Oracle® Communications Convergent Charging Controller

High Availability Operations Guide for Solaris

Release 12.0.0

December 2017

Copyright

Contents

About This Document

Scope

This guide provides an overview of Oracle Communications Convergent Charging Controller. It also introduces the general concepts of Convergent Charging Controller using Oracle Clusterware. This document is not intended as a detailed configuration guide and is not certified on any specific version of Oracle Cluster Server.

Audience

This guide is intended for system administrators and system integrators who have some experience with implementing high-availability services and have an understanding of Convergent Charging Controller.

Related documents

For more information, see the following document sets:

- Oracle Communications Convergent Charging Controller:
    - *Oracle Communications Convergent Charging Controller Release Notes*
    - *Oracle Communications Convergent Charging Controller Installation Guide*
- Oracle Database:
    - *Oracle Database High Availability Overview 12c Release 1*
    - *Oracle Database High Availability Best Practices 12c Release 1*
- Oracle Clusterware:
    - *Oracle Solaris Cluster 4.3 Software Installation Guide*
    - *Oracle Solaris 11 System Administration Guide: IP Services*
    - *Oracle Solaris Cluster Data Service for Oracle Solaris Zones Guide*

Document Conventions

Typographical Conventions

The following terms and typographical conventions are used in the Oracle Communications Convergent Charging Controller documentation.

Formatting Convention	Type of Information
Special Bold	Items you must select, such as names of tabs. Names of database tables and fields.
Italics	Name of a document, chapter, topic or other publication. Emphasis within text.
Button	The name of a button to click or a key to press. **Example:** To close the window, either click **Close**, or press **Esc**.
Key+Key	Key combinations for which the user must press and hold down one key and then press another. Example: **Ctrl+P** or **Alt+F4**.
`Monospace`	Examples of code or standard output.
`Monospace Bold`	Text that you must enter.
variable	Used to indicate variables or text that should be replaced with an actual value.
menu option > menu option >	Used to indicate the cascading menu option to be selected. Example: **Operator Functions > Report Functions**
hypertext link	Used to indicate a hypertext link.

Specialized terms and acronyms are defined in the glossary at the end of this guide.

Chapter 1

System Overview

Overview

Introduction

This chapter provides a high-level overview of Convergent Charging Controller high availability (HA). It explains the basic functionality of the system and lists the main components.

In this chapter

This chapter contains the following topics.

High Availability Overview

Introduction

An HA environment should have minimal or no downtime caused by unplanned outages. Outages can be caused by disk drive failures, network failures, system processing unit (SPU) failures, improper system configuration, and application software failures due to application errors or temporarily unavailable system resources.

Additionally, an HA environment should minimize downtime required for planned system and application maintenance and upgrades. Routine system and application upgrades (such as installing kernel or application patch, or new applications) should occur without taking the critical application services off line.

Oracle highly recommends that client applications be configured so that they detect connection problems and automatically attempt to reconnect when a connection is lost.

Key HA features

Convergent Charging Controller can remain available in various failure conditions. Convergent Charging Controller in an HA environment has the following key features:

- Distributed multiprocess, multi-node, multi-system, and multi-site deployment with application resiliency and fault tolerance
- Application service HA with automatic process recycling and failover
- Hardware HA through redundancy and configuration

Disaster recovery

Disaster recovery requires that you set up a remote instance of Convergent Charging Controller that can be activated in the event of a catastrophic failure at the production site. An HA system for Convergent Charging Controller, consisting of multiple clustered servers, is usually limited by the length of the cables connecting the shared data disk devices and the network interfaces. A remote disaster recovery site that is geographically dispersed requires access to the same resources as the production site, including:

- Network connectivity to clients
- Hardware
- Up-to-date Convergent Charging Controller configuration data
- Dynamic provisioning data

An HA environment requires regular system backups and data replication mechanisms. Data backup must be implemented independent of Convergent Charging Controller.

Hardware requirements for HA

You achieve hardware availability by using redundant backup components for each subsystem that may fail:

- Mirrored dual-port data disks to protect the application from loss of critical data
- Redundant network interfaces and networks to ensure that application clients can connect to the network
- Redundant SPUs to guard against entire system failures

Redundancy by Node Setup

Hardware redundancy on its own does not guarantee the HA of application services offered by Convergent Charging Controller. It is achieved by ensuring that all software components included in the entire solution are built and configured for fault tolerance. When you set up an HA environment, you must eliminate single points of failure that prevent Convergent Charging Controller from processing orders for an extended period of time.

Each Convergent Charging Controller node type has a different redundant architecture to ensure continuous service availability beyond the availability of each hardware element.

Redundancy by SMS Node

Important: HA for Convergent Charging Controller is supported only on Solaris.

The default installation of SMS nodes does not provide HA. The redundant SMS node setup includes the following deployment types:

- **Small deployments:** In small deployments, SMS is deployed as a single node, where SMS is not redundant and its service availability is based on the availability of the underlying server. In such a setup, operational integrity is maintained through the use of a secure backup mechanism. When the SMS node is offline, the network routing offered by the SLC and VWS nodes i.e. the end subscriber handset based services, will be unaffected.
- **Highly available deployment:** More typically, the SMS is deployed in HA setup, which has two separate servers, each able to host the SMS node arranged in an active/failover topology. To achieve this, the SMS is installed in a Solaris Container located on a common disk array, with Oracle Solaris Cluster managing the container as a clustered resource. The disk array hosts a common application file system and storage for a single database instance. Importantly, failover service offered by Solaris Cluster ensures that the SMS instance is active and that the file-system and database are accessed by only a single server. For this reason, the file system will be single user, usually the

native Solaris ZFS file system. With this configuration, a planned or unplanned shutdown of the active SMS instance results in the failover instance starting up. The failover time is mostly taken up by the time taken to start the failover container and to start the Oracle Database Management System (DBMS) for various SMS services.

These options are based around a single site. The clustered option is constrained in their inter-server distance by the physical connections between the servers and the common array. To overcome this constraint, introduce Oracle Data Guard in one of two configurations:

- Provide an additional SMS disaster recovery option at a second site. The SMS disaster recovery option receives database transaction updates directly from the primary node through Oracle Data Guard, which maintains the SMS disaster recovery database up to date with the primary SMS in near real time.

  Note: Activation of the disaster recovery site is typically undertaken following a total and catastrophic loss of the primary site. Activation would typically take 30 to 60 minutes, depending upon the number of connected nodes and the familiarity of the operations staff with the necessary procedures.
- Provide a geographically redundant SMS. The geographically redundant SMS requires separate zones for the SMS database and the SMS application on each server. The SMS database zones host independent SMS database which operate in much the same way as the SMS disaster recovery, with Oracle Data Guard maintaining the standby database up to date in near real time.

  The SMS application zone operates in a similar manner to the HA cluster, except that there is no shared file system. Instead, a copy of the SMS application zone is ‘seeded’ on the standby server. Failover is still managed by Oracle Cluster to ensure that only one instance is active at a time. Either periodically, or following a maintenance activity, there is an operational requirement to refresh the standby image. This is achieved using standard ZFS capabilities.

Failover of the SMS application will typically take 30 to 60 seconds. Failover of the SMS database will typically take between a few tens of seconds and a few minutes depending upon the transaction load on the database.

Redundancy by SLC Node

The SLC hosts the service logic and network interfaces and integrates with an external online charging system (OCS) for rating and charging services. The SLCs rely on the connected network elements to manage the distribution of traffic between nodes. These might be on a load-share or active/standby basis to one or several nodes.

Some service providers dedicate particular groups of SLC nodes to specific traffic types or to some other grouping. Other service providers configure all SLCs to handle all types of traffic. If subscribers are provisioned into Convergent Charging Controller, all SLCs will host all provisioned subscribers.

Transactions started on one SLC node continue to be serviced by that node, that is the transaction data remains local to each SLC node and is not shared between the SLC nodes. In the event of either a planned or unplanned outage of an SLC node, all active transactions on that node are lost. New transactions would then be targeted to one of the other available SLC nodes.

Planned outages (for example, maintenance activities) are typically scheduled during quiet traffic periods. In this situation, it is normal for the network operator to reduce traffic for the selected SLC node so that all the new transactions target other SLC nodes. By allowing the existing transactions to complete on the chosen SLC node, maintenance activity can be undertaken with minimum service interruption.

Voice calls and data sessions have periodic commits, which further minimize the opportunity for revenue loss from a planned or unplanned outage of an SLC node. This is achieved since the revenue loss is limited to the amounts reserved but not committed which, through configuration, will be only part of a session and limited to the most recent reservation chunk within those sessions that remain active.

Geographic redundancy of the SLC nodes is achieved by locating SLC nodes on different sites. The total number of required SLC nodes depends on the number of nodes for the required traffic level, the number of sites, and whether complete site failure in the busy hour is a required scenario, or a maintenance outage of a single site.

The worst case would typically be a dual site setup, where long-term catastrophic failure of one site is a required scenario. In this case, each site would need N+1 nodes, requiring a total of 2(N+1) SLC nodes.

Redundancy by VWS Node

The VWS is exclusively deployed in a 2N mated pair architecture, where one node is active and the second node is a hot standby. Each node has its own separate database, with transaction data copied from the active to the standby at the application layer, such that the active service transactions can be started on one node, and in case a failover occurs, the service transactions can complete on the second node.

The client systems of the VWS are the SMS and SLC nodes. A mated pair of VWS nodes forms a logical construct termed a ‘Domain’. Each VWS Domain hosts one or more voucher batches. Through data mastered on the SMS and replicated to the other nodes, the target VWS Domain can be identified. Each node then maintains connections to both VWS nodes within each domain and exclusively uses the connection to the active node within the required domain. The two nodes within a domain are designated the primary and secondary nodes. If the primary node is available, it will be the active node. When the primary node fails, the secondary node becomes the active node. If the primary node becomes unavailable, due to either planned or unplanned outages, the transactions initiated on one node are continued on the previously standby node. As such, failover between the VWS nodes within a domain is seamless and happens in real time.

When the primary VWS node returns to service, it initially needs to catch up with the active secondary node. It first processes the incoming synchronization files before notifying the client systems that it is now active. After it becomes active, it continues to process any in-bound synchronization files. Geographic redundancy of the VWS nodes is achieved by locating the two nodes in any given domain, on separate sites.

Redundancy by Traffic/Service Type

To assess the impact of the loss of any single node, to look at the impact with respect to each traffic or service type, such as:

- Customer care operations that come to the platform through these interfaces:
  - SMS screens
  - PI on the SMS
  - OSD on the SLC nodes
- Customer care operations can come to the platform through these interfaces:
  - WEB 2.0 through the PI to the SMS
  - WEB 2.0 through OSD to the SLC(s)
  - USSD request to the SLC(s)
  - SMS text to the SLC(s)
  - IVR session managed through the SLC(s)
- Session-based traffic services that are categorized by having a back and forth message exchange between the serving network element and the Convergent Charging Controller that is the controlling network element. For each node:
  - Failure of an SMS node will have negligible impact to an active session or a new. If there is no subscriber data, no network-side updates occur.
  - Failure of the primary VWS node has no affect on active or new sessions, that is any transactions initiated on the primary, but not completed, will complete on the secondary.

  - Where the SLC node serving the session is lost, the connection to the adjacent network element will drop. For established bearer sessions, the controlling network element may hold the session up until either it is terminated by one of the parties involved and/or until the controlling element requires further direction from Convergent Charging Controller. This would typically be an additional reservation of funds to continue the session. At this point, for voice services, the session failure would be recognized and the bearer session dropped. For data services, the serving node may maintain the bearer session and attempt a new transaction to one of the other SLCs. The sessions that were being set up at the time of failure may either fail back through the network, or be re-attempted to one of the other SLC nodes, as determined by the serving network element. The result on the end subscriber is that the session attempt might fail or an established session might be dropped.
- Event-based traffic services are categorized by having a request-response transaction, that is the response concludes the message exchange between the serving network element and the Convergent Charging Controller. Failure of the serving SLC node will result in loss of connection to the serving network element and failure of that request, following a short timeout. In that situation, serving network elements will typically re-attempt the transaction to one of the other serving SLC nodes, that is the end subscriber does not perceive any issue.

Chapter 2

SMS Node Configuration for High Availability

Overview

Introduction

This chapter provides information about configuring the cluster software on an SMS node.

In this chapter

This chapter contains the following topics.

Configuring SMS Node Cluster

Introduction

This topic describes the SMS cluster node configuration.

Oracle Clusterware Configuration

When you configure Oracle Clusterware for Solaris, you specify the cluster nodes, the service groups, and the service group composition. The configuration contains the following aspects.

Refer to the Oracle Clusterware documents listed in Related Documents for details.

Aspect	Description
Include clauses	Used to include standard and custom resource type definitions. The resource type definitions are from other files residing in the current cluster configuration directory.
Cluster definition	Contains attributes that apply to the entire cluster.
System definition	Defines the cluster node and attributes applicable to the defined node. Keyword system followed by the system name.
Service group definitions	Specifies the systems configured to run the services defined in the service group and the list of systems where services are started automatically when the cluster server starts up.
Resource definitions	Defines resource type its resource name and all required attributes expected by the resource agent.
Resource dependencies	Used to control the startup and shutdown sequence of resources that belong to the same service group.

<table>
<tr><th>Aspect</th><th>Description</th></tr>
<tr><td>Group dependencies</td><td>Uses the following keywords to define the dependency between service groups:
<table>
<tr><th>Keyword</th><th>Indicates that the parent group must be...</th></tr>
<tr><td>online global</td><td>online anywhere in the cluster before the child can go online.</td></tr>
<tr><td>online local</td><td>online on the same node before the child can go online.</td></tr>
<tr><td>online remote</td><td>online on a different system in the cluster before the child can go online.</td></tr>
<tr><td>offline local</td><td>offline on the same node before the child can go online.</td></tr>
</table>
</td></tr>
</table>

About Configuring Active and Passive SMS Nodes

Refer to the following guides to find more general information about active/passive setup:

- *Oracle Database High Availability Overview 12c Release 1 (12.1)*
- *Oracle Database High Availability Best Practices 12c Release 1 (12.1)*
- *Oracle Data Guard Concepts and Administration 12c Release 1 (12.1)*
- *Oracle Solaris Cluster 4.3 Software Installation Guide*
- *Oracle Clusterware Administration and Deployment Guide 12c Release 1 (12.1)*
- *Convergent Charging Controller Installation Guide*

Synchronizing System Time

The local system time must be consistent on all cluster nodes. You can properly synchronize the time by enabling the Network Time Protocol (NTP) daemon on each node in the cluster. The NTP daemon configuration is in the **/etc/inet/ntp.conf** file.

You can use the default configuration of the cluster NTP file, found in **/etc/inet/ntp.conf.cluster**.

For more information, see the discussion about how to configure NTP in *Oracle Solaris 4.3 Concepts Guide*.

Connection Configuration

Introduction

You perform the following tasks to configure the Oracle Clusterware inter-node communication services:

- Configure the Oracle Clusterware
- Register interconnects and ports

Configuring the Oracle Clusterware

During the configuration of the Oracle Cluster, when you run the configuration script (**/usr/cluster/bin/scinstall**), the configuration script probes the interconnects between the nodes. If it cannot detect the interconnections or cannot detect the other node over the interconnect due to a problem, you will be prompted with an error message, and the script is paused. After you have corrected the problem, you can continue to run the script and this time it properly detects the interconnects, as shown in the following example:

```
You can either attempt to correct the problem and try the probes again
or manually configure the transport. To correct the problem might
involve re-cabling, changing the configuration, or fixing hardware.
You must configure the transport manually to configure tagged VLAN
adapters and non tagged VLAN adapters on the same private interconnect
VLAN.

Do you want to try again (yes/no) [yes]? yes

The following connections were discovered:
   sms01:ce1  switch1  sms02:ce1
   sms01:ce2  switch2  sms02:ce2
Completed discovery of the cluster transport configuration
```

Registering Interconnects and Ports

This procedure shows an example of how to manually add four more cluster interconnects, by registering each VLAN on the switch and its ports.

Step	Description

1	Register the switches, as root, by typing:
	`/usr/cluster/bin/clinterconnect add switch3`
	`/usr/cluster/bin/clinterconnect add switch4`
	`/usr/cluster/bin/clinterconnect add switch5`
	`/usr/cluster/bin/clinterconnect add switch6`

2	Register the ports, as root, by typing:
	`/usr/cluster/bin/clinterconnect  add sms01:ce3,switch3`
	`/usr/cluster/bin/clinterconnect  add sms02:ce3,switch3`
	`/usr/cluster/bin/clinterconnect  add sms01:ce5,switch4`
	`/usr/cluster/bin/clinterconnect  add sms02:ce5,switch4`
	`/usr/cluster/bin/clinterconnect  add sms01:ce6,switch5`
	`/usr/cluster/bin/clinterconnect  add sms02:ce6,switch5`
	`/usr/cluster/bin/clinterconnect  add sms01:ce7,switch6`
	`/usr/cluster/bin/clinterconnect  add sms02:ce7,switch6`

3	Check the status of the cluster transport paths, by typing: `scstat –W` **Result:** The configuration is displayed.

```
Transport path:   sms02:ce3           sms01:ce3           Path online
Transport path:   sms02:ce2           sms01:ce2           Path online
Transport path:   sms02:ce1           sms01:ce1           Path online
Transport path:   sms02:ce5           sms01:ce5           Path online
Transport path:   sms02:ce7           sms01:ce7           Path online
Transport path:   sms02:ce6           sms01:ce6           Path online
```

Public Network Configuration

Introduction

Oracle Solaris includes standard agents that you can use to manage dynamic IP addresses required by the service groups. There are two ways to manage dynamic IP addresses. You can use:

- NIC and IP: To control a single public interface card
- IP multipathing agents: To control a redundant backup set of network interface cards

Configuring a Single NIC/IP Pair

For Solaris, you configure NIC and IP pairs in two different configuration files: **/etc/inet/netmasks** and **/etc/inet/hosts.**

Step	Action

1	Add IP address and hostname, by typing:
	`vi /etc/hosts`
	`192.168.46.41     sms01`

2	Add the network mask of the added hosts, by typing:
	`vi /etc/netmasks`
	`192.168.46.0     255.255.255.0`

Configuring IP Multipathing Pair

Ensure each network port has a unique MAC address, as shown in the following example.

Step	Action
1	Type: `# eeprom “local-mac-address?=true”`

2	On Node 1, type:
	`# vi /etc/hostname.ce0`
	and insert the following in the file:
	`sms01-mgmt-ce0 netmask + broadcast + group \`
	`mgmt deprecated -failover up addif sms01 netmask + broadcast + \`
	`failover up`

3	Type:
	`# vi /etc/hostname.ce9`
	and insert the following in the file:
	`sms01-mgmt-ce9 netmask + broadcast + group \`
	`mgmt deprecated -failover standby up`

4	On Node 2, type:
	`# vi /etc/hostname.ce0`
	and insert the following in the file:
	`sms02-mgmt-ce0 netmask + broadcast + group \`
	`mgmt deprecated -failover up addif sms02 netmask + broadcast + \`
	`failover up`

5	Type:
	`# vi /etc/hostname.ce9`
	and insert the following in the file:
	`sms02-mgmt-ce9 netmask + broadcast + group \`
	`mgmt deprecated -failover standby up`

Installing a Geo-Redundant SMS Node

This section provides information about a sample deployment of a single SMS node with a highly available standby node.

With high bandwidth, low latency, and dedicated connections, you can extend the distance between nodes within an SMS cluster. This is possible because of the absence of shared storage between the nodes.

Preparing the Installation

Ensure the following tasks are completed:

1. Two identically installed Solaris nodes: the primary SMS node and the secondary SMS node.
2. One standard single SMS node is configured and integrated with all the Convergent Charging Controller nodes.
3. SMS nodes are on the same VLAN/subnet.
4. To allow a cluster to be formed between both nodes without using shared storage, an additional and independent node accessible by both the primary and secondary SMS nodes, must be available to function as a quorum node replacing the quorum device on shared storage.

Installation Process Overview

You install the georedundant SMS node by performing the following tasks:

1. Create the primary SMS database node as a clone of the standard SMS node. See *Cloning the Standard SMS Node to the Primary SMS Database* (on page 11).
2. Install and configure the global cluster on the standard SMS node. See *Installing and Configuring the Global Cluster* (on page 13).
3. Configure for Oracle Data Guard. See *Configuring for Data Guard* (on page 16).
4. Clone the primary SMS database node to the secondary SMS database node. See *Cloning the Primary SMS Database to the Secondary SMS Database* (on page 20).
5. Complete the Data Guard configuration.See *Completing the Dataguard Configuration* (on page 22).
6. Switch the database nodes. See *Switching the Database Modes* (on page 24).
7. Configure the standard SMS node to Database Connections. See *Configuring the Standard SMS Node Database Connections* (on page 25).
8. Configure the SMS database failover from the primary SMS node to the secondary SMS node. See *Configuring the Failover Standard SMS Database* (on page 29).

Cloning the Standard SMS Node to the Primary SMS Database

Follow these steps to clone the standard SMS node to the primary SMS database node.

Step	Action
1	Prepare the standard SMS node by doing the following: a. Log in as the root user. b. Open the **/etc/inittab** file in a text editor: c. Comment out all the entries for the Convergent Charging Controller applications. d. Add the following entry: **`init q`** **`cd /var/spool/cron/crontabs`** e. Save and close the file. f. Open the **cd /var/spool/cron/crontabs** file in a text editor. g. Comment out all the entries for the Convergent Charging Controller applications. h. Save and close the file. i. Run the following command which disables the Oracle database startup on reboot: **`mv /etc/rc3.d/S99oracle /etc/rc3.d/noS99oracle`**
2	Clone the standard SMS node to the primary SMS database by doing the following: a. Log in as the root user. b. Run the following command, which shuts down the standard SMS node: **`zoneadm -z`** *`smsapp`* **`shutdown`** where *`smsapp`* is the standard SMS node.

Step **Action**

c. Run the following command, which copies SMS configuration to the ZFS filesystems:

```
zonecfg -z smsapp export > /tmp/smsapp.zonecfg
```

d. Run the following command, which copy the ZFS filesystems:

```
zone snapshot -r sms_zone@clone
zfs send -rc sms_zone/smsapp@clone | zfs receive
sms_zone/smsdb_1
zfs send -rc sms_zone/smsapp_Parent@clone | zfs receive
sms_zone/smsdb_1_Parent
zfs set mountpoint=/zones/smsdb_1 sms_zone/smsdb_1
zfs mount sms_zone/smsdb_1
```

where *smsdb_1* is the primary SMS database node.

e. Open the **/tmp/smsapp.zonecfg** file in a text editor.

f. Update the zonepath and dataset entries; for example:

```
set zonepath=/zones/smsdb_1
set name=sms_zone/smsdb_1_Parent
```

g. Save and close the file.

h. Run the following commands, which configure the zone:

```
zonecfg -z smsdb_1 -f /tmp/smsapp.zonecfg
zoneadm -z smsdb_1 attach
zoneadm -z smsdb_1 boot
zlogin -C smsdb_1
```

i. Run the following commands, which reset the system configuration:

```
sysconfig unconfigure
sysconfig configure
reboot
```

j. Configure the primary SMS database node; for example:

```
Computer name: smsdb_1
Do not configure DNS
Name service : None
Timezone: UTC/GMT
root password: password
```

where *password* is the password of the root user.

k. Run the following commands, which configure the IP address:

```
ipadm create-ip adm1
ipadm create-addr -T static -a local=IP_address/21
adm1/v4static
ipadm create-ip rep1
ipadm create-addr -T static -a local=IP_address/24
rep1/v4static
ipadm create-ip bill1
ipadm create-addr -T static -a local=IP_address/24
bill1/v4static
route -p add default IP_address
```

where *IP_address* is the IP address of the system

l. Run the following command, which enables the Oracle database startup on restart:

```
mv /etc/rc3.d/S99oracle /etc/rc3.d/noS99oracle
```

Installing and Configuring the Global Cluster

Follow these steps to install and configure the global cluster:

Step **Action**

1 Log in to the primary node as the root user.

2 Run the following commands, which restore the external access to remote procedure call communication:

```
svccfg
svc:> select network/rpc/bind
svc:/network/rpc/bind> setprop config/local_only=false
svc:/network/rpc/bind> quit
svcadm refresh network/rpc/bind:default
svcprop network/rpc/bind:default | grep local_only
```

Expected output:

```
# config/local_only boolean false
```

3 Log in to the secondary SMS node as the root user and repeat step 2.

4 Add HA cluster publisher to the primary SMS node by doing the following:

a. Log in to the primary SMS node as the root user.

b. Run the following command, which verifies the publisher:

```
pkg publisher
```

Expected output:

```
# PUBLISHER                    TYPE     STATUS P LOCATION
#
# solaris                      origin   online F
http://ipkg.us.oracle.com/solaris11/support/
```

c. Run the following command, which adds the publisher:

```
pkg set-publisher -g http://ipkg.us.oracle.com/ha-
cluster/support/ ha-cluster
```

5 Add HA cluster publisher to the secondary SMS node by doing the following:

a. Log in to the secondary SMS node as the root user.

b. Run the following command, which verifies the publisher:

```
pkg publisher
```

Expected output:

```
# PUBLISHER                    TYPE     STATUS P LOCATION
#
# solaris                      origin   online F
http://ipkg.us.oracle.com/solaris11/support/
```

c. Run the following command, which adds the publisher:

```
pkg set-publisher -g http://ipkg.us.oracle.com/ha-
cluster/support/ ha-cluster
```

Step **Action**

6 Add HA cluster publisher to the cluster server by doing the following:

a. Log in to the cluster server as the root user.

b. Run the following command, which verifies the publisher:

```
pkg publisher
```

Expected output:

```
# PUBLISHER                  TYPE     STATUS P LOCATION
#
# solaris                    origin   online F
http://ipkg.us.oracle.com/solaris11/support/
```

c. Run the following command, which adds the publisher:

```
pkg set-publisher -g http://ipkg.us.oracle.com/ha-
cluster/support/ ha-cluster
```

7 Install the cluster server by doing the following.

a. Log in as the root user.

b. Run the following command, which installs the cluster server:

```
pkg install ha-cluster-quorum-server-full
```

c. Run the following command, which starts the cluster server:

```
/usr/cluster/bin/clquorumserver start 9000
```

8 Install the Oracle Solaris Cluster 4.3 software by doing the following:

a. Log in to the primary SMS database node as the root user.

b. Run the following command which install the software:

```
pkg install ha-cluster-full
```

c. Log in to the secondary SMS database node as the root user.

d. Run the following command which install the software:

```
pkg install ha-cluster-full
```

9 Enable the common agent container by doing the following:

a. Log in to the primary SMS node as the root user.

b. Run the following command, which enables the common agent container:

```
/usr/sbin/cacaoadm enable
```

c. Log in to the secondary SMS node as the root user.

d. Run the following command, which enables the common agent container:

```
/usr/sbin/cacaoadm enable
```

10 Create the cluster node by doing the following:

a. Log in to the primary SMS node as the root user.

b. Run the following command, which creates the cluster:

```
/usr/cluster/bin/scinsstall
```

c. Follow the instructions displayed.

d. Run the following command, which verifies the status:

```
/usr/cluster/bin/clnode status
```

Expected output:

```
# ipwaa50                 online
```

e. Run the following command, which sets up the cluster node:

```
/usr/cluster/bin/clsetup
```

f. Follow the instructions displayed.

Step	Action

11	Add the second cluster node by doing the following:
	a. Log in to the secondary SMS node as the root user.
	b. Run the following command, which creates the cluster:
	`/usr/cluster/bin/scinsstall`
	c. Follow the instructions displayed.
	d. (Optional) You can run the command along with the setup entries using the following command:
	`/usr/cluster/bin/scinstall -i -C global_1_global_2_gc -N global_1 -A trtype=dlpi,name=net3 -m endpoint=:net3,endpoint=net3`
	where,
	global_1 is the primary SMS node
	global_2 is the secondary SMS node
	e. Run the following command, which restarts the node:
	`shutdown - g0 -i6 -y`

12	Add the cluster server by doing the following:
	a. Log in to the primary SMS node as the root user.
	b. Run the following command, which creates the cluster:
	`/usr/cluster/bin/clsetup`
	c. Follow the instructions displayed.
	d. Run the following command, which verifies the cluster server:
	`/usr/cluster/bin/clquorumsever sho`

Configuring for Data Guard

Follow these steps to configure the Oracle Data Guard.

Step	Action

1	Prepare the primary SMS database node by doing the following:
	a. Log in to the primary SMS database as the oracle user.
	b. Run the following command, which opens SQL*Plus:
	`sqlplus '/as sysdba'`
	c. Run the following command which verify whether the primary SMS database is in archivelog mode:
	`SELECT log_mode FROM v$database;`
	d. If the database is in nonarchivelog mode, run the following commands, which switch the database to archivelog mode:
	`SHUTDOWM IMMEDIATE;`
	`STARTUP MOUNT;`
	`ALTER DATABASE ARCHIVELOG;`
	`ALTER DATABASE OPEN;`
	e. Run the following command, which enables forced logging:
	`ALTER DATABASE FORCE LOGGING;`
	f. Run the following command, which exits SQL*Plus:
	`EXIT`
	g. If the primary SMS database does not have a password, run the following command on all the cluster nodes, which creates the password file:
	`orapwd file=$ORACLE_HOME/dbs/orapwSMF password=oracle`

Step	Action
2	Create the standby redo logs by doing the following.

Note: Create the standby redo logs with a size as greater than as compared to the online redo logs. The number of standby redo logs must be one more than the online redo logs.

a. Log in to the primary SMS database as the oracle user.

b. Run the following command, which opens SQL*Plus:

```
sqlplus '/as sysdba'
```

c. Run the following command which create the log file:

```
col group# for 9999
col member for a50
set linesize 120
select GROUP#,THREAD#,MEMBERS,BYTES FROM V$LOG;
select GROUP#,MEMBER FROM v$logfile;
select GROUP#, BYTES FROM V$STANDBY_LOG;
ALTER DATABASE ADD STANDBY LOGFILE
('/oracle/redologs/SMF/s_redoSMF01.log') SIZE 104857600;
ALTER DATABASE ADD STANDBY LOGFILE
('/oracle/redologs/SMF/s_redoSMF02.log') SIZE 104857600;
ALTER DATABASE ADD STANDBY LOGFILE
('/oracle/redologs/SMF/s_redoSMF03.log') SIZE 104857600;
ALTER DATABASE ADD STANDBY LOGFILE
('/oracle/redologs/SMF/s_redoSMF04.log') SIZE 104857600;
ALTER DATABASE ADD STANDBY LOGFILE
('/oracle/redologs/SMF/s_redoSMF05.log') SIZE 104857600;
ALTER DATABASE ADD STANDBY LOGFILE
('/oracle/redologs/SMF/s_redoSMF06.log') SIZE 104857600;
ALTER DATABASE ADD STANDBY LOGFILE
('/oracle/redologs/SMF/s_redoSMF07.log') SIZE 104857600;
ALTER DATABASE ADD STANDBY LOGFILE
('/oracle/redologs/SMF/s_redoSMF08.log') SIZE 104857600;
ALTER DATABASE ADD STANDBY LOGFILE
('/oracle/redologs/SMF/s_redoSMF09.log') SIZE 104857600;
ALTER DATABASE ADD STANDBY LOGFILE
('/oracle/redologs/SMF/s_redoSMF10.log') SIZE 104857600;
ALTER DATABASE ADD STANDBY LOGFILE
('/oracle/redologs/SMF/s_redoSMF11.log') SIZE 104857600;
ALTER DATABASE ADD STANDBY LOGFILE
('/oracle/redologs/SMF/s_redoSMF12.log') SIZE 104857600;
ALTER DATABASE ADD STANDBY LOGFILE
('/oracle/redologs/SMF/s_redoSMF13.log') SIZE 104857600;
ALTER DATABASE ADD STANDBY LOGFILE
('/oracle/redologs/SMF/s_redoSMF14.log') SIZE 104857600;
ALTER DATABASE ADD STANDBY LOGFILE
('/oracle/redologs/SMF/s_redoSMF15.log') SIZE 104857600;
ALTER DATABASE ADD STANDBY LOGFILE
('/oracle/redologs/SMF/s_redoSMF16.log') SIZE 104857600;
ALTER DATABASE ADD STANDBY LOGFILE
('/oracle/redologs/SMF/s_redoSMF17.log') SIZE 104857600;

SELECT GROUP#, BYTES FROM V$STANDBY_LOG;
```

Step **Action**

3 Modify the **initSMF.ora** file parameters on the primary SMS database by doing the following:

Note: If you are using **spfile**, you can add these parameters online without shutting down the primary database.

a. Log in to the primary SMS database as the oracle user.

b. Create a backup of the **initSMF.ora** file by running the following commands:

```
cd /u01/app/oracle/product/12.1.0.2/dbs
cp initSMF.ora initSMF.ora.bak
```

c. Open the **initSMF.ora** file in a text editor.

d. Add the following entries:

```
db_unique_name=SMF1
log_archive_config='DG_CONFIG=(SMF1,SMF2)'
log_archive_dest_1='LOCATION=/oracle/archivelogs/SMF/
VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=SMF1'
log_archive_dest_2='SERVICE=smsdb_2 ASYNC
VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE)
DB_UNIQUE_NAME=SMF1'
log_archive_dest_state_1=ENABLE
log_archive_dest_state_2=ENABLE
log_archive_max_processes=30
fal_server=smsdb_2
standby_file_management=AUTO
```

e. Save and close the file.

f. Run the following commands which create the archivelog directory:

```
mkdir /oracle/archivelogs/SMF
```

4 Create a **controlfile** for the standby database by doing the following:

a. Log in to the primary SMS database as the oracle user.

b. Run the following commands which open the SQL*PLUS:

```
sqlplus '/ as sysdba'
```

c. Run the following command, which creates the **controlfile** for the standby database:

```
SHUTDOWN IMMEDIATE;
STARTUP MOUNT;
ALTER DATABASE CREATE STANDBY CONTROLFILE ASS
'/tmp/s_control101.ctl';
SHUTDOWN IMMEDIATE
```

Step	Action
5	Configure the **tnsnames.ora** file by doing the following:

a. Log in to the primary SMS database as the oracle user.

b. Open the **tnsnames.ora** file in a text editor.

c. Add the following lines:

```
smsdb_1=
(DESCRIPTION =
   (ADDRESS_LIST =
     (ADDRESS = (PROTOCOL = TCP)(Host = smsdb_1)(Port =
1521))
   )
 (CONNECT_DATA =
   (SERVICE_NAME = SMF)
 )
)
smsdb_2 =
(DESCRIPTION =
   (ADDRESS_LIST =
     (ADDRESS = (PROTOCOL = TCP)(Host = smsdb_2)(Port =
1521))
   )
 (CONNECT_DATA =
   (SERVICE_NAME = SMF)
 )
)
```

d. Save and close the file.

e. Open the **/etc/hosts** file in a text editor.

f. Add the following lines:

```
x.x.x.x smsdb_1
y.y.y.y smsdb_2
```

g. Save and close the file.

Cloning the Primary SMS Database to the Secondary SMS Database

Follow these steps to clone the primary SMS database node to the secondary SMS database node:

Step	Action
1	Configure the secondary SMS node by doing the following: a. Log in to the primary SMS node as the root user. b. Run the following command, which shuts down the primary SMS node: **zoneadm -z** *smsapp* **shutdown** c. Run the following command, which copies SMS configuration to the ZFS file systems: **zonecfg -z** *smsapp* **export > /tmp/***smsapp***.zonecfg** d. Run the following commands: **scp /tmp/***smsdb_1***.zonecfg global_2:/tmp** e. Run the following commands, which copy the ZFS file systems: **zone snapshot -r** *smsdb_1***@clone** **zfs send -rc sms_zone/***smsdb_1***@clone | ssh** *global_2* **zfs receive sms_zone/smsdb_2** **zfs send -rc sms_zone/smsdb_1_Parent@clone | ssh** *global_2* **zfs receive sms_zone/***smsdb_2***_Parent**

Step	Action
2	Configure the secondary SMS node by doing the following:

a. Log in to the secondary SMS node as the root user.

b. Run the following commands:

```
zfs set mountpoint=/zones/smsdb_2 sms_zone/smsdb_2
zfs mount sms_zone/smsdb_2
```

c. Open the **/tmp/smsapp.zonecfg** file in a text editor.

d. Update the zonepath and dataset entries; for example:

```
set zonepath=/zones/smsdb_2
set name=sms_zone/smsdb_2_Parent
zonecfg -z smsdb_2 -f /tmp/smsdb.zonecfg
zoneadm -z smsdb_2 attach
zoneadm -z smsdb_2
```

e. Run the following commands, which reset the system configuration:

```
sysconfig unconfigure
sysconfig configure
reboot
```

f. Configure the secondary SMS database node; for example:

```
Computer name: smsdb_2
Do not configure DNS
Name service : None
Timezone: UTC/GMT
root password: password
```

g. Run the following commands, which configure the IP address:

```
ipadm create-ip adm1
ipadm create-addr -T static -a local=IP_address/21
adm1/v4static
ipadm create-ip rep1
ipadm create-addr -T static -a local=IP_address/24
rep1/v4static
ipadm create-ip bill1
ipadm create-addr -T static -a local=IP_address/24
bill1/v4static
route -p add default IP_address
```

Completing the Dataguard Configuration

Follow these steps to complete the Data Guard configuration.

Step	Action

1	Modify the **initSMF.ora** file parameters on the secondary SMS database node by doing the following:
	a. Log in to the secondary SMS database node as the oracle user.
	b. Create a backup of the **initSMF.ora** file:
	`cd /u01/app/oracle/product/12.1.0.2/dbs`
	`cp initSMF.ora initSMF.ora.bak`
	c. Open the **initSMF.ora** file in a text editor.
	d. Add the following entries:
	`db_unique_name=SMF2`
	`log_archive_config='DG_CONFIG=(SMF1,SMF2)'`
	`log_archive_dest_1='LOCATION=/oracle/archivelogs/SMF/ VALID_FOR=(ALL_LOGFILES,ALL_ROLES) DB_UNIQUE_NAME=SMF2'`
	`log_archive_dest_2='SERVICE=smsdb_1 ASYNC VALID_FOR=(ONLINE_LOGFILES,PRIMARY_ROLE) DB_UNIQUE_NAME=SMF1'`
	`log_archive_dest_state_1=ENABLE`
	`log_archive_dest_state_2=ENABLE`
	`log_archive_max_processes=30`
	`fal_server=smsdb_1`
	`standby_file_management=AUTO`
	e. Save and close the file.

Step **Action**

2 Configure the **tnsnames.ora** file by doing the following:

a. Log in to the primary SMS database as the oracle user.

b. Open the **tnsnames.ora** file in a text editor.

c. Add the following lines:

```
smsdb_1=
(DESCRIPTION =
   (ADDRESS_LIST =
     (ADDRESS = (PROTOCOL = TCP)(Host = smsdb_1)(Port =
1521))
   )
 (CONNECT_DATA =
   (SERVICE_NAME = SMF)
 )
)
smsdb_2 =
(DESCRIPTION =
   (ADDRESS_LIST =
     (ADDRESS = (PROTOCOL = TCP)(Host = smsdb_2)(Port =
1521))
   )
 (CONNECT_DATA =
   (SERVICE_NAME = SMF)
 )
)
```

d. Save and close the file.

e. Open the **/etc/hosts** file in a text editor.

f. Add the following lines:

```
x.x.x.x smsdb_1
y.y.y.y smsdb_2
```

g. Save and close the file.

3 On both the primary and secondary SMS nodes, test the configuration by running the following commands:

```
tnsping smsdb_1
tnsping smsdb_2
```

Step **Action**

4 Restart the databases by doing the following:

a. Log in to the primary SMS database node as the oracle user.

b. Run the following command, which opens SQL*Plus:

```
sqlplus '/as sysdba'
```

c. Run the following command, which starts the primary SMS database:

```
STARTUP
```

d. Log in to the secondary SMS database node as the oracle user.

e. Run the following command, which opens SQL*Plus:

```
sqlplus '/as sysdba'
```

f. Run the following command, which start the secondary SMS database:

```
STARTUP MOUNT;
```

g. Run the following commands, which make the required modifications:

```
ALTER DATABASE OPEN READ ONLY;
 ALTER DATABASE RECOVER MANAGED STANDBY DATABASE USING
CURRENT LOGFILE DISCONNECT FROM SESSION;
```

5 Test the Data Guard configuration by doing the following:

a. Log in to the primary SMS database node as the oracle user.

b. Run the following command, which opens SQL*Plus:

```
sqlplus '/as sysdba'
```

c. Run the following commands, which test the database configuration:

```
SELECT sequence#, first_time, next_time FROM v$archived_log
ORDER BY sequence#;
ALTER SYSTEM SWITCH LOGFILE;
SELECT sequence#, first_time, next_time FROM v$archived_log
ORDER BY sequence#;
```

d. Log in to the secondary SMS database node as the oracle user.

e. Run the following command, which opens SQL*Plus:

```
sqlplus '/as sysdba'
```

f. Run the following commands, which test the database configuration:

```
SELECT sequence#, first_time, next_time, applied FROM
v$archived_log ORDER BY sequence#;
```

Note: Ensure the database listeners are running on both the primary and secondary SMS nodes.

Switching the Database Modes

A database can be either primary mode or standby mode. These modes can be altered at run time without loss of data or resetting the redo logs. This process is known as a switchover.

Follow these steps to switch the database modes.

Step **Action**

1 Run the following command, which connects to the primary database:

```
CONNECT / AS SYSDBA
```

2 Run the following command, which converts the primary database to the standby database:

```
ALTER DATABASE COMMIT TO SWITCHOVER TO STANDBY WITH SESSION
SHUTDOWN;
```

Step	Action
3	Run the following command, which shuts down the primary database: `SHUTDOWN IMMEDIATE`

4	Run the following commands, which start the old primary database as the standby database:
	`STARTUP MOUNT;`
	`ALTER DATABASE MOUNT STANDBY DATABASE;`
	`ALTER DATABASE OPEN READ ONLY;`
	`ALTER DATABASE RECOVER MANAGED STANDBY DATABASE DISCONNECT FROM SESSION;`

5	Run the following command, which connects to the original standby database: `CONNECT / AS SYSDBA`
6	Run the following command, which converts the standby database to the primary database: `ALTER DATABASE COMMIT TO SWITCHOVER TO PRIMARY`
7	Run the following command, which shuts down the standby database: `SHUTDOWN IMMEDIATE`

8	Run the following commands, which start the standby database as the primary database:
	`EXIT;`
	`sqlplus '/as sysdba'`
	`STARTUP;`

Configuring the Standard SMS Node Database Connections

Configuring standard SMS node database connections involves the following:

- Configuring Oracle Wallet
- Configuring the SMS database

Configuring Oracle Wallet

Follow these steps to configure Oracle Wallet:

Step	Action

1	Log in to the standard SMS node as the root user.
	`su-oracle`
	`cd $ORACLE_HOME/network/admin`
	`vi`

2	Open the **$ORACLE_HOME/network/admin/sqlnet.ora** file in a text editor.

3	Add the following lines:
	`WALLET_LOCATION =`
	`  (SOURCE = (METHOD = FILE)`
	`    (METHOD_DATA =`
	`      (DIRECTORY = /u01/app/oracle/product/12.1.0.2/network/admin)))`
	`SQLNET.WALLET_OVERRIDE = TRUE`

4	Save and close the file.

Step	Action

5	Run the following commands, which create the wallet store and follow the instructions displayed:
	`mkstore -wrl . -create`
	`mkstore -wrl . -createCredential SMF smf SMF`
	`mkstore -wrl . -createCredential ACS_ADMIN_SMF acs_admin ACS_ADMIN`
	`mkstore -wrl . -createCredential CCS_ADMIN_SMF ccs_admin CCS_ADMIN`
	`sqlplus /@CCS_ADMIN_SMF`
	`show user`

6	Run the following command, which set the required file permissions: `chmod 777 *wallet*`
7	Log in as the smf_user user.

8	Run the following commands, which display the users for the SMF database:
	`sqlplus /@SMF`
	`show user;`

9	Run the following commands, which display the users for the ACS ADMIN SMF database:
	`sqlplus /@ACS_ADMIN_SMF`
	`show user;`

10	Run the following commands, which display the users for the CCS ADMIN SMF database:
	`sqlplus /@CCS_ADMIN_SMF`
	`show user;`

Configuring the SMS Database

Follow these steps to configure the application startup.

Step	Action
1	Log in as the root user.

2	Add the following entry:
	`"-u /@SMF"`
	in the following files:
	`/IN/service_packages/SMS/bin/smsNamingServerStartup.sh`
	`/IN/service_packages/SMS/bin/smsAlarmRelayStartup.sh`
	`/IN/service_packages/SMS/bin/smsReportsDaemonStartup.sh`
	`/IN/service_packages/SMS/bin/smsReportSchedulerStartup.sh`
	`/IN/service_packages/SMS/bin/smsStatsThresholdStartup.sh`
	`/IN/service_packages/SMS/bin/smsTaskAgentStartup.sh`
	Note: Add the following line where applicable:
	`Waiting for DB SMF`

Step	Action

3	Add the following entry:
	`"-u /@CCS_ADMIN_SMF"`
	in the following files:
	`/IN/service_packages/CCS/bin/ccsProfileDaemonStartup.sh`
	`/IN/service_packages/CCS/bin/ccsVoucherStartup.sh`
	`/IN/service_packages/CCS/bin/ccsAccountStartup.sh`
	`/IN/service_packages/CCS/bin/ccsCDRTrimDBStartup.sh`
	Note: Add the following line where applicable:
	`Waiting for DB SMF`

4	Open the **/IN/service_packages/ACS/bin/acsDbCleanup.sh** file in a text editor and do the following:
	a. Search for the following line:
	`connect /`
	b. Replace the line with the following:
	`connect /@ACS_ADMIN_SMF`
	c. Save and close the file.

5	Open the **/IN/service_packages/ACS/bin/acsSetCurrentDate.sh** file in a text editor and add the following lines:
	`/IN/service_packages/ACS/bin/acsProfile -u / -U -G 1 -W 0x2b0b90 -L $MONTH`
	`/IN/service_packages/ACS/bin/acsProfile -u / -U -G 1 -W 0x2b0b91 -L $DAY`
	`/IN/service_packages/ACS/bin/acsProfile -u /@ACS_ADMIN_SMF -U -G 1 -W 0x2b0b90 -L $MONTH`
	`/IN/service_packages/ACS/bin/acsProfile -u /@ACS_ADMIN_SMF -U -G 1 -W 0x2b0b91 -L $DAY`
	Save and close the file.

6	Open the **/IN/service_packages/CCS/bin/ccsPmxGlobalLimitExpiry_job.sh** file in a text editor.
	a. Search for the following line:
	`conn /`
	b. Replace with the following line:
	`conn /@CCS_ADMIN_SMF`
	c. Save and close the file.

7	Open the **/IN/service_packages/CCS/bin/ccsbt_deactive_cleanup.sh** file in a text editor.
	a. Search for the following line:
	`connect /`
	b. Replace with the following line:
	`connect /@CCS_ADMIN_SMF`
	c. Save and close the file.

8	Open the **/IN/service_packages/CCS/bin/ccsbt_execute.sh** file in a text editor.
	a. Search for the following line:
	`ORACLEUSERPASS="/"`
	b. Replace with the following line:
	`ORACLEUSERPASS="/@CCS_ADMIN_SMF"`
	c. Save and close the file.

Step	Action

9	Open the **/IN/service_packages/CCS/bin/smsDbCleanup.sh** file in a text editor.
	a. Search for the following line:
	`connect /`
	`sqlplus -s / <<END`
	`connect /`
	`connect /`
	`connect /`
	b. Replace with the following line:
	`connect /@SMF`
	`sqlplus -s /@SMF <<END`
	`connect /@SMF`
	`connect /@SMF`
	`connect /@SMF`
	c. Save and close the file.

10	Open the **/IN/service_packages/CCS/bin/fetchSubs.sh** file in a text editor and add the following lines:
	`CONNECT /`
	`CONNECT /@SMF`
	Save and close the file.

11	Open the **/IN/service_packages/CCS/bin/fetchSubs.sh** file in a text editor.
	a. Search for the following line:
	`CONNECT/"`
	b. Replace with the following line:
	`CONNECT /@SMF`
	c. Save and close the file.

12	Open the **eservconfig** file in a text editor and add the following entries:
	`triggering.oracleLogin = "/@SMF"`
	`CCS.oracleUserAndPassword = "/@CCS_ADMIN_SMF"`
	`CCS.ccsCDRLoader.dbUserPass = "/@CCS_ADMIN_SMF"`
	`CCS.ccsCDRFileGenerator.OracleUsernamePassword = "/@CCS_ADMIN_SMF"`
	`CCS.ccsPeriodicCharge.OracleUserAndPassword = "/@CCS_ADMIN_SMF"`
	`CCS.ccsPeriodicCharge.mergeWalletsOptions.oracleLogin = "/@CCS_ADMIN_SMF"`
	`SES.dbCleanup.password = "@SMF"`
	`pi.general.oraUser = "/@SMF"`
	`pi.PIbeClient.oracleLogin = "/@SMF"`
	Save and close the file.

Step **Action**

13 Open the **sms.html** or **sms.jnlp** file in a text editor and add the following parameters:

```
<param name=clusterDatabaseHost value="
(DESCRIPTION =
(LOAD_BALANCE=NO)
(FAILOVER=YES)
(ENABLE=BROKEN)
(ADDRESS_LIST =
(ADDRESS = (PROTOCOL=TCP)(HOST=smsdb_1)(PORT=1521))
(ADDRESS = (PROTOCOL=TCP)(HOST=smsdb_2)(PORT=1521))
)
(CONNECT_DATA =
(SERVICE_NAME = SMF_RW)
(SERVER=DEDICATED)
)
)
">
```

Save and close the file.

14 Open the **resyncServer** file in a text editor and add the following entries:

```
${BASEDIR}/bin/inputBootstrap --node-id $1 --hex-address $2 --
port $3 -u "/" $ENHANCED \
${BASEDIR}/bin/smsCompareResyncServer ${EXTRA_FLAGS} -u "/" <
"/tmp/resyncServerInput.$1.${DATE}" >> ${LOGFILE} 2>&1 &
${BASEDIR}/bin/inputBootstrap --node-id $1 --hex-address $2 --
port $3 -u "/" $ENHANCED \
${BASEDIR}/bin/smsCompareResyncServer ${EXTRA_FLAGS} --inform-
master -u "/" \
${BASEDIR}/bin/inputBootstrap --node-id $1 --hex-address $2 --
port $3 -u "/@SMF" $ENHANCED \
${BASEDIR}/bin/smsCompareResyncServer ${EXTRA_FLAGS} -u "/@SMF"
< "/tmp/resyncServerInput.$1.${DATE}" >> ${LOGFILE} 2>&1 &
${BASEDIR}/bin/inputBootstrap --node-id $1 --hex-address $2 --
port $3 -u "/@SMF" $ENHANCED \
${BASEDIR}/bin/smsCompareResyncServer ${EXTRA_FLAGS} --inform-
master -u "/@SMF" \
```

Save and close the file.

15 Open the **/IN/service_packages/UIS/etc/cdrLoader.conf** file in a text editor and add the following entries:

```
username=smf
password=SMF
nsname=SMF
```

Save and close the file.

Configuring the Failover Standard SMS Database

Follow these steps to configure SMS node failover from the primary SMS database node to the secondary SMS database node.

Step	Action
1.	Clone the SMS node from the primary SMS node to the secondary SMS node by doing the following. a. Log in to the primary SMS node as the root user. b. Run the following command, which shuts down the standard SMS node: **zoneadm -z** *smsapp* **shutdown** c. Run the following command, which copies the SMS configuration to the ZFS file system: **zonecfg -z** *smsapp* **> /tmp/***smsapp***.zonecfg** **scp /tmp/smsapp.zoneconfig global_2:/tmp** d. Run the following commands, which copy the ZFS file systems: **zone snapshot -r sms_zone@clone** **zfs send -rc sms_zone/***smsapp***@clone | ssh** *global_2* **zfs receive sms_zone/***smsapp* **zfs send -rc sms_zone/***smsapp***_Parent@clone | ssh** *global_2* **zfs receive sms_zone/***smsapp***_Parent** e. Log in to the secondary SMS node as the root user. f. Run the following commands: **zonecfg -z** *smsapp* **-f /tmp/***smsapp***.zonecfg** **zoneadm -z** *smsapp* **attach** **zoneadm -z** *smsapp* **boot** **Note:** Ensure the test node is working and nodes can connect.
2	Configure the standard SMS node failover from the primary SMS node to the secondary SMS node by doing the following: a. Log in to the secondary SMS node as the root user. b. Run the following command, which shuts down the standard SMS node: **zoneadm -z** *smsapp* **shutdown** c. Log in to the primary SMS node as the root user. d. Run the following commands: **zonecfg -z** *smsapp* **set autoboot=false** **add attr** **set name=osc-ha-zone** **set type=boolean** **set value=true** **end** **verify** **commit** e. Log in to the secondary SMS node as the root user. f. Run the following commands: **zonecfg -z** *smsapp* **set autoboot=false** **add attr** **set name=osc-ha-zone** **set type=boolean** **set value=true** **end** **verify** **commit**

Step	Action
	g. Log in to the primary SMS node as the root user.

	h. Run the following commands:
	`clrt register SUNW.gds`
	`/usr/cluster/bin/clrg create smsapp-zone-rg`
	`/usr/cluster/bin/clrg online -emM -n global_1 smsapp-zone-rg`
	`cd /opt/SUNWsczone/sczbt/util`
	`cp -p sczbt_config sczbt_config.smsapp-fz-rs`

	i. Open the **sczbt_config.smsapp-fz-rs** file in a text editor.

	j. Set the following parameters:
	`RS=smsapp-fz-rs`
	`RG=smsapp-zone-rg`
	`PARAMETERDIR=/etc/zones/cluster_pfiles`
	`SC_NETWORK=false`
	`FAILOVER=false`
	`Zonename="smsapp"`
	`Zonebrand="solaris"`

	k. Run the following commands:
	`mkdir /etc/zones/cluster_pfiles`
	`./sczbt_register -f ./sczbt_config.smsapp-fz-rs`

	l. Save and close the file.
	m. Log in to the secondary SMS node as the root user.

	n. Run the following commands:
	`mkdir /etc/zones/cluster_pfiles`
	`cd /etc/zones/cluster_pfiles`
	`scp dfabrice@global_1:/etc/zones/cluster_pfiles/*`

	o. Log in to the primary SMS node as the root user.

	p. Run the following commands:
	`clrs enable smsapp-fz-rs`
	`/usr/cluster/bin/clrg offline smsapp-zone-rg`
	`zoneadm list -cv | grep smsapp`
	`/usr/cluster/bin/clrg online smsapp-zone-rg`
	`zoneadm list -cv | grep smsapp`
	`clrs status -g smsapp-zone-rg`
	`/usr/cluster/bin/clrg switch -n ipwaa53 smsapp-zone-rg`
	`clrs status -g smsapp-zone-rg`

Step **Action**

3 Set up the password-less SSH connections for the primary SMS node and the secondary SMS node by doing the following:

Note: Perform these steps to update the standard SMS node ZFS file systems after applying the patches and configuration changes.

a. Log in to the primary SMS node as the root user.

b. Run the following commands:

```
ssh-keygen -t dsa -N "" -f ~/.ssh/id_dsa
ssh-copy-id -i ~/.ssh/id_dsa.pub global_2
ssh global_2
exit
```

c. Run the following commands, which update the date to the `` `date '+%Y%m%d%H%M%S'` `` format:

```
zfs snapshot -r sms_zone/smsapp_OS@$DATE
zfs snapshot -r sms_zone/smsapp_Parent@$DATE
zfs send -rc -i sms_zone/smsapp_OS@initial
sms_zone/smsapp_OS@$DATE | ssh global_2 zfs recv -Fv
sms_zone/smsapp_OS
zfs send -rc -i sms_zone/smsapp_Parent@initial
sms_zone/smsapp_Parent@$DATE | ssh global_2 zfs recv -Fv
sms_zone/smsapp_Parent
```

d. Run the following command:

```
zfs list -t all -r sms_zone
```

e. Run the following command which search for the latest snapshot date and export as $PREV_DATE:

```
zfs snapshot -r sms_zone/smsapp_OS@$DATE
zfs snapshot -r sms_zone/smsapp_Parent@$DATE
zfs send -rc -i sms_zone/smsapp_OS@$PREV_DATE
sms_zone/smsapp_OS@$DATE | ssh global_2 zfs recv -Fv
sms_zone/smsapp_OS
zfs send -rc -i sms_zone/smsapp_Parent@$PREV_DATE
sms_zone/smsapp_Parent@$DATE | ssh global_2 zfs recv -Fv
sms_zone/smsapp_Parent
```

For example:

```
PREV_DATE="20130913093118"
DATE=`date '+%Y%m%d%H%M%S'
```

Glossary of Terms

AAA

Authentication, Authorization, and Accounting. Specified in Diameter RFC 3588.

ACS

Advanced Control Services configuration platform.

CCS

1) Charging Control Services component.

2) Common Channel Signalling. A signalling system used in telephone networks that separates signalling information from user data.

Convergent

Also “convergent billing”. Describes the scenario where post-paid and pre-paid calls are handed by the same service platform and the same billing system. Under strict converged billing, post-paid subscribers are essentially treated as “limited credit pre-paid”.

cron

Unix utility for scheduling tasks.

Diameter

A feature rich AAA protocol. Utilises SCTP and TCP transports.

DTMF

Dual Tone Multi-Frequency - system used by touch tone telephones where one high and one low frequency, or tone, is assigned to each touch tone button on the phone.

GSM

Global System for Mobile communication.

It is a second generation cellular telecommunication system. Unlike first generation systems, GSM is digital and thus introduced greater enhancements such as security, capacity, quality and the ability to support integrated services.

GUI

Graphical User Interface

HLR

The Home Location Register is a database within the HPLMN (Home Public Land Mobile Network). It provides routing information for MT calls and SMS. It is also responsible for the maintenance of user subscription information. This is distributed to the relevant VLR, or SGSN (Serving GPRS Support Node) through the attach process and mobility management procedures such as Location Area and Routing Area updates.

IN

Intelligent Network

IP

1) Internet Protocol

2) Intelligent Peripheral - This is a node in an Intelligent Network containing a Specialized Resource Function (SRF).

IP address

Internet Protocol Address - network address of a card on a computer.

IVR

Interactive Voice Response - systems that provide information in the form of recorded messages over telephone lines in response to user input in the form of spoken words or, more commonly, DTMF signalling.

MAP

Mobile Application Part - a protocol which enables real time communication between nodes in a mobile cellular network. A typical usage of the protocol would be for the transfer of location information from the VLR to the HLR.

PI

Provisioning Interface - used for bulk database updates/configuration instead of GUI based configuration.

SCTP

Stream Control Transmission Protocol. A transport-layer protocol analogous to the TCP or User Datagram Protocol (UDP). SCTP provides some similar services as TCP (reliable, in-sequence transport of messages with congestion control) but adds high availability.

Session

Diameter exchange relating to a particular user or subscriber access to a provided service (for example, a telephone call).

SLC

Service Logic Controller (formerly UAS).

SMS

Depending on context, can be:

- Service Management System hardware platform
- Short Message Service
- Service Management System platform
- Convergent Charging Controller Service Management System application

SQL

Structured Query Language is a database query language.

SRF

Specialized Resource Function – This is a node on an IN which can connect to both the SSP and the SLC and delivers additional special resources into the call, mostly related to voice data, for example play voice announcements or collect DTMF tones from the user. Can be present on an SSP or an Intelligent Peripheral (IP).

SSP

Service Switching Point

TCP

Transmission Control Protocol. This is a reliable octet streaming protocol used by the majority of applications on the Internet. It provides a connection-oriented, full-duplex, point to point service between hosts.

UIS

USSD Interactive Services

USSD

Unstructured Supplementary Service Data - a feature in the GSM MAP protocol that can be used to provide subscriber functions such as Balance Query.

VLR

Visitor Location Register - contains all subscriber data required for call handling and mobility management for mobile subscribers currently located in the area controlled by the VLR.

VWS

Oracle Voucher and Wallet Server (formerly UBE).

Index