10

Advanced Administration Topics

An enormous amount of material was covered in the previous nine chapters. In essence, each of those chapters can be referenced out of order depending on the area you are focusing on. The same applies to this chapter. Here we cover several advanced, disjoined topics, but all ultimately under the fold of Oracle SOA Suite 11*g* administration.

This chapter covers the following advanced administration topics:

- Patching Oracle SOA Suite Components
- Upgrading from Oracle SOA Suite 10*g*
- Installing a highly available clustered infrastructure
- Silent/scripted installation

Patching is an activity that all administrators will certainly be involved in. Often, any of the software components used within the Oracle SOA Suite 11*g* infrastructure may need to be patched (or upgraded) to address vulnerabilities within them, fix product bugs, or simply to continue running on a supported version. This chapter covers a methodology that can be adopted for patching and upgrading all components within your environment and is meant to help understand the few postpatch and postupgrade activities that are needed to be performed.

Upgrading from Oracle SOA Suite 10*g* to 11*g* is a tremendous effort involving code migration and infrastructure upgrade, as well as cutover planning and execution. We describe various considerations and explain what needs to be done from an administration standpoint during the upgrade effort.

We cover, in painstaking detail how to set up a 2-node cluster to enable Oracle SOA Suite 11*g* to run in a highly available mode. Thus, if one node experiences a crash, there is no impact on transactional activity as the other node remains available. All sessions and in-flight transactions are replicated across all nodes of the cluster transparently and are handled by the infrastructure. Furthermore, by adding additional nodes, the overall capacity of the infrastructure is increased, making it capable of handling larger loads as well. Architecting a 2-node cluster can take many shapes and forms, and we offer one approach that leverages shared binaries (where multiple domains share the same installed components).

Our last topic describes how to perform silent installations using scripts, which is the ability to install the entire Oracle SOA Suite 11*g* stack without manual intervention or going through a GUI based installer. This is very helpful as you have a set of common scripts that can instantaneously provision an environment as well as have the ability to create an exact same environment across multiple sites.

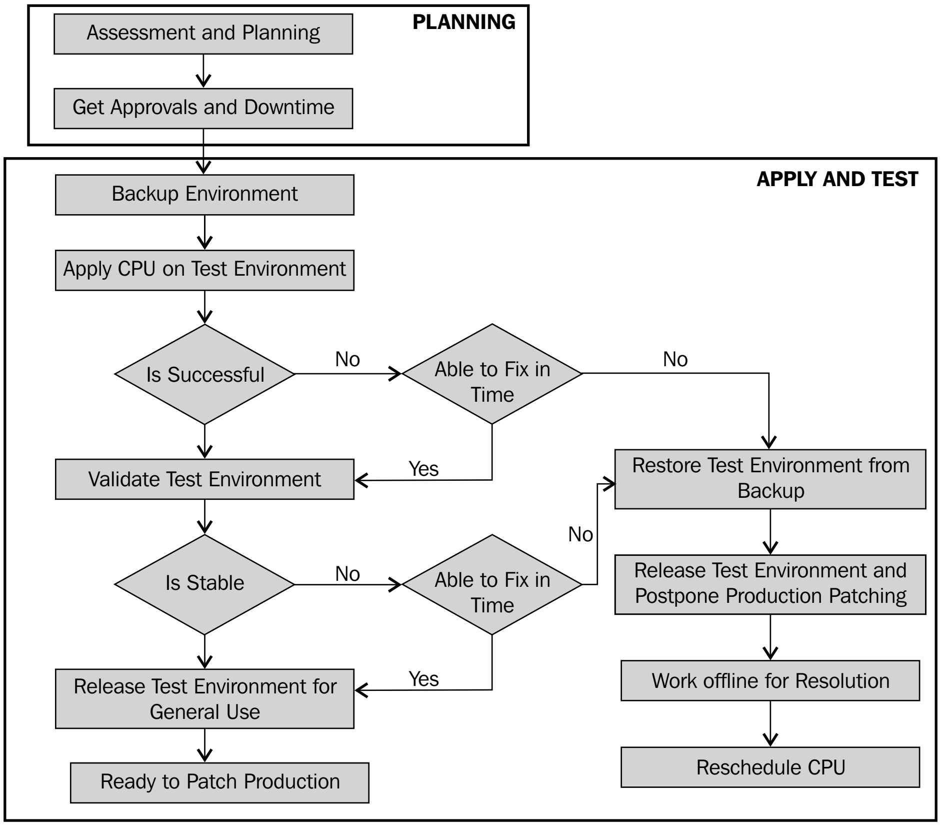
Patching Oracle SOA Suite components

As an administrator of the SOA infrastructure, you need to ensure business continuity and plug all known vulnerabilities in the infrastructure by having a comprehensive strategy of both proactive and reactive patching. **Proactive patching** involves aligning the SOA platform patching with **Oracle Critical Patch Updates** (**CPU**) and product release cycles. **Reactive patching** is installing one-off patches that address product specific bugs discovered during development, testing, or in production, or if the actual behavior of the product is not consistent with what is documented in the product documentation library. Patching production environments is sometimes challenging as there may be a very small window to do so, if any at all. There is always a risk associated with patching, albeit a small one, that may result in destabilization post patching. It thus becomes absolutely important that the process of patching follows a set mechanism in order to be driven to closure. We discuss a patching strategy that you can refer to, build upon, and have in place across the infrastructure that you maintain.

It is important to know that any patch implementation typically consists of three main phases:

- Assessment and planning
- Test deployment and testing
- Production deployment

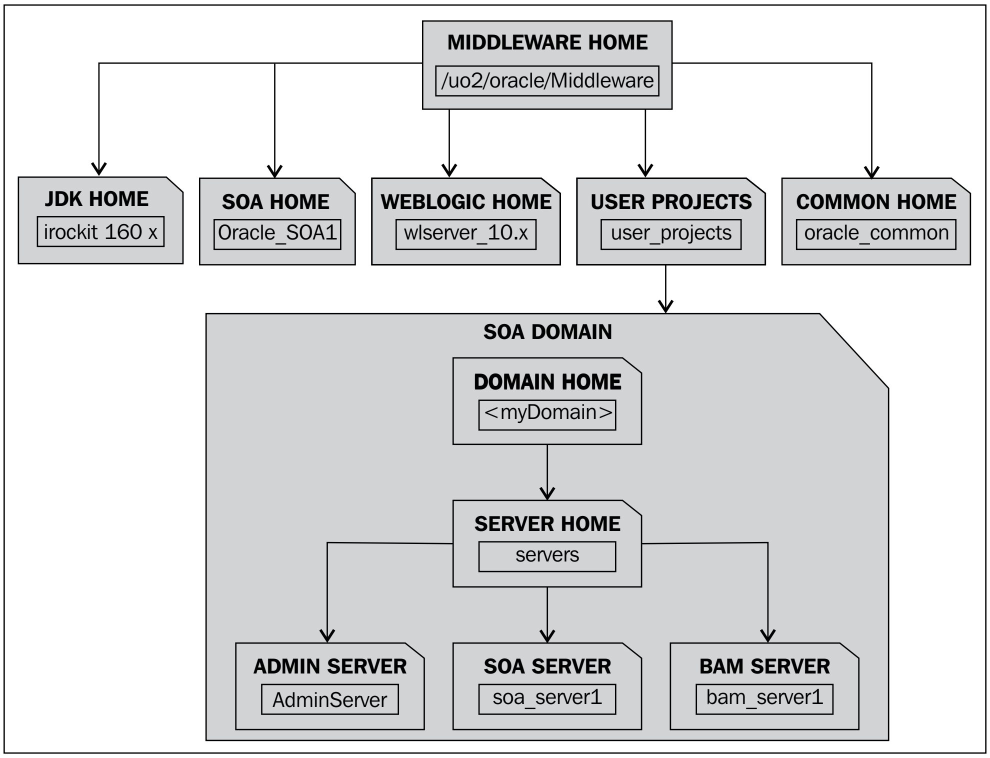
The following figure depicts a workflow of the patch implementation process, beginning with the planning before moving to the testing phase. Application of the patch on non-production environments is performed followed by testing and validation. In some cases, it is not possible to rollback patches, hence if a patch did not apply successfully or did not resolve the particular issue it is meant to address, a restoration of the environment may be required.


PLANNING
Assessment and Planning
Get Approvals and Downtime
APPLY AND TEST
Backup Environment
Apply CPU on Test Environment
Is Successful
No
Able to Fix in Time
No
Yes
Validate Test Environment
Restore Test Environment from Backup
No
Is Stable
No
Able to Fix in Time
Release Test Environment and Postpone Production Patching
Yes
Release Test Environment for General Use
Work offline for Resolution
Ready to Patch Production
Reschedule CPU

Most one-off patches can be applied using the **OPatch** utility. A more complex scenario is where you would have to upgrade an existing installation to a new release or patch set.

Shortly, we will cover a case study that involves upgrading an environment running on Oracle SOA Suite 11*g* PS4 to PS5, to give you a complete step-by-step overview of all activities to be performed in any upgrade scenario. Before embarking on this activity, it is important to understand how the Oracle SOA Suite 11*g* installation is structured.

The Middleware Home is a top-level folder that contains software libraries such as Java Home, Oracle WebLogic Server Home, Oracle Common Home, Oracle SOA Home, and usually the domain(s) folder. The Middleware Home directory is created when Oracle WebLogic Server is first installed. The Oracle Common Home contains the binary and library files required for Oracle Enterprise Manager Fusion Middleware Control and **Java Required Files** (**JRF**), whereas the Oracle SOA Home maintains libraries and extensions required to execute composite applications. Having these separate homes helps in the ongoing maintenance and patching, as depending upon what functionality needs patching, the applicable Oracle Home can be pointed to that component's specific directory. For example, if a patch needs to be applied to Oracle WebLogic Server, then the `ORACLE_HOME` environment variable should point to the `$MW_HOME/wlserver_10.3` folder. All other folders are unaffected. The following figure presents a commonly applied diagrammatic view of the installation directories:



The steps covered in this case study are generic and can be applied to any ad hoc scenarios. The first step in patching/upgrading an environment is to make sure that you maintain components that are certified to work with each other. The Oracle Fusion Middleware certification matrix, found at `http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html`, lists all supported configurations across all Oracle Fusion Middleware products. To upgrade any of the components, locate the product Oracle Fusion Middleware 11*g* Release 1 and download its matrix in Microsoft Excel format.

Also note that upgrading Oracle WebLogic Server and Oracle SOA Suite go hand-in-hand. If Oracle WebLogic Server is Version 10.3.5, then Oracle SOA Suite must be Version 11.1.1.5. If Oracle WebLogic Server is Version 10.3.6, then Oracle SOA Suite must be at 11.1.1.6, and so on. So plan accordingly. One-off patches, such as those that address specific product or component bugs, do not have this same requirement.

Upgrading JDK

From the certification matrix, obtain the certified release of Sun JDK and/or JRockit JDK that is compatible with your Oracle SOA Suite 11*g* Version. In this example, we upgrade Oracle JRockit 1.6.0_26 to 1.6.0_31 on Linux.

To upgrade the JDK in a particular host machine, perform the following:

1. Shutdown all the running managed servers and Node Manager processes.
2. Locate where your existing JDK is installed. For example, it may be under the Middleware Home:`/u02/oracle/middleware/jrockit1.6.0_26`.
3. Download the `jrockit-jdk1.6.0_31-R28.2.3-4.1.0-linux-x64.bin` file to the Middleware Home and ensure that it has execute permissions:

   ```
   cd $MW_HOME
   chmod 750 jrockit-jdk1.6.0_31-R28.2.3-4.1.0-linux-x64.bin
   ```

4. Start the installer: `jrockit-jdk1.6.0_31-R28.2.3-4.1.0-linux-x64.bin`.
5. The JDK binaries will be extracted to this new folder:`/u02/oracle/middleware/jrockit1.6.0_31`.
6. Edit the `setDomainEnv.sh` startup script (typically located under `$DOMAIN_HOME/soa_domain/bin`).
7. For Oracle JRockit, ensure that the following are set, replacing the highlighted text with your specific folder and version:

   ```
   JAVA_VENDOR="Oracle"
   JAVA_HOME="/u02/oracle/middleware/jrockit1.6.0_31"
   ```

For Sun JDK, ensure that the following are set, replacing the highlighted text with your specific folder and version:

```
JAVA_VENDOR="Sun"
JAVA_HOME="/u02/oracle/middleware/jdk1.6.0_29"
```

8. Start the Node Manager and the managed servers.
9. In the server startup log verify that the correct JDK is selected during start up.

If there are any custom certificates applicable for the domain, you would need to perform additional steps to import those certificates in the keystore used by the JDK:

1. Navigate to the following older JDK directory:
   ```
   cd $MW_HOME/jrockit1.6.0_26/jre/lib/security
   ```
2. Execute the following command to export the certificates:
   ```
   keytool -list -v -keystore cacerts -storepass changeit >>
   cacertList.txt
   ```
3. Find out how many entries are in the keystore:
   ```
   cat cacertList.txt | grep entries
   ```
4. The output will appear similar to the following:
   ```
   Your keystore contains 76 entries
   ```

By default, a JDK has a specific number of entries in its keystore. For example, Oracle JRockit 1.6.0_26 contains exactly 76 default entries. If the number outputted for that particular JRockit Version is more than 76, you would need to identify the custom certificates and extract them based on their alias name. They would then have to be imported into the `cacerts` keystore of the new JDK.

Also check for additional `.jks` files that may exist in any of the old JRockit's directory:

```
find . -name /u02/oracle/middleware/jrockit1.6.0_26 "*.jks"
```

If any are found, they need to be copied to the new JRockit's equivalent directory.

Upon starting up the managed servers, the output logs (for example, `soa_server1.out`) can be used to confirm that the correct JDK has been referenced. It would appear similar to the following:

```
starting weblogic with Java version:
java version "1.6.0_31"
Java(TM) SE Runtime Environment (build 1.6.0_31-b05)
Oracle JRockit(R) (build R28.2.3-13-149708-1.6.0_31-20120327-1523-
linux-x86_64, compiled mode)
```

Upgrading the Oracle WebLogic Server Home

Similar to the JDK upgrade, you would need to refer to the Oracle Fusion Middleware certification matrix to download the certified and compatible version of Oracle WebLogic Server. To upgrade the WebLogic Server Home, perform the following steps:

1. Download the applicable Oracle WebLogic Server patch. For example, to download the 10.3.6 patch:
   a. Log in to `http://support.oracle.com`.
   b. Click on the **Patches & Updates** tab.
   c. Click on the **Product or Family (Advanced)** link.
   d. For example, if upgrading from Oracle WebLogic Server 10.3.5 to 10.3.6, search for the **Oracle WebLogic Server** product and the **WLS 10.3.6** release.
   e. Download the appropriate patch for your operating system: `p13529623_1036_Generic.zip` (for 64-bit operating systems).
2. Shutdown the AdminServer.
3. Unzip the patch to a temporary folder. For example:

   ```
   unzip p13529623_1036_Generic.zip
   ```
4. Set the Java Home by specifying the new JDK directory:

   ```
   export JAVA_HOME=/u02/oracle/middleware/jrockit1.6.0_31
   ```
5. Start the installer (the `-d64` argument is only required for 64-bit product sets and the `-Djava.io.tmpdir` argument is optionally used to specify an alternate temporary folder to extract the software):

   ```
   java -d64 -Djava.io.tmpdir=/var/tmp –jar wls1036_upgrade_generic.
   jar
   ```
6. Follow the simple wizard instructions to upgrade your software.
7. Repeat the same process for all other Oracle WebLogic Server Homes (if running a cluster without shared binaries).

Upgrading the Oracle SOA Suite Home

To upgrade to a later release of Oracle SOA Suite, say from 11.1.1.5 to 11.1.1.6, merely download the 11.1.1.6 software from the Oracle Technology Network website. For example, the 64-bit versions of the software can be downloaded from here:

- `http://download.oracle.com/otn/nt/middleware/11g/111160/ofm_soa_generic_11.1.1.6.0_disk1_1of2.zip`
- `http://download.oracle.com/otn/nt/middleware/11g/111160/ofm_soa_generic_11.1.1.6.0_disk1_2of2.zip`

Upgrading the software is straightforward:

1. Shutdown all managed servers.
2. Run the following commands, replacing the Java Home with your current version:

   ```
   unzip ofm_soa_generic_11.1.1.6.0_disk1_1of2.zip
   unzip ofm_soa_generic_11.1.1.6.0_disk1_2of2.zip
   ./Disk1/runInstaller –jreLoc /u02/oracle/middleware/
   jrockit1.6.0_31
   ```

3. Navigate through all screens in the wizard without making any changes.
4. Click on **Install** to begin the upgrade.
5. Repeat the same process for all other Oracle SOA Homes (if running a cluster without shared binaries).

Applying postupgrade tasks

Postupgrade tasks include manually upgrading the Java Required Files (JRF) and **Oracle Platform Security Services** (**OPSS**) components as well as updating the policy store and deploying a shared library. These are typically required after any upgrade to the Oracle SOA Suite 11*g* software stack, unless otherwise noted in the patch notes.

Upgrading JRF

To upgrade JRF:

1. Run the following commands:

   ```
   cd $MW_HOME/oracle_common/common/bin
   ./setWlstEnv.sh
   ./wlst.sh
   ```

2. Run the `upgradeJRF()` WLST command on the machine where the AdminServer is running. Update the domain location to reflect actual values of your environment:

```
upgradeJRF('/u02/oracle/middleware/user_projects/domains/soa_
domain')
```

Upgrading OPSS

In the same WLST session, upgrade OPSS by performing the following:

1. Run the `upgradeOpss()` command on the machine where the AdminServer is running. Update the location of the `jps-config.xml` and `system-jazn-data.xml` to reflect your environment:

```
upgradeOpss(jpsConfig="/u02/oracle/middleware/user_projects/
domainssoa_domain/config/fmwconfig/jps-config.xml", jaznData="/
u02/oracle/middleware/user_projects/domainssoa_domain/config/
fmwconfig/system-jazn-data.xml")
```

Running soa-upgrade.py

To update your policy store and deploy a required shared library, run the `soa-upgrade.py` WLST script. This adds a new role grant and policy grant within the `soa-infra` application stripe in the policy store. It also deploys the `oracle.soa.worklist.webapp` shared library to the SOA domain. Simply perform the following steps:

1. Start up the AdminServer and the managed server(s).
2. Run the following commands, replacing the highlighted text with values reflective of your environment:

```
cd $MW_HOME/Oracle_SOA1/common/bin
./wlst.sh ../../bin/soa-upgrade.py --username weblogic --password
welcome1 --wlsHost soahost --adminServerListenPort 7001
--targetList soa_server1
```

3. Repeat this for all SOA managed servers (if running in a cluster).

Upgrading the OWSM Policy Repository

Each new patch set of **Oracle Web Services Manager** (**OWSM**) can contain new predefined web services policies. The default policy repository must be updated to install additional new policies to the domain as follows:

1. Use WLST `connect()` to connect to the server running the **owsm-pm** application.
2. Run the `upgradeWSMPolicyRepository()` WLST command to upgrade the policy repository.

Upgrading the Oracle SOA Suite database schemas

These instructions describe the steps required to upgrade the Dehydration Store and MDS after a successful upgrade of Oracle SOA Suite. For updating schemas from one version to another you would need to use the **Patch Set Assistant** (**PSA**).

Download and install the compatible version of **Oracle Repository Creation Utility** (**RCU**) and follow the steps:

1. Set the environment and replace highlighted text with your environment specific settings:

   ```
   export ORACLE_HOME=/u02/oracle/middleware/Oracle_SOA1
   cd $ORACLE_HOME/bin
   ```

2. Run the PSA, replacing highlighted text with your database connection information and entering the applicable passwords when prompted:

   ```
   ./psa -dbType Oracle -dbConnectString //[dbhost]:[dbport]/
   [servicename] -dbaUserName sys -schemaUserName [prefix]_SOAINFRA
   ./psa -dbType Oracle -dbConnectString //[dbhost]:[dbport]/
   [servicename] -dbaUserName sys -schemaUserName [prefix]_MDS
   ```

Install the latest purge scripts. For example, if you have just upgraded to 11.1.1.6, download the `ofm_rcu_linux_11.1.1.6.0_disk1_1of1.zip` RCU ZIP file:

1. Unzip the RCU file to your database server and change to the required directory:

   ```
   unzip ofm_rcu_linux_11.1.1.6.0_disk1_1of1.zip
   cd ./rcuHome/rcu/integration/soainfra/sql/soa_purge
   ```

2. Run SQL*Plus against the SOAINFRA schema:

   ```
   sqlplus [prefix]_soainfra@orcl @soa_purge_scripts.sql
   ```

Verifying the database upgrade

Connect to the database to confirm that the database upgrade has completed successfully by running the following SQL statements and verifying the output:

```
SQL> SELECT version, status, upgraded FROM schema_version_registry
WHERE owner='[PREFIX]_SOAINFRA';

VERSION      STATUS      UPGRADED
------------ ----------- --------
11.1.1.6.0   VALID       Y
```

```
SQL> SELECT version, status, upgraded FROM schema_version_registry
WHERE owner='[PREFIX]_MDS';

VERSION      STATUS      UPGRADED
------------ ----------- --------
11.1.1.6.0   VALID       Y
```

To verify that the purge scripts have been updated successfully, run the `soa_verify_scripts.sql` script which is located under the RCU Home:

1. Change to the required directory:

   ```
   cd ./rcuHome/rcu/integration/soainfra/sql/verify
   ```

2. Run SQL*Plus against the SOAINFRA schema:

   ```
   sqlplus [prefix]_soainfra@orcl @soa_verify_scripts.sql
   Checking for invalid database objects
   ```

If you are using an Oracle database, a recompilation of database objects is required after running the PSA. This is performed by simply connecting to the database as SYS and running the following command:

```
SQL> @?/rdbms/admin/utlrp.sql
```

This command compiles the database objects that were updated by the PSA and the following query is executed to ensure there are no longer any invalid database objects:

```
SELECT owner, object_name FROM all_objects WHERE status='INVALID';
```

Ideally none of the database objects for the updated schema should be invalid. If there are any, rerun the `utlrp.sql` command again and recheck or engage your DBA.

Upgrading from Oracle SOA Suite 10*g*

The latest released version of Oracle SOA Suite 10*g* is 10.1.3.5.2. At the time of writing this book, the latest version of Oracle SOA Suite 11*g* is 11.1.1.6. Regardless of the 10*g* release you are upgrading from and the 11*g* you are upgrading to, the approach is the same.

Moving Oracle SOA Suite 10*g* to Oracle SOA Suite 11*g* is both a migration and an upgrade. The **migration** involves converting the existing 10*g* code to its 11*g* equivalent to enable it to execute on Oracle SOA Suite 11*g*. The **upgrade**, which is the primary emphasis of the SOA Suite administrator, is the action of moving to a new version of the same product.

We recommend familiarizing yourself with Oracle's upgrade documentation. They do a good job of addressing many, though not all, of the scenarios you may encounter. These documents include:

- *Oracle Fusion Middleware Upgrade Planning Guide 11g Release 1 (11.1.1)* at `http://docs.oracle.com/cd/E23943_01/upgrade.1111/e10125/toc.htm`.
- *Oracle Fusion Middleware Upgrade Guide for Oracle SOA Suite, WebCenter, and ADF 11g Release 1 (11.1.1)* at `http://docs.oracle.com/cd/E14571_01/upgrade.1111/e10127/toc.htm`.
- *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite 11g Release 1 (11.1.1)* at `http://docs.oracle.com/cd/E14571_01/integration.1111/e10224/toc.htm`.

To summarize, the upgrade activities that an administrator will be involved in include:

1. Installing Oracle SOA Suite 11*g*.
2. Configuring Oracle WebLogic Server 11*g*.
3. Creating partitions.
4. Migrating BPEL domain properties.
5. Importing artifacts to the **Metadata Store** (**MDS**).
6. Migrating **Domain Value Maps** (**DVMs**).
7. Migrating BPEL fault policies.
8. Migrating the code.
9. Cutting over to Oracle SOA Suite 11*g*.

Installing Oracle SOA Suite 11*g*

Performing a major release upgrade from Oracle SOA Suite 10*g* to 11*g* is not an in-place upgrade; a new installation in a separate home or server is required. It is also required to maintain your 10*g* environment for the duration of the upgrade and migration process.

Installing Oracle SOA Suite 11*g* involves installing Oracle Database 11*g* and creating a database, running the Repository Creation Utility (RCU) to create all SOA required database schemas, followed by the installation of Oracle WebLogic Server 11*g* and Oracle SOA Suite 11*g*, and finally concluding with creating the SOA domain. For scalability purposes, we always recommend installing Oracle SOA Suite 11*g* on a 64-bit operating system with a 64-bit JVM.

You can install Oracle SOA Suite 11*g* manually or by following the silent installation instructions listed later in this chapter.

Upgrading the Oracle SOA Suite 10*g* infrastructure to 11*g* involves installing a completely new stack, as it is not an in-place upgrade (that is, the existing 10*g* infrastructure is untouched). Thus, once developers migrate their SOA 10*g* code, it will be deployed on to a completely new environment. This simplifies the development, testing, and cutover efforts as both the 10*g* and 11*g* environments continue to coexist.

Configuring the Oracle WebLogic Server

Certain setup must be performed on the Oracle WebLogic Server 11*g* application server prior to migrating your code. All out-of-the-box configurations need not be touched, and only custom configuration will need to be set identically in the new environment. This mostly includes data sources, messaging queues/topics, and connection factories. Data sources and connection factories are used by application code to connect to a resource or target system without the need to hardcode passwords within the code.

For example, your SOA projects may be leveraging custom created JNDIs that take the following format:

- `jdbc/MyDataSource`
- `eis/DB/MyConnectionFactory`

Keep the following in mind, as it will ultimately affect your production cutover:

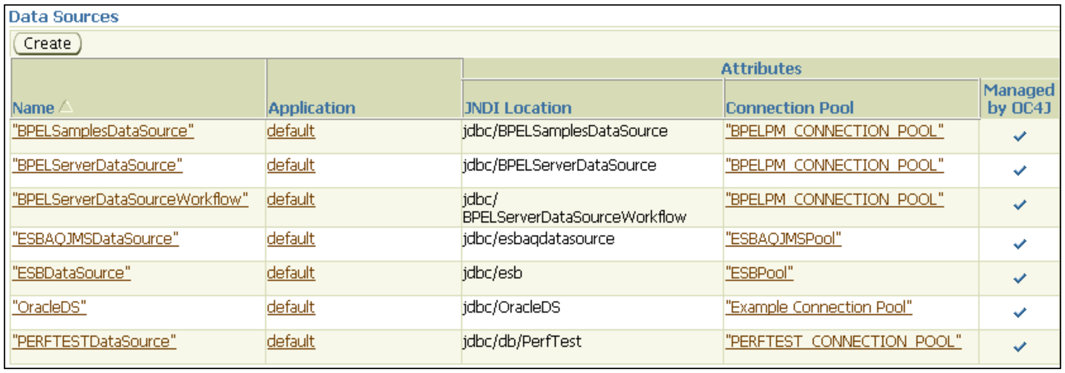
- Data sources and connection factories for both the 10*g* and 11*g* infrastructures can point to the same targets.
- Consider turning off all services on the 10*g* infrastructure that poll (for example, JMS Adapter consume, AQ Adapter dequeue, and DB Adapter polling). If not, polling services will compete on both the old and new infrastructures and you cannot guarantee which of the 10*g* or 11*g* infrastructure will consume the data.

The majority of Oracle SOA Suite 10*g* installations run on Oracle Application Server 10*g* while Oracle SOA Suite 11*g* are usually installed on Oracle WebLogic Server 11*g*, thus the instructions documented in this chapter are specific to those application servers. Instructions may differ if different application servers are utilized in your environment.

Configure custom data sources

Follow these instructions to configure your data sources in Oracle WebLogic Server 11*g* using the same settings as in Oracle Application Server 10*g*:

1. In Oracle Application Server 10*g*, log in to the Enterprise Manager console (this URL is specified in the `setupinfo.txt` file located under the `$ORACLE_HOME/Apache/Apache` directory of your installation).
2. Navigate to **Cluster Topology** | **oc4j_soa** | **Administration** | **JDBC Resources**.



Data Sources

Name △	Application	Attributes: JNDI Location	Attributes: Connection Pool	Managed by OC4J
"BPELSamplesDataSource"	default	jdbc/BPELSamplesDataSource	"BPELPM_CONNECTION_POOL"	✓
"BPELServerDataSource"	default	jdbc/BPELServerDataSource	"BPELPM_CONNECTION_POOL"	✓
"BPELServerDataSourceWorkflow"	default	jdbc/BPELServerDataSourceWorkflow	"BPELPM_CONNECTION_POOL"	✓
"ESBAQJMSDataSource"	default	jdbc/esbaqdatasource	"ESBAQJMSPool"	✓
"ESBDataSource"	default	jdbc/esb	"ESBPool"	✓
"OracleDS"	default	jdbc/OracleDS	"Example Connection Pool"	✓
"PERFTESTDataSource"	default	jdbc/db/PerfTest	"PERFTEST_CONNECTION_POOL"	✓

3. Note the data source name and JNDI location.
4. Click on the **Connection Pool** and note its settings.
5. In Oracle WebLogic Server 11*g*, log in to the WebLogic Server Administration Console.
6. Navigate to **soa_domain** | **Services** | **Data Sources**.
7. Create the data sources and connection pools using the settings captured from 10*g* earlier.

Migrating custom connection factories

The following instructions configure your connection factories in Oracle WebLogic Server 11*g* using the same settings as in Oracle Application Server 10*g*:

1. In Oracle Application Server 10*g*, log in to the Enterprise Manager console.
2. Navigate to **Cluster Topology** | **oc4j_soa** | **Applications** | **default** | **[adapter]** | **Connection Factories**.
3. Note the connection factory settings.

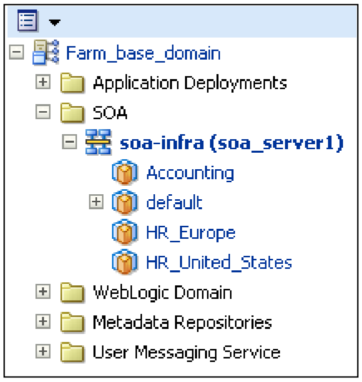
4. In Oracle WebLogic Server 11*g*, log in to the WebLogic Server Administration Console.
5. Navigate to **soa_domain** | **Deployments** | **[Adapter]** | **Configuration** | **Outbound Connection Pools**.
6. Create the connection factory using the settings captured earlier.

Creating partitions

BPEL domains in Oracle SOA Suite 10*g* are used to group BPEL processes. Each BPEL domain has its own configuration and logging and is created by logging in to the BPEL Admin console at a URL similar to the following: `http://<host>:7777/BPELAdmin`.

There is no concept of BPEL domains in Oracle SOA Suite 11*g*. Technically, there is a single BPEL domain. Beginning with Oracle SOA Suite 11*g* PS2, the concept of a partition is introduced. Partitions, unlike BPEL domains, do not have their own configuration or logging. However, from a code grouping perspective, they can be treated identically to BPEL domains.

For example, if you have a BPEL domain called **Accounting**, simply create a partition called Accounting. The following screenshot shows an example of Oracle SOA Suite 11*g* partitions:



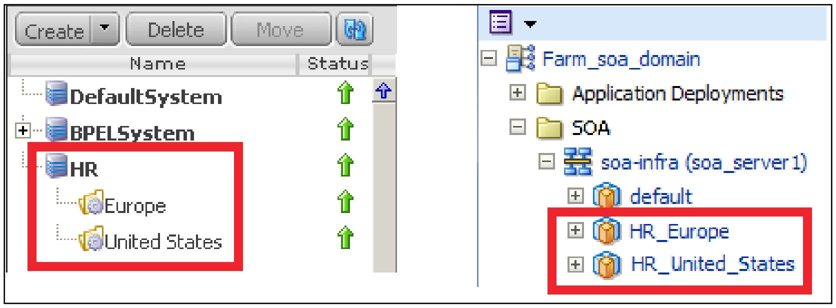
To create 11*g* partitions mimicking your 10*g* BPEL domains, perform the following steps:

1. Log in to BPEL Admin 10*g* at `http://<host>:<port>/BPELAdmin`.
2. Note the BPEL domains.

3. In Oracle SOA Suite 11*g*, log in to Oracle Enterprise Manager Fusion Middleware Control.
4. Create a partition for each BPEL domain by right-clicking on **soa-infra** and clicking on **Manage Partitions**.

ESB Systems and Service Groups are used in Enterprise Service Bus 10*g* to group ESB services. There is no concept of Systems and Service Groups in Oracle SOA Suite 11*g*. Partitions, unlike ESB Systems and Service Groups, do not have listeners. However, from a code grouping perspective, they can be treated similarly to Systems and Service Groups.

One major difference is the fact that Services Groups in ESB 10*g* can be cascading (that is, a Service Group can host another Service Group). Thus, cascading hierarchies will need to be flattened. For example, if you have a System called **HR** with two Service Groups **Europe** and **United States**, it will have to be flattened as shown in the following screenshot:



To create 11*g* partitions mimicking your 10*g* Systems and Service Groups hierarchy, perform the following steps:

1. Log in to ESB Control 10*g* at `http://<host>:<port>/esb`.
2. Expand all Systems and Service Groups and make a note of them.
3. In Oracle SOA Suite 11*g*, log in to Oracle Enterprise Manager Fusion Middleware Control.
4. Create a partition for each flattened Systems and Service Group by right-clicking on **soa-infra** and clicking on **Manage Partitions**.

> Though ESB 10*g* Systems and Service Groups can have spaces in their names, partition names in Oracle SOA Suite 11*g* do not support spaces. Consider using underscores instead of spaces when creating partitions.

Migrating BPEL domain properties

There is technically a single domain in Oracle SOA Suite 11*g* (versus potentially multiple domains in SOA Suite 10*g*), so domain level properties will affect the entire server (affecting both BPEL and Mediator components):

1. Begin by obtaining a copy of all BPEL Process Manager 10*g* properties for all domains. This is located on the Oracle_Home of the BPEL Process Manager 10*g* in the following location: `$ORACLE_HOME/bpel/domains/[domain]/config/domain.xml`

   If you have multiple BPEL domains and different settings for the properties on each of those domains, you will have to decide which setting you want to use in 11*g*.

2. The properties should then be manually added to one of these 11*g* configuration files:

   ```
   $DOMAIN_HOME/config/soa-infra/configuration/bpel-config.xml
   $DOMAIN_HOME/config/[domain_name]/configuration/soa-infra-config.xml
   ```

The following table shows properties that have changed between Oracle SOA Suite 10*g* and 11*g*. All other properties remain the same.

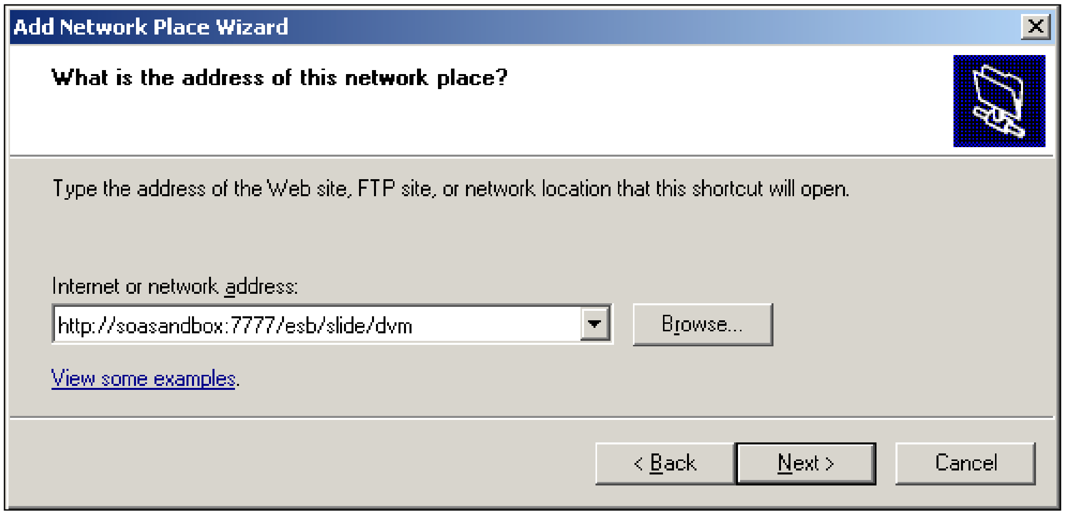
10*g* Property	**11*g* Property**
`dspInvokeAllocFactor`	No longer exists
`dspMaxThreads`	No longer exists
`processCheckSecs`	No longer exists
`txDatasourceJndi`	No longer exists
`uddiLocation`	`oracle.soa.uddi.registry.inquiryUrl`
`uddiPassword`	`oracle.soa.uddi.registry.password`
`uddiUsername`	`oracle.soa.uddi.registry.username`

Migrating DVMs

Although there are numerous ways to migrate DVMs from Oracle SOA Suite 10*g* to 11*g*, the approach here is one of the quicker and simpler ways:

1. In Windows XP, double-click on **My Network Places**, then double-click on **Add Network Place**.

2. Navigate through the screens, and provide the URL to your ESB slide as shown in the following screenshot:



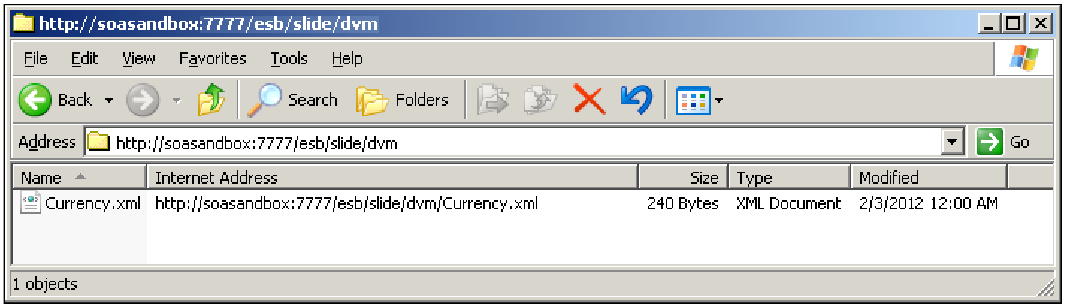
Add Network Place Wizard

What is the address of this network place?

Type the address of the Web site, FTP site, or network location that this shortcut will open.

Internet or network address: http://soasandbox:7777/esb/slide/dvm

3. The contents of the ESB slide are now accessible via Windows Explorer:



http://soasandbox:7777/esb/slide/dvm

Address: http://soasandbox:7777/esb/slide/dvm

Name	Internet Address	Size	Type	Modified
Currency.xml	http://soasandbox:7777/esb/slide/dvm/Currency.xml	240 Bytes	XML Document	2/3/2012 12:00 AM

1 objects

4. Download the DVMs and copy them to your local filesystem.
5. Copy the DVMs to your Linux server.
6. Run the following command to rename all files with extension from `.xml` to `.dvm`:

```
rename .xml .dvm *.xml
```

7. Perform a global search-and-replace to change the `<dvm>` attribute:

```
find . -type f -exec sed -i "s%isNew=\"true\"%xmlns=\"http://
xmlns.oracle.com/dvm\"%" {} \;
```

This replaces an attribute in the DVM to convert it to the 11*g* compatible version. For example:

```
10g: <dvm name="Country_Code" isNew="true">
11g: <dvm name="Country_Code" xmlns="http://xmlns.oracle.com/dvm">
```

8. Create a folder on the local filesystem, and copy the DVMs to this location: `c:\apps\CustomMetaData\dvm`.
9. Zip up the contents of this folder (from the top level `c:\apps\`) and use ant to import it to the MDS (refer to *Chapter 2, Management of SOA Composite Applications,* for details on how to import artifacts to the MDS with ant).

   In our example, this would import all DVMs to the MDS under the following location: `oramds:/apps/CustomMetaData/dvm`.

Importing shared schemas and WSDLs

If you have shared schemas and/or WSDLs, import them to the MDS, maintaining the same hierarchy. If you have imports that are fully qualified references, they must be changed to local references accordingly.

Let's say you have two schemas on your shared server in the following locations:

```
http://soasandbox:7777/Account/Account.xsd
```

```
http://soasandbox:7777/Custom/CustomAccount.xsd
```

And let's say that `Account.xsd` imports the `CustomAccount.xsd` schema as an HTTP reference as follows:

```
<schema import="http://soasandbox:7777/Custom/CustomAccount.xsd">
```

This reference in `Account.xsd` will need to be changed to a local reference, as:

```
<schema import="../Custom/CustomAccount.xsd">
```

Now both these schemas should be copied to your local filesystem, in preparation for import to the MDS:

```
c:\ant\CustomMetaData\xsd\Account\Account.xsd
c:\ant\CustomMetaData\xsd\Custom\CustomAccount.xsd
```

Finally, zip up the contents of this folder and use ant to import it to the MDS. This would be imported to the MDS under the following location:

```
oramds:/apps/CustomMetaData/xsd/Account/Account.xsd
oramds:/apps/CustomMetaData/xsd/Custom/CustomAccount.xsd
```

Migrating BPEL fault policies

It is possible to maintain multiple fault policies in Oracle SOA Suite 11*g*. This section describes, at a high-level, how to migrate a single, custom fault policy from 10*g* to 11*g*:

1. Identify all BPEL Process Manager 10*g* domains that have fault policies applied to them. For example, the default domain will have its own fault policy that may have been customized. The two relevant policy files are located here:

   ```
   $ORACLE_HOME/bpel/domains/default/config/fault-bindings.xml
   $ORACLE_HOME/bpel/domains/default/config/fault-policies/
   DefaultPolicy.xml
   ```

2. Change the version in `fault-bindings.xml` from **2.0.1** to **3.0** and change the `<process>` element to `<composite>`, as highlighted in the following code:

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <faultPolicyBindings version="3.0"
        xmlns="http://schemas.oracle.com/bpel/faultpolicy"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
     <composite faultPolicy="DefaultPolicy"/>
   </faultPolicyBindings>
   ```

3. Rename the 10*g* fault policy file to `fault-policies.xml`, change the version from "2.0.1" to "3.0", and add a top level `<faultPolicies>` node as shown in the highlighted text:

   ```
   <?xml version="1.0" encoding="UTF-8"?>
   <faultPolicies
        xmlns="http://schemas.oracle.com/bpel/faultpolicy"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
     <faultPolicy version="3.0" id="DefaultPolicy"
         xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"
          xmlns:xs="http://www.w3.org/2001/XMLSchema"
          xmlns="http://schemas.oracle.com/bpel/faultpolicy"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
          .
          .
          .
     </faultPolicy>
   </faultPolicies>
   ```

4. Create a folder on the local filesystem, and copy both files to this location: `c:\apps\CustomMetaData\faultPolicies`.

5. Zip up the contents of this folder (from the top level `c:\apps\`) and use ant to import it to the MDS.

   This would be imported to the MDS under the following location: `oramds:/apps/CustomMetaData/faultPolicies`.

Migrating the code

To migrate BPEL and ESB projects developed originally in Oracle JDeveloper 10*g* to their 11*g* equivalent involves going through multiple steps before and after the actual migration of the project itself. The end result is a single 11*g* SOA composite that will consist of a single BPEL or Mediator component. Oracle JDeveloper 11*g* provides a **Migration Wizard** that will automatically upgrade your 10*g* SOA projects.

The code migration is mostly a development activity and not addressed in this book, as the emphasis here is on the infrastructure side of things. Your development team can refer to the *Oracle Fusion Middleware Upgrade Guide for Oracle SOA Suite, WebCenter*, and *ADF 11g Release 1 (11.1.1)* (mentioned earlier) for additional information.

Cutting over to Oracle SOA Suite 11*g*

Cutting over to your production environment from an already running Oracle SOA Suite 10*g* to 11*g* will vary depending on the types of integrations you have and the downtime you can afford. The simplest approach is to install your Oracle SOA Suite 11*g* cluster using the same load balancer as the 10*g* environment, deploy all the migrated SOA projects, and bring them in load. Essentially, both your 10*g* and 11*g* infrastructures can run in parallel (since they are effectively running the same code) as you bring the 10*g* servers out of load. If your SOA code is using inbound adapters to consume messages, you can turn them off on the 10*g* side so that they may get flushed out as you bring the 11*g* into load.

However, there are certain factors that must be considered which may render this cutover approach invalid. The following questions must be evaluated when designing your cutover process:

- Do you rely on long running processes such as human workflow?
- Do you rely on inbound adapters that consume messages?
- Do you have a gateway that virtualizes your services?

Based on these answers, you may have to alter your cutover approach. Refer to the Oracle documentation listed earlier which covers various possible scenarios for cutover.

Installing a highly available cluster

High availability refers to the ability to maintain operational integrity if a server failure occurs. This can be achieved by installing two or more Oracle SOA Suite 11*g* nodes in a cluster to protect against failure. Server failures could be software induced, such as a JVM crashing due to being out of memory, or a hardware failure, such as a local disk going bad. Installing an Oracle SOA Suite 11*g* cluster requires additional setup and configuration different from that of a single-node installation.

There are many ways in which you can set up a highly available SOA infrastructure such as vertical and/or horizontal scaling over physical machines, hardware partitioning using virtualization, and cloud computing. This section will limit itself to setting up a highly available two node cluster across physical machines as this topology is one of the more common ones.

There are numerous approaches to laying out the filesystem in an Oracle SOA Suite 11*g* cluster, with a share all, share some, or share little approach regarding the binaries. In the instructions we describe here, we follow a share all approach, which means that the majority of the software binaries are shared between the two servers except what is not possible. This is convenient as performing an upgrade or applying a patch takes effect on all nodes of the cluster without having to repeat the activity multiple times.

You can also get creative on how you wish to set up your load balancers. Ideally, you would want to set up two sets of load balancing URLs; for example, `http://admin.mydomain.com` which is dedicated for administration traffic (such as the admin consoles) and `http://soahost.mydomain.com`, which is dedicated to transactional activity. You may also opt to configure them as SSL if transport level security is to be enforced and then use SSL acceleration if needed.

Oracle's Enterprise Deployment Guide (links provided as follows) is the standard reference documentation that you can refer to for most activities related to setting up high availability.

- *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle SOA Suite 11g Release 1 (11.1.1)* at `http://docs.oracle.com/cd/E25054_01/core.1111/e12036/toc.htm`.
- *Oracle Fusion Middleware High Availability Guide 11g Release 1 (11.1.1)* E10106-02 at `http://docs.oracle.com/cd/E16764_01/core.1111/e10106/title.htm`.

High availability installation overview

Installing a 2-node Oracle SOA Suite 11*g* cluster on a 64-bit Linux operating system requires performing the following steps:

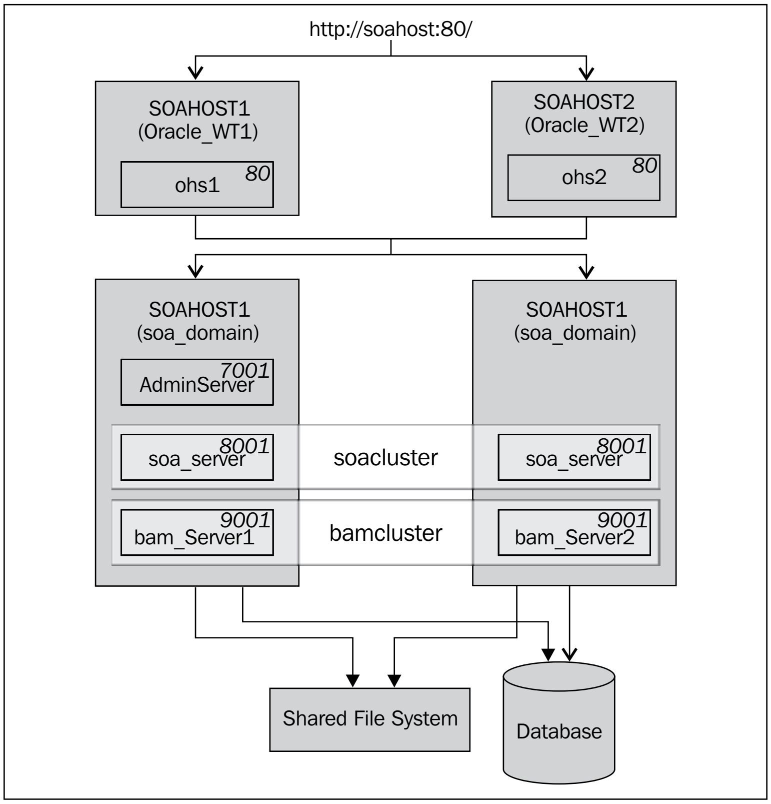
- Apply operating system prerequisites
- Install Oracle JRockit JDK 1.6.0_31 64-bit
- Install Oracle Web Tier 11*g* (11.1.1.6) 64-bit
- Install Oracle WebLogic Server 11*g* (10.3.6) 64-bit
- Install Oracle SOA Suite 11*g* (11.1.1.6) 64-bit

Architectural considerations

The following architectural considerations are made with the instructions provided in this section:

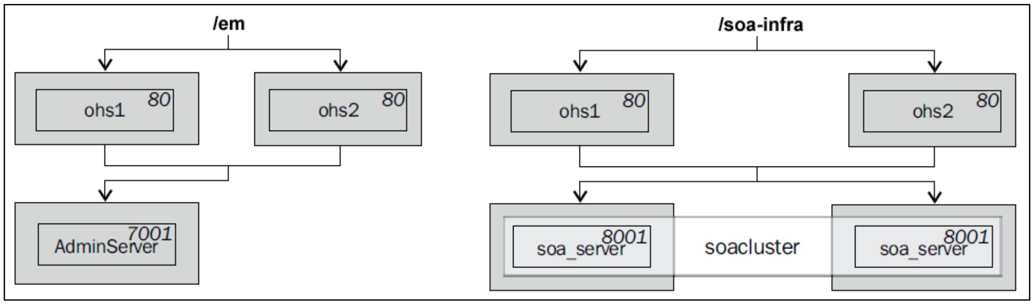
- Both console and transactional activity use the same load balancer URL.
- No floating IP address will be utilized, thus the AdminServer is tied exclusively to `SOAHOST1`. If `SOAHOST1` is down and irrecoverable, additional work is needed to reconfigure `SOAHOST2` to bring up the AdminServer.
- A shared filesystem is utilized for transaction logs, deployment descriptors (plan files), and persistent stores as well as software binaries that include Oracle JRockit JDK, Oracle WebLogic Server, and Oracle SOA Suite.
- 64-bit installations are performed. Though 64-bit JVMs have a slight performance hit compared to their 32-bit equivalents, memory space is potentially unlimited. Oracle SOA Suite 11*g* has considerably more memory requirements than 10*g*, thus you do not want to be limited by a 3 GB JVM.
- Oracle Web Tier 11*g* is installed which includes the **Oracle HTTP Server** (**OHS**) based on Apache. In the absence of a load balancer, OHS will provide basic routing, security, and control features that may be utilized.
- Oracle Web Services Manager (OWSM) is deployed to the SOA managed servers and not on its own dedicated managed servers.

- The database can be single node or RAC.



The preceding figure depicts the architecture for our 2-node Oracle SOA Suite 11*g* cluster. We rely on two physical servers called `SOAHOST1` and `SOAHOST2`. A single domain, **soa_domain**, is created. The SOA managed servers are within a cluster called **soacluster** while the BAM managed servers are in **bamcluster**. The Oracle HTTP Server is listening on port 80 and routes console requests to the AdminServer and transactional requests to the SOA managed servers. Each node of the cluster relies on a shared filesystem and shared database.

By leveraging context based routing in OHS, requests are routed to the appropriate application. For example, the following figure shows that requests intended for `/em` (Oracle Enterprise Manager Fusion Middleware Control) are routed to the deployed application on AdminServer, while transactional requests beginning from `/soa-infra` in the URL are alternated between both SOA managed servers. This routing is controlled by adding directives to the `mod_wl_ohs.conf` plugin configuration file, described later. Thus, we are able to maintain a single point of entry through the load balancer and route the incoming requests according to the context.



Applying prerequisites

Key prerequisites need to be applied prior to the installation of the Oracle SOA Suite 11*g* cluster. These steps must be performed on both physical servers, `SOAHOST1` and `SOAHOST2`.

Verifying temp and SWAP space

Ensure that both `/tmp` has at least 1 GB available disk space and that SWAP is configured to be at least 1 GB in size.

Creating operating system accounts

These instructions assume that the entire software stack will be installed by the operating system user `oracle` having a group of `oinstall`. As **root**, run the following commands to create the operating system account:

```
groupadd oinstall
groupadd dba
useradd -c "Oracle software owner" -g oinstall -G dba oracle
passwd oracle
```

Verifying mount points

Create the mount point `/u01/app` on both servers, having at least 10 GB. This will store all non-shared binaries and domain configuration. As **root**, modify the permissions accordingly to allow the oracle user full read/write access to this top-level directory. As **root**, create the mount points and change the permissions appropriately:

```
mkdir /u01/app
chown oracle:oinstall /u01/app
```

Updating the hosts file

You may already have both soahost1 and soahost2 configured in your hosts file or it may be already configured within your network's DNS. Both are acceptable. Edit the /etc/hosts file to include adminhost, which is an alias to the AdminServer, and set it to the same IP address as soahost1, as exemplified here:

```
" 192.168.2.107  soahost1  adminhost
" 192.168.2.108  soahost2
```

Since AdminServer can only be started on a single machine within the cluster, we have chosen to bind it to soahost1.

Configuring operating system prerequisites

Certain recommended Linux operating system prerequisites are needed before installing Oracle SOA Suite 11*g* and creating domains. The steps here are specific to the 64-bit versions of Oracle Enterprise Linux 5 or Red Hat Linux Enterprise Server 5. All instructions here must be run as the **root** user:

1. As **root**, add the following to the end of /etc/sysctl.conf on both servers, using your server domain and hostname in place of the highlighted text:

   ```
   # semaphores: semmsl, semmns, semopm, semmni
   kernel.shmmni = 4096
   kernel.sem = 256 32000 100 128
   fs.file-max = 6815744
   fs.aio-max-nr = 1048576
   kernel.hostname = soahost1.mydomain.com
   kernel.domainname = mydomain.com
   net.ipv4.ip_local_port_range = 9000 65500
   net.core.rmem_default = 262144
   net.core.wmem_default = 262144
   net.core.rmem_max = 4194304
   net.core.wmem_max = 1048576
   kernel.msgmax = 8192
   kernel.msgmnb = 65535
   kernel.msgmni = 2878
   ```

2. Add the following to the end of the /etc/security/limits.conf file:

   ```
   oracle  soft    nofile  4096
   oracle  hard    nofile  65536
   oracle  soft    nproc   2047
   oracle  hard    nproc   16384
   ```

3. Add the following to the end of `/etc/pam.d/login`:

   ```
   session    required     /lib/security/pam_limits.so
   ```

4. Add the following to the end of `/etc/profile`:

   ```
   if [ $USER = "oracle" ]; then
     if [ $SHELL = "/bin/ksh" ]; then
       ulimit -Su 16383
       ulimit -Hu 16383
       ulimit -Sn 63535
       ulimit -Hn 63535
     else
       ulimit -Hn 63535 -Sn 63535 -Hu 16383 -Su 16383
     fi
   fi
   ```

5. Ensure that the latest versions of the following **Red Hat Package Manager** packages, or **RPMs**, are installed. Note that versions of these RPMs may vary depending on the specific release of your operating system.

   ```
   rpm -ivh compat-db-4.2.52-5.1.i386.rpm
   rpm -ivh compat-db-4.2.52-5.1.x86_64.rpm
   rpm -ivh binutils-2.17.50.0.6-14.el5-x86_64.rpm
   rpm -ivh compat-libstdc++-33-3.2.3-61.i386.rpm
   rpm -ivh compat-libstdc++-33-3.2.3-x86_64.rpm
   rpm -ivh elfutils-libelf-0.137-3.el5.i386.rpm
   rpm -ivh elfutils-libelf-devel-0.137-3.el5-x86_64.rpm
   rpm -ivh elfutils-libelf-devel-static-0.137-3.el5.i386.rpm
   rpm -ivh gcc-4.1.2-50.el5-x86_64.rpm
   rpm -ivh gcc-c++-4.1.2-50.el5-x86_64.rpm
   rpm –ivh glibc-2.5-12-i686.rp
   rpm -ivh glibc-2.5-12-x86_64.rpm
   rpm -ivh glibc-common-2.5-58-x86_64
   rpm –ivh glibc-devel-2.5-12-i386.rpm
   rpm -ivh glibc-devel-2.5-x86_64.rpm
   rpm -ivh glibc-headers-2.5-49.i386.rpm
   rpm -ivh ksh-20100202-1.el5.i386.rpm
   rpm -ivh libaio-0.3.106-5.i386.rpm
   rpm –ivh libaio-0.3.106-x86_64.rpm
   rpm -ivh libaio-devel-0.3.106-5.i386.rpm
   rpm –ivh libaio-devel-0.3.106-5.x86_64.rpm
   rpm -ivh libgcc-4.1.2-48.el5.i386.rpm
   rpm –ivh libgcc-4.1.2-50.el5-x86_64.rpm
   rpm -ivh libgomp-4.4.0-6.el5.i386.rpm
   rpm –ivh libstdc++-4.1.1-x86_64.rpm
   rpm -ivh libstdc++-4.1.2-48.el5.i386.rpm
   ```

```
rpm -ivh libstdc++-devel-4.1.2-48.el5.i386.rpm
rpm –ivh libstdc++-devel-4.1.2-50.el5-x86_64
rpm -ivh libXp-1.0.0-8.1.el5.i386.rpm
rpm -ivh libXp-1.0.0-8.1.el5.x86_64.rpm
rpm –ivh make-3.81-3.el5-x86_64.rpm
rpm –ivh sysstat-7.0.2-3.el5.x86_64.rpm
rpm –ivh unixODBC-2.2.11-7.1.i386.rpm
rpm -ivh unixODBC-2.2.11-7.1.x86_64.rpm
rpm -ivh unixODBC-devel-2.2.11-7.1.i386.rpm
rpm -ivh unixODBC-devel-2.2.11-7.1.x86_64.rpm
```

Configuring .bash_profile preferences

Update `.bash_profile` to create shortcuts which are used throughout the installation and to simplify directory navigation. The remaining instructions throughout this section are based on these environment settings:

1. As the **oracle** user, edit the `.bash_profile` script (default location is `/home/oracle/.bash_profile`) and add the following towards the bottom:

```
############################################
# Common SOA Suite 11g environment shortcuts
############################################
export PS1="\u@\h:\$(pwd)> "
export TMPDIR=/tmp
export ORACLE_BASE=/u01/app/oracle
export ORACLE_SHARE=/u01/share
export DOMAIN=soa_domain
export INSTALL_SOFTWARE=$ORACLE_SHARE/install_software
export ORAINVENTORY=$ORACLE_BASE/oraInventory
export DOMAIN_HOME=$ORACLE_BASE/user_projects/domains/${DOMAIN}
export MW_HOME=$ORACLE_SHARE/middleware
export WL_HOME=$MW_HOME/wlserver_10.3
export JAVA_HOME=$MW_HOME/jrockit1.6.0_31
export ORACLE_HOME=$MW_HOME/Oracle_SOA1
# Next line in SOAHOST1 only
export OHS_HOME=$MW_HOME/Oracle_WT1/instances/ohs_instance1
# Next line in SOAHOST2 only
export OHS_HOME=$MW_HOME/Oracle_WT2/instances/ohs_instance2
export PATH=$JAVA_HOME/bin:$PATH:.
```

Set up shared storage

The shared filesystem will contain the Java Home, Middleware Home, Oracle WebLogic Server Home, and Oracle SOA Home. This is to allow a single set of software binaries to be shared across both servers. This allows for a quicker installation process as well as reduced maintenance when it comes to patching and upgrading. Transaction logs and JMS persistent stores must also reside on this shared storage, but for operational reasons. To set up shared storage, create the shared folder `/u01/share` accessible to both `SOAHOST1` and `SOAHOST2` and modify the following permissions accordingly:

```
mkdir -p $INSTALL_SOFTWARE
mkdir -p $MW_HOME
mkdir -p $ORACLE_SHARE/soa_domain/soacluster/jms
mkdir -p $ORACLE_SHARE/soa_domain/soacluster/tlogs
chown oracle:oinstall $ORACLE_SHARE/*
```

Setting up the database

Install Oracle Database 11*g* (11.2.0 and higher) and create a database. This can be a single node, a grid, or RAC setup. The default number of processes and cursors after installing Oracle Database 11*g* do not meet the minimum required for Oracle SOA Suite 11*g* dehydration store. After installation of the database, execute the following commands on the database server as SYS to increase them:

```
ALTER SYSTEM SET processes=1000 SCOPE=spfile;
ALTER SYSTEM SET open_cursors=1000 SCOPE=spfile;
```

Setting up the load balancer

Contact your network team and create a load balanced URL that routes all requests (using round robin algorithm) from `http://soahost:80` to `http://soahost1:80` and `http://soahost2:80`.

Downloading the software

Download all software listed in the table below to the `/u01/share/install_software` directory. These versions are specific to the latest 11.1.1.6 release of the Oracle Fusion Middleware stack.

Prior to clicking on any of the links in the table below, navigate to the main download page at `http://www.oracle.com/technetwork/middleware/soasuite/downloads/index.html` and sign in to the Oracle Technology Network.

Product	Link
Oracle JRockit JDK 1.6.0_31 (64-bit)	`http://download.oracle.com/otn/bea/jrockit/jrockit-jdk1.6.0_31-R28.2.3-4.1.0-linux-x64.bin`
Oracle Repository Creation Utility (RCU) 11*g* (11.1.1.6)	`http://download.oracle.com/otn/linux/middleware/11g/111160/ofm_rcu_linux_11.1.1.6.0_disk1_1of1.zip`
Oracle Web Tier Utilities 11*g* (11.1.1.6) 64-bit	`http://download.oracle.com/otn/linux/middleware/11g/111160/ofm_webtier_linux_11.1.1.6.0_64_disk1_1of1.zip`
Oracle WebLogic Server 11*g* (10.3.6) Generic	`http://download.oracle.com/otn/nt/middleware/11g/wls/1036/wls1036_generic.jar`
Oracle SOA Suite 11*g* (11.1.1.6) Generic Disk 1	`http://download.oracle.com/otn/nt/middleware/11g/111160/ofm_soa_generic_11.1.1.6.0_disk1_1of2.zip`
Oracle SOA Suite 11*g* (11.1.1.6) Generic Disk 2	`http://download.oracle.com/otn/nt/middleware/11g/111160/ofm_soa_generic_11.1.1.6.0_disk1_2of2.zip`

> The direct links in the preceding table point to the latest versions of the required products and components, but note that Oracle often retires older versions of their software and keeps only links to the latest versions for public download.

Installing the software

Now that all prerequisites are set and the software has been downloaded, it is time to begin the actual installation and configuration of Oracle JRockit, Oracle Web Tier, Oracle WebLogic Server, and Oracle SOA Suite.

Sourcing .bash_profile

All instructions assume that the `.bash_profile` is sourced so that all shortcuts and environment variables are appropriately set. This can be done by logging off your current Unix session and logging back in (assuming your operating system user is using the Bash shell) or by sourcing the `.bash_profile`:

1. Source the `.bash_profile`:

   ```
   cd
   . .bash_profile
   ```

2. Verify that the environment variables are valid by validating a few of them:

   ```
   echo $MW_HOME
   echo $JAVA_HOME
   echo $ORACLE_HOME
   ```

Installing Oracle JRockit 1.6.0

Oracle JRockit will be installed on the shared filesystem and shared across both machines:

1. On `SOAHOST1` only, run the following commands to start the Oracle JRockit installer on the shared filesystem:

   ```
   cd $MW_HOME
   chmod 750 $ORACLE_SHARE/install_software/jrockit-jdk1.6.0_31-
   R28.2.3-4.1.0-linux-x64.bin
   $ORACLE_SHARE/install_software/jrockit-jdk1.6.0_31-R28.2.3-4.1.0-
   linux-x64.bin
   ```

   The command line version of the **Oracle JRockit Installer - Oracle JRockit JDK R28.2.3 for Java SE 6 with JRMC 4.1** will now begin.

2. On the **Welcome** page, enter Next:

   ```
   Welcome:
   --------
   Enter [Exit][Next]> Next
   ```

3. On the **Choose Product Installation Directory** page, enter the new installation directory `/u01/share/middleware/jrockit1.6.0_31`:

   ```
   Choose Product Installation Directory:
   --------------------------------------
   Enter new Product Installation Directory OR [Exit][Previous]
   [Next]> /u01/share/middleware/jrockit1.6.0_31
   ```

4. On the next **Choose Product Installation Directory** page, enter `1`:

   ```
   Choose Product Installation Directory:
   --------------------------------------
   Use above value or select another option:
       1 - Enter new Product Installation Directory
       2 - Change to default [/home/oracle/jrockit-
   jdk1.6.0_31-R28.2.3-4.1.0]
   Enter option number to select OR [Exit][Previous][Next]> 1
   ```

5. On the next **Choose Product Installation Directory** page, enter `Next` to accept the previously entered installation directory:

```
Choose Product Installation Directory:
--------------------------------------
    "Product Installation Directory" = [/u01/share/middleware/
jrockit1.6.0_31]
Enter new Product Installation Directory OR [Exit][Previous]
[Next]> Next
```

6. On the **Optional Components 1** page, enter `2` to skip the installation of the demos and samples:

```
Optional Components 1
Do you want to install Demos and Samples?
    1 - Yes, install Demos and Samples.
  ->2 - No, do not install.
Enter a number or [Exit][Previous][Next]> 2
```

7. On the **Optional Components 2** page, enter `2` to skip the installation of the source code:

```
Optional Components 2
Do you want to install Source Code?
    1 - Yes, install Source Code.
  ->2 - No, do not install.
Enter a number or [Exit][Previous][Next]> 2
```

8. On the **Installation Complete** page, press Enter to exit the installer:

```
Installation Complete
Congratulations! Oracle JRockit JDK R28.2.3 for Java SE 6 with
JRMC 4.1 has been successfully installed to /u01/share/middleware/
jrockit1.6.0_31.
Press [Enter] to continue or type [Exit]>
```

Installing Oracle Web Tier 11*g*

Oracle Web Tier 11*g* should be installed on both `SOAHOST1` and `SOAHOST2` on the shared filesystem but under different Oracle Homes and different OHS instances for each server:

1. Execute the following commands on both `SOAHOST1` and `SOAHOST2` to install Oracle Web Tier:

```
cd $INSTALL_SOFTWARE
unzip ofm_webtier_linux_11.1.1.6.0_64_disk1_1of1.zip
```

2. Edit the `staticports.ini` file:

   ```
   vi ./Disk1/stage/Response/staticports.ini
   ```

3. Uncomment this line:

   ```
   OHS Port = 8888
   ```

Start the installer:

```
./Disk1/runInstaller
```

4. On the **Specify Inventory** directory screen, enter the following:

   Inventory Directory: `/u01/app/oracle/oraInventory`

   Operating System Group: `oinstall`

   Open a new shell window as **root** and run the following command:

   ```
   /u01/app/oracle/oraInventory/createCentralInventory.sh
   ```

5. On the **Install Software Updates** screen, select **Skip Software Updates**.
6. On the **Install Software Updates** screen, choose **Install and Configure**.
7. On the **Specify Installation Location** screen, enter the following:

   Oracle Middleware Home: `/u01/share/middleware`

   Oracle Home Directory: Oracle_WT1 (on `SOAHOST1`)
   Oracle_WT2 (on `SOAHOST2`)

8. On the **Specify Security Updates** screen, uncheck **I wish to receive security updates via My Oracle Support**.
9. On the **Configure Components** screen:
   - Check **Oracle HTTP Server**.
   - Uncheck **Oracle Web Cache**.
   - Uncheck **Associate Selected Components with WebLogic Domain**.

10. On the **Specify Component Details** screen, enter the following:

    Instance Home Location (on SOAHOST1):

    `/u01/share/middleware/Oracle_WT1/instances/ohs_instance1`

    Instance Home Location (on SOAHOST2):

    `/u01/share/middleware/Oracle_WT2/instances/ohs_instance2`

    Instance Name: `ohs_instance1` (on `SOAHOST1`)
    `ohs_instance2` (on `SOAHOST2`)

    OHS Component Name: `ohs1` (on `SOAHOST1`)
    `ohs2` (on `SOAHOST2`)

11. On the **Configure Ports** screen, choose the **Specify Ports using Configuration** file and enter the following **File name**: /u01/share/install_software/Disk1/stage/Response/staticports.ini
12. After installation is complete, execute the following commands:
    ```
    cd $OHS_HOME/bin
    ./opmnctl stopall
    ```
13. Execute the following commands to delete the extracted software:
    ```
    cd $INSTALL_SOFTWARE
    rm -rf readme.htm Disk1 Disk2 Disk3
    ```

Configuring Oracle Web Tier 11*g*

These instructions must be executed on both SOAHOST1 and SOAHOST2 to configure Oracle Web Tier to listen on port 80 and route incoming requests to the appropriate managed servers. For example, the applications responsible for serving WebLogic Server Administration Console and Oracle Enterprise manager Fusion Middleware Control are deployed to AdminServer. Thus, requests to the /em and /console URLs are explicitly routed to AdminServer only which is running on host adminhost on port 7001. Perform the following to configure OHS:

1. Shutdown OHS:
   ```
   cd $OHS_HOME/bin
   ./opmnctl stopall
   ```
2. Modify the Apache binaries to allow them to run on port 80 (Unix based operating systems limit services from running on ports less than 1024 with non-privileged owners):
   ```
   su
   chown root $MW_HOME/Oracle_WT?/ohs/bin/.apachectl
   chmod 6750 $MW_HOME/Oracle_WT?/ohs/bin/.apachectl
   exit
   ```
3. Edit both HTTP configuration files:
   ```
   vi $OHS_HOME/config/OHS/ohs?/httpd.conf
   ```
4. Replace the following:
   ```
   OLD:    Listen 8888
   NEW:    Listen 80
   ```

5. Add the following to the end of the file, where `SOAHOST` represents the load balancer URL name:

   ```
   NameVirtualHost *:80
   <VirtualHost *:80>
     ServerName http://soahost:80
     RewriteEngine On
     RewriteOptions inherit
   </VirtualHost>
   ```

6. Edit both WLS module configuration files:

   ```
   vi $OHS_HOME/config/OHS/ohs?/mod_wl_ohs.conf
   ```

7. Add the following to the end of the file:

   ```
   # SOA
   <Location /soa-infra>
     SetHandler weblogic-handler
     WebLogicCluster soahost1:8001,soahost2:8001
   </Location>
   # BPM Worklist
   <Location /integration>
     SetHandler weblogic-handler
     WebLogicCluster soahost1:8001,soahost2:8001
   </Location>
   # B2B Console
   <Location /b2bconsole>
     SetHandler weblogic-handler
     WebLogicCluster soahost1:8001,soahost2:8001
   </Location>
   # UMS Preferences
   <Location /sdpmessaging/userprefs-ui>
     SetHandler weblogic-handler
     WebLogicCluster soahost1:8001,soahost2:8001
   </Location>
   # Default to-do taskflow
   <Location /DefaultToDoTaskFlow>
     SetHandler weblogic-handler
     WebLogicCluster soahost1:8001,soahost2:8001
   </Location>
   # Workflow
   <Location /workflow>
     SetHandler weblogic-handler
     WebLogicCluster soahost1:8001,soahost2:8001
   </Location>
   ```

```
# Required if attachments are added for workflow tasks
<Location /ADFAttachmentHelper>
  SetHandler weblogic-handler
  WebLogicCluster soahost1:8001,soahost2:8001
</Location>
# SOA Composer console
<Location /soa/composer>
  SetHandler weblogic-handler
  WebLogicCluster soahost1:8001,soahost2:8001
</Location>
# EM console
<Location /em>
  SetHandler weblogic-handler
  WebLogicCluster adminhost:7001
</Location>
# WLS console
<Location /console>
  SetHandler weblogic-handler
  WebLogicCluster adminhost:7001
</Location>
# Console Help
<Location /consolehelp>
  SetHandler weblogic-handler
  WebLogicCluster adminhost:7001
</Location>
# WSM-PM
<Location /wsm-pm>
  SetHandler weblogic-handler
  WebLogicCluster soahost1:8001,soahost2:8001
</Location>
# BAM console
<Location /OracleBAM>
  SetHandler weblogic-handler
  WebLogicCluster soahost1:9001,soahost2:9001
</Location>
<Location /OracleBAMWS>
  SetHandler weblogic-handler
  WebLogicCluster soahost1:9001,soahost2:9001
</Location>
```

Installing Oracle WebLogic Server 11*g*

The Oracle WebLogic Server 11*g* binaries will be installed on the shared filesystem and need only to be installed once from `SOAHOST1`:

1. Run the following commands:

   ```
   cd $MW_HOME
   java -d64 -Djava.io.tmpdir=/u01/share/middleware -jar $INSTALL_
   SOFTWARE/wls1036_generic.jar
   ```

2. On the **Choose Middleware Home Directory** screen, select **Create a new Middleware Home** and enter the following:

   Middleware Home Directory: `/u01/share/middleware`

3. Ignore the warning.
4. On the **Register for Security Updates** screen, uncheck **I wish to receive security updates via My Oracle Support**.
5. On the **Choose Install Type** screen, select **Custom**.
6. On the **Choose Products and Components** screen, uncheck **Evaluation Database**.
7. On the **JDK Selection** screen, **Oracle 1.6.0_31 (/u01/share/middleware/jrockit1.6.0_31)** should already be selected.
8. On the **Choose Product Installation Directories** screen, keep all defaults.
9. When installation is complete, unselect **Run QuickStart**.

Running the Repository Creation Utility (RCU)

The RCU needs to be executed only from `SOAHOST1` to create all required database schemas:

1. Run the following commands:

   ```
   cd $INSTALL_SOFTWARE
   unzip ofm_rcu_linux_11.1.1.5.0_disk1_1of1.zip
   ./rcuHome/bin/rcu
   ```

2. On the **Create Repository** screen, select **Create**.
3. On the **Database Connection Details** screen, enter the database connection information as follows:

   Database Type: `Oracle Database`

   Hostname: `<hostname>`

   Port: `<port>`

Service Name:	`<service name>`
Username:	`sys`
Password:	`<password>`
Role:	`SYSDBA`

4. On the **Select Components** screen, enter a prefix name such as DEV in the **Create a new Prefix** field and check **SOA and BPM Infrastructure**.
5. On the **Schema Passwords** screen, check **Use same passwords for all schemas** and enter the password information. Choose whether you want to connect to a RAC or non-RAC database and enter values accordingly.
6. On the **Map Tablespaces** screen, accept all defaults.
7. On the **Summary** screen, click **Create** to begin the repository creation.
8. After the installer finishes, execute the following commands to delete the extracted software:

```
cd $INSTALL_SOFTWARE
chmod -R 700 rcuHome
rm -rf rcuHome
```

Installing Oracle SOA Suite 11*g*

Oracle SOA Suite 11*g* binaries will be installed on the shared filesystem, and need only to be installed once from SOAHOST1:

1. Run the following commands:

```
cd $INSTALL_SOFTWARE
unzip ofm_soa_generic_11.1.1.6.0_disk1_1of2.zip
unzip ofm_soa_generic_11.1.1.6.0_disk1_2of2.zip
./Disk1/runInstaller -jreLoc $JAVA_HOME
```

2. Enter the following during installation:
3. On the **Install Software Updates** screen, select **Skip Software Updates**.
4. On the **Specify Installation Location** screen, enter the following:

Oracle Middleware Home:	`/u01/share/middleware`
Oracle Home Directory:	`Oracle_SOA1`

5. On the **Application Server screen,** select **WebLogic Server**.

6. After the installer finishes, execute the following commands to delete the extracted software:

   ```
   cd $INSTALL_SOFTWARE
   rm -rf readme.htm Disk1 Disk2 Disk3 Disk4 Disk5
   ```

Creating the SOA domain

The SOA domain will only be created on `SOAHOST1`:

1. Run the following commands only on `SOAHOST1`:

   ```
   export CONFIG_JVM_ARGS="-DTemplateCatalog.enable.selectable.
   all=true"
   cd $WL_HOME/common/bin
   ./config.sh
   ```

2. On the **Welcome** screen, select **Create a new WebLogic domain**.
3. On the **Select Domain Source** screen, check the following:
   - Oracle SOA Suite - 11.1.1.0 [Oracle_SOA1]
   - Oracle Enterprise Manager - 11.1.1.0 [oracle_common]
   - Oracle Business Activity Monitoring - 11.1.1.0 [Oracle_SOA1]
4. On the **Specify Domain Name and Location** screen, enter the following:

   Domain name: `soa_domain`

   Domain location: `/u01/app/oracle/user_projects/domains`

   Application location: `/u01/app/oracle/user_projects/applications`

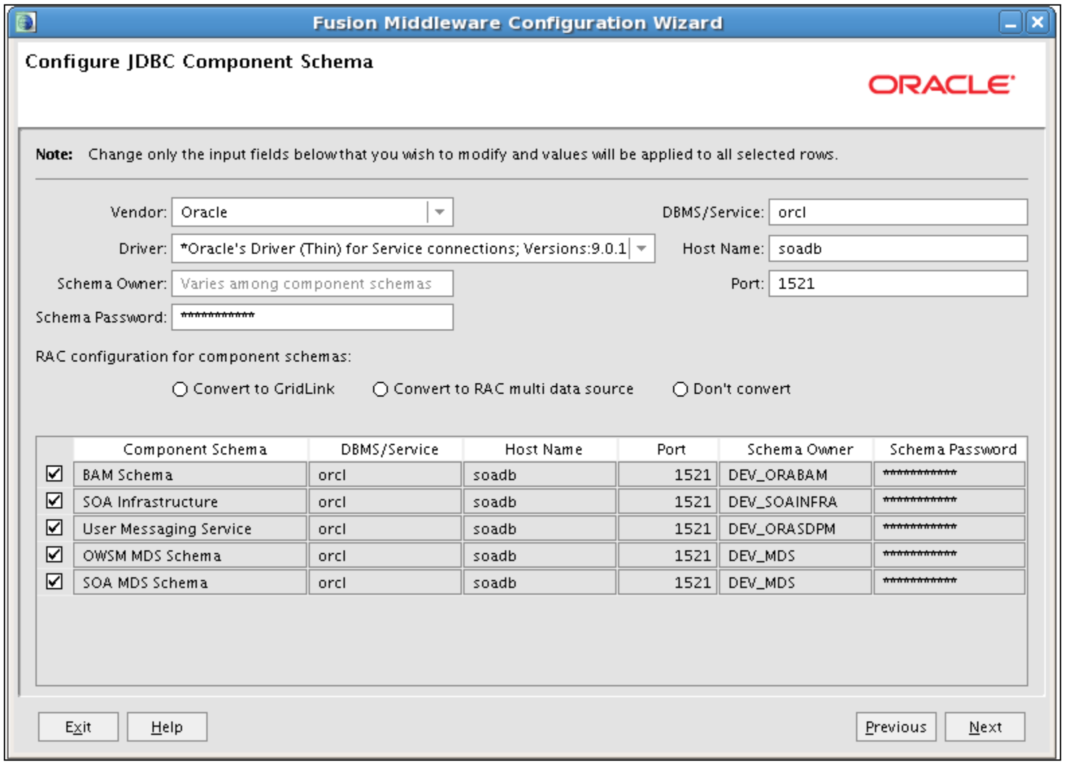
5. On the **Configure Administrator User Name and Password** screen, enter the username and password of the weblogic admin account you wish to create:

   Name: `weblogic`

   Password: `welcome1`

6. On the **Configure Server Start Mode and JDK** screen, perform the following:
   a. Under **WebLogic Domain Startup** Mode, select **Production Mode**.
   b. Under **JDK Selection**, select Available JDKs **JRockit SDK 1.6.0_31 @ /u01/share/middleware/jrockit1.6.0_31.**

7. On the **Configure JDBC Component Schema** screen, shown in the following screenshot, enter the database connection information:



Configure JDBC Component Schema

Note: Change only the input fields belowthat you wish to modify and values will be applied to all selected rows.

Vendor:	Oracle	DBMS/Service:	orcl
Driver:	*Oracle's Driver (Thin) for Service connections; Versions:9.0.1	Host Name:	soadb
Schema Owner:	Varies among component schemas	Port:	1521
Schema Password:	*************		

RAC configuration for component schemas:

○ Convert to GridLink ○ Convert to RAC multi data source ○ Don't convert

	Component Schema	DBMS/Service	Host Name	Port	Schema Owner	Schema Password
☑	BAM Schema	orcl	soadb	1521	DEV_ORABAM	*************
☑	SOA Infrastructure	orcl	soadb	1521	DEV_SOAINFRA	*************
☑	User Messaging Service	orcl	soadb	1521	DEV_ORASDPM	*************
☑	OWSM MDS Schema	orcl	soadb	1521	DEV_MDS	*************
☑	SOA MDS Schema	orcl	soadb	1521	DEV_MDS	*************

> Typically the ***Oracle's Driver (Thin) for Service Connections** is selected for single-node databases while ***Oracle's Driver (Thin) for RAC Service-Instance connections** is selected for RAC databases.

8. On the **Select Optional Configuration** screen, select:
   - Administration Server
   - JMS Distributed Destination
   - Managed Servers, Clusters and Machines
   - Deployments and Services
   - JMS File Store

9. On the **Configure the Administration Server** screen, enter the following:
   - Name: `AdminServer`
   - Listen address: `soahost1`
   - Listen port: `7001`
   - SSL enabled: (unchecked)
10. On the **Select JMS Distributed Destination Type** screen, accept all defaults.
11. On the **Configure Managed Servers** screen, enter the following for **Name**, **Listen address**, **Listen port** and **SSL**:

    `soa_server1`, `soahost1`, `8001`, SSL enabled (unchecked)

    `soa_server2`, `soahost2`, `8001`, SSL enabled (unchecked)

    `bam_server1`, `soahost1`, `9001`, SSL enabled (unchecked)

    `bam_server2`, `soahost2`, `9001`, SSL enabled (unchecked)
12. On the **Configure Clusters** screen, add the following Name and Cluster messaging mode:

    `soacluster`, `unicast`

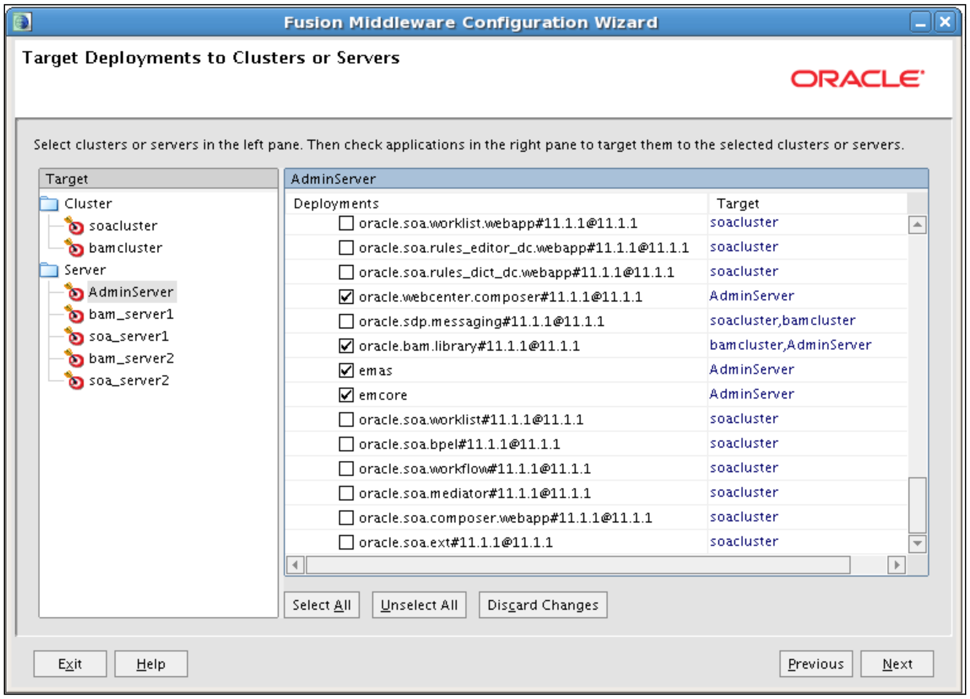
    `bamcluster`, `unicast`
13. On the **Assign Servers to Clusters** screen, perform the following:
    a. Assign **soa_server1** and **soa_server2** to **soacluster**.
    b. Assign **bam_server1** and **bam_server2** to **bamcluster**.
14. On the **Configure Machines** screen, perform the following:
    a. Delete LocalMachine.
    b. Click on the **Unix Machine** tab and add the following **Name**, **Node manager listen address**, and **Node manager listen port**:

       `soamachine1`, `soahost1`, `5556`

       `soamachine2`, `soahost2`, `5556`
15. On the **Target Deployments to Clusters or Servers** screen, perform the following:
    a. Assign **AdminServer**, **soa_server1**, and **bam_server1** to **soamachine1**.
    b. Assign **soa_server2** and **bam_server2** to **soamachine2**.

16. On the **Target Deployments to Clusters or Servers** screen, click on **AdminServer** under Target, and perform the following:
    1. Under the first Deployments section, ensure that only **DMS Application#11.1.1.1.0**, **wsil-wls**, and **em** are targeted to **AdminServer**.
    2. Under **Library**, ensure that everything is targeted to **AdminServer** except for all **oracle.sdp.***, **oracle.soa.***, **oracle.rules.***, and **oracle.bam.***.

Verify that the **Target Deployments to Clusters or Servers** looks like the one shown in the following screenshot:



Target Deployments to Clusters or Servers

Select clusters or servers in the left pane. Then check applications in the right pane to target them to the selected clusters or servers.

Target
- Cluster
  - soacluster
  - bamcluster
- Server
  - AdminServer
  - bam_server1
  - soa_server1
  - bam_server2
  - soa_server2

AdminServer

Deployments	Target
□ oracle.soa.worklist.webapp#11.1.1@11.1.1	soacluster
□ oracle.soa.rules_editor_dc.webapp#11.1.1@11.1.1	soacluster
□ oracle.soa.rules_dict_dc.webapp#11.1.1@11.1.1	soacluster
☑ oracle.webcenter.composer#11.1.1@11.1.1	AdminServer
□ oracle.sdp.messaging#11.1.1@11.1.1	soacluster,bamcluster
☑ oracle.bam.library#11.1.1@11.1.1	bamcluster,AdminServer
☑ emas	AdminServer
☑ emcore	AdminServer
□ oracle.soa.worklist#11.1.1@11.1.1	soacluster
□ oracle.soa.bpel#11.1.1@11.1.1	soacluster
□ oracle.soa.workflow#11.1.1@11.1.1	soacluster
□ oracle.soa.mediator#11.1.1@11.1.1	soacluster
□ oracle.soa.composer.webapp#11.1.1@11.1.1	soacluster
□ oracle.soa.ext#11.1.1@11.1.1	soacluster

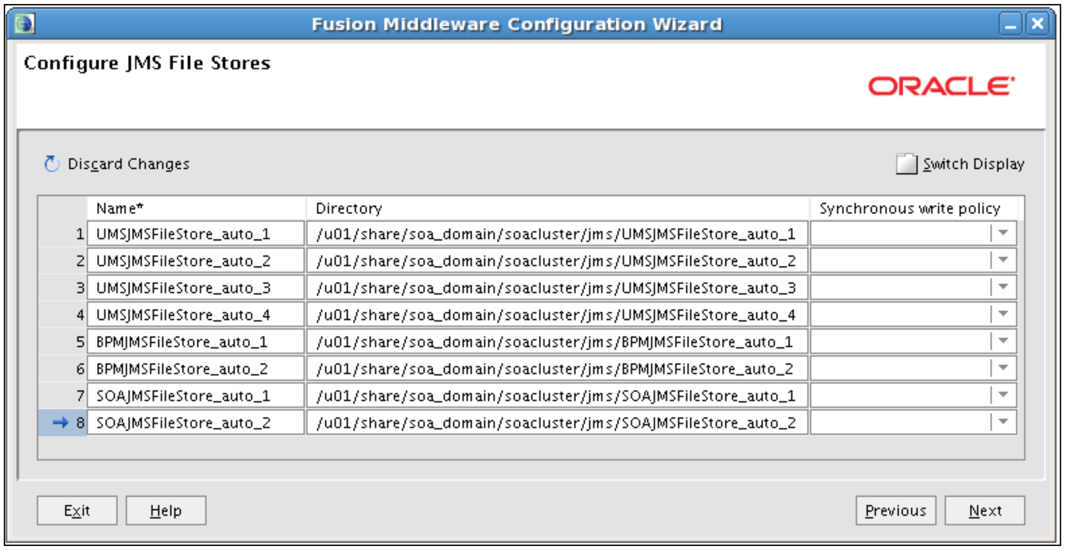
17. On the **Target Services to Clusters or Servers** screen, click on **bamcluster** under **Target**, and perform the following:
    1. Remove **mds-owsm** from **bamcluster**.
    2. Leave the rest of the services as targeted by the installer.
18. On the **Configure JMS File Store** screen, enter the following:

    UMSJMSFileStore_auto_1:

    /u01/share/soa_domain/soacluster/jms/UMSJMSFileStore_auto_1

UMSJMSFileStore_auto_2:
/u01/share/soa_domain/soacluster/jms/UMSJMSFileStore_auto_2
WseeFileStore_auto_1:
/u01/share/soa_domain/soacluster/jms/WseeFileStore_auto_1
WseeFileStore_auto_2:
/u01/share/soa_domain/soacluster/jms/WseeFileStore_auto_2
FileStore_auto_1:
/u01/share/soa_domain/soacluster/jms/FileStore_auto_1
FileStore_auto_2:
/u01/share/soa_domain/soacluster/jms/FileStore_auto_2
BPMJMSFileStore_auto_1:
/u01/share/soa_domain/soacluster/jms/BPMJMSFileStore_auto_1
BPMJMSFileStore_auto_1:
/u01/share/soa_domain/soacluster/jms/BPMJMSFileStore_auto_2
SOAJMSFileStore_auto_1:
/u01/share/soa_domain/soacluster/jms/SOAJMSFileStore_auto_1
SOAJMSFileStore_auto_1:
/u01/share/soa_domain/soacluster/jms/SOAJMSFileStore_auto_2

A screenshot of this screen is shown in the following figure:


Fusion Middleware Configuration Wizard
Configure JMS File Stores
ORACLE
Discard Changes
Switch Display
Exit
Help
Previous
Next

	Name*	Directory	Synchronous write policy
1	UMSJMSFileStore_auto_1	/u01/share/soa_domain/soacluster/jms/UMSJMSFileStore_auto_1	
2	UMSJMSFileStore_auto_2	/u01/share/soa_domain/soacluster/jms/UMSJMSFileStore_auto_2	
3	UMSJMSFileStore_auto_3	/u01/share/soa_domain/soacluster/jms/UMSJMSFileStore_auto_3	
4	UMSJMSFileStore_auto_4	/u01/share/soa_domain/soacluster/jms/UMSJMSFileStore_auto_4	
5	BPMJMSFileStore_auto_1	/u01/share/soa_domain/soacluster/jms/BPMJMSFileStore_auto_1	
6	BPMJMSFileStore_auto_2	/u01/share/soa_domain/soacluster/jms/BPMJMSFileStore_auto_2	
7	SOAJMSFileStore_auto_1	/u01/share/soa_domain/soacluster/jms/SOAJMSFileStore_auto_1	
→ 8	SOAJMSFileStore_auto_2	/u01/share/soa_domain/soacluster/jms/SOAJMSFileStore_auto_2	

19. On the **Summary** screen, click **Create** to begin the domain creation.

Disabling hostname verification

To avoid receiving server certificate errors (as is the case if you have not set up the appropriate certificates to authenticate all nodes with the AdminServer), execute the following instructions (only on `SOAHOST1`) to disable hostname verification, perform the following steps:.

1. Run the following commands, and enter the weblogic username and password when prompted:

   ```
   cd $DOMAIN_HOME
   ./startWebLogic.sh
   ```

2. Wait until the server mode has changed to running.
3. Log in to the WebLogic Administration Console at `http://soahost:7001/console`.
4. Click on **Lock & Edit**.
5. Navigate to **Environment** | **Servers** | **AdminServer(admin)** | **Configuration** | **SSL**.
6. Click on **Advanced**.
7. Set **Hostname Verification** to `None`.
8. Save changes.
9. Repeat steps 4 to 7 for **soa_server1**, **soa_server2**, **bam_server1**, and **bam_server2**.
10. Save and **Activate changes**.
11. Shutdown the AdminServer by pressing *Ctrl*+*C* on the shell window.

Creating boot.properties

Create the `boot.properties` file so that you are not prompted for the Weblogic username and password every time the AdminServer is started up. This is only done on `SOAHOST1`: Execute the following steps, using the password created earlier:

```
cd $DOMAIN_HOME/servers/AdminServer
mkdir security
cd security
echo "username=weblogic" > boot.properties
echo "password=welcome1" >> boot.properties
```

Starting Node Manager on SOAHOST1

At this point, Node Manager is to be configured and started up on SOAHOST1, as follows:

1. Create the Node Manager property file:

   ```
   cd $MW_HOME/oracle_common/common/bin
   ./setNMProps.sh
   ```

2. Configure Node Manager:

   ```
   cd $WL_HOME/common/nodemanager
   vi nodemanager.properties
   ```

3. Set the following properties:

   ```
   StartScriptEnabled=true
   StopScriptEnabled=true
   ```

4. Execute the following commands to start up Node Manager on SOAHOST1:

   ```
   cd $WL_HOME/server/bin
   nohup ./startNodeManager.sh &
   ```

Copying domain configuration to SOAHOST2

The SOA domain configuration is now packed and copied from SOAHOST1 to SOAHOST2, where it is extracted there, as follows:

1. Execute the following commands on SOAHOST1 to pack the SOA domain:

   ```
   cd $WL_HOME/common/bin
   ./pack.sh -managed=true -domain=/u01/app/oracle/user_projects/
   domains/soa_domain -template=soadomaintemplate.jar -template_
   name=soa_domain_template
   ```

2. Unpack the template JAR file on SOAHOST2:

   ```
   cd $WL_HOME/common/bin
   ./unpack.sh -domain=/u01/app/oracle/user_projects/domains/soa_
   domain -template=soadomaintemplate.jar
   ```

Starting Node Manager on SOAHOST2

Now it is time to start up Node Manager on SOAHOST2. Execute the following commands to start up Node Manager on SOAHOST2, as follows:

```
cd $WL_HOME/server/bin
nohup ./startNodeManager.sh &
```

Starting AdminServer on SOAHOST1

The AdminServer is now started up on SOAHOST1 to continue configuration. Run the following commands on SOAHOST1:

```
cd $DOMAIN_HOME
nohup ./startWebLogic.sh &
```

Registering Oracle Web Tier

The two instances of Oracle Web Tier (Oracle HTTP Server) will be registered with the domain. Start up OHS and register the instance. Execute these commands on both SOAHOST1 and SOAHOST2 and enter the weblogic password when prompted:

```
cd $OHS_HOME/bin
./opmnctl startall
./opmnctl registerinstance -adminHost soahost1 -adminPort 7001
-adminUsername weblogic
```

Apply JRF template to SOA cluster and enable Java Object Cache

Apply the JRF template to the soacluster and enable **Java Object Cache** (**JOC**). These instructions need to be executed only from SOAHOST1, as follows:

1. Run the following commands on SOAHOST1:

   ```
   cd $MW_HOME/oracle_common/common/bin
   ./wlst.sh
   ```

2. Enter the following highlighted entries to connect to the AdminServer:

   ```
   wls:/offline> connect()
   Please enter your username : weblogic
   Please enter your password : welcome1
   Please enter your server URL [t3://localhost:7001] : t3://
   soahost1:7001
   Connecting to t3://adminhost:7001 with userid weblogic ...
   Successfully connected to Admin Server 'AdminServer' that belongs
   to domain 'soa_domain'.
   ```

3. Execute the following WLST command to apply JRF to the SOA cluster:

   ```
   applyJRF('soacluster','/u01/app/oracle/user_projects/domains/soa_
   domain','true')
   ```

4. Execute the following WLST command to enable Java Object Cache (JOC):

   ```
   execfile('/u01/share/middleware/oracle_common/bin/configure-joc.
   py')
   ```

5. Enter the following highlighted entries:

   ```
   Enter Hostnames (eg host1,host2) : soahost1,soahost2
   Do you want to specify a cluster name (y/n) <y>y
   Enter Cluster Name : soacluster
   Enter Discover Port : 9991
   Enter Distribute Mode (true|false) <true> : true
   Do you want to exclude any server(s) from JOC configuration (y/n)
   <n>n
   ```

6. Press *Ctrl+C* to exit the window.

Configuring the SOA cluster

Specific configuration needs to be applied to the SOA cluster, all of which is performed via the WebLogic Server Administration Console.

1. Log into the WebLogic Server Administration Console at `http://soahost:80/console`.
2. Click on **Lock & Edit**.
3. Navigate to **Environment** | **Clusters** | **soacluster** | **Configuration** | **General** and set the following:

   Cluster Address: `soahost1:8001,soahost2:8001`

4. Navigate to **Environment** | **Clusters** | **soacluster** | **Configuration** | **HTTP** and set the following:

   Frontend Host: `soahost`

   Frontend HTTP Port: `80`

   Frontend HTTPS Port: `8889`

Configuring Oracle Coherence for deploying composites

Oracle Coherence must be configured to support deployment of composites onto a cluster:

1. Navigate to **Environment** | **Servers** | **soa_server1** | **Configuration** | **Server Start** and set **Arguments** to:

   ```
   -Dtangosol.coherence.wka1=soahost1 -Dtangosol.coherence.
   wka2=soahost2 -Dtangosol.coherence.localhost=soahost1
   ```

2. Navigate to **Environment** | **Servers** | **soa_server2** | **Configuration** | **Server Start** and set **Arguments** to:

   ```
   -Dtangosol.coherence.wka1=soahost1 -Dtangosol.coherence.
   wka2=soahost2 -Dtangosol.coherence.localhost=soahost2
   ```

Configuring default persistent store

The default persistent store must be configured for transaction recovery, as follows:

1. Navigate to **Environment** | **Servers** | **soa_server1** | **Configuration** | **Services** and set **Directory** to the following:

   ```
   /u01/share/soa_domain/soacluster/tlogs
   ```

2. Navigate to **Environment** | **Servers** | **soa_server2** | **Configuration** | **Services** and set **Directory** to the following:

   ```
   /u01/share/soa_domain/soacluster/tlogs
   ```

3. Save and **Activate changes**.

Targeting deployments to the BAM server

Not all components in Oracle BAM can be clustered. Only the front ending web host can be targeted to the cluster. Specific deployments must be targeted to the appropriate BAM targets, as follows:

1. On the WebLogic Server Administration Console, click on **Lock & Edit**.
2. Click on **Deployments**.
3. Click on the **oracle-bam** application.
4. Click **Targets**.
5. Change the targets as per the following table:

Component	**Type**	**Target**
oracle-bam(11.1.1)	Enterprise Application	bamcluster
/oracle/bam	WEBAPP	bam_server1
`oracle-bam-adc-ejb.jar`	EJB	bam_server1
`oracle-bam-ems-ejb.jar`	EJB	bam_server1
`oracle-bam-eventengine-ejb.jar`	EJB	bam_server1

Component	Type	Target
`oracle-bam-reportcache-ejb.jar`	EJB	bam_server1
`oracle-bam-statuslistener-ejb.jar`	EJB	bam_server1
OracleBAM	WEBAPP	bamcluster
OracleBAMWS	WEBAPP	bamcluster
`sdpmessagingclient-ejb.jar`	EJB	bam_server1

6. Save and **Activate changes**.
7. Start up **bam_server1** and **bam_server2**.

Configuring the BAM properties

The BAM application URL must be set to the load balancer and the server name to the server running the Active Data Cache, which is `SOAHOST1`, as follows:

1. Log into the Oracle Enterprise Manager Fusion Middleware Control at `http://soahost:80/em`.
2. Expand **BAM**, right-click on **OracleBamWeb(bam_server1)**, then click on **BAM Web Properties**.
3. Update the **BAM** web properties as follows:

   Application URL: `http://soahost:80`

   Server Name: `soahost1`
4. Right-click on **OracleBamWeb(bam_server2)** then click on **BAM Web Properties**.
5. Update the **BAM** web properties as follows:

   Application URL: `http://soahost:80`

   Server Name: `soahost1`

Configuring Active Data Cache (ADC)

Active Data Cache (**ADC**) runs in a singleton mode and hence is configured only to point to `SOAHOST1`:

1. Log into the Oracle Enterprise Manager Fusion Middleware Control at `http://soahost:80/em`.

2. Expand **BAM** and right-click on **OracleBamWeb(bam_server1)**.
3. Click on **System MBean Browser**.
4. Under **Application Defined Beans**, expand **oracle.bam.server**.
5. Update the following properties as follows:

ADCServerName:	`soahost1`
ADCServerPort:	`9001`

Configure callback and server URLs

The Oracle SOA Suite 11*g* callback URLs are configured from Oracle Enterprise Manager Fusion Middleware Control. Configuring the callback server URL property is essential if there are composites deployed to the infrastructure that have asynchronous services waiting to be notified from received activities. In a clustered setup, this is generally set to the load balancer address. But before we configure it, the AdminServer must be restarted so that the numerous changes made earlier take effect. The configuration is done as follows:

1. Log into the WebLogic Server Administration Console at `http://soahost:80/console`.
2. Shutdown the AdminServer from the console.
3. Start up AdminServer on `SOAHOST1` from the Unix prompt as follows:

   ```
   cd $DOMAIN_HOME
   nohup ./startWebLogic.sh &
   ```

4. Log into the WebLogic Server Administration Console again, and start up **soa_server1**.
5. Log into the Oracle Enterprise Manager Fusion Middleware Control at `http://soahost:80/em`.
6. Right-click on **soa-infra** and navigate to **SOA Administration | Common Properties**.
7. Configure the **Callback Server URL** and **Server URL** as follows:

Callback Server URL:	`http://soahost:80`
Server URL:	`http://soahost:80`

8. Click on **Apply**.
9. Log into the WebLogic Server Administration Console and shutdown and restart **soa_server1** and **soa_server2**.

Verifying the installation

Now that the 2-node cluster is installed, some basic verification can be performed by logging in to each of the consoles. Log in to and confirm that the following URLs are accessible:

- `http://soahost/console`
- `http://soahost/em`
- `http://soahost/integration/worklistapp`
- `http://soahost/soa/composer`
- `http://soahost/wsm-pm`
- `http://soahost/OracleBAM` (only works in Microsoft Internet Explorer)
- `http://soahost1:7001/dms/Spy`
- `http://soahost2:7001/dms/Spy`

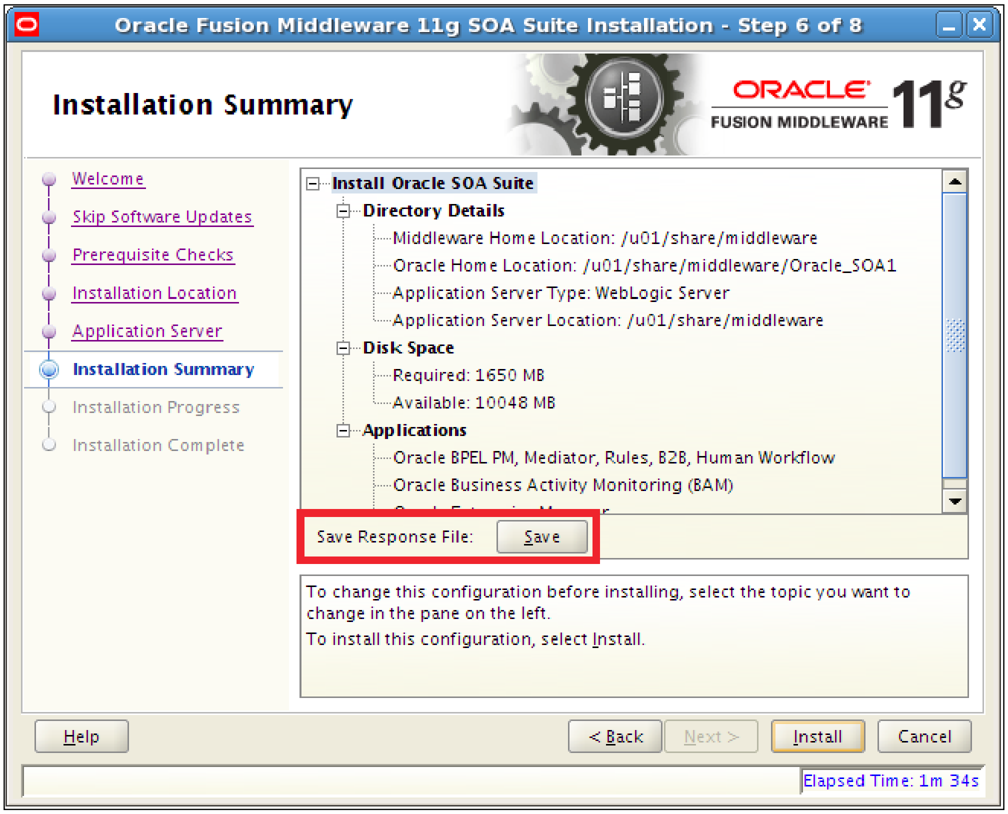
That's it! You now have a fully functional 2-node cluster.

Silent Installation

The majority of Oracle products support silent installations, wherein the installation can take place without a graphical user interface and without user intervention. This is useful when an end-to-end scripted approach to infrastructure installation is needed especially in cases where many environments need to be created and/or recreated.

Silent installations rely on response files which contain environment specific information such as directories, port numbers, and other parameters required by the installer. Essentially all settings that can be modified in the graphical installer will have a corresponding configuration setting in the response file. Response files are basic text files and the simplest way to create one is to go through the graphical installer, and on the last screen, it will provide a checkbox asking you whether you want the response file saved or not.

This can be seen on the last screen of the Oracle SOA Suite 11*g* installer as shown in the following screenshot:



Oracle Fusion Middleware 11g SOA Suite Installation - Step 6 of 8

Installation Summary

- Install Oracle SOA Suite
  - Directory Details
    - Middleware Home Location: /u01/share/middleware
    - Oracle Home Location: /u01/share/middleware/Oracle_SOA1
    - Application Server Type: WebLogic Server
    - Application Server Location: /u01/share/middleware
  - Disk Space
    - Required: 1650 MB
    - Available: 10048 MB
  - Applications
    - Oracle BPEL PM, Mediator, Rules, B2B, Human Workflow
    - Oracle Business Activity Monitoring (BAM)

Save Response File: Save

To change this configuration before installing, select the topic you want to change in the pane on the left.
To install this configuration, select Install.

Elapsed Time: 1m 34s

The approach to silently installing Oracle SOA Suite 11*g* requires a few pre-installation activities that include applying all operating system prerequisites and generating the response file templates to be used as the basis for future silent installations.

Afterwards, silently installing Oracle SOA Suite 11*g* involves:

- Customizing the response files, updating them with environment specific settings
- Installing the JDK via a scripted approach
- Silently installing Oracle WebLogic Server
- Silently running RCU to create the corresponding component schemas
- Silently installing Oracle SOA Suite
- Creating the SOA domain via a scripted approach

In this section, we demonstrate the commands involved in silently installing Oracle SOA Suite 11*g* in a single node architecture. Clustered installations require additional effort not described here, as there are many pre and post steps that must be performed across multiple servers (as seen in the previous section).

Applying operating system prerequisites

Apply all operating system prerequisites first. Refer to the *Applying Prerequisites* section under *Installing a highly available cluster* earlier in this chapter.

Creating the oraInst.loc file

For Unix-based installations, the `oraInst.loc` file must be manually created first. This is located in the `/etc` folder in Linux:

Determine the location of your Oracle Inventory (such as, `/u01/app/oracle/oraInventory`).

Determine the operating system group of accounts that will be installing the software (for example, oinstall).

Log in to the server as **root**, and run the following commands:

```
echo "inventory_loc=/u01/app/oracle/oraInventory" > /etc/oraInst.loc
echo "inst_group=oinstall" >> /etc/oraInst.loc
chown oracle:oinstall /etc/oraInst.loc
```

Customizing the response files

Prior to silently installing the products, you must customize the response files. The JDK installation and creation of the SOA domain are not reliant on response files and can be performed via scripted installations. Oracle WebLogic Server, Oracle SOA Suite, and the RCU rely on response files that must be customized.

Oracle WebLogic Server response file

Ideally, you should capture and save the response file from the GUI based installer. Save the file as `silentInstallWLS.xml` and replace all highlighted text for the Middleware Home and the Oracle WebLogic Server installation directory with values reflective of your environment. The response file for Oracle WebLogicServer will have a format similar to the following:

```
    <?xml version="1.0" encoding="UTF-8"?>
    <bea-installer>
      <input-fields>
        <data-value name="BEAHOME" value="/u02/oracle/middleware" />
        <data-value name="WLS_INSTALL_DIR" value="/u02/oracle/middleware/
    wlserver_10.3" />
        <data-value name="INSTALL_NODE_MANAGER_SERVICE" value="yes" />
        <data-value name="NODEMGR_PORT" value="5559" />
      </input-fields>
    </bea-installer>
```

Oracle SOA Suite response file

Similar to Oracle WebLogic Server, the Oracle SOA Suite response file should ideally be saved from the GUI based installer. Save the file as `silentInstallSOASuite.xml` and replace all highlighted text for the Middleware Home and Oracle Home with values reflective of your environment. The response file for Oracle SOA Suite will have a format similar to the following:

```
[ENGINE]
Response File Version=1.0.0.0.0
[GENERIC]
ORACLE_HOME=/u02/oracle/middleware/Oracle_SOA1
MIDDLEWARE_HOME=/u02/oracle/middleware
CONFIG_WIZARD_RESPONSE_FILE_LOCATION=0
[SYSTEM]
[APPLICATIONS]
[RELATIONSHIPS]
```

Silently installing the products

Oracle JRockit, by default, supports command line installations. Oracle WebLogic Server and Oracle SOA Suite are installed silently. Finally, the SOA domain is created via a scripted approach.

Silently installing Oracle JRockit

Both Sun JDK and Oracle JRockit can be installed via the command line:

1. Ensure that the binary file has the appropriate permissions (on Linux) and start the installer from the Middleware Home directory to immediately extract it there:

   ```
   cd $MW_HOME
   chmod 750 jrockit-jdk1.6.0_31-R28.2.3-4.1.0-linux-x64.bin
   jrockit-jdk1.6.0_31-R28.2.3-4.1.0-linux-x64.bin
   ```

The command line version of the **Oracle JRockit Installer - Oracle JRockit JDK R28.2.3 for Java SE 6 with JRMC 4.1** will now begin.

2. On the **Welcome** page, enter Next:

   ```
   Welcome:
   --------
   Enter [Exit][Next]> Next
   ```

3. On the **Choose Product Installation Directory** page, enter the new installation directory `/u01/share/middleware/jrockit1.6.0_31`:

   ```
   Choose Product Installation Directory:
   --------------------------------------
   Enter new Product Installation Directory OR [Exit][Previous]
   [Next]> /u01/share/middleware/jrockit1.6.0_31
   ```

4. On the next **Choose Product Installation Directory** page, enter `1`:

   ```
   Choose Product Installation Directory:
   --------------------------------------
   Use above value or select another option:
       1 - Enter new Product Installation Directory
       2 - Change to default [/home/oracle/jrockit-
   jdk1.6.0_31-R28.2.3-4.1.0]
   Enter option number to select OR [Exit][Previous][Next]> 1
   ```

5. On the next **Choose Product Installation Directory** page, enter `Next` to accept the previously entered installation directory:

   ```
   Choose Product Installation Directory:
   --------------------------------------
       "Product Installation Directory" = [/u01/share/middleware/
   jrockit1.6.0_31]
   Enter new Product Installation Directory OR [Exit][Previous]
   [Next]> Next
   ```

6. On the **Optional Components 1** page, enter `2` to skip the installation of the demos and samples:

   ```
   Optional Components 1
   Do you want to install Demos and Samples?
       1 - Yes, install Demos and Samples.
     ->2 - No, do not install.
   Enter a number or [Exit][Previous][Next]> 2
   ```

7. On the **Optional Components 2** page, enter `2` to skip the installation of the source code:

   ```
   Optional Components 2
   Do you want to install Source Code?
       1 - Yes, install Source Code.
     ->2 - No, do not install.
   Enter a number or [Exit][Previous][Next]> 2
   ```

8. On the **Installation Complete** page, press Enter to exit the installer:

   ```
   Installation Complete
   Congratulations! Oracle JRockit JDK R28.2.3 for Java SE 6 with
   JRMC 4.1 has been successfully installed to /u01/share/middleware/
   jrockit1.6.0_31.
   Press [Enter] to continue or type [Exit]>
   ```

In fact, once JRockit (or Sun JDK) is installed the first time, there is no need to install it again on any of your other servers. Simply zip up the folder, and unzip it to your other servers.

Silently installing Oracle WebLogic Server

Installing Oracle WebLogic Server silently differs from a standard installation by merely adding the `-mode` and `-silent-xml` arguments to the Java command as shown:

```
export JAVA_HOME=/u02/oracle/middleware/jrockit1.6.0_31
export PATH=$JAVA_HOME/bin:$PATH
java -Djava.io.tmpdir=/var/tmp -jar wls1036_generic.jar -mode=silent
-silent_xml=silentInstallWLS.xml
```

The Java Home and path should be set beforehand. For 64-bit installations, add the `-d64` Java argument.

Silently creating RCU schemas

The RCU provides a silent script to install the Dehydration Store schemas in a silent mode too. Modify the highlighted parameters to reflect your environment:

```
$RCU_HOME/bin/rcu -silent –createRepository -connectString
[dbhost]:[dbport]:[servicename] –dbUser sys –dbRole sysdba –
schemaPrefix [prefix] –component SOA
```

The preceding command will create the required schemas for Oracle SOA and BPM infrastructure and prompt for the database and component schema passwords from the command line. Alternatively, to avoid interaction from the command line, you can create a text file containing all the necessary passwords (one password per line) and then use the -f option to pass this password file to the command line.
For example, if you create a file called passwordfile.txt, you can use the following command:

```
$RCU_HOME/bin/rcu -silent –createRepository -connectString
[dbhost]:[dbport]:[servicename] –dbUser sys –dbRole sysdba -
schemaPrefix [prefix] –component SOA -f < passwordfile.txt
```

Silently installing Oracle SOA Suite

To install Oracle SOA Suite silently, the -silent and -response arguments are passed to the runInstaller command as follows:

```
./runInstaller -jreLoc $JAVA_HOME -silent -response
silentInstallSOASuite.rsp -invPtrLoc /etc/oraInst.loc
-log=silentInstallSOASuite.log
```

As you can see from this command, the JRE location is required as well as references to the response and oraInst.loc files. Optionally, the output can be dumped to a log file via the -log argument.

Silently creating the SOA domain

Creating the SOA domain silently (that is, via scripting) requires creating a custom WLST script. Response files are not leveraged here. The scripts included with this chapter include a sample WLST script to use as a reference.

Once the createDomain.py script and all underlying subscripts are customized, the SOA domain can be created by running the following commands:

```
sh $WL_HOME/common/bin/wlst.sh createDomain.py
```

The preceding command will create and configure the SOA domain in its entirety.

Refer to the ReadMe.txt file provided with the code bundle of this chapter for prerequisites required prior to running the preceding command.

Summary

This chapter covers four core areas that most administrators will likely address at some point; patching, upgrading from 10*g*, installing a high availability cluster, and performing silent installs. These areas are large enough to dedicate a single chapter to each of them, but we have attempted to cover in enough detail the steps to assist you in understanding and executing those activities.

With regards to patching, we described the following:

- Providing a patching strategy and approach
- Patching or upgrading JDK, Oracle WebLogic Server, and Oracle SOA Suite
- Applying applicable post-upgrade activities
- Updating the database schemas and deploying the latest purge scripts
- Verifying the database schema upgrade

To upgrade from Oracle SOA Suite 10*g* to 11*g*, we covered the following:

- Summarizing the upgrade approach
- Configuring the application server, including creation of custom data sources and connection factories
- Migrating BPEL domains and ESB Systems and Service Groups to partitions
- Migrating BPEL domain properties, DVMs, and fault policies
- Importing shared schemas and WSDLs to the MDS

Installing a 2-node cluster is a large effort, but when broken down, is not very complicated. One of the more important aspects is designing how you want the final architecture to look. The installation approach detailed in this chapter consisted of shared binaries for the JDK, Oracle WebLogic Server, and Oracle SOA Suite. Oracle Web Tier was used as the frontend web server, and the entire installation process was detailed with step-by-step instructions.

Finally, if a scripted and non-interactive approach to installing Oracle SOA Suite 11*g* and its related products is required, silent installation is the way to go. We covered:

- Describing response files and how to generate them
- Applying required prerequisites such as the creation of the `oraInst.loc` file
- Customizing the response file and running the silent installers

This was a large chapter indeed with a lot to cover! Fortunately, each of these four main areas can be read individually and used as a reference when needed.