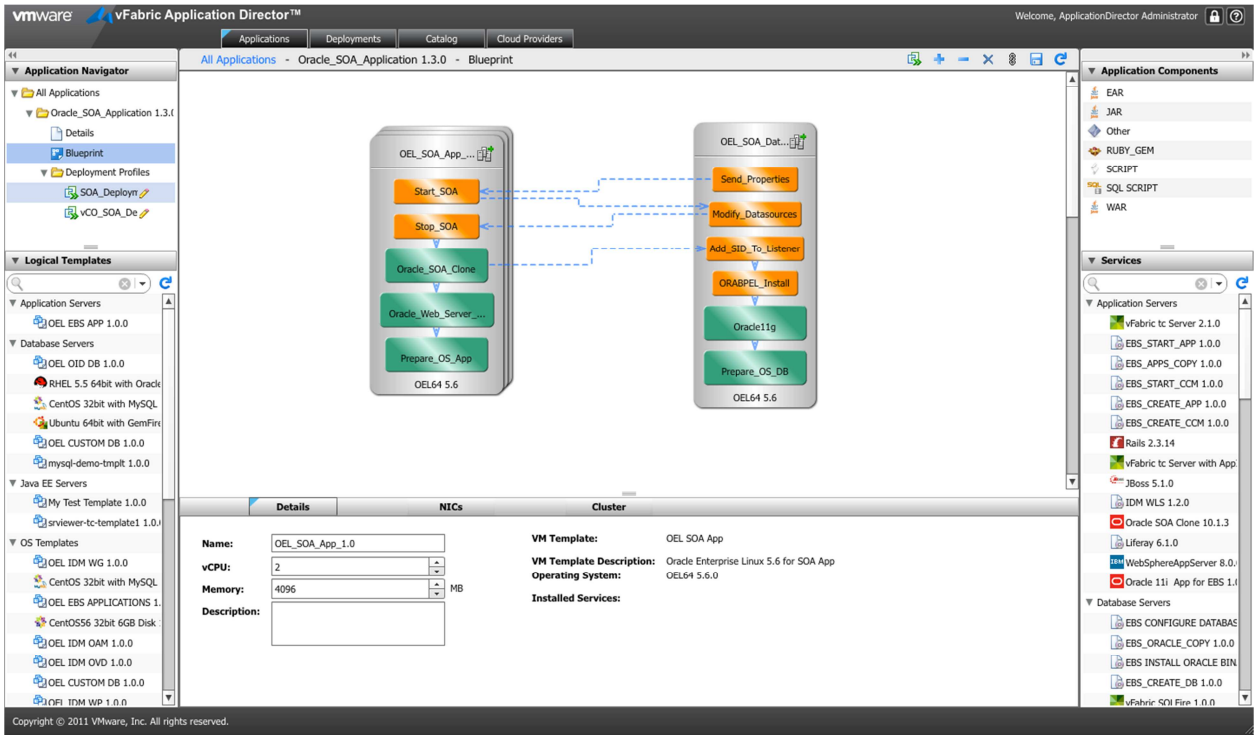
Oracle SOA Suite - Blueprint Information

1. OVERVIEW

The purpose of this document is to describe the Oracle SOA Suite blueprint in detail. The goal of the blueprint is to allow for provisioning and configuration of an Oracle SOA Suite instance in an automated and repeatable fashion.

The following components and versions are installed as part of the SOA Suite instance:

(1) Oracle SOA Suite 10g on the app tier
(2) Oracle Database 11.2.0.2 on the database tier
(3) JDK 1.6.0



2. HIGH LEVEL BLUEPRINT DESIGN

Conceptually, the blueprint design consists of the tasks described below. These tasks are listed in chronological order:

(1) Prepare OS (Database node & Application node)
(2) Install Oracle Web Server clone (on Application node)
(3) Install Oracle SOA Suite clone (on Application node)
(4) Install Oracle 11g (on database node)
(5) Install orabpel schema (on database node)

2.1 Files

You will need access to create the following file repository that contains the necessary installers and configuration files needed. We will refer to this repository as the Dropbox Home in this document.

$DROPBOX_HOME/SOACLONE:

clone-hostname-06-21-2012_1226.zip	Clone of SOA and OHS prepared from hostname
clone.properties	Clone.properties file for SOA clone
clone.properties.OHS	Clone.properties file for OHS clone
OHS-endpoint-map.xml	Endpoint map file for OHS clone
SOA-endpoint-map.xml	Endpoint map file for SOA clone
scripts/	Database scripts for orabpel schema
clone.xml	Ant-based main driver file for cloning
replaceSOATokens.xsl	XSLT script to replace configuration tokens
install-schemas.sh	Script to install orabpel schema in database
replaceToken.xsl	XSLT script to replace configuration tokens

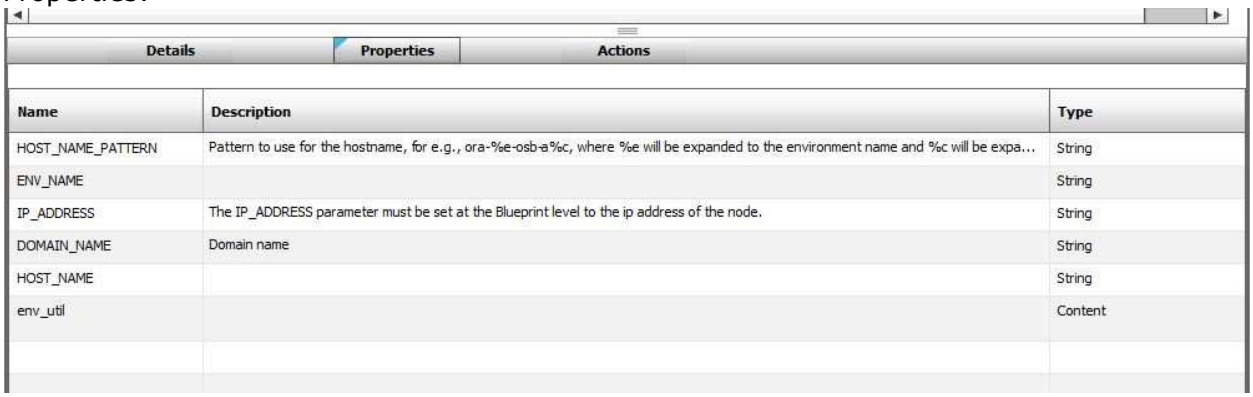
2.2 Prepare OS Properties

The Prepare OS task sets the hostname for the machine that's provisioned. The following properties govern the execution of the blueprint:

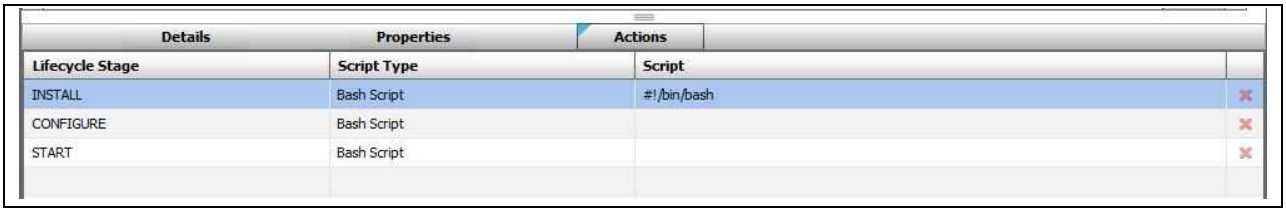
Property	**Description**	**Notes**
HOST_NAME_PATTERN	Set this to be a hostname pattern based on the environment (for e.g., dev1).	%e in the pattern will be substituted by the environment name
IP_ADDRESS	Set this property to the IP address of the machine	Self:NIC0_ip
DOMAIN_NAME	Set this property to be the domain name	
HOST_NAME	The eventual hostname will be set in this property. Therefore, you can use this property within other	

	modules in this blueprint	

Properties:



Actions:



Install Script:

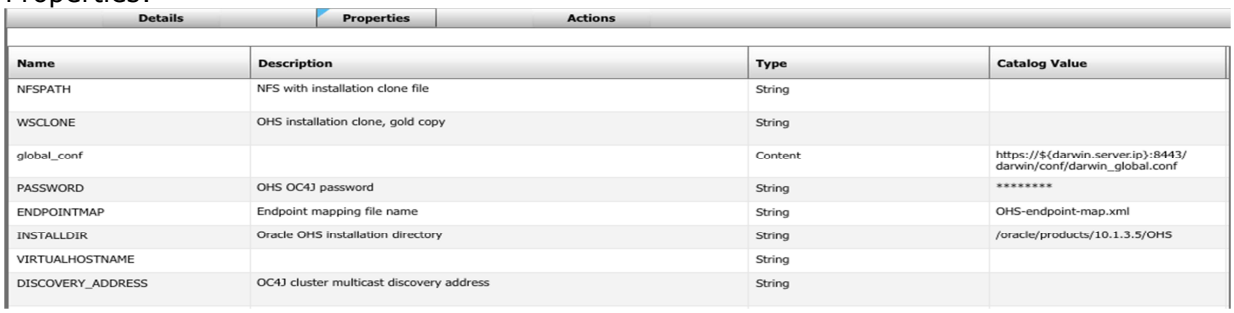
```
#!/bin/bash

# Set hostname variable

HOST_NAME=`echo $HOST_NAME_PATTERN | sed "s/%e/$ENV_NAME/g"`

# Change VM hostname

hostname $HOST_NAME

# Replace hostname in /etc/hosts

sed -i.bak "s/$IP_ADDRESS.*/$IP_ADDRESS $HOST_NAME.$DOMAIN_NAME $HOST_NAME/g" /etc/hosts

sed -i.bak "s/HOSTNAME=.*/HOSTNAME=$HOST_NAME.$DOMAIN_NAME/g" /etc/sysconfig/network

# set HOST_NAME INCLUDING DOMAIN

HOST_NAME="$HOST_NAME"."$DOMAIN_NAME"

echo "Setting hostname to $HOST_NAME"
```

2.3 Install Oracle Web Server (OHS) Clone

The Install Oracle Web Server (i.e. OHS) clone applies the cloning process to create an OHS clone. It's governed by the following properties:

Property	Description	Notes
NFSPATH	Location of the dropbox file repository	NFS mount where installation binaries & deployment scripts are located.
WSCLONE	OHS installation clone file	
PASSWORD	OC4J admin password for this installation	
ENDPOINTMAP	Endpoint mapping file for OHS cloning process	Used to configure endpoints in the environment
INSTALLDIR	Directory in which to install OHS	
VIRTUALHOSTNAME	Virtual hostname	
DISCOVERY_ADDRESS	OC4J cluster multicast address	

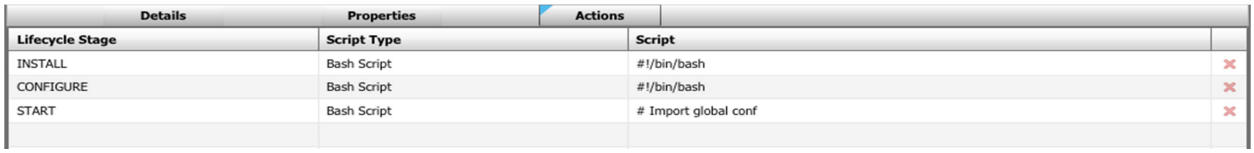
Properties:



Details | Properties | Actions

Name	Description	Type	Catalog Value
NFSPATH	NFS with installation clone file	String	
WSCLONE	OHS installation clone, gold copy	String	
global_conf		Content	https://${darwin.server.ip}:8443/darwin/conf/darwin_global.conf
PASSWORD	OHS OC4J password	String	********
ENDPOINTMAP	Endpoint mapping file name	String	OHS-endpoint-map.xml
INSTALLDIR	Oracle OHS installation directory	String	/oracle/products/10.1.3.5/OHS
VIRTUALHOSTNAME		String	
DISCOVERY_ADDRESS	OC4J cluster multicast discovery address	String	

Actions:



Details | Properties | Actions

Lifecycle Stage	Script Type	Script
INSTALL	Bash Script	#!/bin/bash
CONFIGURE	Bash Script	#!/bin/bash
START	Bash Script	# Import global conf

Install Script:

```
#!/bin/bash

export PATH=$PATH:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

set -e

MOUNTPOINTLOCATION=/tmp/mount

mkdir -p $MOUNTPOINTLOCATION

mount $NFSPATH $MOUNTPOINTLOCATION
```

```
mkdir /oracle/WSCLONE

cp $MOUNTPOINTLOCATION/SOACLONE/clone.xml /oracle/WSCLONE/

cp $MOUNTPOINTLOCATION/SOACLONE/clone.properties.OHS /oracle/WSCLONE/clone.properties

cp $MOUNTPOINTLOCATION/SOACLONE/OHS-endpoint-map.xml /oracle/WSCLONE/

cd /oracle/WSCLONE

echo "Run Oracle SOA Clone Installer"

$MOUNTPOINTLOCATION/apache-ant-1.8.3/bin/ant -f clone.xml -Dsrc.ohs.oc4jadmin.password=$PASSWORD

-Dsrc.ohs.oracle.home=$INSTALLDIR

-Dclone.zip.filename=$MOUNTPOINTLOCATION/SOACLONE/$WSCLONE apply
```

Configure Script:

```
#!/bin/bash

# Import global conf

. $global_conf

export PATH=$PATH:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

set -e

MOUNTPOINTLOCATION=/tmp/mount

cd /oracle/WSCLONE

echo "Run Oracle OHS Clone Configuration"

$MOUNTPOINTLOCATION/apache-ant-1.8.3/bin/ant -f clone.xml -Ddir=$INSTALLDIR -
Dfilename=$ENDPOINTMAP -Dsrc.lbr.virtual.hostname=$VIRTUALHOSTNAME -
Ddest.xmlupdate1.value=$DISCOVERY_ADDRESS apply-endpoints

chown -R oracle $INSTALLDIR

echo "Delete clone directory"

cd ..

rm -r -f WSCLONE

umount $MOUNTPOINTLOCATION
```

Start Script:

```
su - oracle -c "$INSTALLDIR/opmn/bin/opmnctl startall"
```

2.4 Install SOA Suite clone

The Install SOA Suite clone applies the clone and creates a SOA Suite instance. It's governed by the following properties:

Properties:

Details | Properties | Actions

Name	Description	Type	Catalog Value
NFSPATH	NFS path to clone gold image	String	
SOACLONE	SOA Clone intallation zip file, gold copy	String	
global_conf		Content	https://${darwin.server.ip}:8443/ darwin/conf/darwin_global.conf
PASSWORD	OC4J and orabpel password	String	********
DBSERVICE	Dehydration Store DB service name	String	ORCL
DBPORT	Dehydration store DB port	String	1521
DBHOST	Dehydration store DB host	String	localhost
INSTALLDIR	SOA Installation DIrectory	String	/oracle/products/10.1.3.5/SOA
ENDPOINTMAP		String	SOA-endpoint-map.xml
DBPASSWORD		String	********
DISCOVERY_ADDRESS	OC4J cluster multicast discovery address	String	

Actions:

Details | Properties | Actions

Lifecycle Stage	Script Type	Script
INSTALL	Bash Script	#!/bin/bash
CONFIGURE	Bash Script	#!/bin/bash
START	Bash Script	# Import global conf

Install Script:

```
MOUNTPOINTLOCATION=/tmp/mount

mkdir -p $MOUNTPOINTLOCATION

mount $NFSPATH $MOUNTPOINTLOCATION

mkdir /oracle/SOACLONE

cp $MOUNTPOINTLOCATION/SOACLONE/clone.xml /oracle/SOACLONE/

cp $MOUNTPOINTLOCATION/SOACLONE/clone.properties /oracle/SOACLONE/

cd /oracle/SOACLONE

$MOUNTPOINTLOCATION/apache-ant-1.8.3/bin/ant -f clone.xml  -Dsrc.db.hostname=$DBHOSTNAME -
Dsrc.db.port=$DBPORT -Dsrc.db.servicename=$DBSERVICE -Dsrc.db.orabpel.password=$DBPASSWORD -
Dsrc.soa.oc4jadmin.password=$PASSWORD -Dsrc.db.orabpel.password=$PASSWORD -
Dsrc.soa.oracle.home=$INSTALLDIR -
Dclone.zip.filename=$MOUNTPOINTLOCATION/SOACLONE/$SOACLONE apply
```

Configure Script:

```
cp $MOUNTPOINTLOCATION/SOACLONE/replaceSOATokens.xml /oracle/SOACLONE/



#Modify the build.properties file

echo "bpel.db.user=orabpel" >> /oracle/SOACLONE/build.properties

echo "bpel.db.password=$DBPASSWORD" >> /oracle/SOACLONE/build.properties

echo
"bpel.db.url=jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(LOAD_BALANCE=on)(ADDRESS=(PROTOCOL
=tcp)(HOST=$DBHOST)(PORT=$DBPORT)))(CONNECT_DATA=(SERVICE_NAME=$DBSERVICE)))" >>
/oracle/SOACLONE/build.properties

$MOUNTPOINTLOCATION/apache-ant-1.8.3/bin/ant -f replaceSOATokens.xml -propertyfile build.properties -
Dfilename=$ENDPOINTMAP -Dsoaclonelocation=$MOUNTPOINTLOCATION/SOACLONE

$MOUNTPOINTLOCATION/apache-ant-1.8.3/bin/ant -f clone.xml -Ddir=$INSTALLDIR -
Ddest.xmlupdate1.value=$DISCOVERY_ADDRESS -Dfilename=$ENDPOINTMAP apply-endpoints
```

Start Script:

```
#Start SOA

su - oracle -c "$INSTALLDIR/opmn/bin/opmnctl startall"
```

2.5 Install Oracle 11g

This task installs Oracle database 11g on the database node.

Properties:

Details | Properties | Actions

Name	Description	Type	Catalog Value
global_conf	URL to download Darwin global configuration for each node	Content	https://${darwin.server.ip}:8443/darwin/conf/darwin_global.conf
SELECTED_LANGUAGES	Select one of the following: en fr ar bn pt_BR bg fr_CA ca hr cs da nl ar_EG en_G...	String	en
ORACLE_BASE	Base directory to install files	String	/disk2/oracle
INSTALL_EDITION	Installation edition. The option can be one of: EE SE SEONE	String	EE
EMAIL_ADDRESS	Administrator email address	String	administrator@example.com
LISTENER_PROTOCOL	Listener protocal to use. Select one of: TCP TCPS NMP IPC VI	String	TCP
LISTENER_PORT	Port to bind to	String	1521
GDBNAME	Databse name	String	exampledb.us.oracle.com
SID	Database ID	String	exampledb
SYSTEMPASSWORD	Password for the system user	String	**********
SYSPASSWORD	Password for the sys user	String	**********
NFSPATH	NFS mount path to installation files. Both linux_11gR2_database_1of2.zip and lin...	String	
ORACLE_HOSTNAME	Hostname to bind to	String	localhost.localdomain

Actions:

Details | Properties | Actions

Lifecycle Stage	Script Type	Script	
INSTALL	Bash Script	#!/bin/bash	✖
CONFIGURE	Bash Script		✖
START	Bash Script		✖

Install Script:

```
#!/bin/bash

export
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/opt/vmware/bin:/opt/vmware/bin

# Tested on CentOS

if [ -x /usr/sbin/selinuxenabled ] && /usr/sbin/selinuxenabled; then

    if [ -x /usr/sbin/setenforce ]; then

        /usr/sbin/setenforce Permissive

    else

        echo 'SELinux is enabled. This may cause installation to fail.'

    fi

fi
```

```
#########make swapfile for installer##############
#Oracle installer will test for a swap size of 150M, this guarantees we have it
dd if=/dev/zero of=/tmp/swapfile bs=1024 count=150000
mkswap /tmp/swapfile
swapon /tmp/swapfile
#########PARAMETERS FROM APPLICATION DIRECTOR - DEFAULTS#########
#Putting central inventory in root partition
INVENTORY_LOCATION="$ORACLE_BASE/product/11.2.0/oraInventory"
#Defining home in ORACLE_BASE to avoid problems
ORACLE_HOME="$ORACLE_BASE/product/11.2.0/db_1"
#########SCRIPT INTERNAL PARAMETERS#########
MOUNTPOINTLOCATION=/tmp/mount
ORCALEINSTALLERLOCATION="$ORACLE_BASE/orclinstaller"
SYSCTLCONF="/etc/sysctl.conf"
ORACLEINSTALLSCRIPT="$ORCALEINSTALLERLOCATION/orclscript_runinstaller.sh"
ORACLENETCACONFIGURATIONSCRIPT="$ORCALEINSTALLERLOCATION/orclscript_netca_configuration.sh
"
ORACLEDBCACONFIGURATIONSCRIPT="$ORCALEINSTALLERLOCATION/orclscript_dbca_configuration.sh"
#Response File location
DBFILE="$ORCALEINSTALLERLOCATION/database/response/db_install.rsp"
NETCAFILE="$ORCALEINSTALLERLOCATION/database/response/netca.rsp"
DBCAFILE="$ORCALEINSTALLERLOCATION/database/response/dbca.rsp"
#Response File Backup location
DBFILE_BACKUP="$ORCALEINSTALLERLOCATION/database/response/db_install-original.rsp"
NETCAFILE_BACKUP="$ORCALEINSTALLERLOCATION/database/response/netca-original.rsp"
DBCAFILE_BACKUP="$ORCALEINSTALLERLOCATION/database/response/dbca-original.rsp"
PROGNAME=`basename $0`


#Array
```

```
INSTALL_EDITION_ARRAY=(EE SE SEONE)

LISTENER_PROTOCOL_ARRAY=(TCP TCPS NMP IPC VI)

LANG_ARRAY=(en fr ar bn pt_BR bg fr_CA ca hr cs da nl ar_EG en_GB et fi de el iw hu is in it ja ko es lv lt ms
es_MX no pl pt ro ru zh_CN sk sl es_ES sv th zh_TW tr uk vi)

###########Parameter Validation Functions##################

## Function To Display Error and Exit

function check_error()

{

  if [ ! "$?" = "0" ]; then

    error_exit "$1";

  fi

}



function error_exit()

{

  echo "${PROGNAME}: ${1:-"Unknown Error"}" 1>&2

  exit 1

}



## Function To Validate Integer

function valid_int()

{

  local data=$1

  if [[ $data =~ ^[0-9]{1,9}$ ]]; then

    return 0;

  else

    return 1

  fi

}
```

```
function check_install_edition()
{
  len=${#INSTALL_EDITION_ARRAY[*]}
  local temp=$1;
  for (( i=0; $i < $len; i++ )) do
    if [ "${INSTALL_EDITION_ARRAY[$i]}" = "$temp" ]; then
      return;
    fi
  done
  error_exit "Invalid Install Edition"
}

function check_listener_protocol()
{
  len=${#LISTENER_PROTOCOL_ARRAY[*]}
  local temp=$1;
  for (( i=0; $i < $len; i++ )) do
    if [ "${LISTENER_PROTOCOL_ARRAY[$i]}" = "$temp" ]; then
      return;
    fi
  done
  error_exit "Invalid Listener Protocol"
}

function check_lang()
{
  len=${#LANG_ARRAY[*]}
```

```
  local temp=$1;
  for (( i=0; $i < $len; i++ )) do
    if [ "${LANG_ARRAY[$i]}" = "$temp" ]; then
      return;
    fi
  done
  error_exit "Invalid Language"
}

function check_email()
{
  local email=$1;
  case $email in
  *@?*.?*) ;;
  *) error_exit "Invalid EMAIL ID"; false ;;
  esac
}

function echo_d()
{
   CURDATE=`date +%H:%M:%S`
   echo -e $CURDATE  "$*"
}



####################SCRIPT EXECUTION ##################
echo_d "Parameter Validation"
```

```
if [ "x${ORACLE_HOSTNAME}" = "x" ]; then
   error_exit "ORACLE_HOSTNAME not set."
fi

if [ "x${INVENTORY_LOCATION}" = "x" ]; then
   error_exit "INVENTORY_LOCATION not set."
fi

if [ "x${SELECTED_LANGUAGES}" = "x" ]; then
   error_exit "SELECTED_LANGUAGES not set."
else
  check_lang $SELECTED_LANGUAGES
fi

if [ "x${ORACLE_HOME}" = "x" ]; then
   error_exit "ORACLE_HOME not set."
fi

if [ "x${ORACLE_BASE}" = "x" ]; then
   error_exit "ORACLE_BASE not set."
fi

if [ "x${INSTALL_EDITION}" = "x" ]; then
   error_exit "INSTALL_EDITION not set."
else
   check_install_edition $INSTALL_EDITION;
fi
```

```
if [ "x${EMAIL_ADDRESS}" = "x" ]; then
    error_exit "EMAIL_ADDRESS not set."
else
   check_email $EMAIL_ADDRESS;
fi

if [ "x${LISTENER_PROTOCOL}" = "x" ]; then
    error_exit "LISTENER_PROTOCOL not set."
else
   check_listener_protocol $LISTENER_PROTOCOL
fi

if [ "x${LISTENER_PORT}" = "x" ]; then
    error_exit "LISTENER_PORT not set."
else
   if ! valid_int $LISTENER_PORT; then
      error_exit "Invalid parameter LISTENER_PORT"
   fi
fi

if [ "x${GDBNAME}" = "x" ]; then
    error_exit "GDBNAME not set."
fi

if [ "x${SID}" = "x" ]; then
    error_exit "SID not set."
fi
```

```
if [ "x${SYSTEMPASSWORD}" = "x" ]; then
   error_exit "SYSTEMPASSWORD not set."
fi

if [ "x${SYSPASSWORD}" = "x" ]; then
   error_exit "SYSPASSWORD not set."
fi

if [ "x${NFSPATH}" = "x" ]; then
   error_exit "NFSPATH not set."
fi

echo_d "Parameter Validation -- DONE"

basearch=`uname -p`

echo_d "Updating pre-requiste packages"

#yum --nogpgcheck --noplugins -y update make compat-db gcc gcc-c++ libstdc++ pdksh sysstat compat-libstdc++-33 elfutils-libelf-devel elfutils-libelf-devel-static unixODBC unixODBC-devel libaio-devel nfs-utils unzip

#yum --nogpgcheck --noplugins -y install make compat-db gcc gcc-c++ libstdc++ pdksh sysstat compat-libstdc++-33 elfutils-libelf-devel elfutils-libelf-devel-static unixODBC unixODBC-devel libaio-devel nfs-utils unzip

YUM_COMMONS="unzip"

YUM_ORACLE_DEPS="binutils compat-db elfutils-libelf-devel-static compat-libstdc++-33 compat-libstdc++ elfutils-libelf elfutils-libelf-devel gcc gcc-c++ glibc glibc glibc-common glibc-devel glibc-devel libaio libaio-devel libgcc libgcc libstdc++ lbXtst libstdc++ libstdc++-devel make numactl-devel sysstat nfs-utils pdksh unixODBC unixODBC-devel"

YUM_ORACLE_DEPS+=" glibc.i686"

yum --nogpgcheck -y install $YUM_COMMONS

yum --nogpgcheck -y install $YUM_ORACLE_DEPS
```

```
echo_d "Updating pre-requiste packages -- DONE"


mkdir -p $MOUNTPOINTLOCATION

mkdir -p $ORCALEINSTALLERLOCATION


echo_d "Mounting Oracle Installer"

#Mounting the NFS Drive

DIST=`cat /etc/redhat-release |sed s/\ release.*//`

if [ "$DIST" = "CentOS" ] ; then

   /sbin/service portmap start

else

   /sbin/service rpcbind start

fi

mount $NFSPATH $MOUNTPOINTLOCATION

check_error "Errors during mounting oracle installer.";

echo_d "Mounting Oracle Installer -- DONE"


# Copy Oracle 11g Installer

echo_d "Copying Oracle Installer"

if [ $basearch == "i686" ] ; then

  cp $MOUNTPOINTLOCATION/linux_11gR2_database_1of2.zip
$ORCALEINSTALLERLOCATION/linux_11gR2_database_1of2.zip

  check_error "Errors during copying zip 1of2 of Oracle installer.";

  cp $MOUNTPOINTLOCATION/linux_11gR2_database_2of2.zip
$ORCALEINSTALLERLOCATION/linux_11gR2_database_2of2.zip

  check_error "Errors during copying zip 2of2 of Oracle installer.";

else
```

```
  cp $MOUNTPOINTLOCATION/linux.x64_11gR2_database_1of2.zip
$ORCALEINSTALLERLOCATION/linux_11gR2_database_1of2.zip

  check_error "Errors during copying zip 1of2 of Oracle installer.";

  cp $MOUNTPOINTLOCATION/linux.x64_11gR2_database_2of2.zip
$ORCALEINSTALLERLOCATION/linux_11gR2_database_2of2.zip

  check_error "Errors during copying zip 2of2 of Oracle installer.";

fi

echo_d "Copying Oracle Installer -- DONE"


echo_d "Extracting Oracle Installer"

unzip -q $ORCALEINSTALLERLOCATION/linux_11gR2_database_1of2.zip -d $ORCALEINSTALLERLOCATION

check_error "Errors during extracting 1of2 of Oracle installer.";

unzip -q $ORCALEINSTALLERLOCATION/linux_11gR2_database_2of2.zip -d $ORCALEINSTALLERLOCATION

check_error "Errors during extracting 2of2 of Oracle installer.";

echo_d "Extracting Oracle Installer -- DONE"


echo_d "Modifying the response files"

#Creating backup of property files

cp --backup --force -- $DBFILE $DBFILE_BACKUP

cp --backup --force -- $NETCAFILE $NETCAFILE_BACKUP

cp --backup --force -- $DBCAFILE $DBCAFILE_BACKUP


#Changing the parameter in db.rsp

sed -i "s~ORACLE_HOSTNAME=*~ORACLE_HOSTNAME=$ORACLE_HOSTNAME~g" $DBFILE

sed -i "s~INVENTORY_LOCATION=*~INVENTORY_LOCATION=$INVENTORY_LOCATION~g" $DBFILE

sed -i "s~SELECTED_LANGUAGES=*~SELECTED_LANGUAGES=$SELECTED_LANGUAGES~g" $DBFILE

sed -i "s~ORACLE_HOME=*~ORACLE_HOME=$ORACLE_HOME~g" $DBFILE

sed -i "s~ORACLE_BASE=*~ORACLE_BASE=$ORACLE_BASE~g" $DBFILE

sed -i "s~oracle.install.db.InstallEdition=*~oracle.install.db.InstallEdition=$INSTALL_EDITION~g" $DBFILE
```

```
sed -i
"s~oracle.install.db.config.starterdb.dbcontrol.emailAddress=*~oracle.install.db.config.starterdb.dbcontrol.emailAddress=$EMAIL_ADDRESS~g" $DBFILE

sed -i
"s~oracle.install.db.config.starterdb.dbcontrol.enableEmailNotification=.*~oracle.install.db.config.starterdb.dbcontrol.enableEmailNotification=true~g" $DBFILE

sed -i "s~oracle.install.option=*~oracle.install.option=INSTALL_DB_SWONLY~g" $DBFILE

sed -i "s~UNIX_GROUP_NAME=*~UNIX_GROUP_NAME=oinstall~g" $DBFILE

sed -i "s~oracle.install.db.DBA_GROUP=*~oracle.install.db.DBA_GROUP=dba~g" $DBFILE

sed -i "s~oracle.install.db.OPER_GROUP=*~oracle.install.db.OPER_GROUP=oinstall~g" $DBFILE

sed -i "s~oracle.install.db.config.starterdb.type=*~oracle.install.db.config.starterdb.type=GENERAL_PURPOSE~g"
$DBFILE

sed -i
"s~SECURITY_UPDATES_VIA_MYORACLESUPPORT=*~SECURITY_UPDATES_VIA_MYORACLESUPPORT=false~g" $DBFILE

sed -i "s~DECLINE_SECURITY_UPDATES=*~DECLINE_SECURITY_UPDATES=true~g" $DBFILE



#Changing the parameter in netca.rsp

LISTENERPROTOCOL="{\"$LISTENER_PROTOCOL;$LISTENER_PORT\"}"

sed -i "s~LISTENER_PROTOCOLS=*~#LISTENER_PROTOCOLS=~g" $NETCAFILE

echo "LISTENER_PROTOCOLS = $LISTENERPROTOCOL" >> $NETCAFILE



#Changing the parameter in dbca.rsp

sed -i "s~GDBNAME=*~#GDBNAME~g" $DBCAFILE

sed -i "s~SID=*~#SID~g" $DBCAFILE

sed -i "150iGDBNAME = $GDBNAME" $DBCAFILE

sed -i "150iSID = $SID" $DBCAFILE

sed -i "150iSYSTEMPASSWORD = $SYSTEMPASSWORD" $DBCAFILE

sed -i "150iSYSPASSWORD = $SYSPASSWORD" $DBCAFILE

echo_d "Modifying the response files -- DONE"
```

```
#Setup Standard Users Ands Groups

echo_d "Adding Oracle User"

#/usr/sbin/groupdel oinstall

#/usr/sbin/groupdel dba

#/usr/sbin/userdel oracle


/usr/sbin/groupadd oinstall

/usr/sbin/groupadd dba

/usr/sbin/useradd -m -g oinstall -G dba oracle

#check_error "Errors during setting up user accounts.";

echo_d "Adding Oracle User -- DONE"


echo_d "Setting Kernel Parameters"

#Set Kernel Parameters

echo "kernel.shmmni=4096" >> $SYSCTLCONF

echo "kernel.sem=250 32000 100 128" >> $SYSCTLCONF

echo "fs.file-max=6815744" >> $SYSCTLCONF

echo "net.ipv4.ip_local_port_range=9000 65500" >> $SYSCTLCONF

echo "net.core.rmem_default=262144" >> $SYSCTLCONF

echo "net.core.wmem_default=262144" >> $SYSCTLCONF

echo "net.core.rmem_max=4194304" >> $SYSCTLCONF

echo "net.core.wmem_max=1048576" >> $SYSCTLCONF

echo "fs.aio-max-nr=1048576" >> $SYSCTLCONF


# Upload Changed Kernel Parameters

/sbin/sysctl -p

echo_d "Setting Kernel Parameters -- DONE"
```

```
#Basic Directory Structure

mkdir -p $ORACLE_BASE

mkdir -p $ORACLE_HOME

mkdir -p $INVENTORY_LOCATION

chown -R oracle:oinstall $ORACLE_BASE

chown -R oracle:oinstall $ORACLE_HOME

chown -R oracle:oinstall $INVENTORY_LOCATION

chmod -R 775 $ORACLE_BASE

chmod -R 775 $ORACLE_HOME

chmod -R 775 $INVENTORY_LOCATION

check_error "Errors during creating basic directory structure.";


#Modifying the shell limit and bash profile

echo_d "Modifying the shell limit and bash profile"

echo "oracle soft nproc 2047" >> /etc/security/limits.conf

echo "oracle hard nproc 16384" >> /etc/security/limits.conf

echo "oracle soft nofile 1024" >> /etc/security/limits.conf

echo "oracle hard nofile 65536" >> /etc/security/limits.conf

echo "Modifying the shell limit -- DONE"


#Modifying pam.d login

echo "session required pam_limits.so" >> /etc/pam.d/login


#Modify the bash profile oracle user

echo "export ORACLE_HOSTNAME=$ORACLE_HOSTNAME" >> /home/oracle/.bash_profile

echo "export ORACLE_UNQNAME=$SID" >> /home/oracle/.bash_profile

echo "export ORACLE_BASE=$ORACLE_BASE" >> /home/oracle/.bash_profile

echo "export ORACLE_HOME=$ORACLE_HOME" >> /home/oracle/.bash_profile
```

```
echo "export ORACLE_SID=$SID" >> /home/oracle/.bash_profile
echo "export TNS_ADMIN=$ORACLE_HOME/network/admin" >> /home/oracle/.bash_profile
echo "export PATH=/usr/sbin:$ORACLE_HOME/bin:$PATH" >> /home/oracle/.bash_profile
echo "export LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib:/usr/lib64" >> /home/oracle/.bash_profile
echo "export CLASSPATH=$ORACLE_HOME/jlib:$ORACLE_HOME/rdbms/jlib" >> /home/oracle/.bash_profile
echo_d "Modifying the shell limit and bash profile -- DONE"

# RUN INSTALLER
touch $ORACLEINSTALLSCRIPT
chmod 777 $ORACLEINSTALLSCRIPT
echo "#!/bin/bash" >> $ORACLEINSTALLSCRIPT
echo "cd $ORCALEINSTALLERLOCATION/database" >> $ORACLEINSTALLSCRIPT
echo "./runInstaller -silent -force -ignorePrereq -responseFile $DBFILE" >> $ORACLEINSTALLSCRIPT
echo "exit \"$?\"" >> $ORACLEINSTALLSCRIPT

touch $ORACLENETCACONFIGURATIONSCRIPT
chmod 777 $ORACLENETCACONFIGURATIONSCRIPT
echo "#!/bin/bash" >> $ORACLENETCACONFIGURATIONSCRIPT
echo "cd $ORACLE_HOME/bin/" >> $ORACLENETCACONFIGURATIONSCRIPT
echo "./netca -silent -responseFile $NETCAFILE" >> $ORACLENETCACONFIGURATIONSCRIPT
echo "exit \"$?\"" >> $ORACLENETCACONFIGURATIONSCRIPT

touch $ORACLEDBCACONFIGURATIONSCRIPT
chmod 777 $ORACLEDBCACONFIGURATIONSCRIPT
echo "#!/bin/bash" >> $ORACLEDBCACONFIGURATIONSCRIPT
echo "cd $ORACLE_HOME/bin/" >> $ORACLEDBCACONFIGURATIONSCRIPT
echo "./dbca -silent -responseFile $DBCAFILE" >> $ORACLEDBCACONFIGURATIONSCRIPT
echo "exit \"$?\"" >> $ORACLEDBCACONFIGURATIONSCRIPT
```

```
#Switch Oracle User
echo_d "Starting Oracle Installer"
#xhost +
su - oracle -c $ORACLEINSTALLSCRIPT
check_error "Unsuccessfull Database Installation";

echo ""
echo_d "Wait 3 minutes ..."
sleep 180

for (( ; ; ))
do
  grep -q "Shutdown Oracle Database 11g Release 2 Installer" $INVENTORY_LOCATION/logs/installActions*.log
  if [ $? -eq 0 ]
  then
     grep -q "Exit Status is 0" $INVENTORY_LOCATION/logs/installActions*.log
     if [ $? -eq 0 ]
     then
       export DISPLAY=:0
       echo_d "*** Success: Oracle Installed - running Post-Install scripts ..."
       $INVENTORY_LOCATION/orainstRoot.sh
       $ORACLE_HOME/root.sh
       echo_d "Configuring NetCA and creating starter DB"
       su - oracle -c $ORACLENETCACONFIGURATIONSCRIPT
       check_error "Unsuccessfull NETCA Configuration";
       su - oracle -c $ORACLEDBCACONFIGURATIONSCRIPT
       check_error "Unsuccessfull DBCA Configuration";
```

```
        echo_d "Configuring Oracle -- DONE"
        break
      else
        echo_d "*** Error: Oracle Not Installed"
        exit 1
      fi
  else
    sleep 30
  fi
done

sleep 10
echo_d "Checking starter DB connectivity"
SQLPLUS_TESTSCRIPT="/home/oracle/sqlplus.sh"

# The number 2718281828459045 should never occur otherwise in the sqlplus output
echo "#!/bin/bash" >> $SQLPLUS_TESTSCRIPT
echo "$ORACLE_HOME/bin/sqlplus / as sysdba <<EOF" >> $SQLPLUS_TESTSCRIPT
echo "select 2718281828459045 from dual;" >> $SQLPLUS_TESTSCRIPT
echo "EOF" >> $SQLPLUS_TESTSCRIPT
chmod +x $SQLPLUS_TESTSCRIPT

# Check that we are finally able to connect to the Oracle service (ORA-01034 err code for ORCL service unavailable)
CMD_OUTPUT_SUC=`su - oracle -c /home/oracle/sqlplus.sh |grep 2718281828459045|wc -l`
CMD_OUTPUT_FAIL=`su - oracle -c /home/oracle/sqlplus.sh |grep ORA-01034|wc -l`
if [ $CMD_OUTPUT_SUC == "1" ]; then
  if [ $CMD_OUTPUT_FAIL = "1" ]; then
    echo_d "*** Error: Unable to connect to $SID"
```

```
    exit 1
  else
    echo_d "*** Success: Connected to $SID"
  fi
else
  echo_d "*** Error: Unable to connect to $SID"
  exit 1
fi


rm $ORACLEINSTALLSCRIPT
rm $ORACLENETCACONFIGURATIONSCRIPT
rm $ORACLEDBCACONFIGURATIONSCRIPT
rm $SQLPLUS_TESTSCRIPT
```

2.6 Install Orabpel Schema

This task installs the orabpel schema necessary for the execution of SOA Suite. The following properties govern the execution of the blueprint:

Properties:

Details | Properties | Actions

Name	Description	Type	Blueprint Value
GDBNAME		String	self:Oracle11g:GDBNAME
ORACLE_BASE		String	self:Oracle11g:ORACLE_BASE
global_conf		Content	self:Oracle11g:global_conf
SID		String	self:Oracle11g:SID
BPEL_PASSWORD		String	*******
BPEL_ACCOUNT		String	orabpel
SYSPASSWORD		String	self:Oracle11g:SYSPASSWORD
PORT		String	self:Oracle11g:LISTENER_PORT
HOSTNAME		String	self:Oracle11g:ORACLE_HOSTNAME

Actions:

Details | Properties | Actions

Lifecycle Stage	Script Type	Script
INSTALL	Bash Script	#!/bin/bash
CONFIGURE	Bash Script	
START	Bash Script	

Install Script:

```
#!/bin/bash

SOADBSCRIPT="/tmp/mount/SOACLONE/install-schemas.sh"

su oracle -c $SOADBSCRIPT
```