

Installing Highly Available SAP system with Failover Oracle Database on Solaris Cluster

ORACLE WHITE PAPER | MAY 2016

SAP® Certified
Integration with SAP NetWeaver®





Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.



ORACLE®



Table of Contents

Disclaimer	1
1. Introduction	1
2. Preparation	2
3. Set Up File Systems for SAP and Oracle	2
3.1. SAP File Systems	2
3.2. Oracle File Systems	3
4. Create Solaris Cluster Resource Groups and Storage Resources	6
5. Install High-Availability SAP System on logical hosts	6
6. Take the Oracle Database Server and Client under Solaris Cluster Control	8
7. Take the (A)SCS and ERS instances under Solaris Cluster control	11
8. Take the Application Server Instances under Solaris Cluster Control	16

1. Introduction

Oracle Solaris Cluster is a high availability (HA) cluster hardware and software product for the Oracle Solaris operating system. It is used to improve the availability of hardware and software services and business continuity. Oracle Solaris Cluster operates by integrating redundant storage, network switches, computers (known as cluster nodes), the Solaris Cluster framework software, and cluster data services (known as agents) for the applications to create a HA environment. Applications are administrated and monitored in resource groups which consist of one or more resources. Resource groups can be configured as fail over, scalable or multi-mastered, depending on the application requirement.

Oracle Solaris Cluster 3.3 and 4.x (x=0,1,2,3) supports all the SAP products based on SAP NetWeaver 7.0x (x=0,1,2,3) and 7.x (x=1,11,2,3,31,4,5) with SAP kernel version updated to at least 720/720_EXT. For more information of supported SAP products on Solaris Cluster, please refer to the Solaris Cluster Compatibility Guide and corresponding SAP notes.

A typical High Availability SAP system contains the following instances:

- » **(A)SCS** – The Standalone Central Services instance
- » **ERS** – The Enqueue Replication Server instance
- » **Database** – The Oracle Database instance can be failover or with Oracle RAC. In this document, the Oracle database is single instance that can failover from one node to the other.
- » **PAS** – The Primary Application Server instance
- » **AAS** – (Optional) One or more Additional Application Server instances

The Oracle Solaris Cluster Data Service for SAP NetWeaver (or short: NetWeaver Agent) is fully qualified by Oracle and also certified by SAP. The agent provides administrative and monitoring functions for (A)SCS, ERS, Database, PAS and AAS instances to ensure that these instances are highly available. The agent is also integrated with SAP HA script connector via sapstartsrv.



2. Preparation

2.1. Prepare Oracle Solaris Cluster hardware with storage.

2.2. Install Solaris operating system, Solaris Cluster framework and the data service software. Install the latest patches.

2.3. Download the SAP product software. The downloaded files can be extracted with unrar, unzip and SAPCAR. After unzipping the kernel package, you may find the SAPCAR executable in the extracted kernel DVD.

2.4. Create the necessary users, groups and projects for the SAP system and Oracle database. If the users and groups are created centrally, make sure they are available on all the cluster nodes. If the users and groups are created locally on each node, make sure the users and groups are identical on all the nodes, especially with same uid and gid.

2.5. Create the project with required parameters for the users and groups. For more information, see SAP note 724713.

2.6. Prepare the required logical hosts for each SAP and Oracle instance. It is recommended to put them into `/etc/hosts` with short name and also Full Qualified Domain Name.

3. Set Up File Systems for SAP and Oracle

On Oracle Solaris Cluster, ZFS Storage Appliance or Global File System are recommended to be used as HA file systems. The file systems can be attached or mounted to all the cluster nodes so that the subfolders and files are shared among cluster nodes. Failover of file system is not necessary.

3.1. SAP File Systems

The whole SAP system can be installed on the same file system `/usr/sap`, or each SAP instance can have its own file system under `/usr/sap/<SID>/<Instance>`. We recommend putting all the SAP instances on the same file system `/usr/sap` for each Solaris Cluster or zone cluster. Prepare the following file systems for the SAP instances. Mount them on all the cluster nodes for the SAP instances.

- » **/usr/sap** – shared file system, mounted on corresponding zones that SAP instances may run.
- » **/sapmnt/<SID>** – shared file system, mounted on all zones that SAP instances may run. If the Oracle database is on different cluster as the SAP instances, this file system need to be mounted on the zones for Oracle database as well.
- » **/usr/sap/trans** – shared file system, mounted on all zones that SAP instances may run.

NFS mount options on SAP zones according to SAP recommendation:

`rw,bg,hard,rsize=32768,wsiz=32768,proto=tcp,vers=3`

NFS mount options on Oracle DB zones according to Oracle recommendation:

`rw,bg,hard,nointr,rsiz=1048576,wsiz=1048576,vers=3,proto=tcp,forcedirectio`


The mount option `nointr` is not recommended for SAP on Solaris Cluster. Setting `nointr` may cause Solaris Cluster not able to kill and restart the `sapstartsv` processes when NFS is hanging. This will cause the whole SAP system hanging.

SAP FILE SYSTEMS

SAP File System	Mounted on	Mount Options
/sapmnt/<SID>	All zones SAP instances may run	<code>rw,bg,hard,rsiz=32768,wsiz=32768,proto=tcp,vers=3</code>
	Oracle Database zones	<code>rw,bg,hard,nointr,rsiz=1048576,wsiz=1048576,vers=3,proto=tcp,forcedirectio</code>
/usr/sap (one share or multiple shares)	All zones SAP instances may run	<code>rw,bg,hard,rsiz=32768,wsiz=32768,proto=tcp,vers=3</code>
/usr/sap/trans	All zones SAP instances may run	<code>rw,bg,hard,rsiz=32768,wsiz=32768,proto=tcp,vers=3</code>

3.2. Oracle File Systems

Prepare the following file systems on ZFS SA or as Global File System for the Oracle database instance. Mount them on the cluster nodes for the Oracle database. Make sure the file system has the



permissions 755 and belongs to ora<sid>:dba (when using user ora<sid>) or oracle:oinstall (when using user oracle).

- » **/oracle** – shared file system. The /oracle can be put on shared file system or on local file system. The benefit of using shared file system for /oracle is easier to configuration, installation and administration. The benefit of putting /oracle on local file system is to avoid nested mount points. For Oracle RAC, we recommend to put /oracle on local file system, because the Grid Infrastructure is required to be installed on each node locally in the folder /oracle/grid. For single instance failover Oracle Database, it is recommended to put /oracle on shared file system.
- » **/oracle/client** – shared file system. If the Oracle database and the SAP instances are on the same cluster, the Oracle Client can be on /oracle file system. If the SAP instances are on different (Zone) Clusters as the Oracle database, it is recommended to put Oracle Client on a shared file system which is separated from /oracle. Then /oracle/client should be mounted on both the cluster for Oracle database and also the cluster for SAP instances.
- » **/oracle/<SID>/<Oracle_database_version>** – shared file system, mounted on all Oracle database zone
- » **/oracle/<SID>/origlogA** – shared file system, mounted on all Oracle database zone
- » **/oracle/<SID>/origlogB** – shared file system, mounted on all Oracle database zone
- » **/oracle/<SID>/mirrlogA** – shared file system, mounted on all Oracle database zone
- » **/oracle/<SID>/mirrlogB** – shared file system, mounted on all Oracle database zone
- » **/oracle/<SID>/oraarch** – shared file system, mounted on all Oracle database zone
- » **/oracle/<SID>/sapreorg** – shared file system, mounted on all Oracle database zone
- » **/oracle/<SID>/sapdata1** – shared file system, mounted on all Oracle database zone
- » **/oracle/<SID>/sapdata2** – shared file system, mounted on all Oracle database zone
- » **/oracle/<SID>/sapdata3** – shared file system, mounted on all Oracle database zone
- » **/oracle/<SID>/sapdata4** – shared file system, mounted on all Oracle database zone
- » **/oracle/<SID>/cfgtoollogs** – shared file system, mounted on all Oracle database zone
- » **/oracle/<SID>/oraflash** – shared file system, mounted on all Oracle database zone
- » **/oracle/<SID>/saparch** – shared file system, mounted on all Oracle database zone
- » **/oracle/<SID>/sapbackup** – shared file system, mounted on all Oracle database zone
- » **/oracle/<SID>/sapcheck** – shared file system, mounted on all Oracle database zone
- » **/oracle/<SID>/sapprof** – shared file system, mounted on all Oracle database zone
- » **/oracle/<SID>/saptrace** – shared file system, mounted on all Oracle database zone

NFS mount options for /oracle, /oracle/client and /oracle/<SID>/<Oracle_database_version> which hold binaries:

```
rw,bg,hard,nointr,rsize=1048576,wsiz=1048576,vers=3,proto=tcp,suid
```

NFS mount options for Oracle data and log files:

```
rw,bg,hard,nointr,rsize=1048576,wsiz=1048576,vers=3,forcedirectio,proto=tcp,suid
```

The mount option noac is not recommended for single instance Oracle database. It is only required by Oracle RAC.

FILE SYSTEMS FOR SINGLE INSTANCE FAILOVER ORACLE DATABASE (WITHOUT ASM)

Oracle File System	Mounted on	Mount Options
/oracle	Every Oracle DB zone	rw,bg,hard,nointr,rsize=1048576,wsiz=1048576,vers=3,proto=tcp,suid
/oracle/client	Every Oracle DB zone, every zone for SAP PAS and AAS instances	rw,bg,hard,nointr,rsize=1048576,wsiz=1048576,vers=3,proto=tcp,suid
/oracle/<SID>/<release>	Every Oracle DB zone	rw,bg,hard,nointr,rsize=1048576,wsiz=1048576,vers=3,proto=tcp,suid
/oracle/<SID>/origlogA	Every Oracle DB zone	rw,bg,hard,nointr,rsize=1048576,wsiz=1048576,vers=3,forcedirectio,proto=tcp,suid
/oracle/<SID>/origlogB	Every Oracle DB zone	rw,bg,hard,nointr,rsize=1048576,wsiz=1048576,vers=3,forcedirectio,proto=tcp,suid
/oracle/<SID>/mirrlogA	Every Oracle DB zone	rw,bg,hard,nointr,rsize=1048576,wsiz=1048576,vers=3,forcedirectio,proto=tcp,suid
/oracle/<SID>/mirrlogB	Every Oracle DB zone	rw,bg,hard,nointr,rsize=1048576,wsiz=1048576,vers=3,forcedirectio,proto=tcp,suid
/oracle/<SID>/oraarch	Every Oracle DB zone	rw,bg,hard,nointr,rsize=1048576,wsiz=1048576,vers=3,forcedirectio,proto=tcp,suid
/oracle/<SID>/sapreorg	Every Oracle DB zone	rw,bg,hard,nointr,rsize=1048576,wsiz=1048576,vers=3,forcedirectio,proto=tcp,suid
/oracle/<SID>/sapdata1	Every Oracle DB zone	rw,bg,hard,nointr,rsize=1048576,wsiz=1048576,vers=3,forcedirectio,proto=tcp,suid
/oracle/<SID>/sapdata2	Every Oracle DB zone	rw,bg,hard,nointr,rsize=1048576,wsiz=1048576,vers=3,forcedirectio,proto=tcp,suid
/oracle/<SID>/sapdata3	Every Oracle DB zone	rw,bg,hard,nointr,rsize=1048576,wsiz=1048576,vers=3,forcedirectio,proto=tcp,suid
/oracle/<SID>/sapdata4	Every Oracle DB zone	rw,bg,hard,nointr,rsize=1048576,wsiz=1048576,vers=3,forcedirectio,proto=tcp,suid
/oracle/<SID>/saparch	Every Oracle DB zone	rw,bg,hard,nointr,rsize=1048576,wsiz=1048576,vers=3,forcedirectio,proto=tcp,suid
/oracle/<SID>/oraflash	Every Oracle DB zone	rw,bg,hard,nointr,rsize=1048576,wsiz=1048576,vers=3,forcedirectio,proto=tcp,suid

/oracle/<SID>/sapcheck	Every Oracle DB zone	rw,bg,hard,nointr,rsize=1048576,wsiz=1048576,vers=3,forcedirectio,proto=tcp,suid
/oracle/<SID>/saptrace	Every Oracle DB zone	rw,bg,hard,nointr,rsize=1048576,wsiz=1048576,vers=3,forcedirectio,proto=tcp,suid
/oracle/<SID>/sapprof	Every Oracle DB zone	rw,bg,hard,nointr,rsize=1048576,wsiz=1048576,vers=3,forcedirectio,proto=tcp,suid
/oracle/<SID>/sapbackup	Every Oracle DB zone	rw,bg,hard,nointr,rsize=1048576,wsiz=1048576,vers=3,forcedirectio,proto=tcp,suid
/oracle/<SID>/cfgtoollogs	Every Oracle DB zone	rw,bg,hard,nointr,rsize=1048576,wsiz=1048576,vers=3,forcedirectio,proto=tcp,suid

For information about configuration of file systems with Oracle RAC, please refer to the Oracle document “Configuring an Oracle ZFS Storage Appliance for SAP systems with Oracle Database”.

For Oracle Database file systems with ASM, please refer to the corresponding document.

4. Create Solaris Cluster Resource Groups and Storage Resources

1. Create the scalable resource group for the file systems; Add resource of each file system to the resource group; Take the resource group online and enable the resources; Double check to make sure the file systems are mounted on all the required cluster nodes.

2. Create failover resource groups for each SAP and Oracle instances; Add logical host resource to each resource group accordingly; Take the resource group online and enable the resources.

For more information please refer to the [Oracle Solaris Cluster Data Services Planning and Administration Guide](#).

5. Install High-Availability SAP System on logical hosts

1. Make sure the logical hosts for each SAP instances are running on the installation node. Take the logical hosts to the status “not deprecated”.

```
clrs status -g scs-rg ers-rg
```

```
ifconfig -a
```



```
ifconfig <interface> -deprecated up
```

2. Create a temporary folder for sapinst log files of installing the (A)SCS instance. Switch to this folder and start the sapinst from this folder. From SAPINST GUI, Choose <SAP Product> -> Oracle -> SAP Systems -> <Product Type> -> High-Availability System -> (A)SCS Instance. Perform as described in the SAP installation guide to install the (A)SCS instance. If sapinst is stopped during the installation, you may start it from this folder again and the sapinst will continue the installation from where it stopped.

```
mkdir <ASCS log folder>
cd <ASCS log folder>
<SWPM folder>/sapinst SAPINST_USE_HOSTNAME=<ASCS logical host>
```

3. Create a temporary folder for sapinst log files of installing the ERS instance. Switch to this folder and start the sapinst from this folder. From SAPINST GUI, Choose <SAP Product> -> Oracle -> SAP Systems -> <Product Type> -> High-Availability System -> Enqueue Replication Server Instance. Perform as described in the SAP installation guide to install the ERS instance.


```
mkdir <ERS log folder>
cd <ERS log folder>
<SWPM folder>/sapinst SAPINST_USE_HOSTNAME=<ERS logical host>
```

4. Create a temporary folder for sapinst log files of installing the Oracle Database instance. Switch to this folder and start the sapinst from this folder. From SAPINST GUI, Choose <SAP Product> -> Oracle -> SAP Systems -> <Product Type> -> High-Availability System -> Database Instance. Perform as described in the SAP installation guide to install the Oracle Database instance.

```
mkdir <database log folder>
cd <database log folder>
<SWPM folder>/sapinst SAPINST_USE_HOSTNAME=<database logical host>
```

5. When there is a popup window during the step “Install database server”, if the Oracle Database Software has been preinstalled, set the link from the installation location (normally in subfolder of /u01) to the required location /oracle/<SID>/<release>. If there is no Oracle Software preinstalled, perform as described in the popup window to install the Oracle Database software. Click OK to continue the SAP installation.

6. Create a temporary folder for sapinst log files of installing the Primary Application Server Instance. Switch to this folder and start the sapinst from this folder. From SAPINST GUI, Choose <SAP Product> -> Oracle -> SAP Systems -> <Product Type> -> High-Availability System -> Primary



Application Server Instance (or Central Instance in older SAP release). Perform as described in the SAP installation guide to install the Primary Application Server Instance.

```
mkdir <PAS log folder>
cd <PAS log folder>
<SWPM folder>/sapinst SAPINST_USE_HOSTNAME=<PAS logical host>
```

7. Create a temporary folder for sapinst log files of installing the Additional Application Server Instance. Switch to this folder and start the sapinst from this folder. From SAPINST GUI, Choose <SAP Product> -> Oracle -> SAP Systems -> <Product Type> -> High-Availability System -> Additional Application Server Instance (or Dialog Instance in older SAP release). Perform as described in the SAP installation guide to install the Additional Application Server Instance.

```
mkdir <AAS log folder>
cd <AAS log folder>
<SWPM folder>/sapinst SAPINST_USE_HOSTNAME=<AAS logical host>
```

6. Take the Oracle Database Server and Client under Solaris Cluster Control

1. Stop the SAP system and the Oracle database, including the SAP sapstartsv processes, the Oracle listener and the SAP host agent.


2. Open a terminal on each cluster node for the database instance, logon as root user.

3. Since /oracle folder is shared on all cluster nodes, the subfolders and files on /oracle are identical on all the cluster nodes. But the files in /var/opt/oracle and /usr/local/bin are on local file system.. To make them identical, run the following scripts on the other node as well (These scripts have already run on the installation node during the Oracle installation.)

```
/oracle/<SID>/<release>/root.sh
/oracle/orainventory/orainstRoot.sh
```

4. Sync the following file from the installation node to the other cluster nodes. Replace the hostname with your cluster node host name.

```
rsync -avz /etc/inet/services root@epp1-db-02:/etc/inet
```



5. If the users home folders are on local file system, sync the user's home directory from the installation node to the other nodes. Replace the hostname with your cluster node host name.

```
rsync -avz /export/home/* root@epp1-db-02:/export/home
```

6. Switch the resource group db-rg to the second node. Replace the hostname with your cluster node host name. Replace the resource group name with your resource group name.

```
clrg switch -n epp1-db-02 db-rg
```

7. Logon as ora<sid> user and start the Oracle listener and the Oracle database server on the second node. Replace the hostname with your cluster node host name. Replace the Oracle user name with your Oracle user name.

```
epp1-db-02:oracle> lsnrctl start  
epp1-db-02:oracle> sqlplus / as sysdba  
SQL> startup
```


8. If everything works correctly, create a user to be used for Solaris Cluster fault monitoring according the document http://docs.oracle.com/cd/E56676_01/html/E56737/chdijbgb.html#scrolltoc .

```
SQL> create user user identified by password;  
SQL> alter user user default tablespace system quota 1m on system;  
SQL> grant select on v_$sysstat to user;  
SQL> grant select on v_$archive_dest to user;  
SQL> grant select on v_$database to user;  
SQL> grant create session to user;  
SQL> grant create table to user;  
SQL> create profile profile limit PASSWORD_LIFE_TIME UNLIMITED;  
SQL> alter user user identified by password profile profile;
```

9. If everything works correctly, stop the Oracle database server and the Oracle listener.

```
SQL> shutdown immediate  
SQL> exit  
epp1-db-02:oracle> lsnrctl stop
```

10. Repeat steps 6, 7 and 9 on every other cluster node for Oracle Database Server, to make sure the Oracle Database can be started and stopped successfully on each cluster node.



11. Logon as user root on one node, register the resource types for Oracle database server and Oracle listener.

```
clrt register oracle_server
clrt register oracle_listener
```

11. Create Oracle database server resource in the failover resource group for Oracle Database. Replace the parameter values with your parameter values.

```
clrs create -d -g db-rg -t oracle_server \
-p alert_log_file=/oracle/PP1/saptrace/diag/rdbms/pp1/PP1/trace/alert_PP1.log \
-p connect_string=Mary/Mary \
-p oracle_sid=PP1 \
-p oracle_home=/oracle/PP1/112_64 \
-p resource_dependencies_offline_restart=scal-oracle-rs \
oracle-rs
```

12. Create Oracle listener resource in the failover resource group for Oracle Database. Replace the parameter values with your parameter values.

```
clrs create -d -g db-rg -t oracle_listener \
-p listener_name=LISTENER \
-p oracle_home=/oracle/PP1/112_64 \
-p resource_dependencies_offline_restart=scal-oracle-rs \
listener-rs
```

13. Enable the Oracle database resources. Replace the resource names with your resource names.

```
clrs enable oracle-rs
clrs enable listener-rs
```

14. Check the Oracle Database resources status. Replace the resource group name with your resource group name.

```
clrs status -g db-rg
```

15. If everything works fine on the above node, switch the Oracle Database resource group to other nodes one by one for test purpose. Replace the hostname with your cluster node hostname. Replace the resource group name with your resource group name.

```
clrg switch -n epp1-db-01 db-rg
clrs status -g db-rg
```

7. Take the (A)SCS and ERS instances under Solaris Cluster control

1. Open one terminal for each cluster node for the (A)SCS and ERS instances, logon as user root. Make sure the SAP system, except the Oracle database, has been fully stopped.

2. Since `/usr/sap`, `/sapmnt/<SID>`, `/usr/sap/trans` and `/oracle/client` folders are shared on all required cluster nodes, the subfolders and files on these file systems are identical on the cluster nodes. But the SAP Host Agent in folder `/usr/sap/hostctrl` need to be moved to local file system of each cluster node. Perform the following commands to move the SAP Host Agent to local file system. Replace the hostname with your cluster hostname.

```
root@epp1-scs-01:~# mkdir -p /usr/local/sap
root@epp1-scs-02:~# mkdir -p /usr/local/sap

root@epp1-scs-01:~# mv /usr/sap/hostctrl /usr/local/sap/
root@epp1-scs-01:~# rsync -avz /usr/local/sap/hostctrl root@epp1-scs-02:/usr/local/sap
root@epp1-scs-01:~# ln -s /usr/local/sap/hostctrl /usr/sap/hostctrl
```

3. Synchronize the `/etc/inet/services` file from the installation node to the other cluster nodes. Replace the hostname with your cluster node hostname. If the (A)SCS and ERS instances share the same cluster (zones) with the Oracle Database, this step can be skipped since the file has been synchronized in the above section.

```
rsync -avz /etc/inet/services root@epp1-scs-02:/etc/inet
```

4. If the user home folders are on local file system, sync the user's home directory from the installation node to the other nodes. Replace the hostname with your cluster node hostname. If the (A)SCS and ERS instances share the same cluster (zones) with the Oracle Database, this step can be skipped since the folders/files have been synchronized in the above section.

```
rsync -avz /export/home/* root@epp1-scs-02:/export/home
```

5. Modify SAP instance profile of ASCS instance to avoid uncontrolled enqueue server process restarting on the same node. Change the Restart_Program to Start_Program for SAP enqueue server in the file /sapmnt/<SID>/profile/<SID>_<Instance>_<hostname> or START_<Instance>_<hostname>.

```
#-----  
# Start SAP enqueue server  
#-----  
_EN = en.sap$(SAPSYSTEMNAME)_$(INSTANCE_NAME)  
Execute_04 = local rm -f $_EN  
Execute_05 = local ln -s -f $(DIR_EXECUTABLE)/enserver$(FT_EXE) $_EN  
#Restart_Program_01 = local $_EN pf=$_PF  
Start_Program_01 = local $_EN pf=$_PF
```

6. On all the installation node(s), rename the file /etc/rc3.d/S90sapinit to avoid uncontrolled restarting of the sapstartsrv processes.

```
cd /etc/rc3.d  
mv S90sapinit s90sapinit.notneeded
```

7. Switch the resource groups of (A)SCS and ERS instances to the second node. Replace the hostname with your cluster node host name. Replace the resource group names with your resource group names.

```
clrg switch -n epp1-scs-02 scs-rg  
clrg switch -n epp1-scs-02 ers-rg
```

8. Logon the second node as <sid>adm, start the SAP instances SCS00 and ERS10. Make sure the instances can be started correctly. Replace the instance numbers with your SAP system instance numbers. Replace the SID with your SAP system SID.

```
sapcontrol -nr 00 -function StartService PP1  
sapcontrol -nr 10 -function StartService PP1  
sapcontrol -nr 00 -function Start  
sapcontrol -nr 10 -function Start  
sapcontrol -nr 00 -function GetProcessList  
sapcontrol -nr 10 -function GetProcessList
```

9. If everything works correctly, stop the SAP instances. Make sure all SAP processes have been stopped. Replace the instance numbers with your SAP system instance numbers. Replace the SID with your SAP system SID.

```
sapcontrol -nr 00 -function Stop
sapcontrol -nr 10 -function Stop
sapcontrol -nr 00 -function GetProcessList
sapcontrol -nr 10 -function GetProcessList
sapcontrol -nr 00 -function StopService
sapcontrol -nr 10 -function StopService
ps -ef | grep -i sap | grep -v cluster
```

10. (Optional) Enable SAP HA Connector to allow the Solaris Cluster resource can be started/stopped via SAP commands. If this is not enabled, the Solaris Cluster resources can only be managed with Solaris Cluster command. After enabling the SAP HA Connector, the resources can also be managed with sapcontrol command and SAP Management Console. To do so, add the following three lines to the instance profile of each Solaris Cluster controlled instance. If all the SAP instances are under Solaris Cluster control, you may also add the three lines in the DEFAULT.PFL file instead. Note the second parameter and its value should be entered in one line. Please note this is the SAP certified solution configuration and is recommend by SAP. Replace the <SID> with your SAP system SID.

```
#
# SAP HA Script Connector
#
service/halib = /usr/sap/<SID>/SYS/exe/run/saphascriptco.so
service/halib_cluster_connector =
/opt/ORCLscsapnetw/saphacmd/bin/sap_orcl_cluster_connector
service/halib_debug_level = 1
```

11. If SAP HA Connector is enabled, grant the cluster administration privilege to the <sid>adm user on all the clustered zones. Run the following command with user root.

```
usermod -A solaris.cluster.admin <sid>adm
```

12. Register the required Solaris Cluster resource types.

```
clrt register ORCL.sapstartsrv
clrt register ORCL.sapcentr
clrt register ORCL.saprepnq
clrt register ORCL.saprepnq_preempt
```

13. Create the resources for (A)SCS instance. Replace the parameter values with your parameter values.


```
clrs create -d -g scs-rg -t ORCL.sapstartsrv \  
-p SID=PP1 \  
-p sap_user=pp1adm \  
-p instance_number=00 \  
-p instance_name=SCS00 \  
-p host=epp1-scs-lh \  
-p child_mon_level=6 \  
-p resource_dependencies_offline_restart=scal-usr-sap-rs,scal-sapmnt-PP1-rs \  
-p timeout_return=20 \  
scs-startsrv-rs
```

```
clrs create -d -g scs-rg -t ORCL.sapcentr \  
-p SID=PP1 \  
-p sap_user=pp1adm \  
-p instance_number=00 \  
-p instance_name=SCS00 \  
-p host=epp1-scs-lh \  
-p retry_count=0 \  
-p resource_dependencies=scs-startsrv-rs \  
-p resource_dependencies_offline_restart=scal-usr-sap-rs,scal-sapmnt-PP1-rs \  
-p yellow=20 \  
scs-rs
```

14. Create the resources for ERS instance. Replace the parameter values with your parameter values.

```
clrs create -d -g ers-rg -t ORCL.sapstartsrv \  
-p SID=PP1 \  
-p sap_user=pp1adm \  
-p instance_number=10 \  
-p instance_name=ERS10 \  
-p host=epp1-ers-lh \  
-p child_mon_level=6 \  
-p resource_dependencies_offline_restart=scal-usr-sap-rs,scal-sapmnt-PP1-rs \  
-p timeout_return=20 \  
ers-startsrv-rs
```

```
clrs create -d -g ers-rg -t ORCL.saprepenq \  
-p sid=PP1 \  
-p sap_user=pp1adm \  

```

```
-p instance_number=10 \  
-p instance_name=ERS10 \  
-p host=epp1-ers-lh \  
-p resource_dependencies=ers-startsrv-rs \  
-p resource_dependencies_offline_restart=scal-usr-sap-rs,scal-sapmnt-PP1-rs \  
-p yellow=20 \  
ers-rs
```

15. Create the preempter resource in the resource group for (A)SCS instance. Replace the parameter values with your parameter values.

```
clrs create -d -g scs-rg -t ORCL.saprepenq_preempt \  
-p sid=PP1 \  
-p sap_user=pp1adm \  
-p repenqres=ers-rs \  
-p enq_instnr=00 \  
-p resource_dependencies_offline_restart=scs-rs \  
preempter-rs
```

16. Declare weak positive affinity of the scs-rg resource group to the ers-rg resource group. This is to ensure the scs-rg resource group fails over to the node where the ers-rg resource group is running. Replace the resource group names with your resource group names.


```
clrg set -p RG_affinities+=+ers-rg scs-rg  
clrg show -p RG_affinities scs-rg
```

17. Declare positive affinity of the instance resource groups to the storage resource group. Replace the resource group names with your resource group names.

```
clrg set -p RG_affinities+=+++scalmnt-rg ascs-rg  
clrg show -p RG_affinities ascs-rg  
clrg set -p RG_affinities+=+++scalmnt-rg rep-rg  
clrg show -p RG_affinities rep-rg
```

18. (Optional) Set pingpong_interval of the ascs-rg and ers-rg from default one hour to 10 minutes.

```
clrg set -p pingpong_interval=600 scs-rg  
clrg set -p pingpong_interval=600 ers-rg
```



19. Enable the newly created resources. The resources should be able to acknowledge the running instances and shows the status online. It will take a while before preempter switches the rep-rg to the other node. Replace the resource group names with your resource group names.

```
clrs enable +  
clrs status -g scs-rg,ers-rg
```

8. Take the Application Server Instances under Solaris Cluster Control

1. Open a terminal on each cluster node, logon as user root.

2. If PAS and AAS instances are on different Solaris (Zone) Cluster as the (A)SCS and ERS instances, perform the steps 2, 3, 4, and 6 in the above section for (A)SCS and ERS instances on the cluster nodes for PAS and AAS instances.

3. Switch the resource groups for PAS and AAS instances to the second node. Replace the hostname with your cluster node hostname. Replace the resource group names with your resource group names.


```
clrg switch -n epp1-haapps-02 pas-rg  
clrg switch -n epp1-haapps-02 aas-rg
```

4. Logon the second node as <sid>adm, start the PAS and AAS instances. Make sure the SAP instances have been started successfully.

```
sapcontrol -nr 01 -function StartService PP1  
sapcontrol -nr 02 -function StartService PP1  
sapcontrol -nr 01 -function Start  
sapcontrol -nr 02 -function Start  
sapcontrol -nr 01 -function GetProcessList  
sapcontrol -nr 02 -function GetProcessList
```

5. Stop the SAP instances. Make sure all SAP processes have been stopped successfully.

```
sapcontrol -nr 01 -function Stop  
sapcontrol -nr 02 -function Stop  
sapcontrol -nr 01 -function GetProcessList  
sapcontrol -nr 02 -function GetProcessList
```



```
sapcontrol -nr 01 -function StopService
sapcontrol -nr 02 -function StopService
ps -ef | grep -i sap | grep -v cluster
```

6. If there are more Solaris Cluster nodes, repeat the steps 3-5 to make sure the SAP instances can be started and stopped successfully on each of the cluster nodes.

7. (Optional) Enable SAP HA Connector to allow the Solaris Cluster resource can be started/stopped via SAP commands. If this is not enabled, the Solaris Cluster resources can only be managed with Solaris Cluster command. After enabling the SAP HA Connector, the resources can also be managed with sapcontrol command and SAP Management Console. To do so, add the following three lines to the instance profile of each Solaris Cluster controlled instance. If all the SAP instances are under Solaris Cluster control, you may also add the three lines in the DEFAULT.PFL file instead. Note the second parameter and its value should be entered in one line.

```
#
# SAP HA Script Connector
#
service/halib = /usr/sap/<SID>/SYS/exe/run/saphascriptco.so
service/halib_cluster_connector =
/opt/ORCLscsapnetw/saphacmd/bin/sap_orcl_cluster_connector
service/halib_debug_level = 1
```

8. If SAP HA Connector is enabled, grant the cluster administration privilege to the <sid>adm user Run the following command with user root on all the clustered zones. This step can be skipped if the PAS and AAS are on the same Solaris (Zone) Cluster as the (A)SCS and ERS instances.

```
usermod -A solaris.cluster.admin <sid>adm
```

9. Register the required Solaris Cluster resource types.

```
clrt register ORCL.sapstartsrv
clrt register ORCL.sapia
```

10. Create the resources for PAS instance. Replace the parameter values with your parameter values.

```
clrs create -d -g pas-rg -t ORCL.sapstartsrv \
-p SID=PP1 \
-p sap_user=pp1adm \
-p instance_number=01 \
```

```
-p instance_name=DVEBMGS01 \  
-p host=epp1-pas-lh \  
-p child_mon_level=6 \  
-p resource_dependencies_offline_restart=scal-usr-sap-rs,scal-sapmnt-PP1-rs \  
-p timeout_return=20 \  
pas-startsrv-rs
```

```
clrs create -d -g pas-rg -t ORCL.sapdia \  
-p SID=PP1 \  
-p sap_user=pp1adm \  
-p instance_number=01 \  
-p instance_name=DVEBMGS01 \  
-p host=epp1-pas-lh \  
-p resource_dependencies=pas-startsrv-rs,scal-oracle-client-rs \  
-p resource_dependencies_offline_restart=scal-usr-sap-rs,scal-sapmnt-PP1-rs \  
-p yellow=20 \  
pas-rs
```

11. Create the resources for AAS instance. Replace the parameter values with your parameter values.

```
clrs create -d -g aas01-rg -t ORCL.sapstartsrv \  
-p SID=PP1 \  
-p sap_user=pp1adm \  
-p instance_number=02 \  
-p instance_name=D02 \  
-p host=epp1-aas01-lh \  
-p child_mon_level=6 \  
-p resource_dependencies_offline_restart=scal-usr-sap-rs,scal-sapmnt-PP1-rs \  
-p timeout_return=20 \  
aas-startsrv-rs
```

```
clrs create -d -g aas-rg -t ORCL.sapdia \  
-p SID=PP1 \  
-p sap_user=pp1adm \  
-p instance_number=02 \  
-p instance_name=D02 \  
-p host=epp1-aas-lh \  
-p resource_dependencies=J01-startsrv-rs,scal-oracle-client-rs \  
-p resource_dependencies_offline_restart=scal-usr-sap-rs,scal-sapmnt-PP1-rs \  

```



```
-p yellow=20 \  
aas-rs
```

12. Declare positive affinity of the instance resource groups to the storage resource group.

```
clrg set -p RG_affinities+=++scalmnt-rg pas-rg  
clrg show -p RG_affinities pas-rg  
clrg set -p RG_affinities+=++scalmnt-rg aas-rg  
clrg show -p RG_affinities aas-rg
```

13. Enable the newly created resources. If the instances were running, the resources should be able to acknowledge the running instances and shows the status online.

```
clrs enable +  
clrs status
```

14. If the (A)SCS and ERS instance are on different Zone Clusters as the PAS and AAS instances, configure the cross zone resource dependencies from the global zone.

```
clzc list  
clrs set -Z pp1-haapps-zc -p resource_dependencies+=pp1-scs-zc:scs-rs pas-rs  
clrs set -Z pp1-haapps-zc -p resource_dependencies+=pp1-scs-zc:scs-rs aas-rs  
  
clrs show -p resource_dependencies pas-rs aas-rs
```

15. If the Oracle Database is running on different Solaris (Zone) Cluster as the PAS and AAS instances, configure the Oracle External Proxy and set the dependencies accordingly. More information about the Oracle External Proxy can be found at http://docs.oracle.com/cd/E56676_01/html/E52343/index.html







Oracle Corporation, World Headquarters

500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries

Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US

-  blogs.oracle.com/oracle
-  facebook.com/oracle
-  twitter.com/oracle
-  oracle.com

Integrated Cloud Applications & Platform Services

Copyright © 2016, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0116

White Paper Title
January 2016
Author: [OPTIONAL]
Contributing Authors: [OPTIONAL]