIN                     B.11.23        HP IA-64 static binary instrumentation tool
SOE                     B.11.23        SOE
SW-DIST                 B.11.23        HP-UX Software Distributor
SecPatchChk             B.01.03.01     HP-UX Security Check Tools
Secure_Shell            A.03.10.007    HP-UX Secure Shell
SecurityMon             B.11.23        SecurityMon
SecurityTools           B.01.00.03     The security Tools that Bastille can Configure
Serviceguard            A.11.15.00     Serviceguard
SourceControl           B.11.23        SourceControl
Spelling                B.11.23        Spelling
Streams                 B.11.23        HP-UX Streams Product
Streams-TIO             B.11.23        HP-UX_Streams-TIO_Product
Sup-Tool-Mgr            B.11.23.01.21  Support Tools Manager for HPUX systems
SysMgmtAgent            B.03.00.03     servicecontrol manager Agent Product
SysMgmtServer           B.03.00.03     servicecontrol manager Server Product
SystemAdmin             B.11.23        HP-UX System Administration Tools
SystemComm              B.11.23        System Communication utilities - ct,cu,ptydaemon,vt,kermit
TechPrintServ           B.11.23        HP-UX Technical Image Printing Service
TerminalMngr            B.11.23        TerminalMngr
TextEditors             B.11.23        TextEditors
TextFormatters B.11.23        TextFormatters
USB00                   B.11.23        HP Object Oriented USB Driver
UUCP                    B.11.23        Unix to Unix CoPy
Update-UX               B.11.23        HP-UX Update-UX
Upgrade                 B.11.23        Upgrade
UserLicense             B.11.23        HP-UX User License
VRTSfspro               3.5_2_ga03     VERITAS File System Management Services Provider
VRTSob         3.0.2.IA.273.003 VERITAS Enterprise Administrator Service
VRTSobgui               3.0.2.IA.273.003 VERITAS Enterprise Administrator
VRTSvlic                3.01.IA.001c.003 VERITAS License Utilities
VRTSvmdoc               3.5~IA.004     VERITAS Volume Manager Documentation
VRTSvmpro               3.5~IA.004     VERITAS Volume Manager Management Services Provider
VRTSvxvm                3.5~IA.004     Base VERITAS Volume Manager 3.5 for HP-UX
WBEMServices            A.01.05.05     WBEM Services CORE Product
WDB                     B.11.23        HP Wildebeest (HP WDB) Debugger
WDB-GUI                 B.11.23        GUI for the HP WDB Debugger
WLM-Toolkits            A.01.04.01     HP-UX Workload Manager Toolkits
Workload-Mgr            A.02.01.01     HP-UX Workload Manager
X11                     B.11.23        HP-UX X Window Software
X11MotifDevKit B.11.23.03     HP-UX Developer's Toolkit - X11, Motif, and Imake
```

```
Xserver                 B.11.23         HP-UX Xserver
hpuxwsAPACHE            B.1.0.06.01     HP-UX Apache-based Web Server
hpuxwsTOMCAT            B.1.0.06.01     HP-UX Tomcat-based Servlet Engine
hpuxwsWEBMIN            A.1.0.06.01     HP-UX Webmin-based Admin
hpuxwsXML               A.1.0.06.01     HP-UX XML Web Server Tools
iCOD                    B.06.00         Instant Capacity On Demand
mysql                   3.23.54a.00     MySQL open-source database
scsiU320                B.11.23         PCI SCSI U320; Supptd HW=CoreIO
tusc                    7.5             tusc
```

Appendix 22: Output of vgdisplay on rx4640a

```
--- Volume groups ---
VG Name                     /dev/vg00
VG Write Access             read/write
VG Status                   available
Max LV                      255
Cur LV                      8
Open LV                     8
Max PV                      16
Cur PV                      1
Act PV                      1
Max PE per PV               4328
VGDA                        2
PE Size (Mbytes)            16
Total PE                    4318
Alloc PE                    3168
Free PE                     1150
Total PVG                   0
Total Spare PVs             0
Total Spare PVs in use      0

   --- Logical volumes ---
   LV Name                     /dev/vg00/lvol1
   LV Status                   available/syncd
   LV Size (Mbytes)            512
   Current LE                  32
   Allocated PE                32
   Used PV                     1

   LV Name                     /dev/vg00/lvol2
   LV Status                   available/syncd
   LV Size (Mbytes)            8192
   Current LE                  512
   Allocated PE                512
   Used PV                     1

   LV Name                     /dev/vg00/lvol3
   LV Status                   available/syncd
   LV Size (Mbytes)            512
   Current LE                  32
   Allocated PE                32
   Used PV                     1

   LV Name                     /dev/vg00/lvol4
   LV Status                   available/syncd
   LV Size (Mbytes)            8192
   Current LE                  512
   Allocated PE                512
   Used PV                     1

   LV Name                     /dev/vg00/lvol5
   LV Status                   available/syncd
   LV Size (Mbytes)            8192
   Current LE                  512
   Allocated PE                512
   Used PV                     1

   LV Name                     /dev/vg00/lvol6
   LV Status                   available/syncd
   LV Size (Mbytes)            8192
   Current LE                  512
   Allocated PE                512
   Used PV                     1

   LV Name                     /dev/vg00/lvol7
   LV Status                   available/syncd
   LV Size (Mbytes)            8192
   Current LE                  512
   Allocated PE                512
   Used PV                     1

   LV Name                     /dev/vg00/lvol8
   LV Status                   available/syncd
   LV Size (Mbytes)            8704
```

```
   Current LE                  544
   Allocated PE                544
   Used PV                     1


   --- Physical volumes ---
   PV Name                     /dev/dsk/c2t0d0s2
   PV Status                   available
   Total PE                    4318
   Free PE                     1150
   Autoswitch                  On


VG Name                     /dev/db2inst1_0
VG Write Access             read/write
VG Status                   available, exclusive
Max LV                      255
Cur LV                      1
Open LV                     1
Max PV                      16
Cur PV                      1
Act PV                      1
Max PE per PV               17502
VGDA                        2
PE Size (Mbytes)            4
Total PE                    17499
Alloc PE                    17499
Free PE                     0
Total PVG                   0
Total Spare PVs             0
Total Spare PVs in use      0

   --- Logical volumes ---
   LV Name                     /dev/db2inst1_0/db2inst1_0
   LV Status                   available/syncd
   LV Size (Mbytes)            69996
   Current LE                  17499
   Allocated PE                17499
   Used PV                     1


   --- Physical volumes ---
   PV Name                     /dev/dsk/c7t10d0
   PV Status                   available
   Total PE                    17499
   Free PE                     0
   Autoswitch                  On


VG Name                     /dev/db2inst2_0
VG Write Access             read/write
VG Status                   available, exclusive
Max LV                      255
Cur LV                      1
Open LV                     1
Max PV                      16
Cur PV                      1
Act PV                      1
Max PE per PV               17502
VGDA                        2
PE Size (Mbytes)            4
Total PE                    17499
Alloc PE                    17499
Free PE                     0
Total PVG                   0
Total Spare PVs             0
Total Spare PVs in use      0

   --- Logical volumes ---
   LV Name                     /dev/db2inst2_0/db2inst2_0
   LV Status                   available/syncd
   LV Size (Mbytes)            69996
   Current LE                  17499
   Allocated PE                17499
   Used PV                     1


   --- Physical volumes ---
```

```
PV Name                     /dev/dsk/c7t11d0
PV Status                   available
Total PE                    17499
Free PE                     0
Autoswitch                  On
```

Appendix 23: HP's Partner Technology Access Center High Availability Implementation and Validation Services Data Sheet

Hewlett-Packard Company has a wide range of powerful high availability tools and services to assist ISVs in the validation of their applications in a highly available, mission-critical Hewlett-Packard environment.

HP's High Availability software tool suite, including Serviceguard, is a specialized facility for protecting mission-critical applications from hardware and software failures. With Serviceguard, multiple nodes (systems) are organized into an enterprise cluster that is capable of delivering highly available application services to LAN attached clients.

For ease of management and outstanding flexibility, Serviceguard allows all of the resources needed by an application to be organized into entities called "application packages". Application packages consist of any resource needed to support a specific application service, such as disks, network resources, and application or system processes. Packages are the entities that are managed and moved within the enterprise cluster.

When an ISV delivers an application that will be run in a mission critical environment, it is important to certify that the application has been configured and tested in an Serviceguard environment.

To aid ISVs in certifying their applications in a highly available environment, the HP Partner Technology Access Center (PTAC) provides hardware and consulting services. The following are the types of services provided:

- Analysis of Application Environment:
  - system resources used by the application
  - application design and number of packages
  - use of raw, HFS or JFS volumes
  - application recovery methods
  - data loss specifications
  - checkpointing or buffer flushing frequency
  - shared versus replicated file systems for code and/or data
  - use of memory-based data
  - capacity requirements
  - issues affecting failover time
- Cluster configuration of PTAC hardware to match your application needs
- Define application packages and resources required
- Create application package control scripts:
  - application startup & shutdown
  - application monitoring
  - application error handling
  - application restart options
  - application IP addresses
  - service names
  - volume group handling
  - data recovery procedures
- Create, verify and execute a test plan which will exercise defined failure scenarios
- Demonstrate the functionality of the highly available cluster

The HP Partner Technology Access Center maintains the following hardware cluster, available for High Availability application validation:

- (4) RX4640 4-way SMP processors

- 8GB RAM per machine
- 6 LAN interfaces per machine
- 200GB Usable Disk Space

Contact the Partner Technology Access Center for more information about these and additional services through your local HP sales team.

Appendix 24: HP's Partner Technology Access Center High Availability Implementation and Validation Services Process and Methodology

To aid Independent Software Vendors (ISVs) in certifying their applications in a highly available environment, the Hewlett-Packard (HP) Partner Technology Access Center (PTAC) provides hardware, HA training, and consulting services. The following are the types of services provided:

- Analysis of Application Environment
- Cluster configuration of PTAC hardware to match application needs
- Define application packages and resources required
- Create application package control scripts
- Create, verify and execute a test plan which will exercise defined failure scenarios
- Demonstrate the functionality of the highly available cluster

To perform this service, and to offer it as a repeatable deliverable, the PTAC has defined a process and methodology. The completion of this process will result in an ISVs application being validated by HP as being able to perform in a highly available environment.

An ISVs customer either:

- View the ISVs application as mission-critical
- Mandates that the application must be run in a highly available state

However, a typical ISV has limited experience with architecting and testing HA solutions. Additionally, hardware for failover testing is often unavailable.

Initial Activities

- Contact PTAC administration via your local HP sales team and request the HA Validation Services Package.
- Read and share the package with your ISV.
- A knowledgeable engineer from the ISV must be present at the PTAC lab during the entire HA validation.
- If the ISV has an assigned HP TC, that TC may optionally be present for the HA validation.
- Call PTAC administration and schedule PTAC HA Lab time.

Homework

The HA Validation Services Package contains:

- Serviceguard (MC/SG) manual
- "Designing Highly Available Cluster Applications" white paper
- MS/SG product brief
- Example validation write-up
- Example cluster and package configuration scripts.

The goal of this step is that the ISV should understand HP's HA product family and understand how to architect his application for HA validation.

Planning

- What should the cluster look like during normal operations?
- What is the standard configuration of most customers?
- Can application modules be spread across multiple systems? Is this normal?
- Do all pieces of the application failover together to the failover machine?
- Can applications running on different machines failover to a shared failover machine?
- Is there any HA mechanism already built in to the app?

- What are the customers expectations of HP's HA product suite?

Technical Evaluation

- Evaluate the app per HA design rules
- Discuss each rule with the application engineers
- What does the app do today to handle a system panic or reboot?
- Does the app use any system specific calls (e.g. uname, gethostbyname, SPU_ID, etc.)?

The deliverable of this step is a write-up of any issues.

App Setup Without MC/SG

- Setup the system without Serviceguard
- Install the app on the primary system
- Install all shared data on separate external volume groups
- Use JFS file systems as appropriate
- Test the app on the primary system
- Perform a "standard" ISV-provided test to ensure the app is running correctly
- If possible, connect to the app through a client
- Crash the primary system, reboot it, and test how the app starts
- Document any manual procedures
- Can everything start from rc scripts?
- Write a script which brings up the app and all required services

The goal of this step is to ensure that the app can automatically be started and shutdown.

The deliverable of this step is the tasks or scripts which start the app automatically.

No MC/SG, 2 Systems

Try to failover the app to the failover system by hand:

- Connect the volume group to the second system, vgimport, create mount points, etc.
- Document what has to be created on the failover system for the validation whitepaper
- With the app NOT running on the primary system, try to bring it up on the failover system
- Repeat this process until the app will run on the failover system

The goal of this step is to ensure that the failover can occur manually.

Hands On with MC/SG

Configure the MC/SG Cluster:

- Cluster configuration
- Create package(s)
- Create package script
- Compile package configuration scripts and distribute
- Use these scripts as the "customer_defined" functions in the package control scripts

The deliverable of this step is the cluster and package scripts.

Testing

Testing should be performed with a client connected and under system load. This is to test how well and how quickly the application recovers from a failover when a large amount of "work" is queued.

- Halt the package on the primary system and move it to the failover system.
- Move the package back to the primary system.

- Fail one of the systems (e.g. power off, kill monitored process, LAN disconnect)
- Ensure that the package starts on the failover system
- Repeat failover from the failover system back to the primary
- Be sure to test all combinations of app load during testing
- Repeat the failover process under different application states (i.e. heavy user load, no user load, batch jobs, online transactions)
- Keep timing records of how long it takes to completely failover the app
- The customer of the ISV will want to know the failover timing as part of the validation process

Application Monitoring

- MC/SG can monitor the health of processes which are critical to the correct running of the app
- Or, a custom monitor script can be written to monitor specific ISV processes
- Monitor script can be written now, or be written at each customer site

Support and Write-up

- The ISV will own the MC/SG scripts, but the PTAC will keep copies for our records
- Determine whether the ISV will want to come back to test new application releases
- HP supports MC/SG, and the ISV supports the concept of failover with its application

The PTAC HA engineer will work with the ISV to produce the following deliverables. A copy of these will be placed on the HP Advanced Technology Center (ATC) High Availability web page:

- Whitepaper with technical details of the failover, known issues and recommended configurations
- Package control script
- Package configuration file (ASCII)
- Press release on the integration

Example Timetable

To understand the progression of the HA validation process, here is a typical ISV validation scenario:

- Day 1: ISVs HP contact calls PTAC information line
- Day 2: HA Validation Services information package is e-mailed to HP contact
- Day 3: HP contact reviews documentation, and provides to ISV
- Day 4: ISV reviews documentation to understand what must be done prior to coming to the PTAC HA Lab
- Day 5: ISV reports to HP contact when they will be ready to begin validation process
- Day 5: HP contact schedules PTAC HA Lab and engineering time
- Day 6-10: ISV performs necessary homework in preparation for validation process. ISV gathers all non-HP-UX materials that will be needed for the validation. If so desired, ISV prepares a client machine for delivery to the PTAC HA Lab.
- Day 11: ISV travels to PTAC HA Lab in Paramus, NJ
- Day 12: ISV is given a half day of training on HPs HA product suite
- Day 12: ISV and PTAC engineer begin installation of ISV application on PTAC hardware
- Day 13-Day 17: ISV and PTAC engineer follow the HA validation process noted earlier in this document

Notes:

- This is an example scenario. Timing and process progression will be different for each ISV.
- If the ISV wishes to test a client connection to his application, the ISV must supply the client machine. The PTAC lab will provide PC display monitors and PC keyboards.

- The ISV must provide all non-HP-UX material on 4mm DAT tape or CD-ROM. The PTAC HA Lab servers are on a private subnet, and cannot contact machines outside of the PTAC HA Lab.