**WebLogic Server 12.2.1.1.0: Create, configure and tune a Domain**

**Introduction**

This post focuses on the creation, configuration and tuning of an initial WebLogic Server 12.2.1.1.0 Domain.

The WebLogic Server is a necessary requirement, and must already be installed. For instruction how to install the WebLogic Server 12.2.1.1 please refer to my previous post: [WebLogic Server 12.2.1.1.0: Installation on the Linux OS](https://borysneselovskyi.wordpress.com/2016/07/31/weblogic_server_12-2-1-1_installation)

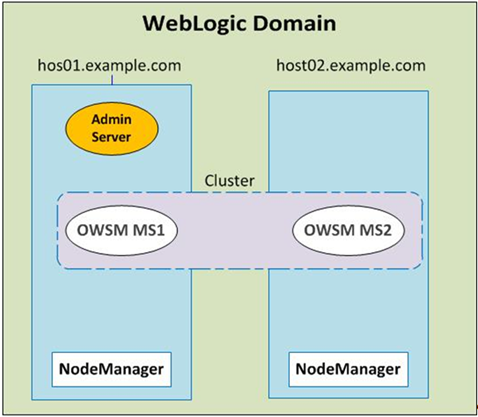
The following tasks will be implemented:

* Preparing the Operating System for the installation of the WebLogic Server 12.2.1.1, Installation JDK 1.8 and the WebLogic Server 12.2.1.1: please refer to [WebLogic Server 12.2.1.1.0: Installation on the Linux OS](https://borysneselovskyi.wordpress.com/2016/07/31/weblogic_server_12-2-1-1_installation)
* Creating of database schemas
* Creating an initial WebLogic Domain
* Post-Installation Tasks:
  + Configuring of components
  + Creating of Start & Stop scripts for the automatic start/stop of components
* WebLogic Domain Tuning and Troubleshooting

The Domain will be reated on two servers (OS OEL 6.7):

* host01.example.com
* host02.example.com

We will configure two managed Server for Web Services in cluster:



**Create Database Schemas (as OS User oracle)**

Note: The creating of the database schemas must be running only once on the first server: host01.

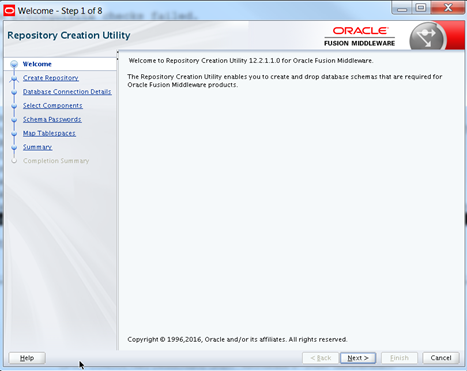
The database must be already created and prepared for the WebLogic Server Domain. For more information about the preparing the database please refer to the [Documentation](http://docs.oracle.com/html/E73100_01/toc.htm#GUID-85F0EAF9-3ACE-421D-BD72-F1A8E6B91116)

Login to the server as user oracle, navigate to oracle\_common directory, and start the Repository Creation Utility (RCU):

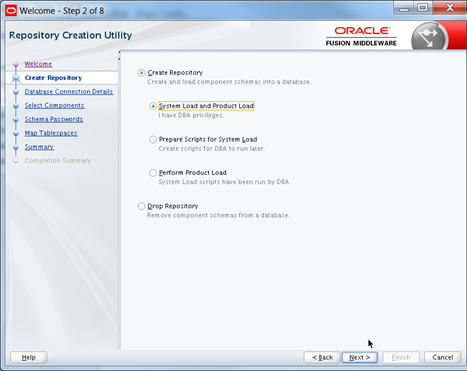
[oracle@host01 oracle]$ cd /u01/app/oracle/product/FMW/Oracle\_Home/oracle\_common/bin

[oracle@host01 bin]$ ./rcu

Step 1: the Welcome page: click „Next“:



Step 2: Create Repository: For the new WebLogic or SOA or another Middleware product choose „Create Repository“ and „System Load and Production Load“:



Step 3: Database Connection Details: On this Page provide the database connection information:

Database Type: Oracle Database

Host Name (database host): host02.example.com

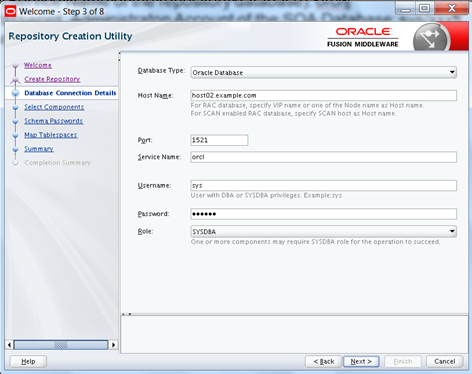
Port (Database Listener Port: 1521)

Service Name (Service Name of the repository database): orcl

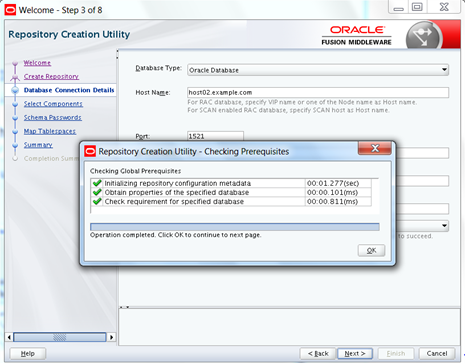
Username & Password: Administration Account of the SOA Database: sys/xxx

Role (Database Admin Role: SYSDBA or SYSOPER): SYSDBA

… „Next“:



The Installer checks the database connectivity:



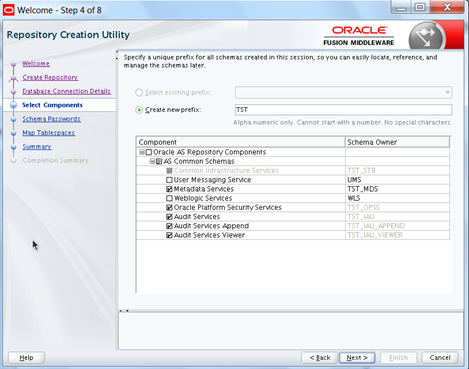
Step 4: Select Components:

Following are mandatory selected components:

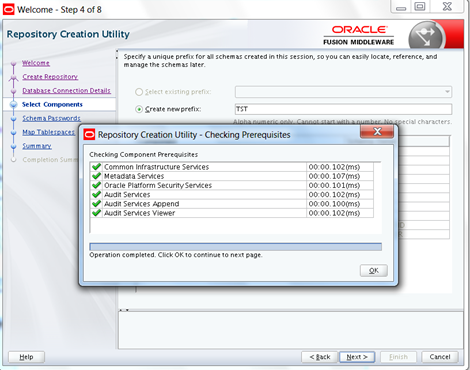
* Common Infrastructure Services (Mandatory selected components cannot be deselected)

Additionally I selected „Oracle Platform Security Services“, „User Messaging Service“ and „Metadata Services“.

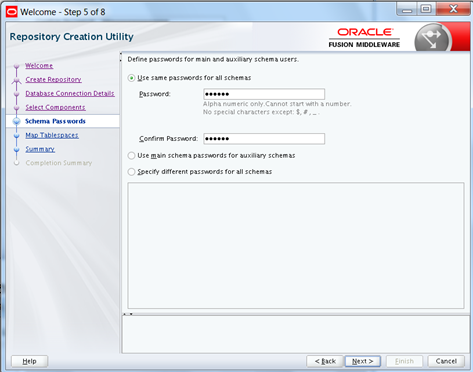
Set the Prefix for database schemas and activate following components…click „Next“:



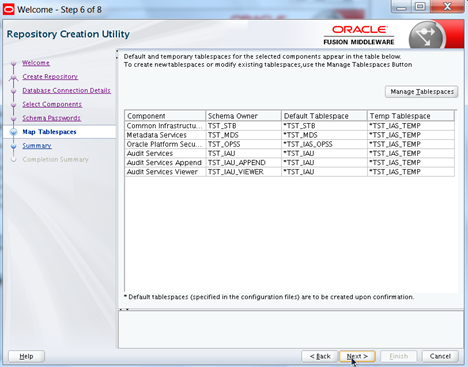
The installer will check database prerequisites:



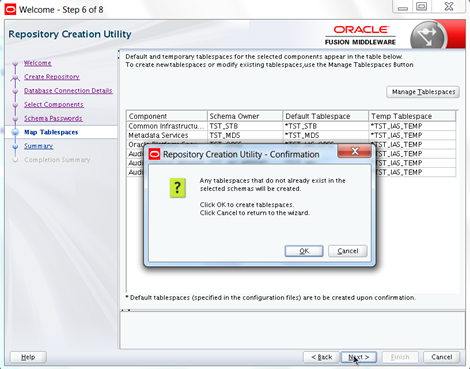
Step 5: Set the password for all database schemas:



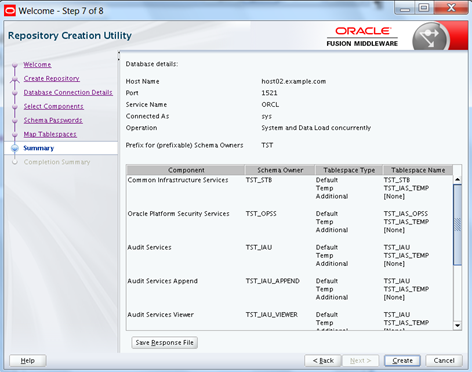
Step 6: Map Tablespaces: On this page you can change the tablespace settings:



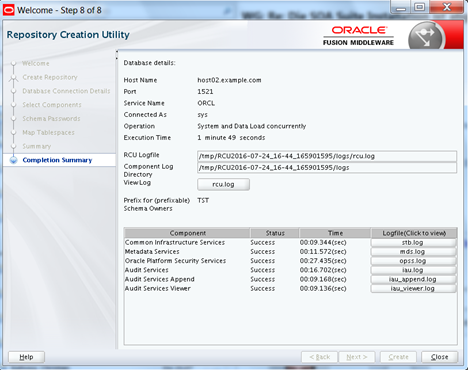
Click „OK“ to create the new tablespace:



Step 7: Summary: check the details and click „Create“:



Check Completion Summary and click „Close“:



The Repository Creation is now completed

**Create initial Domain (as OS User oracle)**

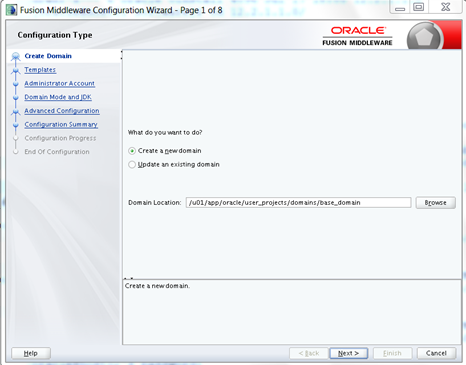
Note: The creating of the WebLogic Server Domain must be running only once on the first server: host01.

Navigate to the oracle\_common directory and start the Configuration Wizard:

[oracle@host01 bin]$ cd /u01/app/oracle/product/FMW/Oracle\_Home/oracle\_common/common/bin

[oracle@host01 bin]$ ./config.sh

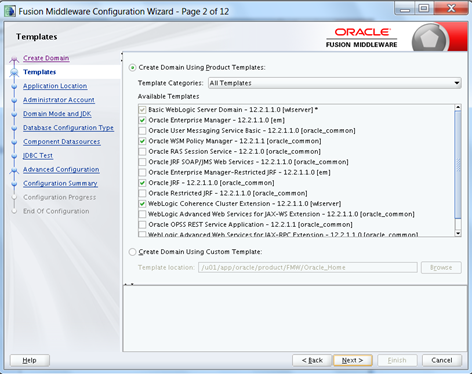
Choose „Create a new Domain“ and set the „Domain Location“ to /u01/app/oracle/user\_projects/domains/base\_domain:



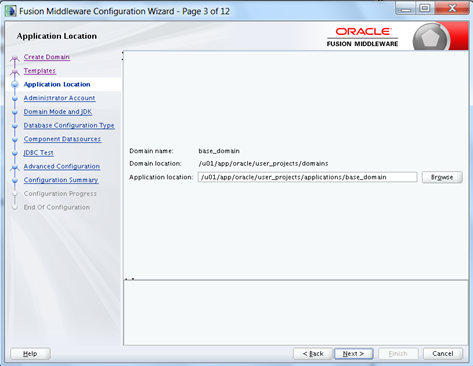
On this page you can add several WebLogic Server templates.

For more information about templates please refer to [Oracle Documentation](https://docs.oracle.com/middleware/1221/wls/WLDTR/fmw_templates.htm#WLDTR227)

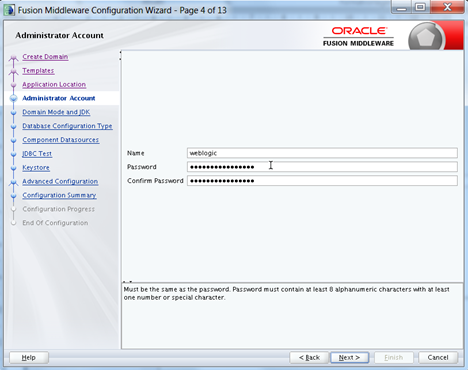
Select OWSM Policy Manager and Enterprise Manager (additional prerequisite components Coherence and JRF will be selected automatically):



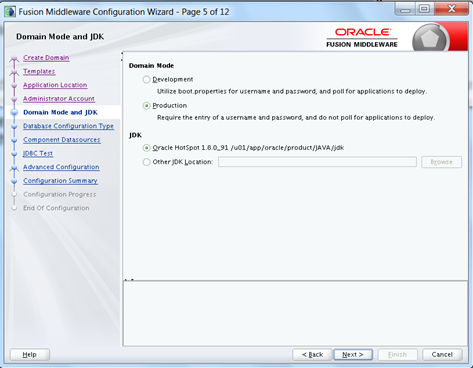
Specify an „Application Location“: /u01/app/oracle/user\_projects/applications/base\_domain:



Set the Name and Password for the WebLogic Administrator:



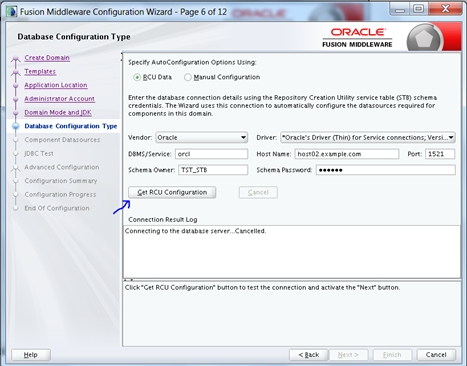
Choose „Domain Mode“: Production and check the JDK location:



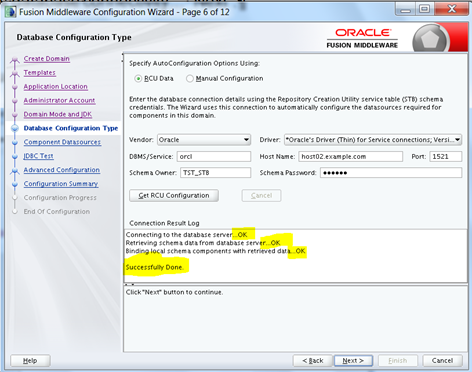
Configure the database details:

Specify the correct schema prefix: TST\_STB and the information what were specified when you previously ran RCU (Chapter „Create Database Schemas“).

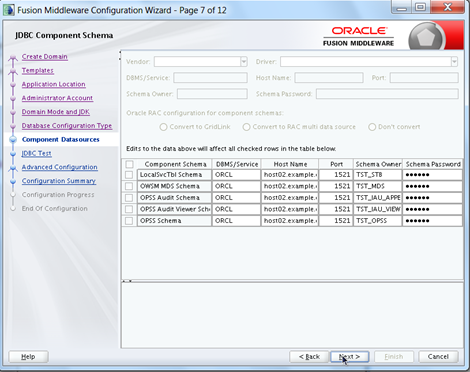
Then click on the Button „Get RCU Configuration“:



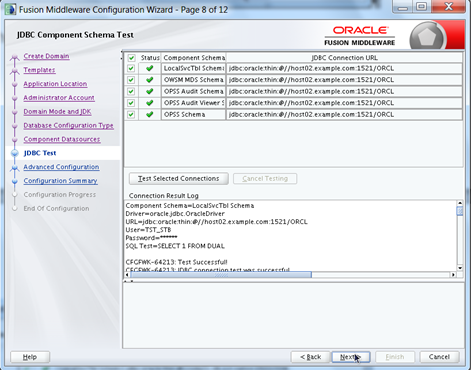
Installer checks database connectivity… „Next“:



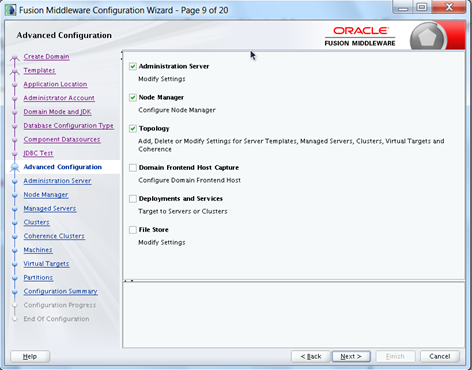
Let default values and click „Next“:



The installer checks the database connections:



Activate check boxes for Administration Server, Node Manager and Topology for the advanced configuration:

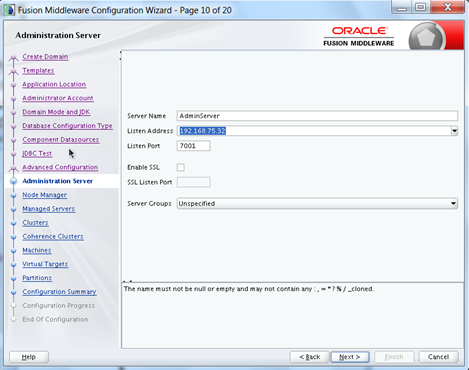


On the next page configure the Administration Server:

Name: AdminServer

Listen Address: 192.168.75.32 (IP Address Server host01.example.com)

Listen Port: 7001

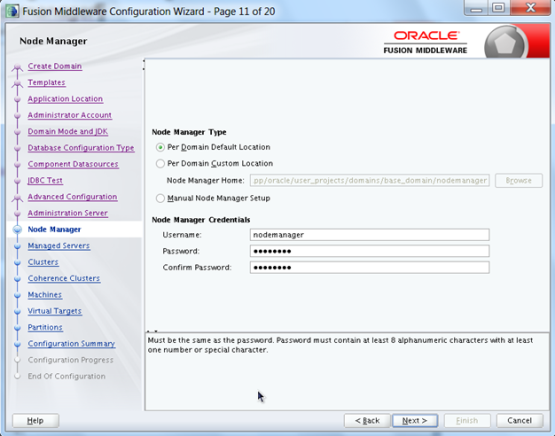


Configure the NodeManager:

Set Node Manager Type to „Per Domain Default Location“ – default

Enter Node Manager Credentials:

Username: nodemanager



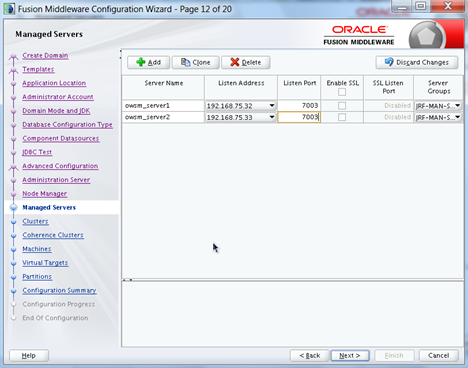
On the page „Managed Servers“ add two Managed Servers with following settings:

|  |  |  |  |
| --- | --- | --- | --- |
| **Managed server** | **Listen Address** | **Port** | **Server Group** |
| owsm\_server1 | 192.168.75.32 | 7003 | JRF-MAN-SVR, WSM-CACHE-SVR,WSMPM-MAN-SRV |
| owsm\_server2 | 192.168.75.33 | 7003 | JRF-MAN-SVR, WSM-CACHE-SVR,WSMPM-MAN-SRV |

In this example we will create two managed servers for web services applications.

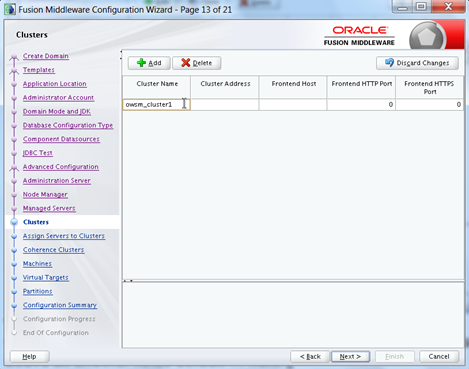
We will place the first managed server owsm\_server1 on the server host01.example.com (IP Address: 192.168.75.32) and the second managed server owsm\_server2 on the host02.example.com (IP Address: 192.168.75.32).

Server groups JRF-MAN-SVR and WSM-CACHE-SVR / WSMPM-MAN-SRV ensure that Oracle JRF and Oracle Web Services Manager (OWSM) services target Managed Servers you are creating. For more info about Server Groups please refer to [Oracle Documentation](https://docs.oracle.com/middleware/1221/core/INFIN/GUID-CA80A6E9-8903-4E19-81D7-A3647A11D0A6.htm#INFIN332)

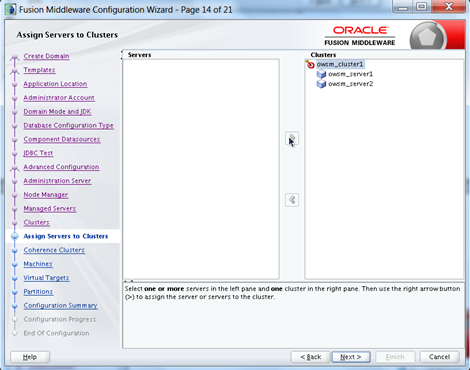


On the Page „Cluster“ click „Add“ and specify Cluster Name „owsm\_cluster1):

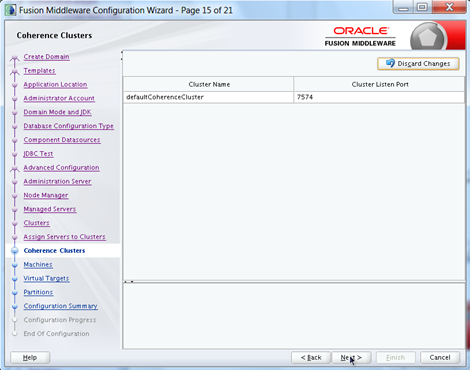
Leave all another fields default … „Next“:



Add two managed servers to the cluster owsm\_cluster1:

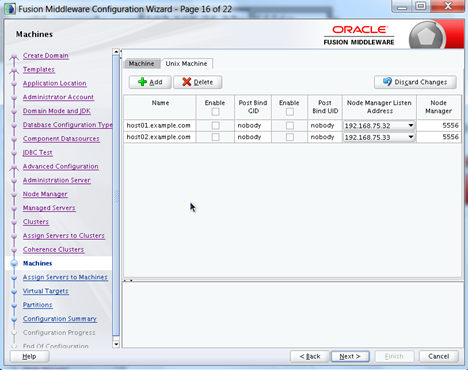


Specify the Coherence Cluster Listen Port:



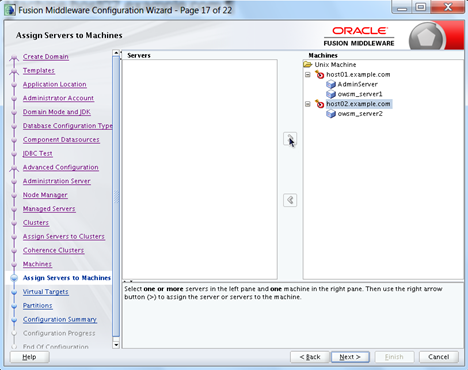
Create two UNIX Machines: one for the host host01.example.com and the second UNIX Machine for the host02.example.com):

|  |  |  |
| --- | --- | --- |
| **Name** | **Listen Address** | **Port** |
| host01.example.com | 192.168.75.32 | 5556 |
| host02.example.com | 192.168.75.33 | 5556 |

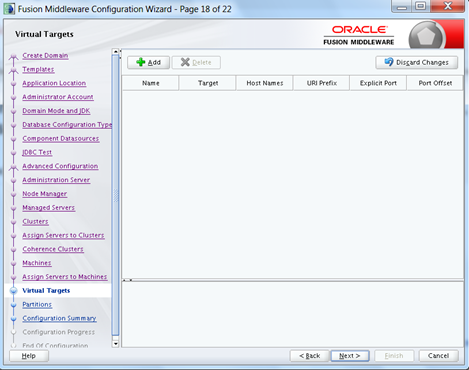


Add the AdminServer and the Managed Server owsm\_server1 to the UNIX Machine host01.example.com.

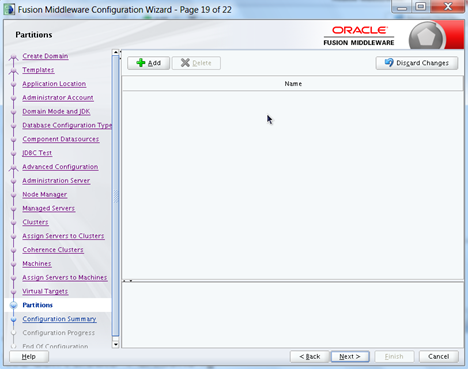
Add the managed server owsm\_server2 to the second UNIX Machine host02.example.com:



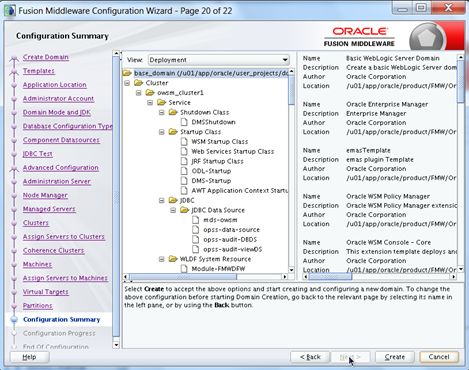
In this example we don’t configure „Virtual Targets“:



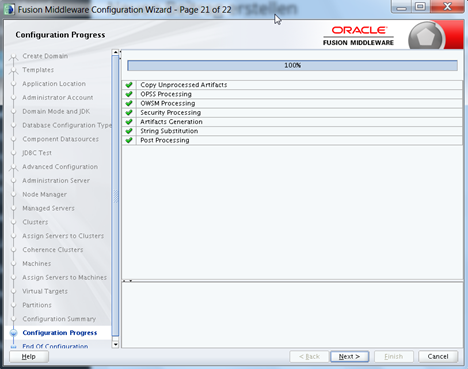
We don’t create „Partitions“:



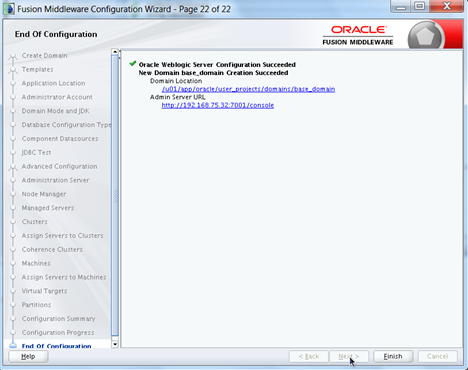
Check configuration summary and press the button „Create“:



The Creation of initial Domain is now completed… „Next“



The WebLogic Server Domain is now created, …“Finish“:



**Start the AdminServer via script.**

Enter the values for the Administration User (weblogic) and passwort when prompted:

[oracle@host01 bin]$ cd /u01/app/oracle/user\_projects/domains/base\_domain/bin

[oracle@soa01 bin]$ ./startWebLogic.sh

...

Enter username to boot WebLogic server:weblogic

Enter password to boot WebLogic server:xxx

**Post-Installation Tasks**

**Disabling the Derby Database**

Note: All tasks described in this chapter must be running on both servers: host01.example.com and host02.example.com

[oracle@host01]$ cd /u01/app/oracle/product/FMW/Oracle\_Home/wlserver/common/derby/lib/

[oracle@host01 lib]$ mv derby.jar disable\_derby.jar

**Create file boot.properties**

Note: All tasks described in this chapter must be running only on the server host01

[oracle@host01]$ cd /u01/app/oracle/user\_projects/domains/base\_domain/servers/AdminServer

[oracle@host01 AdminServer]$ mkdir security

[oracle@host01 AdminServer]$ cd security/

[oracle@host01 security]$ echo -e "username=weblogic\npassword=welcome1" > boot.properties

[oracle@host01 security]$ cat boot.properties

username=weblogic

password=welcome1

**Copying the Domain configuration from the server host01 to the host02**

Save the Domain configuration on the server host01:

[oracle@host01 oracle]$ cd /u01/app/oracle/product/FMW/Oracle\_Home/oracle\_common/common/bin

[oracle@host01 bin]$ ./pack.sh -domain=/u01/app/oracle/user\_projects/domains/base\_domain -template=/home/oracle/base\_domain.jar -template\_name=base\_domain

Output:

<< read domain from "/u01/app/oracle/user\_projects/domains/base\_domain"

>> succeed: read domain from "/u01/app/oracle/user\_projects/domains/base\_domain"

<< write template to "/home/oracle/base\_domain.jar"

..............................

>> succeed: write template to "/home/oracle/base\_domain.jar"

<< close template

>> succeed: close template

Copy the file host01:/home/oracle/base\_domain.jar to the server host02

[oracle@host01 ~]$ scp /home/oracle/base\_domain.jar host02.example.com:/home/oracle

Extract the domain configuration on the server host02.example.com:

[oracle@host02 oracle]$ mkdir -p /u01/app/oracle/user\_projects/domains

[oracle@host02 oracle]$ mkdir -p /u01/app/oracle/user\_projects/applications

[oracle@host02 oracle]$ cd /u01/app/oracle/product/FMW/Oracle\_Home/oracle\_common/common/bin

[oracle@host02 bin]$ ./unpack.sh -user\_name=weblogic -password=welcome1 -domain=/u01/app/oracle/user\_projects/domains/base\_domain -overwrite\_domain=true -template=/home/oracle/base\_domain.jar -log\_priority=DEBUG -log=/tmp/unpack.log -app\_dir=/u01/app/oracle/user\_projects/applications

Output:

Warning: Option "-password" is deprecated. Use "-walletDir" to specify wallet where pw is stored.

Warning: -overwrite\_domain is ignored if this is not a managed server template.

<< read template from "/home/oracle/base\_domain.jar"

>> succeed: read template from "/home/oracle/base\_domain.jar"

<< set config option AppDir to "/u01/app/oracle/user\_projects/applications"

>> succeed: set config option AppDir to "/u01/app/oracle/user\_projects/applications"

<< find User "weblogic" as u1\_CREATE\_IF\_NOT\_EXIST

>> succeed: find User "weblogic" as u1\_CREATE\_IF\_NOT\_EXIST

<< set u1\_CREATE\_IF\_NOT\_EXIST attribute Password to "\*\*\*\*\*\*\*\*"

>> succeed: set u1\_CREATE\_IF\_NOT\_EXIST attribute Password to "\*\*\*\*\*\*\*\*"

<< set u1\_CREATE\_IF\_NOT\_EXIST attribute IsDefaultAdmin to "true"

>> succeed: set u1\_CREATE\_IF\_NOT\_EXIST attribute IsDefaultAdmin to "true"

<< write Domain to "/u01/app/oracle/user\_projects/domains/base\_domain"

...............................................................................................

>> succeed: write Domain to "/u01/app/oracle/user\_projects/domains/base\_domain"

<< close template

>> succeed: close template

NodeManager configuration (as OS User oracle)

Server host1:

Edit file nodemanager.properties:

[oracle@host01 ~]$ cd /u01/app/oracle/user\_projects/domains/base\_domain/nodemanager

[oracle@host01 nodemanager]$ vi nodemanager.properties

Change parameter:

Old:

CrashRecoveryEnabled=false

New:

CrashRecoveryEnabled=true

Start NodeManager:

[oracle@host01 bin]$ cd /u01/app/oracle/user\_projects/domains/base\_domain/bin

[oracle@host01 bin]$ nohup ./startNodeManager.sh &

Server host02:

Edit file nodemanager.properties:

[oracle@host02 ~]$ cd /u01/app/oracle/user\_projects/domains/base\_domain/nodemanager

[oracle@host02 nodemanager]$ vi nodemanager.properties

Change parameter:

Old:

ListenAddress=192.168.75.32

CrashRecoveryEnabled=false

New:

ListenAddress=192.168.75.33

CrashRecoveryEnabled=true

Start NodeManager:

[oracle@host02 bin]$ cd /u01/app/oracle/user\_projects/domains/base\_domain/bin

[oracle@host02 bin]$ nohup ./startNodeManager.sh &

**Stop AdminServer (as OS User oracle)**

Server host01:

[oracle@host01 bin]$ cd /u01/app/oracle/user\_projects/domains/base\_domain/bin

[oracle@host01 bin]$ ./stopWebLogic.sh

**Start AdminServer via NodeManager and WLST on the Server host01 (as OS User oracle)**

Start wlst:

[oracle@host01 oracle]$ /u01/app/oracle/product/FMW/Oracle\_Home/oracle\_common/common/bin/wlst.sh

Connect to NodeManager:

wlst> nmConnect('nodemanager','welcome1', '192.168.75.32','5556','base\_domain','/u01/app/oracle/user\_projects/domains/base\_domain')

Start AdminServer:

wls:/nm/base\_proddomain> nmStart('AdminServer')

Output:

Starting server AdminServer ...

Successfully started server AdminServer ...

**Execute nmEnroll commando on the second server: host02.example.com (as OS User oracle)**

Start wlst:

[oracle@host02]$ /u01/app/oracle/product/FMW/Oracle\_Home/oracle\_common/common/bin/wlst.sh

Connect to AdminServer (AdminServer must be running):

wls:/offline> connect('weblogic','welcome1','t3://192.168.75.32:7001')

Output:

Connecting to t3://192.168.75.32:7001 with userid weblogic ...

Successfully connected to Admin Server "AdminServer" that belongs to domain "base\_domain".

Warning: An insecure protocol was used to connect to the server.

To ensure on-the-wire security, the SSL port or Admin port should be used instead.

Execute nmEnroll commando:

wls:/base\_domain/serverConfig/> nmEnroll('/u01/app/oracle/user\_projects/domains/base\_domain','/u01/app/oracle/user\_projects/domains/base\_domain/nodemanager')

Output:

Enrolling this machine with the domain directory at /u01/app/oracle/user\_projects/domains/base\_domain ...

Successfully enrolled this machine with the domain directory at /u01/app/oracle/user\_projects/domains/base\_domain.

**Restart the NodeManager**

[oracle@host02 bin]$ cd /u01/app/oracle/user\_projects/domains/base\_domain/bin

[oracle@host02 bin]$ ./stopNodeManager.sh

[oracle@host02 bin]$ nohup ./startNodeManager.sh &

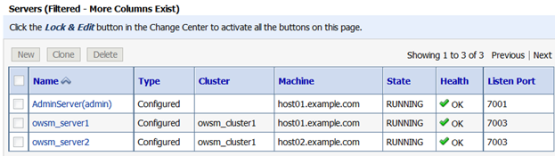
**Start all Managed Servers via AdminConsole**

Open the Browser and go to <http://192.168.75.32:7001/console>

Starting all Managed Servers via WebLogic Server Admin Console:

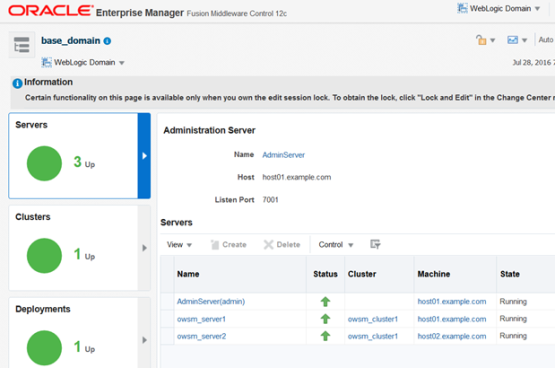


All Managed Servers are running:



Get your familiar with the new look of the Enterprise Manager:

Open the page: <http://192.168.75.32:7001/em>:



**Prepare the WebLogic Server domain for the auto start**

Server host01:

Create directory (as OS User oracle):

[oracle@host01 ~]$ mkdir /u01/app/oracle/scripts

[oracle@host01 ~]$ chmod 744 /u01/app/oracle/scripts

[oracle@host01 ~]$ cd /u01/app/oracle/scripts

Create scripts in the directory /u01/app/oracle/scripts (as OS User oracle)

Create script /home/oracle/scripts/fmw\_nodemanager.sh:

#!/bin/bash

## fmw\_nodemanager.sh

DOMAIN\_HOME=/u01/app/oracle/user\_projects/domains/base\_domain

WORK\_DIR=/u01/app/oracle/scripts

export DOMAIN\_HOME WORK\_DIR

case "$1" in

start)

nohup ${DOMAIN\_HOME}/bin/startNodeManager.sh > ${WORK\_DIR}/nodemanager.out 2>&1 &

;;

stop)

${DOMAIN\_HOME}/bin/stopNodeManager.sh

;;

\*)

echo "usage: $0 [start|stop]"

exit 2

;;

esac

Create script /u01/app/oracle/scripts/fmw\_admin\_and\_ms.sh:

#!/bin/bash

## fmw\_admin\_and\_ms.sh

DOMAIN\_HOME=/u01/app/oracle/user\_projects/domains/base\_domain

ORACLE\_HOME=/u01/app/oracle/product/FMW/Oracle\_Home

WORK\_DIR=/u01/app/oracle/scripts

export DOMAIN\_HOME ORACLE\_HOME WORK\_DIR

WLST=${ORACLE\_HOME}/oracle\_common/common/bin/wlst.sh

export WLST

case "$1" in

start)

${WLST} ${WORK\_DIR}/start\_admin.py

${WLST} ${WORK\_DIR}/start\_ms.py

;;

start\_admin)

${WLST} ${WORK\_DIR}/start\_admin.py

;;

start\_ms)

${WLST} ${WORK\_DIR}/start\_ms.py

;;

stop)

${WLST} ${WORK\_DIR}/stop\_ms.py

${WLST} ${WORK\_DIR}/stop\_admin.py

;;

stop\_admin)

${WLST} ${WORK\_DIR}/stop\_admin.py

;;

stop\_ms)

${WLST} ${WORK\_DIR}/stop\_ms.py

;;

\*)

echo "usage: $0 [start|start\_admin|start\_ms|stop|stop\_admin|stop\_ms]"

exit 2

;;

esac

Create script /u01/app/oracle/scripts/start\_admin.py:

nmConnect('nodemanager','welcome1, '192.168.75.32','5556','base\_domain','/u01/app/oracle/user\_projects/domains/base\_domain')

nmStart('AdminServer')

Create script /u01/app/oracle/scripts/stop\_admin.py:

nmConnect('nodemanager','welcome1, '192.168.75.32','5556','base\_domain','/u01/app/oracle/user\_projects/domains/base\_domain')

nmKill('AdminServer')

Create script /u01/app/oracle/scripts/start\_ms.py:

nmConnect('nodemanager','welcome1, '192.168.75.32','5556','base\_domain','/u01/app/oracle/user\_projects/domains/base\_domain')

nmStart('owsm\_server1')

Create script /u01/oracle/scripts/stop\_ms.py:

nmConnect('nodemanager','welcome1, '192.168.75.32','5556','base\_domain','/u01/app/oracle/user\_projects/domains/base\_domain')

nmKill('owsm\_server1')

Set permissions:

[oracle@host01 scripts]$ chmod -R 747 /u01/app/oracle/scripts

Integrate start-stop-scripts in Linux run-level (as OS user root)

Create script /etc/init.d/oracle (as root):

#!/bin/sh

# chkconfig: 345 99 10

# description: Oracle auto start-stop script.

# Set ORA\_OWNER to the user id of the owner of the

export HOME=/u01/app/oracle/scripts

ORA\_OWNER=oracle

if [ ! -f ${HOME}/fmw\_nodemanager.sh \

-o ! -f ${HOME}/fmw\_admin\_and\_ms.sh   \

-o ! -f ${HOME}/start\_admin.py     \

-o ! -f ${HOME}/stop\_admin.py     \

-o ! -f ${HOME}/start\_ms.py     \

-o ! -f ${HOME}/stop\_ms.py ]

then

echo "Oracle FMW startup: cannot start"

exit

fi

case "$1" in

'start')

su $ORA\_OWNER -c "$HOME/fmw\_nodemanager.sh start"

sleep 120

su $ORA\_OWNER -c "$HOME/fmw\_admin\_and\_ms.sh start"

;;

'stop')

su $ORA\_OWNER -c "${HOME}/fmw\_admin\_and\_ms.sh stop"

su $ORA\_OWNER -c "${HOME}/fmw\_nodemanager.sh stop"

;;

esac

Integrate the script /etc/init.d/oracle in run-levels (as root):

[root@host01 init.d]# chmod 700 /etc/init.d/oracle

[root@host01 init.d]# cd /etc/init.d

[root@host01 init.d]# chkconfig --add oracle

Check ( as root):

[root@host01 init.d]# chkconfig --list oracle

oracle 0:off 1:off 2:off 3:on 4:on 5:on 6:off

Server soa02

Create directory (as OS User oracle):

[oracle@host02 ~]$ mkdir /u01/app/oracle/scripts

[oracle@host02 ~]$ chmod 744 /u01/app/oracle/scripts

[oracle@host02 ~]$ cd /u01/app/oracle/scripts

Create scripts in the directory /u01/app/oracle/scripts (as OS User oracle)

Create script /u01/app/oracle/scripts/fmw\_nodemanager.sh:

#!/bin/bash

## fmw\_nodemanager.sh

DOMAIN\_HOME=/u01/app/oracle/user\_projects/domains/base\_domain

WORK\_DIR=/u01/app/oracle/scripts

export DOMAIN\_HOME WORK\_DIR

case "$1" in

start)

nohup ${DOMAIN\_HOME}/bin/startNodeManager.sh > ${WORK\_DIR}/nodemanager.out 2>&1 &

;;

stop)

${DOMAIN\_HOME}/bin/stopNodeManager.sh

;;

\*)

echo "usage: $0 [start|stop]"

exit 2

;;

esac

Create script /u01/app/oracle/scripts/fmw\_ms.sh:

#!/bin/bash

## fmw\_ms.sh

DOMAIN\_HOME=/u01/app/oracle/user\_projects/domains/base\_domain

ORACLE\_HOME=/u01/app/oracle/product/FMW/Oracle\_Home

WORK\_DIR=/u01/app/oracle/scripts

export DOMAIN\_HOME ORACLE\_HOME WORK\_DIR

WLST=${ORACLE\_HOME}/oracle\_common/common/bin/wlst.sh

export WLST

case "$1" in

start)

${WLST} ${WORK\_DIR}/start\_ms.py

;;

stop)

${WLST} ${WORK\_DIR}/stop\_ms.py

;;

\*)

echo "usage: $0 [start|stop]"

exit 2

;;

esac

Create script /u01/app/oracle/scripts/start\_ms.py:

nmConnect('nodemanager','welcome1, '192.168.75.33','5556','base\_domain','/u01/app/oracle/user\_projects/domains/base\_domain')

nmStart('owsm\_server2')

Create script /u01/app/oracle/scripts/stop\_ms.py:

nmConnect('nodemanager','welcome1, '192.168.75.33','5556','base\_domain','/u01/app/oracle/user\_projects/domains/base\_domain')

nmKill('owsm\_server2')

Set permissions:

[oracle@host02 scripts]$ chmod -R 747 /home/oracle/scripts

Integrate start-stop-scripts in Linux run-level (as OS user root)

Create script /etc/init.d/oracle (as root):

#!/bin/sh

# chkconfig: 345 99 10

# description: Oracle auto start-stop script.

# Set ORA\_OWNER to the user id of the owner of the

export HOME=/u01/app/oracle/scripts

ORA\_OWNER=oracle

if [ ! -f ${HOME}/fmw\_nodemanager.sh \

-o ! -f ${HOME}/fmw\_ms.sh     \

-o ! -f ${HOME}/start\_ms.py     \

-o ! -f ${HOME}/stop\_ms.py ]

then

echo "Oracle FMW startup: cannot start"

exit

fi

case "$1" in

'start')

su $ORA\_OWNER -c "$HOME/fmw\_nodemanager.sh start"

sleep 120

su $ORA\_OWNER -c "$HOME/fmw\_ms.sh start"

;;

'stop')

su $ORA\_OWNER -c "${HOME}/fmw\_ms.sh stop"

su $ORA\_OWNER -c "${HOME}/fmw\_nodemanager.sh stop"

;;

esac

Integrate the script /etc/init.d/oracle in run-levels (as root):

[root@host02 init.d]# chmod 700 /etc/init.d/oracle

[root@host02 init.d]# cd /etc/init.d

[root@host02 init.d]# chkconfig --add oracle

Check (as root):

[root@host02 init.d]# chkconfig --list oracle

oracle 0:off 1:off 2:off 3:on 4:on 5:on 6:off

**Tuning and Troubleshooting the WebLogic Server Domain**

**Tuning JVM (as OS User oracle)**

We will customize the settings for the Java Heap size and for WebLogic Server Start parameters by the creating the File setUserOverrrides.sh.

We will tune the Java heap size for the Admin Server and for each Managed Server.

We will set some WebLogic Server start Parameter:

* -Dweblogic.MaxMessageSize=300000000    # increase the MessageSize
* -Dweblogic.threadpool.MinPoolSize=1024    # Tuning the WebLogic ThreadPool

For more information about the customizing of Java settings please refer to my article

[How to customize Java Virtual Machine Settings in Oracle WebLogic Server 12 on OS Linux / UNIX](https://borysneselovskyi.wordpress.com/2016/07/24/how-to-customize-java-virtual-machine-settings-in-oracle-weblogic-server-12-on-os-linux-unix)

Creating the File setUserOverrides.sh on both hosts: host01.example.com and host02.example.com:

[oracle@host01 ~]$ cd /u01/app/oracle/user\_projects/domains/base\_domain/bin

[oracle@host01 bin]$ vi setUserOverrides.sh

Content of setUserOverrides.sh:

#!/bin/bash

echo "Setting from UserOverrides.sh"

# global settings (for all servers)

export JAVA\_OPTIONS="$JAVA\_OPTIONS -Dweblogic.MaxMessageSize=300000000 -Dweblogic.threadpool.MinPoolSize=1024"

# customer settings for each server

# Admin Server:

if [ "${SERVER\_NAME}" = "AdminServer" ]

then

echo "Customizing USER\_MEM\_ARGS for SERVER\_NAME ${SERVER\_NAME}"

export USER\_MEM\_ARGS="-Xms2g –Xmx2g"

export JAVA\_OPTIONS="$JAVA\_OPTIONS -Djava.awt.headless=true"

fi

# OWSM Managed Server:

if [ "${SERVER\_NAME}" = "owsm\_server1" -o "${SERVER\_NAME}" = "owsm\_server2" ]

then

echo "Customizing USER\_MEM\_ARGS for SERVER\_NAME ${SERVER\_NAME}"

export USER\_MEM\_ARGS="-Xms3g –Xmx3g -XX:+UseParNewGC -XX:+UseConcMarkSweepGC -XX:NewSize=1g"

export JAVA\_OPTIONS="$JAVA\_OPTIONS -Djava.awt.headless=true"

fi

echo "End setting from UserOverrides.sh"

**Tuning Fusion Middleware Control (as OS User oracle)**

Reference: MOS Note „How to Enable Discovery Cache To Avoid Long Delay During Login To Fusion Middleware Control (Doc ID 1423893.1)“

On the server host01 login to wlst with weblogic user and connect to domain environment:

[oracle@host01 ~]$ /u01/app/oracle/product/FMW/Oracle\_Home/oracle\_common/common/bin/wlst.sh

Connect to the AdminServer:

wls:/offline> connect('weblogic','welcome1','t3://192.168.75.32:7001')

Change location to domain custom tree

wls:/base\_domain/serverConfig/> domainCustom()

Set variables:

wls:/base\_domain/domainCustom/> myMBean=ObjectName('emoms.props:Location=AdminServer,name=emoms.properties,type=Properties,Application=em')

wls:/base\_domain/domainCustom/> types=['java.lang.String', 'java.lang.String']

wls:/base\_domain/domainCustom/> type=['java.lang.String']

Set property oracle.sysman.emas.discovery.wls.FMW\_DISCOVERY\_USE\_CACHED\_RESULTS=true:

wls:/base\_domain/domainCustom/> params=['oracle.sysman.emas.discovery.wls.FMW\_DISCOVERY\_USE\_CACHED\_RESULTS', 'true']

wls:/base\_domain/domainCustom/> mbs.invoke(myMBean,'setProperty',params,types)

Set property oracle.sysman.emas.discovery.wls.FMW\_DISCOVERY\_MAX\_CACHE\_AGE=7200000:

wls:/base\_domain/domainCustom/> params=['oracle.sysman.emas.discovery.wls.FMW\_DISCOVERY\_MAX\_CACHE\_AGE','7200000/base\_domain/domainCustom/> mbs.invoke(myMBean,'setProperty',params,types)

Set property oracle.sysman.emas.discovery.wls.FMW\_DISCOVERY\_MAX\_WAIT\_TIME=10000:

wls:/base\_domain/domainCustom/> params=['oracle.sysman.emas.discovery.wls.FMW\_DISCOVERY\_MAX\_WAIT\_TIME','10000/base\_domain/domainCustom/> mbs.invoke(myMBean,'setProperty',params,types)

Invoke getProperty operations to display and confirm parameter values (Values returned by wlst are in red):

Display changes added:

wls:/base\_domain/domainCustom/> param=['oracle.sysman.emas.discovery.wls.FMW\_DISCOVERY\_USE\_CACHED\_RESULTS']

wls:/base\_domain/domainCustom/> mbs.invoke(myMBean,'getProperty',param,type)

'**true**'

wls:/base\_domain/domainCustom/> param=['oracle.sysman.emas.discovery.wls.FMW\_DISCOVERY\_MAX\_CACHE\_AGE']

wls:/base\_domain/domainCustom/> mbs.invoke(myMBean,'getProperty',param,type)

'**7200000**'

wls:/base\_domain/domainCustom/> param=['oracle.sysman.emas.discovery.wls.FMW\_DISCOVERY\_MAX\_WAIT\_TIME']

wls:/base\_domain/domainCustom/> mbs.invoke(myMBean,'getProperty',param,type)

'**10000**'

**Tuning the WebLogic Server Start: Avoiding JVM Delays Caused by Random Number Generation**

Change the JVM random generation from /dev/random to /dev/urandom. For more Information refer to [Oracle Documentation](https://docs.oracle.com/cd/E13209_01/wlcp/wlss30/configwlss/jvmrand.html)

Go to the file java.security and change the parameter securerandom.source:

Server host01 (as OS User oracle):

[oracle@host01 ~]$ vi /u01/app/oracle/product/JAVA/jdk/jre/lib/security/java.security

Old value:

securerandom.source=file:/dev/random

Change to:

securerandom.source=file:/dev/urandom

Repeat the step on the Server host02.example.com.

Restart the Admin and all Managed Servers.

Declare i=0

Declare j=0

Advertisemen