Oracle Utilities Network Management System

Installation Guide

Release 2.5.0.2

F54378-01

March 2022

ORACLE

Installation Guide, Release 2.5.0.2

F54378-01

Contents

Preface

The information in this document is intended to guide you through a successful Oracle Utilities Network Management System installation or upgrade.

Audience

This document is intended for administrators and engineers responsible for installing and upgrading Oracle Utilities Network Management System.

Related Documents

For more information, see the following documents in the Oracle Utilities Network Management System Release 2.5.0.2.0 documentation set:

- *Oracle Utilities Network Management System Adapters Guide*
- *Oracle Utilities Network Management System Advanced Distribution Management System Implementation Guide*
- *Oracle Utilities Network Management System Configuration Guide*
- *Oracle Utilities Network Management System for Water User's Guide*
- *Oracle Utilities Network Management System Licensing Information User Manual*
- *Oracle Utilities Network Management System Operations Mobile Application Installation and Deployment Guide*
- *Oracle Utilities Network Management System Quick Install Guide*
- *Oracle Utilities Network Management System Release Notes*
- *Oracle Utilities Network Management System Security Guide*
- *Oracle Utilities Network Management System User's Guide*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Chapter 1

Introduction

The information in the *Installation Guide* is intended to take you through a successful Oracle Utilities Network Management System installation or upgrade.

- Chapter 2 - Pre-Installation describes the prerequisite steps to ensure that your environment is ready to install the Oracle Utilities Network Management System. It is important that you understand and complete these prerequisite steps before you begin the Oracle Utilities Network Management System installation.
- Chapter 3 - System Installation provides step-by-step installation and setup instructions for the Oracle Utilities Network Management System.
- Appendix A - Applying Migrations for a New Release provides information for comparing and merging your configuration files with a new Oracle Utilities Network Management System release.
- Appendix B - Configuring a Translated User Interface provides information about installing and configuring a language translation.

Resources

- Information about Third Party software packages can be found in the *Oracle Utilities Network Management System Licensing Information User Manual.*
- Further information on configuring your system can be found in the *Oracle Utilities Network Management System Configuration Guide.* Note, however, that you must complete the tasks described in this guide, in the required order, before proceeding to any post-installation tasks.
- Information on installation and deployment of the Oracle Network Management System Operations Mobile Application can be found in the *Oracle Network Management System Operations Mobile Application Installation and Deployment Guide*.

Product Release Naming Conventions

Oracle Utilities Network Management System product releases occur as General Availability (GA) Releases, Service Packs, and Patch Bundles.

- **GA Releases** (for example, 2.**3**.0, 2.**4**.0, 2.**5**.0, and so forth) are complete (new) binary releases with full documentation sets and Quality Assurance (QA) processes.
- **Service Packs** (for example, 2.4.0.**1**, 2.4.0.**2**, 2.4.0.**3** …) are complete (new) binary releases containing a limited set of new features and all bug fixes since the last GA or Service Pack release. Service Packs include targeted QA testing, which is based on current customer platforms and module usage.
- **Patch Bundles** (for example, 2.4.0.0.**1**, 2.4.0.0.**2**, 2.4.0.0.**3** …) - These contain only bug fixes and changed files since the last GA release or service pack. This release will go through QA bug regression testing to verify the system will start and perform basic functionality. Patch bundles are cumulative and the latest patch bundle will always contain **all** patches for that service pack.

System Overview

The Oracle Utilities Network Management System can be broken down into individual components. Each component is installed and configured separately. Oracle Utilities Network Management System uses a client/server architecture. The server supports Oracle Utilities Network Management System daemon processes, while the clients display a graphical user interface to allow the user to interact with the system. Internal daemon service process to daemon service process communication is managed with a concurrency management and messaging system called Isis. Isis is the backbone of the communication architecture for an Oracle Utilities Network Management System. The network model, system configuration, and operational data is all stored persistently in an Oracle database.

The table below describes the Oracle Utilities Network Management System components.

Component	Description
Client User Environments	The Java-based end-user environments are configured using a combination of SQL files (RDBMS table based configuration), XML files, and Java properties files. The XML files are based on an NMS-specific XML schema, which provides the foundation for Java user interface customization.
Isis	Clients access services and tools through a central concurrency management and messaging system called Isis. Isis is a real-time implementation of message oriented middleware that helps provide access to the Oracle Utilities Network Management System daemon service processes as well as inter-daemon process communication.
Services	Services maintain and manage the real-time electrical network data model. Services also cache information from the database tables to optimize client information access.
Oracle WebLogic	Oracle WebLogic hosts Oracle Utilities Network Management System specific Enterprise Java Beans (EJBs). These EJBs help cache the network model and process updates/requests to/from Java clients as well as to/from external systems.
Web-Gateway	The Web-Gateway is a CORBA (Common Object Request Broker Architecture) interface between Network Management System daemon processes and WebLogic EJBs.
Oracle Database	The Oracle Database contains the complete network data model, configuration, and operational data history of an Oracle Utilities Network Management System.

Note: Services, applications, and the Oracle RDBMS tablespaces can be spread over multiple servers or run on a single server. The simplest configuration is for everything (Oracle RDBMS, Oracle Utilities Network Management System services and Oracle WebLogic Java Application Server to run on a single (generally SMP) server. Common variations would include the use of a cluster based hardware server to support high-availability (for Oracle RDBMS and Oracle Utilities Network Management System Services). This provides flexibility for system configuration, depending on your needs and hardware.

Chapter 2

Pre-Installation

This chapter provides an overview of the installation requirements for Oracle Utilities Network Management System and provides additional information that you should read before you begin the installation process. This chapter includes the following topics:

- **Installation Prerequisites**
- Operating System and Administrative User Setup

Installation Prerequisites

This section includes the following topics:

- Introduction
- Requirements for Oracle Utilities Network Management System Database
- Requirements for Spatial Landbase
- Requirements for Java Application Server
- Software Requirements for Unzipping Files
- Security Considerations
- Isis Directory and NTP Daemon
- Client Authentication

Introduction

In order to successfully install the Oracle Utilities Network Management System (and underlying environment), you must have a thorough understanding of the following:

- Unix system administration.
- Oracle RDBMS installation and configuration.
- WebLogic installation and EJB deployment.

In addition, you should have at least a cursory knowledge of the Oracle Utilities Network Management System architecture and applications functionality.

The *Oracle Utilities Network Management System Quick Install Guide* provides an overview of the Oracle Utilities Network Management System architecture and lists the supported

hardware and software configurations. Verify that your system meets system requirements prior to attempting the Oracle Utilities Network Management System installation.

Requirements for Oracle Utilities Network Management System Database

The Oracle RDBMS must be installed and configured before beginning the Oracle Utilities Network Management System. It must be installed on a server that is accessible from the Oracle Utilities Network Management System services applications. Time zone settings for the Oracle RDBMS and Oracle Utilities Network Management System services application must be the same.

Oracle recommends that any new installations should use the AL32UTF8 character set.

Refer to the "Database Configuration" chapter of the *Oracle Utilities Network Management System Configuration Guide* for more details regarding the installation and configuration of the Oracle RDBMS for use with the Oracle Utilities Network Management System applications.

> **Note:** Oracle Locator, a standard component of all editions of Oracle RDBMS, is required for the Oracle Utilities Network Management System installation.
>
> To verify that Oracle Locator has not been removed from your Oracle RDBMS, run the following SQL command:
>
> ```
> select count(1) "Rows" from mdsys.cs_srs;
> ```
>
> If this query returns zero or very few rows (1000+ are expected), consult your DBA to (re)install the Oracle Locator package.

Requirements for Java Application Server

The Oracle Utilities Network Management System requires the installation of a WebLogic Application Server. This installation should be done before installing the Oracle Utilities Network Management System software. Refer to *Oracle Fusion Middleware Installation Guide for Oracle WebLogic Server* for instructions.

Requirements for Spatial Landbase

The Oracle Utilities Network Management System recommends installation of the Oracle Map Builder to simplify the process of creating and managing map, theme, and symbology metadata in the spatial database used to render spatial landbase maps. Oracle Map Builder is part of the Oracle Fusion Middleware MapViewer family of products, which is available for download from the Oracle Technology Network website (www.oracle.com/technetwork/).

Requirements for Web Maps Landbase

The Oracle Utilities Network Management System supports the use of commercial web map servers for Viewer background landbases. It is up to each NMS customer to license with a web map provider for access to a web map service. NMS currently supports the following web map services:

- Google Static Maps Basic or Trial - To configure, you will need to agree to a license agreement with Google and they will provide you with an API key. This option is not recommended for production.
- Google Static Maps for Work/Business - To configure, you will need to agree to a license agreement with Google and they will provide you with a process to define Client IDs and Google generated Crypto Keys.
- Bing Maps - To configure, you will need to agree to a license agreement with Bing/Microsoft and they will provide you with a key.

Software Requirements for Unzipping Files

The Oracle Utilities Network Management System and third party software files are compressed in the ZIP format. Most Linux/Unix-based platforms already have the binaries needed to unzip the distribution archives. An unzip utility is also included in the Oracle client under `$ORACLE_HOME/bin`.

Security Considerations

Please refer to the *Oracle Utilities Network Management System Security Guide* for security overview, recommendations, and guidelines when installing Oracle Utilities Network Management System software.

OpenSSL

Oracle Utilities Network Management System now uses OpenSSL to encrypt traffic between SwService (used in CVR and FLISR) and WebLogic server by default. As such, OpenSSL is now required to be installed with the operating system if you will be running SwService. The openssl binary as well as libssl and libcrypto are required.

To make them available, you will need to install the appropriate OS packages on your system:

- **Linux:** openssl and openssl-devel RPM's
- **Solaris:** library/security/openssl package

Isis Directory and NTP Daemon

Isis is the backbone of the Oracle Utilities Network Management System. It is the messaging bus through which all back-end NMS service components communicate.

On any computer using Isis it is important to have an accurate clock, which moves monotonically forward. Many approaches, such as rdate, can cause the clock to jump unpredictably, possibly backwards. This jumping is especially deleterious to Isis timing and timer queues, but can easily be avoided by using the Network Time Protocol (NTP) daemon, which is designed to gracefully synchronize the system clock with any reliable time source.

NTP is available for free on all operating systems and is simple to configure. Even if all of your services and applications run on a single computer, it is important to run NTP there. If you have several computers on the same LAN, you may want to consider running an NTP server (pointing to an external time source) on one of them and pointing all of the other NTP clients on the LAN to it. All NMS servers should be configured to be in the same timezone, and have their time synchronized with ntp, including all WebLogic servers.

Refer to the "Isis Configuration" chapter of the *Oracle Utilities Network Management System Configuration Guide* for more details regarding the configuration of the Isis message bus for use with the Oracle Utilities Network Management System applications.

Client Authentication

Authorization and roles are stored in the database, but the authentication must come from an external source. No matter what the authentication source, each login name must be granted access to the system by using the Configuration Assistant. Active Directory and LDAP are supported authentication sources.

Operating System and Administrative User Setup

In order to install Oracle Utilities Network Management System software, you must have your system administrator configure a Linux/Unix environment and setup the environment for the Oracle Utilities Network Management System administrative user.

Topics in this section include the following:

- Dual-Environment Configuration
- Administrative User Configuration
- Korn Shell Configuration
- Operating System Configuration
- Core File Naming Configuration

Dual-Environment Configuration

In a dual-environment configuration (used to reduce downtime when installing patches), two administrative users are needed on the same host. This is an implementation of a patching strategy that is more commonly called blue/green patching. While the "blue" environment is active, the "green" environment can be staged to minimize the time it takes to make it active - and vice-versa. Both of these users must be members of the same Unix group. Follow the instructions in the Administrative User Configuration section for both users.

Administrative User Configuration

The administrative user (for example, nmsadmin) owns and controls the services. The administrative user also owns and maintains the binaries, and all the configuration standards.

The Administrative Unix user:

- Owns the Isis release directories.
- Starts and stops Isis processes.
- Performs installs.
- Owns the executable directories.
- Has read-write permissions to the production database.
- Owns the service processes (DBService, MTService, etc.) and performs all sms-start commands.

Creating an Administrative User

The administrative user, as the name implies, has central control over many critical aspects of the Oracle Utilities Network Management System. This user is the central controller of:

- Isis: configuration and starting and stopping of the Isis processes. See the *Oracle Utilities Network Management System Configuration Guide* for information on configuring Isis.
- Oracle Utilities Network Management System services: stopping and starting services and administering service logs.
- Oracle Utilities Network Management System files: binaries, configuration files.
- Database connection for example, read/write privileges.
- Model-building data.

It should be noted that for data security, Oracle Utilities Network Management System tools that can be used to directly modify data are installed with permissions set so that only the administrative user is allowed to execute them.

The administrative user has access to critical components of the system. This user owns and maintains the services, the starting of the services, model building, binaries, the database, and the configuration standards. The administrative user maintains the Oracle Utilities Network Management System Unix-based configuration and executables in one location.

A given NMS instance requires at least one NMS admin Unix user/account. For a single NMS admin environment this could be something like *nmsadmin* but for a dual NMS environment (to support expedited NMS patching) there are two NMS Unix admin accounts (*nmsadmin-1* and *nmsadmin-2* – for example). In either case there is only one NMS RDBMS admin user/schema (nms_prod – for example). The NMS Unix admin account is where NMS Services will execute. The NMS RDBMS admin schema is used to construct and maintain the NMS RDBMS schema.

- The NMS RDBMS admin schema handles the SQL Data Definition Language (DDL) statements necessary to construct and maintain (patch) the NMS schema. The RDBMS admin schema/account is NOT generally used directly by NMS Services for a running NMS production instance – only for patching. Once the Unix admin account is setup according to this document the "ISQL –admin" shortcut uses the $RDBMS_ADMIN environment variable to access the admin schema.
  - ISQL -admin
- The $RDBMS_ADMIN environment variable must be different from the $RDBMS_HOST environment variable as they are used to look up and access different credentials for different schemas in the $NMS_HOME/etc/wallet. The $RDBMS_ADMIN and $RDBMS_HOST variables CAN (ultimately) be two alternate "net_service_name" entries in a local tnsnames.ora configuration file – as noted in the example above (PRODSERV01ADM and PRODSERV01).
- NMS Services (and WebLogic) use an "access" schema to perform routine SQL Data Manipulation Language (DML) updates to a production NMS instance. There are generally two NMS RDBMS access schemas (nms_prod_1 and nms_prod_2 – for example). These "access" schemas contain only synonyms that point to desired tables within the nms_prod (admin) schema.
- Only one access schema is fully utilized at a given time for a given NMS instance. One is for production and the other for staging an NMS patch – and they alternate purposes (sometimes called blue/green patching). For example, the nmsadmin-1 Unix account would typically have $NMS_ENVIRONMENT=1 and would leverage the nms_prod_1 access schema for operations and vice-versa for the nmsadmin-2 Unix account (it would have $NMS_ENVIRONMENT=2 and would leverage the nms_prod_2 access schema) – by convention.

  Note: Even if dual NMS admin environments are NOT desired an NMS access schema is still required (typically $NMS_ENVIRONMENT=1). From a NMS Unix admin and RDBMS schema perspective the only difference between a single NMS admin installation and a dual NMS admin installation is the lack of a second NMS Unix admin account and its associated second RDBMS access schema.
- The ISQL utility (with no options) uses the $RDBMS_HOST environment variable to locate a similarly named RDBMS net_service_name entry in the tnsnames.ora file to determine which RDBMS access schema and credentials to use for a given NMS Unix admin environment. Each NMS admin environment manages its own set of Oracle wallets and credentials so the associated access schema will be as well. The RDBMS access schema that a given NMS admin environment uses will be identified by the $ORACLE_READ_WRITE_USER environment variable within a given NMS Services environment. Once an NMS admin environment is properly configured the ISQL utility will allow access to

the default $ORACLE_READ_WRITE_USER access schema within that environment.

- ISQL

For more information on the dual NMS admin account scheme see the NMS Configuration Guide.

Korn Shell Configuration

Oracle Utilities Network Management System uses the Korn Shell to set environment variables and provide the command line interface to the operating system. The Korn Shell, also referred to as ksh, standardizes command line execution and requests, such as running scripts, executing applications, and operating the services. The Korn Shell uses a file called .profile to configure itself. The administrative account needs to have:

- Its default shell set to ksh.
- Its .profile configured to source the Oracle Utilities Network Management System configuration file (.nmsrc).

For your convenience, templates of a generic .profile and .nmsrc file are included in the Oracle Utilities Network Management System software distribution, under $NMS_BASE/templates. These files can be copied to $NMS_HOME/.profile and $NMS_HOME/.nmsrc and then modified to suit your installation.

.profile Configuration

The Korn Shell .profile is a hidden file that exists in the user's home directory. When a user logs in, this file executes, setting environment variables and defining terminal configuration. The following is required for setting up .profile or the Oracle Utilities Network Management System administrative user.

Edit the .profile file to source the user environment configuration file (.nmsrc) by adding the following line to the end of the file (using any text editor).

```
. ~/.nmsrc
```

This runs .nmsrc in the current shell and initializes all of the environment variables within the .nmsrc file in the current working environment.

The .profile file must also execute correctly when called from another script, as well as when the user logs in at a terminal. Anything in .profile that is terminal-specific should be placed in an "if" clause to suppress execution if the .profile is not being run from a terminal.

```
# Set a variable to be true when .profile is
# being run from a terminal rather than a script.
#
if tty -s
then
TTY=true;
else
TTY=false;
fi
#
# Protect items that must only be run from a
# terminal and not from a script.
#
```

```
if $TTY
then
stty Compaq
tset -I -Q
PS1="`hostname`>"
fi
```

Executables/Run-Times

The Oracle Utilities Network Management System Unix-based software is installed in the product home directory ($NMS_BASE/bin). When commands are entered at the prompt, the shell looks for the appropriate bin directory for a matching program. The PATH environment variable determines where the shell looks for the bin directory, so PATH must be modified to include the location of the Oracle Utilities Network Management System software. It is defined in the .nmsrc file located in the user's home directory and it may contain multiple path names, each separated with a colon (:). The shell parses each path name until the corresponding program is located or each path name is exhausted.

> **Note:** The .nmsrc file sets up the PATH environment variable to ensure that the correct executables are discovered in the correct order. If you need to modify the PATH environment variable, it should be done in the .profile, after the .nmsrc is run, and you should only append directories to the end of the list. Doing otherwise could cause problems with your system.

Operating System Configuration

A standard operating system installation will often not be optimally configured to work with an Oracle Utilities Network Management System. Sometimes the user will spawn more processes than allowed by the standard kernel configuration. Other times, a map file may require a larger data segment than the average user. Due to problems like these, you may find that you will have to tweak the operating system configuration, which may include reconfiguring the kernel or some other part of your Unix system.

The values that are specified in this guide are examples only, as the correct values depend on how large your operating model is, how you use the system (for example, as a server, app-server, or client) and what kind of a load is placed on the system. This section should give you an idea of how to change components of the operating system that frequently become a problem running Oracle Utilities Network Management System.

Note on Oracle Support Policy on VMWare: Refer to My Oracle Support knowledge base article 249212.1 for Oracle's support policy on VMWare.

Linux

In Linux, limits to data segment size and the number of files available to the user are defined by the ulimit command. For the most part, these parameters do not need to be tweaked, but should you need to, you can run:

ulimit -d <*datasegment size in kilobytes*>	`256 Mb` (usually sufficient)
ulimit -n <*number of file descriptors*>	`1024` (usually sufficient)

Ensure your host has appropriately sized kernel buffers to support high volumes of NMS UDP messaging. For example, on Linux, you can run the following to see current configuration:

```
/sbin/sysctl -q net.core
```

For a large or very active NMS system ensure the following parameters are set reasonably - below are suggested values:

```
   net.core.rmem_default = 8388608
   net.core.rmem_max = 8388608
   net.core.wmem_default = 524288
   net.core.wmem_max = 1048576
   net.core.netdev_max_backlog = 2000
net.core.optmem_max = 2524287
```

You need to be root to update the Linux `/etc/sysctl.conf` file and reboot after changing it (safest if changes are made). Note you can generally change the above parameters on the fly, but they will NOT be retained on reboot unless you update the `/etc/sysctl.conf` file to match.

Example of how to change one of the noted config values on the fly:

```
sysctl -w net.core.rmem_default=8388608
```

> **Note:** The values above are examples that have been tested and shown to generally yield reasonable results. It is entirely possible that other values may be appropriate for your NMS model and hardware.

Solaris

In Solaris, limits to data segment size and the number of files available to the user are defined by the ulimit command. For the most part, these parameters do not need to be tweaked, but should you need to, you can run:

ulimit -d <*datasegment size in kilobytes*>	`256 Mb` (usually sufficient)
ulimit -n <*number of file descriptors*>	`1024` (usually sufficient)

Core File Naming Configuration

Standard Unix configuration generally names core files as "core" and places it in the directory where the executable was executed. This is problematic because the core file will get overwritten if another core file is generated, which will destroy information that could possibly be used to better track down the source of the problem. Fortunately, there are operating specific steps to have core files be saved with process specific names.

Linux

Note the following may be the default on the Linux distributions supported by Oracle Utilities Network Management System.

```
echo "1" > /proc/sys/kernel/core_uses_pid
```

You should also verify that the ulimit for core files is set to **unlimited**; otherwise, core or truncated core files may not be created:

```
ulimit -c unlimited
```

Solaris

As the root user, edit `/etc/coreadm.conf`:

```
(COREADM_INIT_PATTERN=core.%p)
```

Or run:

```
coreadm -i "core.%p"
```

You should also verify that the ulimit for core files is set to **unlimited**; otherwise, core or truncated core files may not be created:

```
ulimit -c unlimited
```

Chapter 3

System Installation

This chapter describes the Oracle Utilities Network Management System installation. Topics include:

Installation Steps

Before you begin installing Oracle Utilities Network Management System, ensure that you have read and met all pre-installation requirements identified in the previous chapters. Those chapters contain important information with which you must be familiar before you begin the installation so you can avoid potential problems during the installation.

- If this is a first-time installation of Oracle Utilities Network Management System software, follow all steps in this guide starting with **Installing Oracle Utilities Network Management System Software** on page 3-4.
- If you are upgrading from a previous Oracle Utilities Network Management System software release, follow the steps outlined in **Upgrading to NMS v2.5.0.2** on page 3-2.

Upgrading to NMS v2.5.0.2

Prior to Oracle Utilities Network Management System 2.5.0.2, a single Oracle RDBMS schema was used for both Data Definition Language (DDL) and Data Manipulation Language (DML) purposes with a second (read-only) access schema used by Oracle NMS clients. Oracle Utilities Network Management System 2.5+ uses at least three Oracle RDBMS schemas.

1. An "admin" schema that that owns all relevant tables and views and allows DDL (for installations and patching).
2. At least one read-write schema that is essentially a list of synonyms that provide DML access to "admin" schema tables and views. If the dual environment setup is chosen - there will be two (parallel) read-write schemas that can be used for access during expedited patching.
3. A read-only schema that is used by WebLogic to provide read-only access to a subset of "admin" schema tables and views. Like the read-write schema - the read-only schema is strictly a set of synonyms (no actual tables or views in this schema). This schema has not fundamentally changed for the 2.5.0.2 release and is only listed for completeness.

The "admin" schema essentially equates to the DDL portion of the "primary" schema for releases of Oracle Utilities Network Management System prior to 2.5.0.2. The new read-write schema (or schemas) equate to the DML portion of the older (single) "primary" schema. This is an important transition that must be worked through for an Oracle Utilities Network Management System 2.5.0.2 upgrade.

Upgrading the Oracle Utilities Network Management System should be done on a test system prior to attempting an upgrade on a production system. Make a complete copy of the production system on a test system, including the file system and the database. Once the test system is running, follow the steps below to upgrade your test system to Oracle Utilities Network Management System Release 2.5.0.2.0. Follow the instructions based on what release you currently have implemented.

When satisfied with your test system, complete these same steps to upgrade your production system.

1. Log in as the administrative user (for example, nmsadmin).
2. Stop all services including Isis.
   - For releases prior to NMS 2.5.0.0.0, use the following command:

     ```
     sms_stop.ces –ai
     ```

   - For systems running NMS 2.5.0.0.0, use the following command:

     ```
     sms-stop -ai
     ```

     Note: If your system does not support the sms_stop.ces or sms-stop script, use:

     ```
     Action -force any.any stop
     cmd shutdown
     ```

3. Make sure the Naming Service is not running.

   ```
   ps -ef | grep tao_cosnaming
   ```

   If the Naming Service is running, the output will be similar to the following:

```
nmsadmin 348204  1  0  Aug 11   -  0:46 /opt/oms-9.1/bin/
Naming_Service -p /users/oms1/logs/Naming_Service.pid iiop://
server.example.com:17821 -ORBEndpoint
```

If a process is running (user = nmsadmin), kill it:

```
kill [PID]
```

From the example output above, the command would be:

```
kill 348204
```

4. Stop the currently running WebLogic application server.
5. Complete all steps in **Installing Oracle Utilities Network Management System Software** on page 3-4.

   Note: This release uses new templates to help properly configure the software. Please pay careful attention to ensure you use the new templates and their settings.
6. Complete all steps in **Starting Isis** on page 3-7.
7. If your Oracle database version is supported by Oracle Utilities Network Management System 2.5.0.2.0, you can skip the Create Database Environment section. Otherwise, do the following:
   - Backup your current Oracle Utilities Network Management System database.
   - Install the new version of the Oracle RDBMS.
   - Complete all steps in **Create Database Environment** on page 3-7, matching the configuration of your previous RDBMS instance.
   - Import your current Oracle Utilities Network Management System database onto the new RDBMS installation.
8. If upgrading from a release prior to 2.5.0.0.0, drop the old read only schema and create new read only and read-write schemas based on `$NMS_BASE/templates/nms.sql.template`.
9. Complete setting up the project configuration directory and upgrading your model following the steps below:
   - If you have not already done so, move your project configuration files into the $NMS_CONFIG directories as described in **Project Configuration Directory** on page 3-57.
   - Execute the nms-install-config script, which will merge your project configuration with the product configuration and place the results in the runtime directory:

   ```
   nms-install-config --nojava
   ```

10. Follow the procedures in Appendix A-Applying Migrations for a New Release.
11. Execute step 6 of **Validation Model Setup** on page 3-9 to enable write permissions for the user that runs the Java Application Server
12. Complete all steps in **Web Application Configuration** on page 3-10.
13. Complete all steps in **Starting Services** on page 3-59.
14. Complete all steps in the **WebLogic Server Runtime Configuration** on page 3-22.

15. Complete all steps in **Deploying Oracle Utilities Network Management System in WebLogic Server** on page 3-33.
16. If you will be using the web client application installers, complete all steps in **Installing Web Clients** on page 3-37.

Installing Oracle Utilities Network Management System Software

Use the following procedures to install Oracle Utilities Network Management System software.

1. Log in as the administrative user (for example, nmsadmin).
2. Set the NMS_ROOT, NMS_HOME, and NMS_BASE environment variables. For example:

```
   export NMS_ROOT=/users/nmsadmin
   export NMS_HOME=/users/nmsadmin
export NMS_BASE=$NMS_ROOT/nms/product/2.5.0.2.0
```

3. Set the ORACLE_HOME environment variable. For example:

```
export ORACLE_HOME=/users/oracle/product/19.9
```

4. Set the JAVA_HOME environment variable to the 64-bit JDK installation directory. For example:

```
export JAVA_HOME=/opt/java8
```

5. Set PATH and LD_LIBRARY_PATH variables

```
   export PATH=$NMS_BASE/3rdparty/bin:$ORACLE_HOME/bin:$PATH
export LD_LIBRARY_PATH=$NMS_BASE/3rdparty/lib:$ORACLE_HOME/lib
```

6. Unzip the Oracle Utilities Network Management System "Base Software" zip file. For example:

```
unzip /path/to/filename.zip
```

7. In the `/etc` directory, create an empty file named `nmstab`.

   nmstab is used as a repository of NMS installations on a machine. To install NMS, the following must be true:

   - The NMS_HOME environment variable must be set.
   - `nmstab` must exist and be writable.

   The install script (nms-install) will check these requirements and update nmstab.

   - NMS admin users must be members of the same group and nmstab be writable by that group.

   If the group and nmstab requirements cannot be met, nms-install can be run with the `noNmsTab` option. However, if this is done, the nms-list-installs script and OEM integration will not be available.

   `nmsRoot` (found in `$NMS_ROOT/network/bin`) must be run only once per machine as root to create nmstab. The name of the NMS group is passed as a

parameter to this script. Optionally, the name of the owner to use for nmstab can be passed as the second parameter (root is used by default).

```
$NMS_ROOT/network/bin/nmsRoot
```

8. Run the install script:

```
   cd network
./nms-install
```

   Note: this could take several minutes to complete.

9. Remove the installation files before continuing:

```
   cd ..
rm –rf network
```

10. If you already have an existing `.profile`, then append the following line to the bottom:

```
. $NMS_HOME/.nmsrc
```

    This ensures that your environment is set correctly at login.

11. If you have an existing Oracle Utilities Network Management System resource file with a name other than `$NMS_HOME/.nmsrc` (.ces.rc, .cesrc, …), rename it to `$NMS_HOME/.nmsrc`. Move all project-specific environment variables out of the `.nmsrc` file into your .profile file or another resource file.

12. Change the environment variables set in the `$NMS_HOME/.nmsrc` file using the nmsrc configuration script by executing this command:

```
$NMS_BASE/bin/nms-env-config
```

    Set each variable as appropriate for your environment.

    Notes:

    - **Default Values**

      The first time you run nms-env-config you will need to pay close attention to the values that are presented to you as defaults, and ensure that they are set correctly. During subsequent runs you will be presented with the current settings for each variable as the default, and can simply press return at each prompt, reducing the time it takes to run the script.

      When nms-env-config runs, it will flag variables that are not set to the defaults from the standard template. We encourage the use of defaults as much as possible to help facilitate customer support. However, it is up to the customer to decide if deviating from the defaults is appropriate for their environment.

    - **Upgrading from a Release Prior to 2.5.0.0.0**

      The RDBMS_ADMIN variable is new in 2.5.0.0.0. If upgrading from a release prior to 2.5.0.0.0, set RDBMS_ADMIN to the old value of RDBMS_HOST. Set the new value of RDBMS_HOST to be a different value from RDBMS_ADMIN that has an entry in tnsnames.ora that connects to the same database as the entry matching RDBMS_ADMIN.

    - **Oracle Wallet Security Configuration**

      All projects have to configure and maintain Oracle wallets for each Unix user that starts services that connect to the database.

The script prompts you to choose your wallet location, which sets the TNS_ADMIN environment variable (default: ~/etc/wallet). Then it asks you to create passwords and enter the database credentials. Your new Oracle wallet will then map the RDBMS_ADMIN and RDBMS_HOST environment variables to user names and passwords stored in the wallet.

RDBMS_ADMIN and RDBMS_HOST must have different values with entries in tnsnames.ora pointing to the same database instance. Example tnsnames.ora entry:

```
PRODSERV01ADM, PRODSERV01.world =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = db.example.com)
       (PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = PRODSERV01.world)
    )
)
```

IVR_RBDMS_HOST also needs wallet credentials if you are using IVR.

SwService uses a separate Oracle wallet to store credentials it uses to connect to WebLogic Server. The script prompts you to choose a wallet location, and then sets the NMS_WALLET environment variable (default: $NMS_HOME/etc/nms_wallet). If the wallet does not exist it will be created. In that case you will be prompted to create the wallet password. Then the script asks you to enter credentials to be used by SwService. The credentials specified must be a valid WebLogic user that is a member of the group name specified by weblogic-service-group in $NMS_CONFIG/jconfig/build.properties (defaults to nms-service).

Note: WebLogic user authentication configuration is described in Chapter 13 of the *Oracle Utilities Network Management System Configuration Guide*. WebLogic users are defined in Authentication Providers within the security realm. Real users are typically configured via LDAP or some other directory service. However, for performance reasons, it is recommended that the user credentials used by external adapters (including SwService) be defined in a provider that is internal to the WebLogic installation and that the internal provider is configured to be the first one used; this avoids potential delays caused by latency to the network-based directory services. Ensure that the users defined in build.propeties are defined with the internal Weblogic provider: publisher.ejb-user, config.multispeak_runas_user, and config.ws_runas_user

This script should be run for each Unix user that runs NMS services.

Note: After running nms-env-config, you must log out and log back in to set the environment variables.

13. After making the above changes, log out and log in to set the environment variables. For a list of environment variables set by nms-env-config and their descriptions, see the "Environment Configuration" chapter of the *Oracle Utilities Network Management Systems Configuration Guide*.

14. Execute the following commands to copy the templates from $NMS_BASE/templates directory to $NMS_HOME/etc. If you have existing files in $NMS_HOME/etc, they will be backed up to <file>.bak.<timestamp>.

```
nms-install-templates
```

Starting Isis

Please refer to the "Isis Configuration" chapter of the *Oracle Utilities Network Management System Configuration Guide* for details on configuring and optimizing Isis.

1. Start Isis, as follows:

```
    $ nms-isis stop
$ nms-isis start
```

2. When complete (which will take approximately one minute), type:

```
$ nms-isis status
```

This determines if Isis has successfully started and will return 1 if isis is running, 0 if isis is not running.

Create Database Environment

Note: this procedure is only necessary if you do not have an existing Oracle Utilities Network Management System database.

Use the following procedure to create a database environment: for an Oracle Utilities Network Management System.

Note: this step, the process of defining NMS roles, should only be executed once per Oracle RDBMS instance that is supporting one or more NMS instances.

1. Copy the nms_role.sql.template file to a a project specific directory where you can save configuration files that you only run when you are creating new NMS instances. The resulting nms_role.sql file defines the necessary Oracle roles to support an NMS instance.

   Read the comments in the file regarding recommended use. For most projects the contents of this file should not need to be modified and it can be executed essentially as is.

```
$ mkdir $NMS_HOME/my_sql

$ cp $NMS_BASE/templates/nms_role.sql.template $NMS_HOME/my_sql/
nms_role.sql

$ cd $NMS_HOME/my_sql

$ sqlplus system/<system_passwd>@$RDBMS_HOST < nms_role.sql
```

   If this is the first time you have run this, you may see errors about dropping roles that do not exist, which can generally be safely ignored. To be sure run the file in again and make sure there are no errors on the second attempt (the script should drop and create the required roles without error).

2. Read through the comments in the nms.sql.template file. This file **must** be modified for your installation - often significantly. It is a template for creating the necessary Oracle usernames, schemas, and tablespaces to support an Oracle NMS instance.

   For example, if your company name is Oracle Gas & Electric (oge), you might create a copy of the template for a test NMS instance like this:

   ```
   $ cp $NMS_BASE/templates/nms.sql.template $NMS_HOME/my_sql/
   oge_nms_test.sql
   ```

3. Edit *oge*_nms_test.sql and follow the instructions (included as comments in the file) to suit your environment.

4. Run nms.sql as follows:

   ```
   cd $NMS_HOME/my_sql

   sqlplus sys/<system_passwd>@$RDBMS_HOST as sysdba <
   oge_nms_test.sql
   ```

   If this is the first time you have run this, you may see errors about objects that already exist (or may not exist), which can generally be safely ignored. To be sure, run the file in again and make sure there are no errors on the second attempt. On the second attempt the script should drop and create Oracle usernames and tablespaces without error.

5. Log in as the Oracle Utilities Network Management System Oracle RDBMS administrative user and test the connection to Oracle - using appropriate parameters based on what was provided in the oge_nms_test.sql file above. At the prompt, enter something similar to the following:

   ```
   sqlplus oge_nms_test/oge_nms_test_passwd@$RDBMS_HOST
   ```

   If the connection is successful, a `SQL>` prompt will appear. Enter `exit` to return to the shell prompt.

6. Log in as the Oracle Utilities Network Management System Oracle RDBMS read-only user and test the connection to Oracle - using appropriate parameters based on what was provided in the oge_nms_test.sql file above. At the prompt, enter something similar to the following:

   ```
   sqlplus oge_nms_test_ro/oge_nms_test_ro_passwd@$RDBMS_HOST
   ```

   If the connection is successful, a `SQL>` prompt will appear. Enter `exit` to return to the shell prompt.

Validation Model Setup

Use the following procedure to install an Oracle Utilities Network Management System installation verification network data model:

1. If you do not have an existing network data model to load at this point, you can use the OPAL validation model included in the Oracle Utilities Network Management System release.

2. Log in as the administrative user and run `nms-env-config --base-config` using the following variables:

```
NMS_CONFIG_ORDER="OPAL nms"
SYMBOLOGY_SET=${OPERATIONS_MODELS}/SYMBOLS/OPAL_SYMBOLS.sym
NMS_CONFIG=$NMS_HOME/OPAL
```

3. Log out and log back in again, ensuring the variables are set correctly.

   Note: If you have previously installed the validation model, you should backup any existing local modifications before proceeding to step 4.

```
$ mkdir ~/OPAL-backup
$ cd $NMS_CONFIG
$ cp sql/OPAL_parameters.sql jconfig/build.properties jconfig/
build.xml jconfig/nms*keystore jconfig/global/nms-
client.keystore jconfig/global/properties/
CentricityTool.properties jconfig/server/
CentricityServer.properties ~/OPAL-backup/.
```

4. Copy the OPAL configuration in $NMS_BASE to NMS_CONFIG:

```
$ cd $NMS_HOME
$ rm -rf $NMS_CONFIG
$ cp -r -L $NMS_BASE/OPAL $NMS_CONFIG
```

5. If you backed up files from a previously installed validation model (in step 3), merge those files back into the newly copied $NMS_CONFIG directory (from step 4).

   For each file, if the file from the backup does not exist in the new structure, simply copy it into place in $NMS_CONFIG. If $NMS_CONFIG already contains a file with the same name as the backup copy, then you will need to merge changes from the backup copy into the new file.

6. Run nms-make-symbols, nms-install-config, and nms-setup script to load the schema and configuration, as follows:

```
$ nms-make-symbols
$ nms-install-config --nojava
$ nms-setup -clean -reset
```

7. Enable write permissions for the web map directory so the user that runs the Java Application Server (for example, wls) can create files. This is done to enable the distribution of maps to web clients through the application server. If the Java Application Server is running on the same system as the NMS services, this location would be the $OPERATIONS_MODELS/ser directory:

```
$ cd $OPERATIONS_MODELS
$ mkdir ser
$ su
  Password:
# chown wls:users ser
# exit
```

If the Java Application Server is running on a different system than the NMS services, verify the WEB_tempDirectory (as defined in ces_parameters config table where attrib='WEB_tempDirectory') exists on the system where the Java Application Server is running and that the directory is writable to the user running the Java Application Server.

8. Create the directory for the model files:

```
mkdir -p $OPERATIONS_MODELS/mp
```

9. Load the sample data:

```
$ LoadOPALModel
```

The script will load sql files, start a subset of Oracle Utilities Network Management System services, and then build the data model.

Web Application Configuration

Before installing the web applications, follow these steps:

1. Create backups of the following parameters files, if applicable. Note that NMS_PROJECT is the name of your configuration project, for example, OPAL.

```
$ cp $NMS_CONFIG/sql/NMS_PROJECT_parameters.sql
     $NMS_CONFIG/sql/NMS_PROJECT_parameters.sql.bak

$ cp $NMS_CONFIG/sql/NMS_PROJECT_site_parameters_1.sql
$NMS_CONFIG/sql/NMS_PROJECT_site_parameters_1.sql.bak

$ cp $NMS_CONFIG/sql/NMS_PROJECT_site_parameters_2.sql $NMS_CONFIG/
sql/NMS_PROJECT_site_parameters_2.sql.bak
```

2. Copy the parameters files to your $NMS_CONFIG sql directory:

```
$ cp $NMS_BASE/product/sql/nms_parameters.sql
     $NMS_CONFIG/sql/NMS_PROJECT_parameters.sql

$ cp $NMS_BASE/OPAL/sql/OPAL_site_parameters_1.sql $NMS_CONFIG/
sql/NMS_PROJECT_site_parameters_1.sql

$ cp $NMS_BASE/OPAL/sql/OPAL_site_parameters_2.sql $NMS_CONFIG/sql/
NMS_PROJECT_site_parameters_2.sql
```

3. Navigate to the $NMS_CONFIG/sql directory.

```
$ cd $NMS_CONFIG/sql
```

4. In the $NMS_CONFIG/sql directory, modify the parameters (described in the table below) in *NMS_PROJECT*_parameters.sql, *NMS_PROJECT*_site_parameters_1.sql, and *NMS_PROJECT*_site_parameters_2.sql (*NMS_PROJECT*_site_parameters_2.sql is only necessary if using dual-environment configuration). Refer to the backup files made in step 1, if applicable.

Notes

The WebLogic Server (WLS) needs access to the nmsadmin data directory to get *.mad/*.mac files to turn into *.ser files for use by the NMS Viewer. The WLS can either be running on the same machine as the NMS services or on a different machine.

- If the WLS is running on the same system (not normally done), the WLS can be configured to read the nmsadmin data files directly using a specific file path (`WEB_mapDirectory`) and setting `WEB_syncMaps = false`.
- If the WLS is running on a different server than the NMS services (normally recommended), then you will need to set up a httpd server and set `WEB_syncMaps = true`.

NFS does not work as a solution to make the remote NMS server data directory available to the WLS server. NFS can have delays getting files to the WLS meaning the NMS WLS can potentially read old map data causing the NMS Viewer maps to have errors or show incorrect data. You can either install and configure your own httpd server (for example, Apache) or use the **lighttpd http server**, which is shipped with the base NMS product. Usage of **lighttpd** is discussed in the following section. The `nms-lighttpd` script may be used to start and stop the **lighttpd** process; the script is also shipped as part of the base NMS product and can be configured into your normal NMS startup/shutdown/restart process.

An httpd server is also required to serve up log files to the Oracle Utilities Network Management System Application Management Pack for Oracle Enterprise Manager. You can use your own server or use the lighttpd http server. The `nms-lighttpd-oem` script may be used to start and stop this instance of the lighttpd process. This script is also shipped as part of the base NMS product and can be configured into your normal NMS startup/shutdown/restart process.

Web Application Parameters

Element	Description	Example
WEB_intersysName	The `WEB_intersysName` should match the `implName` of the CORBA gateway. Normally this will be `InterSys_{user}`. This is a site-specific parameter.	`InterSys_nmsadmin`
WEB_syncMaps	If false, it will look for the maps using a file location specified in WEB_mapDirectory. If it is true, it will instead load the maps using http and a web server would have to be installed on the NMS server with the data directory exposed. A standard httpd server is provided with the NMS release and can be started with the `nms-lighttpd` command. Configure `WEB_mapDirectory`, `WEB_mapHttpdPort`, `WEB_mapHttpdAllowedIPs`, and `WEB_tempDirectory` before starting the httpd server.	`true`

Element	Description	Example
WEB_mapDirectory	The location of the maps directory from the perspective of the WebLogic server. This can be either a file path or, if `WEB_syncMaps` is set to true, a location that starts with http://. If using the nms-lighttpd process, this would be set to: `http://<nms-server-name>:<WEB_mapHttpdPort>` **Note:** When `WEB_syncMaps=false`, `WEB_mapDirectory` is the same as `$OPERATIONS_MODELS`. In dual-environment configuration, each environment must have a different value for this parameter. This is a site-specific parameter.	`http://nms-vm:8888`
WEB_mapHttpdPort	The port to use for the httpd server started with the nms-lighttpd command start process. This port must be an unique and available port on the NMS C++ Server system. If running multiple NMS environments on the same machine, please verify this value is unique for this machine. This is a site-specific parameter.	`8888`
WEB_mapHttpdAllowedIPs	List of IP addresses where you will be running the WEB Application Servers that are allowed to access the map files from the httpd server. Separate the IP addresses with vertical bar symbols (|) if there are multiple servers. This is used by the nms command start process. If this value is blank, it will not restrict any IP addresses from reading the map files.	`192.168.107.128|192.168.107.1|127.0.0.1`
WEB_mapHttpdHost	The host name or IP address that the lighttpd httpd server listens for connections on. If specified, this should match the server name from the WEB_mapDirectory property. If not specified or left blank, it will default to 0.0.0.0 (bind to all network interfaces). This is a site-specific parameter.	`192.168.107.18`

Element	Description	Example

Note:
If you are using the lighttpd http server and make changes to `WEB_mapDirectory`, `WEB_mapHttpdPort`, `WEB_mapHttpdAllowedIPs`, or `WEB_mapHttpdHost`, you can restart the http server using these commands:
`nms-lighttpd stop`
`nms-lighttpd start`		

WEB_oemHttpdPort	The port to use for the httpd server started with the nms-lighttpd command start process. This port must be an unique and available port on the NMS C++ Server system. If running multiple NMS environments on the same machine, please verify this value is unique for this machine. This is a site-specific parameter.	`8888`
WEB_oemHttpdAllowedIPs	List of IP addresses where you will be running the WEB Application Servers that are allowed to access the map files from the httpd server. Separate the IP addresses with vertical bar (pipe) symbols (|) if there are multiple servers. This is used by the nms command start process. If this value is blank, it will not restrict any IP addresses from reading the map files.	`192.168.107.128|19 2.168.107.1|127.0.0.1`
WEB_oemHttpdHost	The host name or IP address that the lighttpd httpd server listens for connections on. If specified, this should match the server name from the WEB_mapDirectory property. If not specified or left blank, it will default to 0.0.0.0 (bind to all network interfaces). This is a site-specific parameter.	`192.168.107.18`

Note:
If you are using the lighttpd http server and make changes to `WEB_oemHttpdPort`, `WEB_oemHttpdAllowedIPs`, or `WEB_oemHttpdHost`, you can restart the http server using these commands:
`nms-lighttpd-oem stop`
`nms-lighttpd-oem start`		

WEB_tempDirectory	This is the directory to store cached and serialized map files. It should be a directory writable by the WebLogic Managed Server. This is a site-specific parameter.	`/users/nmsadmin/dist/ maps`

Element	Description	Example
WEB_corbaInitRef	The initial reference of the CORBA naming service. It is in the format of `NameService=corbaloc:iiop:1.2@[host]:[port]/NameService`. The {host} and {port} should match the values of the NMS server's CORBA naming service. This is a site-specific parameter.	`NameService=corbaloc:iiop:1.2@server.example.com:17821/NameService`
WEB_watermark	This is transparent text that will display across the windows of an application diagonally. It can be used to make it very obvious what environment you are currently a part of. If you do not wish to use this feature, do not define this value.	`Production` `Test`
WEB_envName	This is the name of the environment that will display in the main window header under the logged in users' name. It can either be the same value as the watermark or an additional name as desired, such as the name of the system.	`Production` `Test`
WEB_envType	This parameter specifies the NMS environment type. Currently, the recognized values are: • **training:** This environment type is required to use the Training Simulator. It allows Trainer user type and the creation/execution of Training Scenario sheets. • **production:** This environment type should be used in production. It disallows the Reset Model functionality (used by Training Simulator, but can be configured for use in non-production environments for testing purposes). Other values can be used, but do not have any effect other than indicating that this is neither a training nor production environment.	`production`
WEB_documents	This parameter controls the disk location where documents are stored for the Manage Documents Window. This is a subdirectory of the $OPERATIONS_MODELS directory and defaults to `drawings`.	`drawings`

Element	Description	Example

MBS_GEO_PROJ_COORDSYS	The projection of the geographic data being loaded into the NMS model. This parameter is required to convert map geographic coordinates to latitude/longitude values. This is used to request maps from web map servers and other functions within the Web Workspace client such as Coordinate display in the Viewer.
	Use the Proj website as a reference on setting this value: https://proj.org/
	Most projects can specify this as a epsg number; you can use this website to find the espg value: https://epsg.io/
	If you need to define a custom projection, that can be done using the +proj= format.
		`epsg:3734`
		
		`or`
		
		`+proj=lcc +lat_1=41.7 +lat_2=40.43333333333333 +lat_0=39.66666666666666 +lon_0=-82.5 +x_0=600000 +y_0=0 +ellps=GRS80 +units=us-ft +no_defs`

MBS_LL_PROJ_COORDSYS	The target Lat/Long specification to be used when converting coordinates from the geographic coordinates to Lat/Long. This parameter is required to convert map geographic coordinates to latitude/longitude values. Us the references specified above in the MBS_GEO_PROJ_COORDSYS to learn how to set this value. Most projects will set this to epsg:4326 fro Lat/Long.	`epsg:4326`

Note: The WEB_ *properties* may be set by adding a startup parameter to the WebLogic managed server's startup parameter. For example, to configured a watermark for a particular managed server, add this:

```
"-Dwatermark=Read Only System"
```

5. If services are not running, you need to run the following command:

   ```
   sms-start
   ```

   Note: The load will fail if services are not running. For example, services would not be running if you are performing a new project installation.

6. When the above changes have been made, run the following commands:

   ```
   $ cp $NMS_CONFIG/sql/NMS_PROJECT_parameters.sql $NMS_HOME/sql
   $ ISQL NMS_PROJECT_parameters.sql
   ```

7. Starting with Java SE 7 Update 21 in April 2013 all Web Start Applications like Oracle Utilities Network Management System are encouraged to be signed with a trusted certificate for the best user experience. Please follow the steps in the **Security Certificates in Oracle Utilities Network Management** section found in the **Security** chapter of the *Oracle Utilities Network Management Security Guide*.

For purposes other than Production, and if you do not have certificates issued by trusted certificate authorities, you may use the nms-gen-keystore script to generate self-signing certificates:

```
$ nms-gen-keystore
```

You will be prompted for your server name (as entered in a browser) and your organization.

Note: Future update releases of Java will require changing Java security settings on the client to continue to run self-signed applications.

8. Verify that the version of WebLogic you are running matches the supported version of WebLogic defined in the `$NMS_CONFIG/jconfig/build.properties`.

```
   # The WebLogic version to deploy to.
weblogic.version = 12.2.1.3.0
```

9. Edit key.server.name to reflect the DNS name or IP address of the host or cluster that the client will connect to.

10. Verify the value of client.url in `$NMS_CONFIG/jconfig/build.properties`. This is the URL that the clients will use to contact the managed server. It should start with `t3s://` if encryption is used (which is recommended) or `t3://` if the link should not be encrypted.

11. The publisher.ejb-user in `$NMS_CONFIG/jconfig/build.properties` should be a user that is defined as part of the nms-service group. This account will be used when WebLogic performs an internal operation that is not done in response to a user action.

    Note: If the build.properties file does not exist in `$NMS_CONFIG/jconfig/`, you will need to create it. See "Create or Modify the Project build.properties File" section in the *Oracle Utilities Network Management System Configuration Guide* Java Application Configuration chapter for details.

12. If you wish to configure WebLogic to not use SSL/HTTPS, then edit `$NMS_CONFIG/jconfig/build.properties`. Add or change (uncomment) the following line:

```
option.no_force_https
```

13. If you will be running multiple instances of Oracle Utilities Network Management System, you will need to create JDBC Data Sources for each WebLogic managed server, each with a unique JNDI name (see WebLogic Runtime Configuration below). To change the JNDI name from the default of *`jdbc/intersys`*, edit `$NMS_CONFIG/jconfig/build.properties` and modify the following line (you may need to uncomment the line):

```
config.datasource = jdbc/intersys/nmsadmin
```

14. If you are running the application as part of a WebLogic cluster, uncomment the following line:

```
enable.cluster = true
```

15. Once all files are in place, build the configuration by running:

```
cd
nms-install-config --java
```

Copy Supporting Files from the NMS Distribution to the WebLogic Domain

Certain files are required to be installed into the domain level of the WebLogic server. Since WebLogic installations vary, it is necessary to manually copy these files to your WebLogic domain.

As you use the instructions below to copy the files, substitute your system's appropriate values for each of:

- `$NMS_BASE`
- `WLS_HOME`
- `domain_name`
- `user`
- `hostname`

Alternative 1

If the WebLogic Server is located on the **same** system as the NMS installation.

1. Copy the contents of the `wls` directory recursively using **cp**:

```
$ cp -L -r $NMS_BASE/dist/install/wls/.
 $WLS_HOME/user_projects/domains/domain_name
```

Alternative 2

If the WebLogic Server is located on a **different** system than the NMS installation.

1. Copy the contents of the `wls` directory recursively using **scp**:

```
$ scp -r user@hostname:$NMS_BASE/dist/install/wls/.
  WLS_HOME/Oracle/Middleware/user_projects/domains/domain_name
```

2. Having copied the files, restart the WebLogic Managed Server that will be running NMS.

Configuring NMS Security Roles

The following are the LDAP groups that are used by NMS. It is recommended that separate groups be defined for each of these for best security. However, it is also possible to use the same group for multiple types.

NMS Generic User

There are up to three generic users that may need to be defined in LDAP, and assigned to the "service_users" mentioned below. If you wish to use different names for any of them, the names can be configured in $NMS_CONFIG/jconfig/build.properties.

- **nms-service:** This account is used to connect to the Corba Publisher, and is set by property "publisher.ejb-user". This account is required in every NMS installation.
- **mobile-proxy:** this account is used by the NMS Web Services ear, and is set by property "config.ws_runas_user". This is not required unless the Web Services ear is deployed.
- **multispeak-user:** this account is used by the NMS MultiSpeak ear, and is set by property "config.multispeak_runas_user". This is not required unless the MultiSpeak ear is deployed.

If these users do not already exist in the WebLogic security domain, they need to be created. See the **User Authentication** chapter in the *Oracle Utilities Network Management System Configuration Guide* for information on how to either configure LDAP or add local users in WebLogic.

NMS Groups

- **view_only_users:** Users that can only view NMS data.
- **call_entry_users:** Users that can update call entry.
- **service_alert_users:** Users that can update service alert.
- **standard_users:** Users that can modify any NMS data.
- **nms-scada-control:** Users that can initiate SCADA outbound field control requests.
- **service_users:** Adapter and the publisher.ejb-user defined in build.properties.
- **nms-twofactor:** Users that should be authenticated using two-factor authentication.

NMS Roles

- **NmsRead:** calls to view NMS data.
- **NmsWrite:** calls to update NMS data.
- **NmsScadaControl:** calls that initiate a SCADA outbound field control request.
- **NmsCallEntry:** calls used only by Call Entry.
- **NmsUpdate:** calls that update the database. These are also protected by application by the allowed_update_tables entries in CentricityServer.properties.
- **NmsService:** calls that should only be run by adapters and internal accounts.
- **NmsCustom1:** This is an unused role that can be used for project specific requirements.
- **NmsCustom2:** This is an unused role that can be used for project specific requirements.
- **NmsCustom3:** This is an unused role that can be used for project specific requirements.
- **NmsMultifactor:** Indicates that the user should use two-factor authentication when logging in.

NMS Roles to Groups Mapping

- **NmsRead:** view_only_users, call_entry_users, service_users, standard_users, nms-scada-control
- **NmsWrite:** standard_users, service_users, nms-scada-control
- **NmsCallEntry:** call_entry_users, standard_users, service_users
- **NmsUpdate:** service_alert_users, standard_users, service_users, nms-scada-control
- **NmsService:** service_users
- **NmsMultifactor:** nms-twofactor

Groups can be configured by modifying and running the `roles.py` python script, which can be found in the scripts directory under the WebLogic domain directory.

> **Note:** see **Copy Supporting Files from the NMS Distribution to the WebLogic Domain** on page 3-17 for information.

Open and review the script. If you have a need to create custom groups, other than what is set for the defaults in NMS, then modify the top portion of the script:

```
view_only_users = "nms-view-only"
call_entry_users = "nms-call-entry"
service_alert_users = "nms-service-alert"
standard_users = "nmsuser"
scada_control_users = "nms-scada-control"
service_users = "nms-service"
mobile_proxy = "nms-mobile"
```

No further changes to the script are required.

Run the following two commands:

```
. $WL_HOME/server/bin/setWLSEnv.sh
```

Note: This script will set the correct environment variables in your terminal.

```
$WL_HOME/../oracle_common/common/bin/wlst.sh roles.py
```

The script will output messages and prompt you for WebLogic details. For example:

```
Initializing WebLogic Scripting Tool (WLST) ...

Welcome to WebLogic Server Administration Scripting Shell

Type help() for help on available commands

WebLogic domain name: nms (this is the specific domain where NMS
will be running)

Please enter your username:  weblogic (this is the administrative
user login for WebLogic)

Please enter your password: (this is the password for your WebLogic
admin user)

Please enter your server URL [t3://localhost:7001]:   (url for the
administration console; the default may work for you)

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

Successfully connected to Admin Server 'AdminServer' that belongs
to domain 'nms'.

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.

NMS Roles updated
```

Once this has completed, you must stop and restart WebLogic.

Please see the "**Manage Security Roles**" topic in the Oracle WebLogic Server Administration Console Online Help for more information.

Modifying or Adding Security for Individual EJB Methods

If there is a requirement to modify the base ejb security configuration, it is possible to do so with deployment descriptors. Before attempting to do this, it is important to have a good understanding of WebLogic groups and roles, as well as deployment descriptors. See the Oracle WebLogic Server documentation for details.

- The deployment descriptor for cesejb.ear should be saved to `$NMS_CONFIG/jconfig/override/cesejb.jar/META-INF/ejb-jar.xml`
- The deployment descriptor for fwserver.ear should be save to `$NMS_CONFIG/jconfig/override/fwserver.jar/META-INF/ejb-jar.xml`

The following example defines a special role for Session.runScript. Note that defining security for a method using deployment descriptors replaces any existing security configuration:

```
<ejb-jar xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/ejb-jar_3_0.xsd"
         version="3.0">
    <assembly-descriptor>
        <security-role>
            <role-name>NmsCustom1</role-name>
        </security-role>
        <method-permission>
            <role-name>NmsCustom1</role-name>
            <method>
                <ejb-name>Session</ejb-name>
                <method-intf>Remote</method-intf>
                <method-name>runScript</method-name>
            </method>
        </method-permission>
    </assembly-descriptor>
</ejb-jar>
```

It is up to the project to define the role using the WebLogic Administration Console. See the Oracle WebLogic Server documentation for details.

WebLogic Server Runtime Configuration

For information on creating and installing to a WebLogic cluster, which requires the Enterprise edition of WebLogic, see the KM document 1911737.1:

```
https://support.oracle.com/rs?type=doc&id=1911737.1
```

For information on using a customer provided load balancer, see KM document 1910405.1.

```
https://support.oracle.com/rs?type=doc&id=1910405.1
```

If you wish to use multiple instances of WebLogic, but not part of a cluster or using a load balancer, see the KM document 1215414.1:

```
https://support.oracle.com/rs?type=doc&id=1215414.1
```

If you have multiple WebLogic servers that aren't part of a WebLogic cluster, enable this setting in `CentricityServer.properties`:

```
# If your project has multiple web gateways that aren't in a
# cluster, then set this to true.
# Also ensure that the gateway are on different servers, or each
has
# a unique servername as specified by the command line
# argument nms-servername:
# -Dnms.servername=my_server_name
multiple_weblogic_servers = true
```

Create a Managed Server

1. Access the WebLogic Server Administration Console by entering the following URL:

   ```
   http://hostname:port/console
   ```

   Here hostname represents the DNS name or IP address of the Administration Server, and port represents the number of the port on which the Administration Server is listening for requests (port 7001 by default).

2. If you have not already done so, in the Administration Console Change Center, click **Lock & Edit**.
3. In the left pane of the Console, expand Environment and select **Servers**.
4. Click **New**.
5. For the **Server Listen Address**, enter an IP address or DNS name that resolves to an IP address of the server.
6. Change the **Server Listen port** to an unused port.
7. Click **Advanced** and then set RMI JDBC Security to **Secure**.
8. Click **Finish**.
9. In the Servers table, click the server name to open the Settings page.
10. Click the **Tuning** tab.
11. Make sure that Enable Native IO is **not** selected.
12. If necessary, click **Advanced** to access advanced tuning parameters.
13. In the **Muxer Class** field, enter:

```
weblogic.socket.NIOSocketMuxer
```

14. Select the **Control** tab, select your managed server, and click the **Start** button to start your managed server.
15. On the server's **Configuration: General** page, click the **View JNDI Tree** link.

    The JNDI tree for the server appears in a new Administration Console window.
16. In the new Administration Console window, select the top node in the JNDI Tree Structure.
17. Select the **Security** tab and then the **Policies** sub-tab.
18. Under **Methods**, choose **lookup**.
    - Add Conditions
    - Select **Allow Access to Everyone**.
    - Click **Finish**
19. Under **Methods**, choose **All**.
    - Click **Add Conditions**, select **Role**, click **Next**.
    - In the Role Argument Name field, enter **Admin**.
    - Click **Add** and then **Finish**.
20. Click **Save**.
21. To activate these changes, in the Administration Console Change Center, click **Activate Changes**.

Configure Database Connectivity

In WebLogic Server, you configure database connectivity by adding data sources to your WebLogic domain. To create a JDBC data source in your domain, you can use the Administration Console:

1. If you have not already done so, in the Administration Console Change Center, click **Lock & Edit**.
2. In the **Domain Structure** tree, expand **Services**, then select **Data Sources**.
3. On the **Summary of JDBC Data Sources** page, click **New**, and then select **Generic Data Source**.
4. On the JDBC Data Source Properties page, enter or select the following information:
    - **Name** - Enter a name for this JDBC data source.
      For example: `JDBC Data Source-nms`.
    - **JNDI Name** - Enter the JNDI path to where this JDBC data source will be bound.
      Use `jdbc/intersys` for the JNDI path.

    Note: If you will have multiple instances of Oracle Utilities Network Management System running from this WebLogic installation, make the JNDI name unique (for example, `jdbc/intersys/nmsadmin`) and change "config.datasource" in `$NMS_CONFIG/jconfig/build.properties` to match this string.

    - **Database Type** - Select Oracle for the DBMS of the database that you want to connect to.

- Click **Next** to continue.
- On the **JDBC Data Source Properties** page, select ***Oracle's Driver (Thin) for Instance connections; Versions:Any** from the **Database Driver** drop-down list.
- Click **Next** to continue.

5. On the **Transactions Options** page
   - Select **Supports Global Transaction**.
   - Select **Emulate Two-Phase Commit**.
   - Click **Save**.
6. On the **Connection Properties** page, enter values for the following properties:
   - **Database Name** - Enter the name of the database that you want to connect to. Exact database name requirements vary by JDBC driver and by DBMS.
   - **Host Name** - Enter the DNS name or IP address of the server that hosts the database.
   - **Port** - Enter the port on which the database server listens for connections requests.
   - **Database User Name** - Enter the database user account name that you want to use for each connection in the data source. This should match the ORACLE_READ_WRITE_USER variable in `.nmsrc`.
   - **Password/Confirm Password** - Enter the password for the database user account.
   - Click **Next** to continue.
7. On the **Test Database Connection** page, review the connection parameters and click **Test Configuration**.
   - WebLogic attempts to create a connection from the Administration Server to the database. Results from the connection test are displayed at the top of the page. If the test is unsuccessful, you should correct any configuration errors and retry the test.
   - If the JDBC driver you selected is not installed on the Administration Server, you should click **Next** to skip this step.
   - Click **Next** to continue.
8. On the **Select Targets** page, select the servers or clusters on which you want to deploy the data source.
9. Click **Finish** to save the JDBC data source configuration and deploy the data source to the targets that you selected.
10. Perform steps 4-10 for the read-only user. The JNDI path for the user should be the same as previously entered, but with a _readonly appended at the end. Therefore, the default should be `jdbc/intersys_readonly`. Be sure to use the read-only user credentials that were created earlier (matching the ORACLE_READ_ONLY_USER variable in `.nmsrc`).
11. Perform steps 4-10 for the spatial database connection if your system uses spatial landbase in the Web Workspace Viewer. The JNDI path should be `jdbc/spatial`.
12. To activate these changes, in the Administration Console Change Center, click **Activate Changes**.

Create a JMS Server in Your Domain

1. If you have not already done so, in the Administration Console Change Center, click **Lock & Edit**.
2. In the Administration Console, expand **Services** and then **Messaging** and then select **JMS Servers**.
3. On the **Summary of JMS Servers** page, click **New**.

   Note: Once you create a JMS server, you cannot rename it. Instead, you must delete it and create another one that uses the new name.
4. On the **Create a New JMS Server** page:
   - In **Name**, enter a name for the JMS server. For example: JMSServer-nms.
   - In **Persistent Store**, leave this field set to **none**, then the JMS server will use the default file store that is automatically configured on each targeted server instance.
   - Click **Next** to proceed to the targeting page.
5. On the **Selects Targets** page, select the server instance or migratable server target on which to deploy the JMS server.
6. Click **Finish**.
7. To activate these changes, in the Administration Console Change Center, click **Activate Changes**.

Create a JMS System Module in Your Domain

1. If you have not already done so, in the Administration Console Change Center, click **Lock & Edit.**
2. In the Administration Console, under **Services** expand **Messaging** and select **JMS Modules**.
3. On the **Summary of JMS Modules** page, click **New**.

   Note: Once you create a module, you cannot rename it. Instead, you must delete it and create another one that uses the new name.
4. On the **Create JMS System Module** page:
   - In **Name**, enter a name for the JMS system module. For example: SystemModule-nms.
   - Click **Next** to proceed to the targeting page.
5. On the **Targets** page, select the server instance or cluster target on which to deploy the JMS system module, and then click **Next**.
6. On the **Add Resources** page, select the checkbox to immediately add resources to the newly created JMS Module.
7. Click **Finish**.
8. On the **Configuration** page, click **New** above the Summary of Resources table.
9. On the **Create a New JMS System Module Resource** page, select **Connection Factory** from the list of JMS resources and then click **Next**.

10. On the **Connection Factory Properties** page, define the connection factory's basic properties:
    - In **Name**, enter a name for the connection factory. For example: *ConnectionFactory-nms*.

    Note: Once you create a connection factory, you cannot rename it. Instead, you must delete it and create another one that uses the new name.

    - In **JNDI Name**, enter **ConnectionFactory**.
11. Click **Next** to proceed to the targeting page.
12. For basic default targeting, accept the default targets presented in the **Targets** box and click **Finish**. The configured connection factory is added to the module's Summary of Resources table, which displays its default targets.
13. On the **Configuration** page, click **New** above the Summary of Resources table.
14. On the **Create a New JMS System Module Resource** page, select **Distributed Topic** from the list of JMS resources, and then click **Next**.
15. On the **JMS Distributed Destination Properties** page, define the distributed topic's basic properties:
    - In **Name**, enter a name for the distributed topic. For example: `MsgBean-nms`.

    Note: Once you create a distributed topic, you cannot rename it. Instead, you must delete it and create another one that uses the new name.

    - In **JNDI Name**, enter *topic/MsgBean*.
16. Click **Next** to proceed to the targeting page.
17. For basic default targeting, accept the default targets presented in the Targets box and then click **Finish**. The JMS system module resource is added to the module's Summary of Resources table, which displays its default targets.
18. On the Configuration page, click New above the Summary of Resources table.
19. On the Create a New JMS System Module Resource page, select Distributed Topic from the list of JMS resources, and then click Next.
20. On the JMS Distributed Destination Properties page, define the distributed topic's basic properties:
    - In **Name**, enter a name for the distributed topic. For example: `MsgRegister-nms`.

    Note: Once you create a distributed topic, you cannot rename it. Instead, you must delete it and create another one that uses the new name.

    - In **JNDI Name**, enter *topic/MsgRegister*.
    - Change the Forwarding policy to: *Partitioned*
21. Click **Next** to proceed to the targeting page.
22. For basic default targeting, accept the default targets presented in the Targets box and then click **Finish**. The JMS system module resource is added to the module's **Summary of Resources** table, which displays its default targets.To activate these changes, in the Administration Console Change Center, click **Activate Changes**.

Configure T3 Protocol

1. If you have not already done so, in the Administration Console Change Center, click **Lock & Edit**.
2. In the left pane of the Console, expand **Environment** and select **Servers**.
3. On the **Servers** page, click on the server name.
4. Select **Protocols > General**.
5. In the **Maximum Message Size** field, enter **50000000**.

   Note: These settings apply to all protocols in the server's default network configuration.
6. Click **Save**.
7. To activate these changes, in the Administration Console Change Center, click **Activate Changes**.

Configure the Arguments to Use When Starting a Server in Your Domain

1. If you have not already done so, in the Administration Console Change Center, click **Lock & Edit**.
2. In the Administration Console, expand **Environment** and select **Servers**.
3. On the Servers page, click the name of the server.
4. Under **Configuration**, select **Server Start** tab.
5. Determine the amount of memory to set aside for WebLogic. This is set by the -Xms and -Xmx parameters. The -Xms is the amount that should be allocated at startup, and the -Xmx is the maximum amount it should use. To ensure that the maximum memory is allocated at start-up and eliminating the need for extra memory allocation during program execution, we recommend setting -Xms and -Xmx to the same value. We recommend aggressive maximum memory (heap) size of between 1/2 and 3/4 of physical memory. The minimum should be 4096m. In the example below, replace 4096m with the value you wish to configure.
6. On the **Server Start** page, add the following JVM parameters:

```
   -Xms4096m
   -Xmx4096m
   -Dweblogic.system.StreamPoolSize=0
   -XX:+UseG1GC
   -javaagent:lib/nms_monitor.jar
   -Djavax.xml.parsers.DocumentBuilderFactory=
   com.sun.org.apache.xerces.internal.jaxp.DocumentBuilderFactoryI
   mpl
   -Djava.awt.headless=true
   -Dweblogic.jndi.allowGlobalResourceLookup=true
   -Dweblogic.jndi.allowExternalAppLookup=true
   -Doracle.xdkjava.security.resolveEntityDefault=false
-d64
```

Configure the location of the log4j configuration file by setting:

```
-Dlog4j.configurationFile=<fiull path to nms-log4j.xml>
```

If it is desired that the hostname be something other than what the operating system returns, add a startup flag to the app server:

```
-Dnms.servername=[server_name]
```

Replace [server_name] with the name you wish to log. Overriding the name may be helpful if multiple app servers are on the same machine.

If using a web map server to supply web maps to the web viewer, and if your WebLogic system requires a proxy server to access the external web map server, and if the system you are running WebLogic on does not have a system wide proxy server configuration, you will need to configure WebLogic to access the web map server using your network proxy server using the JVM start up parameters.

> Note: See the **Java SE Documentation** for details on configuring the JVM for proxies.

Typically, to configure the JVM to use a proxy server, you will need to set JVM startup parameters:

- If the web map server uses http:

```
   -Dhttp.proxyHost=proxy-hostname
   -Dhttp.proxyPort=port#
-Dhttp.nonProxyHosts=
 host1|host2|192.168.107.*|localhost|127.0.0.*
```

- If web map server uses https:

```
   -Dhttp.proxyHost=proxy-hostname
   -Dhttp.proxyPort=port#
   -Dhttp.nonProxyHosts=
 host1|host2|192.168.107.*|localhost|127.0.0.*
```

List the host names, IP addresses, or IP masks to any local system your JVM will need access to that does not require the proxy server. Be sure to include your local machine, DB system, and NMS core server system at a minimum. The nonProxyHosts lines cannot contain quotes or the managed server startup will fail.

Configuring Java Mail

There is a vast amount of information related to this on the web and under the WebLogic documentation. For a simple SMTP configuration, the following steps should be followed.

1. Log into WebLogic Administration Console.
2. In the Domain Structure tree at the center left, expand the **Services** node.
3. Select the **Mail Sessions** sub-node.
4. Click **New**.
5. For the JNDI Name, enter `mail/nmsMail`.
6. In the **JavaMail Properties** field, enter the following properties:

```
   mail.smtp.port=25
   mail.host=internal-mail-router.SomeDomain.com
mail.transport.protocol=smtp
```

> **Note:** You will need to know the port, host and transport protocol of your mail option. For more information, see the WebLogic documentation.

7. Click **Next** and select the server for your NMS environment.

8. Click **Finish**.
9. The managed server will need to be restarted for the changes to properly take affect.
10. In your Java configuration, modify `$NMS_CONFIG/jconfig/CentricityTool.properties`:

```
    # The protocol to send emails from the client. Valid entries
    # are SMTP and MAPI. MAPI will bring up the default email
    # client to send the email. SMTP will bring up a simple dialog
    # from within NMS to send the email.
email_protocol = SMTP
```

11. In your Java configuration, modify `$NMS_CONFIG/jconfig/server/CentricityServer.properties`

```
    # If SMTP is used instead of MAPI to send emails, this is the
    # query to return the email address for a given username.
    # This is also used by Web Switching for sending automated
    # emails normally initiated via a sheet's state transition
    request.
    # If an email_from value is not configured for automated
    switching # sheet emails, then the calling user's ID is used.
    # This query can be used to translate a user ID to an email
    address.
# username_to_email = select ? || '@oracle.com' email from dual
```

Web Switching Singleton Service

The Web Switching Singleton Service is used to process Web Switching specific requests like creating Open and Close Miscellaneous Log steps for SCADA actions. It also processes model verification updates to steps that have been involved in model changes. (This service is disabled when the product **Switching** is not licensed.)

1. If you have a non-clustered project environment where multiple application servers are running, add the following command line argument for **only** the non-Web Switching instances:

```
-Dnms.disable-swman-static-service=true
```

> **Note:** There must be only one server that is running Web Switching unless NMS is running on a WebLogic cluster.

2. Click **Save**.
3. To activate these changes, in the Administration Console Change Center, click **Activate Changes**.

Configure Keystores

Before You Begin

- Obtain private keys and digital certificates from a reputable certificate authority such as Verisign, Inc. or Entrust.net.
- Create identity and trust keystores.
- Load the private keys and trusted CAs into the keystores.

Configure the Identity and Trust Keystores

1. If you have not already done so, in the Administration Console Change Center, click **Lock & Edit.**
2. In the left pane of the Console, expand **Environment** and select **Servers.**
3. Click the name of the server for which you want to configure the identity and trust keystores.
4. Under **Configuration** select **Keystores.**
5. In the **Keystores** field, select the method **Custom Identity and Java Standard Trust** for storing and managing private keys/digital certificate pairs and trusted CA certificates.
6. In the **Identity** section, define attributes for the identity keystore.
    - **Custom Identity Keystore:** The fully qualified path to the identity keystore `nms-ssl.keystore`. This will be in the `$NMS_HOME/java` directory.

    Note: if your WebLogic Server is running on a different server than the NMS installation, `nms-ssl.keystore` will need to be copied to a location where it is accessible to the account running WebLogic.

    - **Custom Identity Keystore Type:** The type of the keystore. Generally, this attribute is Java KeyStore (JKS); if left blank, it defaults to JKS.
    - **Custom Identity Keystore Passphrase:** The password you will enter when reading or writing to the keystore. This attribute is optional or required depending on the type of keystore. All keystores require the passphrase in order to write to the keystore. However, some keystores do not require the passphrase to read from the keystore. WebLogic Server only reads from the keystore so whether or not you define this property depends on the requirements of the keystore.
7. Click **Save.**
8. To activate these changes, in the Administration Console Change Center, click **Activate Changes.**

Configure the SSL Listen Ports for a Server

1. If you have not already done so, in the Administration Console Change Center, click **Lock & Edit.**
2. In the Administration Console, expand **Environment** and select **Servers.**
3. On the Servers page, click the name of the server.
4. Under **Configuration** select **General.**
    - Select SSL Listen Port Enabled so that the server listens on the SSL listen port.
    - If you want to disable the non-SSL listen port so that the server listens only on the SSL listen port, deselect **Listen Port Enabled.**
5. Click **Save.**
6. To activate these changes, in the Administration Console Change Center, click **Activate Changes.**

Configure SSL

1. If you have not already done so, in the Administration Console Change Center, click **Lock & Edit.**
2. In the left pane of the Console, expand **Environment** and select **Servers.**
3. Click the name of the server for which you want to configure SSL.
4. Under **Configuration** select **SSL,** and set the SSL attributes for the Private Key Alias (defaults to `nms-key`) and Private Key Passphrase.
5. Click **Save.**
6. To activate these changes, in the Administration Console Change Center, click **Activate Changes.**

Set the Default Authenticator Control Flag

1. If you have not already done so, in the Administration Console Change Center, click **Lock & Edit.**
2. In the left pane, select **Security Realms,** then click the name of the realm you are configuring. Select myrealm.
3. Under **Providers** select **Authentication.**

   The Authentication Providers table displays the name of the Authentication and Identity Assertion providers.
4. Click the name of the provider you want to configure. Select DefaultAuthenticator.
5. Under **Configuration** select **Common** and set the **Control Flag** to SUFFICIENT.
6. Click **Save.**
7. To activate these changes, in the Change Center click **Activate Changes.**

Create and Configure an Active Directory Authentication Provider

Note that any of the **WebLogic Authentication Provider** types can be used. Here, **ActiveDirectoryAuthenticator** is used as an example.

1. If you have not already done so, in the Administration Console Change Center, **click Lock & Edit.**
2. In the left pane, select **Security Realms** and click the name of the realm you are configuring (defaults to myrealm).
3. Under **Providers** select **Authentication** and click **New.**

   The Create a New Authentication Provider page appears.
4. In the **Name** field, enter a name for the Authentication provider. For example, enter ldap-provider.
5. From the **Type** drop-down list, select the type of the Authentication provider and click **OK**. Select **ActiveDirectoryAuthenticator**.
6. Under **Providers** select **Authentication** and click the name of the new Authentication provider to complete its configuration.
7. Under **Configuration** select **Common** and set the **Control Flag** to SUFFICIENT.
8. Click **Save.**
9. Under **Configuration** select **Provider Specific** and set the desired values for your Active Directory server. The following configuration is given for example purposes only.

For Connection:

Host: server.example.com

Port: 389

Principal: cn=Administrator,cn=Users,dc=example,dc=com

Credential: The credential (usually a password) used to connect to the LDAP server.

For Users:

User Base DN: cn=
Users,dc=example,dc=com

User Name Attribute: Ensure this matches the attribute specified in the User Base DN (for example, "cn").

10. Click **Save**.
11. To activate these changes, in the Change Center, click **Activate Changes**.
12. After you finish configuring Authentication providers, restart WebLogic Server.

> **IMPORTANT**: verify that users and groups from your authenticator are configured by looking at the **Users and Groups** tab for your security realm.

Configure Local LDAP

1. If you have not already done so, in the Administration Console Change Center, click Lock & Edit.
2. Under **Domain**, select [domain], **Security**, **Embedded LDAP**, and then select the **Refresh Replica At Startup** check box.
3. Click **Save**.

Deploying Oracle Utilities Network Management System in WebLogic Server

To deploy the Oracle Utilities Network Management System application in your domain, follow these steps:

1. Login in as the user account that will run the WebLogic Application Server.
2. Access the WebLogic Server Administration Console by entering the following URL:

   `http://hostname:port/console`

   Here `hostname` represents the DNS name or IP address of the Administration Server, and *port* represents the number of the port on which the Administration Server is listening for requests (port 7001 by default).
3. If you have not already done so, in the Administration Console Change Center, click **Lock & Edit**.
4. In the left pane of the Administration Console, select **Deployments**.
5. If there is a deployment from a previous installation, complete the following two actions before proceeding to step 6:
   - Select the check box to the far left of the deployed cesejb application. Click **Stop** and choose **Force Stop Now** to stop the application.
   - Select the check box to the left of the deployed cesejb application. Click **Delete** (located at the top or bottom of the Deployments table), to delete the cesejb application. Click **Yes** to confirm your decision.
6. In the right pane, click **Install**.
7. In the Install Application Assistant, locate the `cesejb.ear` to install. This will be in the `$NMS_HOME/java/deploy` directory.
8. Click Next.
9. Specify that you want to target the installation as an application.
10. Click **Next**.
11. Select the servers and/or cluster to which you want to deploy the application. The cesejb.ear should be deployed to its own managed server or cluster; therefore, either select a managed server/cluster that does not have other applications or interfaces deployed to it or move existing deployments to a separate instance.

    Note: If you have not created additional Managed Servers or clusters, you will not see this assistant page.
12. Click **Next**.
13. Update the following additional deployment setting:
    - Change the deployed name of the application from `cesejb` to something unique.
14. Click **Next**.
15. Review the configuration settings you have specified, and click **Finish** to complete the installation.
16. If you chose to immediately go to the deployment's configuration screen, click the tabs to set additional configuration settings for the application or module.

    If you chose to change this information later, you are returned to the **Deployments** table, which now includes your newly-installed application or module.

17. To activate these changes, in the Administration Console Change Center, click **Activate Changes**.

    Note: Ensure that your deployment is listed as "Active" in the deployments table. If it is "New" or "Prepared," something has not started correctly.

18. A restart of the WebLogic managed server(s) that will be running Oracle Utilities Network Management System is not required for these changes to take effect unless you are instructed to do so at this time.

19. Open a browser and navigate to: `http://hostname:port/nms`

    Here `hostname` represents the DNS name or IP address, and *port* represents the port for the WebLogic Server.

Troubleshooting

1. If there are deployment issues and you want to validate the connection to the CORBA gateway, open the WebLogic log file, which will display diagnostic information including any issues with the database and CORBA gateway configuration.

2. Performance issues: Some WebLogic default parameter settings are not optimal for a significant production NMS installation. Below are some options NMS has successfully deployed to make WebLogic more responsive and resilient. Values provided are representative but should not be considered definitive – variations may be appropriate for your installation.

WebLogic Managed Server Specific Changes

1. On the left, click Environment > Servers
2. Click the managed server name
3. Click the **Server Start** tab
4. Add the option below to the **Arguments** (server startup options) field:
    - -Dweblogic.security.providers.authentication.ldap.socketTimeout=3
        - Specifies the maximum number of seconds to wait for the connection to any one host specified in the Host attribute.
        - Default=0 which sets no socket timeout
5. Click the **Save** button.
6. Restart the managed server for this change to take effect, but wait to restart until after you have made any other potential LDAP Provider Changes noted below.
7. Repeat this process for any other managed server

LDAP Provider Setting Changes

NOTE: These changes will affect all managed servers within a given WebLogic domain.

The default LDAP cache within WebLogic may be disabled or too small. Timeouts may be disabled for accessing an LDAP server and maybe inappropriate for accessing a replicated LDAP server environment (some form of replicated Active Directory access – for example). Modify default values by completing the following steps in the WebLogic Server Administration Console:

1. Select the Provider Specific page for the LDAP Authentication provider
    - Security Realms > myrealm > Providers > Authentication > your LDAP Authentication provider > Provider Specific
2. Scroll down to the General section
3. Suggest the following (or similar) changes:
    - Set "Change Connect Timeout" to 10.
        - Maximum time in seconds to wait for the connection to the LDAP server to be established.
        - Default=0 which means no maximum time limit.

        Most production NMS installations will have an alternate LDAP server if the primary is not responding.
    - Change "Parallel Connect Delay" to 1
        - Delay in seconds when making concurrent attempts to connect to multiple LDAP servers.
        - Default=0 which means it will not retry if there is no response.
    - Change "Results Time Limit" to 1000
        - The maximum number of milliseconds for the LDAP server to wait for results before timing out.
        - Default=0 which means wait indefinitely.
    - Make sure "Cache Enabled" is checked
        - Specifies whether a cache is used with the LDAP server
        - Default is enabled.
    - Change "Cache Size" to 4096KB
        - Size of the cache (in kilobytes) that is used with the LDAP server
        - Default=32KB
    - Change "Cache TTL" to 3600 seconds (1 hour)
        - Time-to-live for cache (in seconds) used with the LDAP server

            Every WebLogic access from an NMS client is authenticated – not just the initial login. The result of this authentication can be cached to reduce load on an external LDAP server but (by default for peak security) this cache is very modest. The default configuration can result in a significant authentication load on your LDAP servers under peak (NMS user count) and load.

            Values from 1 to 4 hours are suggested for consideration.
        - Default is 60 seconds
    - Click the **Save** button.
4. At the top of the page, click on the "Performance" tab.
    - Change "Group Hierarchy Cache TTL" to match the "Cache TTL" value above.

- Maximum number of seconds a group membership hierarchy entry is valid in the LRU cache.
- Default is 60 seconds.

- Click the **Save** button.

5. Restart the WebLogic Managed Server (full domain) for changes to take effect.

Installing Flex Operations

The Flex Operations browser client uses the same architecture as the Operations Mobile Application (OMA). If you are licensed for both applications, just follow the full installation instructions from the *Oracle Utilities Network Management System Operations Mobile Application Installation & Deployment Guide* to set up both applications. If you are only licensed for Flex Operations, follow the instructions below:

- **Mobile Gateway Server Installation**
  - Deploy the Mobile Gateway
  - Configuring WebLogic to Handle HTTP Basic Challenges Correctly
- **NMS Server Configuration**
  - GeoJSON Map Generation
  - Configuring OMA Object Attribute Viewer
  - Configuring OMA For Schematic Maps
  - Configuring OMA Search Options
- **Operations Mobile Application Project Setup**
  - The entire chapter

Authenticating Flex Operations Users

Flex can be configured to either authenticate a user by the configured LDAP provider that is running the Flex Operations gateway (`nms-ws.ear`) or by the server that is running `cesejb.ear`. The default is cesejb.ear. This is controlled by setting the following parameter in `WebService.properties`:

```
   # Where Flex clients should be authenticate. The choices are
   # cesejb and nms-ws
authenticate = cesejb
```

Please note that even if the Flex user is configured in the `nms-ws` domain, the role configuration still must be configured in the domain running `cesejb.ear`. (See **NMS Roles to Groups Mapping** on page 3-19 for information.)

Installing Web Clients

The Oracle Utilities Network Management System Web Clients may be run from a browser as a Java Web Start application or by installing individual Java client applications.

Allowlisting

The following are executables and libraries that are part of the application that may need to be added to allowlists by anti-virus software:

- MapiSend.exe
- msvcp71.dll
- msvcr71.dll
- SendSignal.exe
- SendSignal64.exe
- ShellExecute.exe

They will be installed in `%APPDATA%\Roaming\.nms\exe`

Java Web Start

If the Java Web Start version is chosen, there is no client installer needed. The user opens the NMS application landing page and clicks a link to one of the Java applications, such as Web Workspace.

Example

```
URL: https://[web-gateway]/nms/
```

Web Workspace Java Web Start link:

```
https://[web-gateway]/nms/nmswebstart?appName=WebSwitching.jnlp
```

Java Client Installation

The Java client applications installer is created by the Oracle Utilities Network Management System Configuration Assistant, which is also a Java application. Therefore, to create the installer, the Configuration Assistant must be run (at least once) using Java Web Start.

Install Prerequisite Software

The client installer creation process requires the following applications be pre-installed on the PC that will run the Configuration Assistant.

- **NSIS** (Nullsoft Scriptable Install System) is an open-source Windows installer development tool; project on SourceForge (http://sourceforge.net/projects/nsis/).
- **Launch4j** is a tool that wraps Java applications in a Windows executable file; available on SourceForge (http://launch4j.sourceforge.net/).
- **Java Standard Edition JDK**. This should match the version you wish to include as part of the client installation. Normally you would choose the latest JDK. (http://www.oracle.com/technetwork/java/javase/downloads/index.html)

Note: see the *Oracle Utilities Network Management System Quick Install Guide* for supported software versions.

Create Environment Variables

1. Create the following system environment variables, as necessary, to ensure that the Configuration Assistant can find the applications.

   - **NSIS Environment Variable**

   Name: **NSIS_HOME**

   Value: Path to NSIS.exe.

   Note: If NSIS was installed in the default location, the environment variable does not need to be added.

   - **Launch4j Environment Variable**

   Name: **LAUNCH4J_HOME**

   Value: Path to launch4j.exe.

   Note: If Launch4J was installed in the default location, the environment variable does not need to be added.

   - **JDK Environment Variable**

   Name: **JAVA_HOME**

   Value: Path to the root of the JDK (where the jre and bin subfolders are located).

2. After setting the environment variables, reboot your PC.

Create Installer

To create the installer, open Configuration Assistant and do the following:

1. Select **Create Client Installer...** from the **Actions** menu.

   A save dialog will open that allows you to modify the file name and location; neither the name nor location will affect how the applications are ultimately installed.

2. Click **Save**.

   Note: if the file already exists at that location, you will be asked to confirm replacement.

3. The client installer creation process will call NSIS, which will open and display a log of its activities in the **MakeNSISW** window. When the process is complete, NSIS will allow you to run the installer (using the **Test Installer** function) or **Close** the application.

Install Client Applications

1. If **MakeNSISW** is still open, click **Test Installer** to run the client installer. If it is not open, navigate to the location where the installer was saved and double-click the installer file name or icon (depending on your view). The **Oracle Utilities NMS Setup Wizard** dialog will open.
2. On the **Choose Install Location** page, select the destination folder and click **Next**.
3. On the **Choose Components** page, select the components to install from the list of licensed products. Click **Install**.
4. When the installation is complete, click **Close**.

Note: Start menu application shortcuts will be created under **All Programs** in the **Oracle Utilities NMS** folder.

Updating Clients

Client installers must be recreated whenever a new release (major release, service pack, or patch) is implemented.

1. Uninstall the existing applications from each client PC.
2. Follow the "Create Installer" task using the updated Configuration Assistant.
3. Follow the "Install Client Applications" task.

Deploying Patch Bundles

Steps to Deploy a Patch Bundle in Dual-Environment Configuration

This section describes the general steps for deploying a patch bundle in dual-environment configuration. This involves two NMS Administrative Users:

- Active – the user where NMS is currently running
- Staging – the user where this patch bundle will be installed

Each of these users has a corresponding WebLogic managed server. These will be called the Active managed server and Staging managed server.

All steps are to be performed on the Staging user unless otherwise noted.

Note: See the README.txt that comes with the patch bundle for specific instructions.

1. Take note of the changes for this patch as documented in the Product Fix Design document(s). This patch may require a manual migration.
2. Identify which administrative user is staging. This command will output "A" for the active user and "S" or "None" for the staging user.

   ```
   $ nms-version --get-stage
   ```

3. If you are deploying this to production, copy the new configuration and manual migrations to the production server.
4. Log in as the NMS Administrative User that is staging (for example, nmsadmin2).
5. Remove the previous installation directory, if one exists.

   ```
   $ cd ~
   $ rm -rf network
   ```

6. Unzip the nms.<version>-date-platform.zip file found within the downloaded zip file.

   ```
   $ unzip nms.<version>-date-platform.zip
   ```

7. Run the install script.

   ```
   $ cd network
   $ ./nms-install
   ```

Note: If you have not created an `/etc/nmstab` file, add the `-noNmsTab` option to nms-install.

8. Remove the installation files before continuing.

```
$ cd ~
$ rm -rf network
```

9. Stop any running services and Isis.

```
$ sms-stop -aiS
```

10. Run nms-install-config.

```
$ nms-install-config
```

11. Start nms-setup.

    Note: You can continue to the next step without waiting for this to finish.

```
$ nms-setup
```

12. Undeploy the old ear from the Staging managed server.
13. Restart the Staging managed server.
14. Complete Steps 1 - 19 in Deploying Oracle Utilities Network Management System in WebLogic Server on page 3-30 for the Staging managed server.
15. Wait for nms-setup to complete.
16. Verify that nms-setup succeeded. If there are manual migrations that were missed, the setup will stop. If so, complete any manual migrations and try again starting at step 9.
17. On the Active Administrative User, force users out of the system

```
Action any.pub* ejb enable_login false
```

18. On the Active Administrative User

    Note: You can continue to the next step without waiting for this to finish.

19. Stop the Active Managed Server, stop any running services and Isis.

```
$ sms-stop -aiS
```

    Note: You can continue to the next step without waiting for this to finish.

20. On the Staging Administrative User, run nms-post-setup

```
$ nms-post-setup
```

21. Re-start services. The Staging environment has become the new Active environment.

```
$ sms-start
```

Users will be able to log into the system shortly after sms-start finishes.

Steps to Deploy a Patch Bundle Without Dual-Environment Configuration

This section describes the general steps for deploying a patch bundle.

> **Note:** See the `README.txt` that comes with the patch bundle for specific instructions.

1. Take note of the changes for this patch as documented in the Product Fix Design document(s). This patch may require a manual migration.
2. If you are deploying this to production, copy the new configuration and manual migrations to the production server.
3. Log in as the NMS Administrative User (for example, nmsadmin).
4. Remove the previous installation directory, if one exists.

```
$ cd ~
$ rm -rf network
```

5. Unzip the `nms.[version]-date-platform.zip` file found within the downloaded zip file.

```
$ unzip nms.[version]-date-platform.zip
```

6. Run the install script.

```
$ cd network
$ ./nms-install
```

> **Note:** If you have not created an `/etc/nmstab` file, add the `-noNmsTab` option to the `nms-install` command.

7. Remove the installation files before continuing.

```
$ cd ~
$ rm -rf network
```

8. Force users out of the system.

```
Action any.pub* ejb enable_login false
```

9. Stop any running services and Isis.

```
$ sms-stop -aiS
```

10. Run nms-install-config.

```
$ nms-install-config
```

11. Start nms-setup with the -post option.

> **Note:** You can continue to the next step without waiting for this to finish.

```
$ nms-setup -post
```

12. Undeploy the old ear.
13. Restart the managed server.
14. Complete Steps 1 - 19 in **Deploying Oracle Utilities Network Management System in WebLogic Server** on page 3-33.
15. Wait for nms-setup to complete.

16. Verify that nms-setup succeeded. If there are manual migrations that were missed, the setup will stop. If so, complete any manual migrations and try again starting at step 8.
17. Re-start services.

    ```
    $ sms-start
    ```

    Users will be able to log into the system shortly after sms-start finishes.

Steps to Deploy a Patch Bundle using Failover Patching

This section describes the general steps for deploying an NMS patch bundle using failover patching. This patches an NMS environment on a separate site while users are live at the original site. For this section, environment #1 on Site A is active with live users. The patch will be applied on environment #2 at Site B. In practice, the environment numbers may be swapped. If so then change these steps accordingly.

Prerequisites:

- Two or more NMS sites configured for failover patching as described in the Site Guard chapter of the Configuration Guide.
- Dual-environment configuration on all sites.
- Oracle Flashback Database must be enabled in Oracle RDBMS.
- NMS previously installed and configured at Site B.
- SSH password-less login must be enabled between the NMS servers on Site A and Site B.

All steps are to be performed on the NMS staging (non-active) environment unless otherwise noted.

> **Note:** See the README.txt that comes with the patch bundle for specific instructions.

1. Take note of the changes for the patch as documented in the Product Fix Design document(s). Any given patch may require one or more specific manual migrations.
2. On Site A, identify which administrative user is the staging environment (where the NMS patch will be applied). The command below will output "A" for the Active NMS administrative user environment and "S" or "None" for the Staging NMS administrative environment.

   ```
   $ nms-version --get-stage
   ```

3. If you are deploying to production, copy the new configuration and manual migrations to the NMS staging user environment of Site A.
4. On Site A, log in as the NMS Administrative User that is staging (for example, nmsadmin2).
5. Remove the previous installation directory, if one exists.

   ```
   $ cd ~
   $ rm -rf network
   ```

6. Unzip the nms.<version>-date-platform.zip file found within the downloaded zip file.
   $ unzip nms.<version>-date-platform.zip

7. Run the install script.

```
$ cd network
$ ./nms-install
```

   Note: If you have not already created an /etc/nmstab file, add the -noNmsTab option to nms-install.

8. Remove the installation files before continuing (recommended).

```
$ cd ~
$ rm -rf network
```

9. Stop any running services and the Isis message bus.

```
$ sms-stop -aiS
```

10. Run nms-install-config.

```
$ nms-install-config
```

11. Start nms-setup.

```
$ nms-setup
```

12. Verify that nms-setup succeeded. If there are manual migrations that were missed, the setup will stop. If so, complete any manual migrations and try again starting at step 9.
13. Login as the staging user on Site B and stop NMS services and the Isis message bus there.

```
sms-stop -aiS
```

14. Logout from the staging user at Site B.
15. Undeploy the old ear from the Staging managed server at both Site A and Site B.
16. On Site A, logged in as the NMS Staging administrative user, run nms-sync-site to sync the patch to Site B.

```
$ nms-sync-site patch site-b.example.com
```

17. In Oracle Enterprise Manager, open the Generic System for Site A.
18. Select the menu item Generic System -> Site Guard -> Operations.
19. Select the operation plan for upgrading Site B and click Execute Operation.
20. Click Yes.
21. Click on the link to the procedure to follow the progress. This procedure performs the following steps.
    - Create a guaranteed restore point on the Site A RDBMS.
    - Transition all NMS end users at Site A (as a group) to Read Only mode.
    - Failover RDBMS from Site A to Site B.

    Note this is different than an RDBMS Switchover. In failover mode both RDBMS instances (one at each site) end up as operationally independent read/write RDBMS instances. At this point RDBMS transactions at one site will NOT automatically be propagated to the other site.

    - Start WebLogic Node Manager and AdminServer at Site B.

22. On Site B run the post setup script with the NMS_MODEL_SOURCE environment variable unset.
    Note: You can continue to the next step without waiting for this to finish.

    ```
    $ NMS_MODEL_SOURCE= nms-post-setup
    ```

23. On Site B, run nms-install-config for Java.

    ```
    $ nms-install-config --java
    ```

24. Complete Steps 1 - 19 in Deploying Oracle Utilities Network Management System in WebLogic Server on page 3-29 for the Staging managed server on Site B.
25. Wait for nms-post-setup to complete.
26. Re-start services. The Staging environment on Site B is now active.

    ```
    $ sms-start
    ```

27. Optional: If desired, the environment on Site B can be validated, and then reset back to the current state.
    - Login with the sysdba role on the Site B database and create a guaranteed restore point. Any unique name can be chosen for this restore point (two restore points cannot have the same name).

      ```
      SQL> create restore point "VALIDATE_NMS" guarantee flashback
      database;
      ```

    - Perform core/smoke/sanity tests to validate patch properly installed on Site B.
    - Once validated, stop NMS services and WebLogic managed server.

      ```
      $ sms-stop -a
      $ nms-wls-control stop
      ```

    - Login with the sysdba role on the Site B database and rollback the database to the guaranteed restore point that was set above. Recommend dropping the restore point at the end so the name of the restore point can be recycled when applying the next patch – not required.

      ```
      SQL> shutdown immediate;
      SQL> startup mount;
      SQL> flashback database to restore point "VALIDATE_NMS";
      SQL> shutdown immediate;
      SQL> startup;
      SQL> drop restore point "VALIDATE_NMS";
      ```

    - Start NMS services and WebLogic NMS managed server.

      ```
      $ sms-start
      $ nms-wls-control start
      ```

28. Route relevant external adapters (integrations) to Site B. This will be different for each adapter.
29. Inform users to logout from NMS at site A and login to NMS at site B. This can be done in OEM via the Send Message button in the Application Management Pack for Oracle Utilities NMS or on the command-line with an Action command:

    ```
    $ Action -java any.any display_oem_message "Login to upgraded NMS
    at site-b"
    ```

- Login to the active environment and site A. Stop NMS services and WebLogic managed server

  ```
  $ sms-stop -aiS
  $ nms-wls-control stop
  ```

30. Reinstate the database at Site A as standby. This can be done from Oracle Enterprise Manager with the following steps:
    - In Oracle Enterprise Manager, open the database instance for Site B.
    - Select the menu item Availability -> Data Guard Administration.
    - Click on the Database must be reinstated link in the Data Guard Status for the Site A database.
    - Click **Reinstate**.
    - Click **Yes** on the Confirmation screen.
    - Choose the Database Host Credentials and click **Continue**.
31. Login with the sysdba role on the Site A database and drop the guaranteed restore point created by Site Guard

    ```
    SQL> drop restore point "BEFORE_PATCH_NMS";
    ```

32. At this point, the active environment and the staging environment have switched at all sites. The remaining steps are performed in the active environment at the various sites.
33. On Site B, run nms-sync-site to sync the patch to Site A and any other configured sites.

    ```
    $ nms-sync-site patch site-a.example.com
    $ nms-sync-site patch site-c.example.com
    ```

34. At each site other than Site B, perform the following steps
    - Run nms-install-config for Java to generate an NMS EJB deployment that matches the current patch.

      ```
      $ nms-install-config --java
      ```

    - Complete Steps 1 - 17 in **Deploying Oracle Utilities Network Management System in WebLogic Server** on page 3-33, making sure not to start the managed server.

Installing Oracle Business Intelligence Publisher

Oracle Business Intelligence Publisher is used for printing reports in some Web Workspace tools and in Web Switching. For Oracle BI Publisher documentation, including installation instructions, go to the Oracle Fusion Middleware Documentation page (http://docs.oracle.com/en/middleware/middleware.html) and follow the links to the NMS supported version of Oracle BI Publisher under Products.

> **Note:** If you plan to interface with an existing BI Publisher installation, it must not use any user authentication functions such as Single Sign-On (SSO). If you cannot alter the security protocols for your existing installation of BI Publisher, then you'll need to install an additional instance for the purpose of generating NMS reports.

When installation is complete, proceed to the Oracle Utilities Network Management System Configuration Guide for instructions on how to configure Oracle BI Publisher reports for your system.

Installing and Configuring Optional Components

Spatial Landbase Map Installation

Prerequisite: installation of OPAL Spatial Landbase Maps requires the Map Builder component of Oracle Fusion Middleware MapViewer ; see **Requirements for Spatial Landbase** on page 2-2 for details.

Use the following optional procedure to load the OPAL Spatial Landbase maps:

1. Unzip the OPAL spatial shapefiles and metadata file:

```
$ cd $NMS_CONFIG
$ unzip spatial_landbase.zip
```

2. Start Oracle Map Builder

```
$ cd [directory where mapbuilder.jar is installed]
$ run java -Xms200m -Xmx1000m -jar mapbuilder.jar
```

3. In Map Builder, select **File/New Connection…** Specify the connection information to connect to the server where the spatial data will be served. Then connect to the server.
4. In Map Builder, select **Tools/Import Shapefile…** and click **Next**.
   - Under Data Selection, select the **Multiple Files or Directories** and change the Selection drop down list to **Directory**.
   - Click **Select**; navigate to and **Open** the `$NMS_CONFIG/spatial_landbase` directory.

- Click **Next**.
- Set the SRID to 41100 (Ohio 3401, Northern Zone (1983, US Survey feet)), deselect **Append '_mb' to attribute names in new tables**, and deselect **Append records if table exists**.
- Click **Next**, **Next**, and **Finish**.

5. Import the metadata for the spatial map data. In Map Builder, select **Tools/Import Metadata**.
   - Click **File**; navigate to and **Open** the `NMS_CONFIG/spatial_landbase/SpatialMetadata.dat` file.
   - Select **Styles**, **Themes**, and **Base Maps**, and then click **Ok**.
6. Verify that the OPAL spatial landbase maps and metadata loaded correctly.
   - In the Map Builder left panel, which lists the Metadata directory, expand the **Base Maps** directory and double click the **PRODUCT_PROJECTED_LANDBASE Base Map** icon.
   - From the main panel, in the **PRODUCT_PROJECTED_LANDBASE** top tab, select the **Preview** lower tab, then click the green "**Play**" icon. You should see OPAL landbase data appear in the preview panel.

   Remember to set up a new Generic Data Source in the Oracle WebLogic Server Runtime Configuration to include a name like **JDBC Data Source-spatial** with a JNDI name: **jdbc/spatial** pointing to the same database as the jdbc/intersys connection.

Spatial Outage Summary Installation

This optional installation procedure provides support for displaying spatial outage summary information in the Oracle Utilities Network Management System Web Viewer and supports the interface to Oracle Utilities Customer Self Service.

1. Install the optional Spatial Landbase Maps (from the previous section).
2. Copy `$NMS_BASE/dist/baseconfig/product/ops/viewer/xml/SPATIALLAYERS_SPATIAL_BG_LAYERS.inc` to `$NMS_CONFIG/jconfig/ops/viewer/xml/`.
3. Change `PRODUCT_PROJECTED_LANDBASE` to **`CSS_PROJECTED_LANDBASE`** as shown below:

```
<!-- Used in SpatialLayers.xml
     Used to define the connection string for Oracle WebLogic Server
(WLS)
     to the spatial server -->
<SpatialBGLayers>
    <SpatialBGLayer datasource_name="spatial" jndi_name="jdbc/spatial"
    basemap_name="CSS_PROJECTED_LANDBASE"
    viewer_layer_name="spatial_landbase"/>
</SpatialBGLayers>
```

4. Run the setup script for the OPAL outage summary views:

```
$ OPAL-CSS-setup
```

5. Edit the `$NMS_HOME/etc/system.dat` file by adding or changing the **program TSService** line to have the **outageSumScript** `and` **outageSumPeriod** parameters:

```
program TSService TSService -outageSumScript $NMS_HOME/bin/OPAL-CSS-refresh
-outageSumPeriod 1
```

For demonstration environments, the recommended outageSumPeriod is 1 (minute); for production environments, a value of 10 to 15 is recommended.

6. Remove the **#** (comment) sign at the beginning of the line:

   #`instance <local> TSService`

7. Save the file.
8. Stop and restart SMService:

   ```
   $ Action any.SMService+TSService stop
   $ sms-start
   ```

9. If you are adding this optional feature after installing and configuring the web application server (described in the sections starting with **Web Application Configuration** on page 3-10), install your new java configurations using the following command:

   ```
   $ nms-install-config --java
   ```

   Re-deploy and restart the Web Application in WebLogic NMS. Otherwise you will complete this process in the following sections.

Web Map Server Connection

Google Maps

This optional installation procedure provides support for displaying Google Maps behind the Oracle Utilities Network Management System Web Viewer:

1. Edit or create a project configuration of the `DLG_VIEWER_HIDE_DISPLAY_en_US.properties` file in `$NMS_CONFIG/jconfig/ops/viewer/properties`:

   ```
   $ mkdir -p $NMS_CONFIG/jconfig/ops/viewer/properties
   $ vi  $NMS_CONFIG/jconfig/ops/viewer/properties \
  /DLG_VIEWER_HIDE_DISPLAY_en_US.properties
   ```

   Note: see the Viewer Configuration section in the Java Application Configuration chapter of the *Oracle Utilities Network Management System Configuration Guide* for other Hide/Display configuration options.

2. Add the following lines to the end of the `DLG_VIEWER_HIDE_DISPLAY_en_US.properties` file and save it. These will be the labels in the Web Viewer Hide/Display Spatial Landbase drop-down list:

   ```
   CHBOX_HD_SPATIAL_MAPSRV1_ROAD.text = Google Roadmap
CHBOX_HD_SPATIAL_MAPSRV1_AERIAL.text = Google Satellite
   ```

3. If you don't have a project version of the `SPATIALLAYERS_SPATIAL_BG_LAYERS.inc` file in `$NMS_CONFIG/jconfig/ops/viewer/xml`, copy the product file to your configuration directory:

   ```
   $ mkdir -p $NMS_CONFIG/jconfig/ops/viewer/xml
   $ cp \
   ```

```
  $NMS_BASE/dist/baseconfig/product/ops/viewer/xml/ \
   SPATIALLAYERS_SPATIAL_BG_LAYERS.inc \
$NMS_CONFIG/jconfig/ops/viewer/xml
```

4. Change the project version of the `SPATIALLAYERS_SPATIAL_BG_LAYERS.inc` file to have `SpatialBGLayers` referencing the Google Maps server:

```
<SpatialBGLayers>
  <SpatialBGLayer provider="web1" basemap_name="roadmap"
    viewer_layer_name="google_roadmap"
    misc_params="&amp;style=lightness:10.0"/>

  <SpatialBGLayer provider="web1" basemap_name="hybrid"
    viewer_layer_name="google_satellite"
    misc_params="&amp;style=lightness:10.0"/>
</SpatialBGLayers>
```

5. If you don't have a project version of the `DLG_VIEWER_HIDE_DISPLAY_DECLUTTER.inc` file in `$NMS_CONFIG/jconfig/ops/viewer/xml`, copy the product file to your configuration directory:

```
$ mkdir -p $NMS_CONFIG/jconfig/ops/viewer/xml
$ cp \
  $NMS_BASE/dist/baseconfig/product/ops/viewer/xml/ \
    DLG_VIEWER_HIDE_DISPLAY_DECLUTTER.inc \
  $NMS_CONFIG/jconfig/ops/viewer/xml
```

6. Change the project version of the `DLG_VIEWER_HIDE_DISPLAY_DECLUTTER.inc` file to have `COMBOBOX_HD_SPATIAL` referencing the Google Map Server and layers you wish to configure, for example:

```
<ComboBox name="COMBOBOX_HD_SPATIAL">
   <ComboBoxPlacement start="1,relative" width="1" height="1"
    weight="1,0"/>
   <ComboBoxBehavior
    data_source="DS_VIEWER_DEFAULT.SPATIAL_COMBOBOX">
      <Editable initial="false"/>
      <Keys>
         <Key value="None"/>
         <Key value="Google Roadmap"/>
         <Key value="Google Satellite"/>
      </Keys>
      <SelectPerform>
         <Command value="RefreshCommand"/>
      </SelectPerform>
   </ComboBoxBehavior>
</ComboBox>
```

7. If you don't have a project version of the `CentricityServer.properties` file in `$NMS_CONFIG/jconfig/server`, copy the product file to your configuration directory:

```
$ mkdir -p $NMS_CONFIG/jconfig/server
$ cp $NMS_BASE/dist/baseconfig/product/server/ \
  CentricityServer.properties $NMS_CONFIG/jconfig/server
```

8. Change the project version of the `CentricityServer.properties` file to have parameters referencing the Google Map Server license keys, URL, and configuration information provided to you by Google:

```
# http address:
# http base address required by the web map service
viewer.web1_maps_http_address = http://maps.googleapis.com/maps/
api/staticmap

# http browser address:
# http base browser address
viewer.web1_maps_browser_http_address = http://maps.google.com

# key:
# Key as provided by Google for basic or trial key.
# If using Google Maps for Work/Business, set key to the
# provided crypto key and also set the client_id as provided
viewer.web1_maps_key = AIzzzzzzzzzzzzzzz20-4xxxxxxxxxxxxxxxFg
#viewer.web1_maps_key = SXXXXXxXXxxXXxXXXXXxXXxxxxQ=
#viewer.web1_maps_client_id = gme-oraclenms

# logo:
# name of the logo image file of the map server
viewer.web1_maps_logo = google-map-logo_sm.png

# mapsize:
# Recommened size: 640 - will work
# Max 2048 based on server
viewer.web1_maps_mapsize = 640

# If this is set to true, the map will be downloaded by the server.
# If it is false, it will instead be downloaded by the client
viewer.web1_server_download = true


# scale:
# if map size=640, recommended scale: 2
# if map size 2048, recommended scale: 1
# domain: 1, 2, 4
viewer.web1_maps_scale = 2

# copyright information may be required to be displayed on the
# bottom of the static map image.  This value will define the height
# to clip off the original image and scale to the bottom of
# the clipped image
viewer.web1_maps_copyright_height_percent = 0.016

# This parameter will identify which map type server to use
# for non-spatial map mode, value should be "web1" or "web2",
# when focusing in a browser.
viewer.default_maps_browser_provider = web1

# Please be sure to set the ces_parameters for
# MBS_LL_PROJ_COORDSYS and MBS_GEO_PROJ_COORDSYS values
```

9. If you are adding this optional feature after installing and configuring the web application server (described in the sections starting with Web Application Configuration below), install your new java configurations using the following command:

```
$ nms-install-config --java
```

Re-deploy and restart the Web Application in WebLogic NMS. Please note, you may need to reconfigure your WebLogic application if you now need to configure for a proxy-server. Otherwise you will complete this process in the following sections.

Bing Maps

This optional installation procedure provides support for displaying Bing Maps behind the Oracle Utilities Network Management System Web Viewer:

1. Edit or create a project configuration of the DLG_VIEWER_HIDE_DISPLAY_en_US.properties file in $NMS_CONFIG/jconfig/ops/viewer/properties:

```
$ mkdir -p $NMS_CONFIG/jconfig/ops/viewer/properties
$ vi $NMS_CONFIG/jconfig/ops/viewer/properties/
  DLG_VIEWER_HIDE_DISPLAY_en_US.properties
```

   Note: see the "Viewer Configuration" section in the "Java Application Configuration" chapter of the *Oracle Utilities Network Management System Configuration Guide* for other Hide/Display configuration options.

2. Add the following lines to the end of the `DLG_VIEWER_HIDE_DISPLAY_en_US.properties` file and save it. These will be the labels in the Web Viewer Hide/Display Spatial Landbase drop-down list:

```
COMBOBOX_HD_SPATIAL.bing_road.text = Bing Road
COMBOBOX_HD_SPATIAL.bing_aerial.text = Bing Aerial
```

3. If you don't have a project version of the SPATIALLAYERS_SPATIAL_BG_LAYERS.inc file in $NMS_CONFIG/jconfig/ops/viewer/xml, copy the product file to your configuration directory:

```
$ mkdir -p $NMS_CONFIG/jconfig/ops/viewer/xml
$ cp \
  $NMS_BASE/dist/baseconfig/product/ops/viewer/xml/ \
   SPATIALLAYERS_SPATIAL_BG_LAYERS.inc \
$NMS_CONFIG/jconfig/ops/viewer/xml
```

4. Change the project version of the `SPATIALLAYERS_SPATIAL_BG_LAYERS.inc` file to have `SpatialBGLayers` referencing the Bing Maps server:

```
<SpatialBGLayers>
   <SpatialBGLayer provider="web2" basemap_name="road"
    viewer_layer_name="bing_road"/>
   <SpatialBGLayer provider="web2"
basemap_name="aerialwithlabels"
    viewer_layer_name="bing_aerial"/>
</SpatialBGLayers>
```

5. If you don't have a project version of the `DLG_VIEWER_HIDE_DISPLAY_DECLUTTER.inc` file in `$NMS_CONFIG/jconfig/ops/viewer/xml`, copy the product file to your configuration directory:

```
$ mkdir -p $NMS_CONFIG/jconfig/ops/viewer/xml
$ cp \
  $NMS_BASE/dist/baseconfig/product/ops/viewer/xml/ \
   DLG_VIEWER_HIDE_DISPLAY_DECLUTTER.inc \
$NMS_CONFIG/jconfig/ops/viewer/xml
```

6. Change the project version of the `DLG_VIEWER_HIDE_DISPLAY_DECLUTTER.inc` file to have `COMBOBOX_HD_SPATIAL` referencing the Bing Map Server and layers you wish to configure, for example:

```
<ComboBox name="COMBOBOX_HD_SPATIAL">
   <ComboBoxPlacement start="1,relative" width="1" height="1"
    weight="1,0"/>
   <ComboBoxBehavior
    data_source="DS_VIEWER_DEFAULT.SPATIAL_COMBOBOX">
      <Editable initial="false"/>
      <Keys>
         <Key value="None"/>
         <Key value="Bing Road"/>
         <Key value="Bing Arial"/>
      </Keys>
      <SelectPerform>
         <Command value="RefreshCommand"/>
      </SelectPerform>
   </ComboBoxBehavior>
</ComboBox>
```

7. If you don't have a project version of the `CentricityServer.properties` file in `$NMS_CONFIG/jconfig/server`, copy the product file to your configuration directory:

```
$ mkdir -p $NMS_CONFIG/jconfig/server
$ cp \
    $NMS_BASE/dist/baseconfig/product/server/ \
CentricityServer.properties $NMS_CONFIG/jconfig/server
```

8. Change the project version of the `CentricityServer.properties` file to have parameters referencing the Bing Map Server license keys, URL, and configuration information provided to you by Bing:

```
# http address:
# http base address required by the web map service
viewer.web2_maps_http_address = http://dev.virtualearth.net/
REST/v1/Imagery/Map

# http browser address:
# http base browser address
viewer.web2_maps_browser_http_address = http://www.bing.com/
maps

# key:
# Key as provided by the web map service
viewer.web2_maps_key = Ar5_nnnnnnnn_cc-VwwwwwwwwwwwwwwwwZ-
p5bbbbbbbbbbbbbbbbbbbbbbbbbbZ

# logo:
# name of the logo image file of the map server
viewer.web2_maps_logo = bing_logo_sm.png

# mapsize:
# Recommended size: 834
viewer.web2_maps_mapsize = 834

# If this is set to true, the map will be downloaded by the
server. # If it is false, it will instead be downloaded by the
client
viewer.web2_server_download = true
```

```
   # This parameter will identify which map type server to use for
   # non-spatial map mode, value should be "web1" or "web2", when
   # focusing in a browser.
   viewer.default_maps_browser_provider = web2

   # Please be sure to set the ces_parameters for
# MBS_LL_PROJ_COORDSYS and MBS_GEO_PROJ_COORDSYS values
```

9. If you are adding this optional feature after installing and configuring the web application server (described in the sections starting with Web Application Configuration below), install your new java configurations using the following command:

   ```
   $ nms-install-config --java
   ```

   Re-deploy and restart the Web Application in WebLogic NMS. Please note, you may need to reconfigure your WebLogic application if you now need to configure for a proxy-server. Otherwise you will complete this process in the following sections.

Oracle Locator Server Connection

This optional installation procedure provides support for displaying Oracle Locator maps behind the Oracle Utilities Network Management System Viewer. These instructions assume you have a populated Oracle Locator database accessible to the WebLogic server and that you have set up a datasource in WebLogic to connect to the Locator database schema. If you do not have a Locator database, refer to Oracle Locator and the Oracle (Fusion Middleware) MapViewer Map Builder documentation on building a Locator database.

1. If you do not have a project version of the `SPATIALLAYERS_SPATIAL_BG_LAYERS.inc` file in `$NMS_CONFIG/jconfig/ops/viewer/xml`, copy the product file to your configuration directory:

   ```
   $ mkdir -p $NMS_CONFIG/jconfig/ops/viewer/xml
   $ cp \
     $NMS_BASE/dist/baseconfig/product/ops/viewer/xml/ \
      SPATIALLAYERS_SPATIAL_BG_LAYERS.inc \
   $NMS_CONFIG/jconfig/ops/viewer/xml
   ```

2. Change the project version of the `SPATIALLAYERS_SPATIAL_BG_LAYERS.inc` file to have `SpatialBGLayers` referencing the Bing Maps server:

   ```
   <SpatialBGLayers>
   <SpatialBGLayer datasource_name="spatial"
       jndi_name="jdbc/spatial"
       basemap_name="PRODUCT_PROJECTED_LANDBASE"
       viewer_layer_name="oracle"/>
      <SpatialBGLayer datasource_name="spatial"
       jndi_name="jdbc/spatial"
       basemap_name="CSS_PROJECTED_LANDBASE"
       viewer_layer_name="oracle_with_outage_summary"/>
</SpatialBGLayers>
   ```

   - Set `datasouce_name` to the WebLogic datasouce name to connect to the Locator server.
   - Set `jndi_name` to the WebLogic datasource jndi name to connect to the Locator server.

- Set the basemap_name to the Oracle Locator/Map Builder base map name.
- Set the viewer_layer_name to match the case insensitive, under bar for blanks key name in the SPATIAL_COMBOBOX below.

3. If you do not have a project version of the DLG_VIEWER_HIDE_DISPLAY_DECLUTTER.inc file in $NMS_CONFIG/jconfig/ops/viewer/xml, copy the product file to your configuration directory:

```
$ mkdir -p $NMS_CONFIG/jconfig/ops/viewer/xml
$ cp \
  $NMS_BASE/dist/baseconfig/product/ops/viewer/xml/ \
   DLG_VIEWER_HIDE_DISPLAY_DECLUTTER.inc \
   $NMS_CONFIG/jconfig/ops/viewer/xml
```

Configuring a Web Call Entry-Only Managed Server

The managed server for Web Call Entry can be configured to not require access to the corbagateway or publisher. (Only the database).

Set the managed server up normally with the following changes:

1. Add -DcorbaInitRef=NONE to the managed server startup parameters (See **Configure the Arguments to Use When Starting a Server in Your Domain** on page 3-27.)
2. Change license.properties to have only LicensedProduct=WebCallentry.
3. Create overrides for Web Call configuration files by copying the following files from $NMS_BASE/dist/baseconfig/product/ops/callentry/xml/:
   - CALLENTRY_MENUBAR.inc
   - DLG_SAVE_FUZZY.xml
   - DLG_SAVE_UNCONNECTED.xml

   to $NMS_CONFIG/jconfig/ops/callentry/xml.
4. Edit the new CALLENTRY_MENUBAR.inc by replacing:

```
<Enabled initial="false" when="SAVEABLE or (FUZZY_CALL and
TXTF_ADDRESS_POPULATED) or INTERSECTION_POPULATED"/>
  <PressPerform>
<Command value="ValidateCallDataCommand"/>
<Command value="SaveCommand"/>
```

   with this:

```
<Enabled initial="false" when="SAVEABLE or (FUZZY_CALL and
 TXTF_ADDRESS_POPULATED) or INTERSECTION_POPULATED"/>
<PressPerform>
<Command value="ValidateCallDataCommand"/>
<Command value="SaveCommand">
<Config name="connection" value="database"/>
</Command>
```

5. Edit DLG_SAVE_FUZZY.xml by changing

```
<Enabled initial="false" when="FUZZY_CALL and
TXTF_ADDRESS_POPULATED"/>
<PressPerform>
```

```
<Command value="SaveCommand"/>
```

to this:

```
<Enabled initial="false" when="FUZZY_CALL and
TXTF_ADDRESS_POPULATED"/>
<PressPerform>
   <Command value="SaveCommand">
      <Config name="connection" value="database"/>
</Command>
```

6. Edit `DLG_SAVE_UNCONNECTED.xml` by changing

```
<ButtonBehavior>
  <Enabled initial="false" when="FUZZY_CALL and
   TXTF_ADDRESS_POPULATED"/>
  <PressPerform>
<Command value="SaveCommand"/>
```

to this:

```
<ButtonBehavior>
  <Enabled initial="false" when="FUZZY_CALL and
   TXTF_ADDRESS_POPULATED"/>
  <PressPerform>
   <Command value="SaveCommand">
      <Config name="connection" value="database"/>
</Command>
```

A limitation of running this way is that the managed server will not pick up new control zones that are used for populating fuzzy calls. If a control zone change is done that would affect that function, an administrator would have to run a refresh action on that managed server.

7. Change the project version of the `DLG_VIEWER_HIDE_DISPLAY_DECLUTTER.inc` file to have `COMBOBOX_HD_SPATIAL` referencing the Bing Map Server and layers you wish to configure, for example:

```
<ComboBox name="COMBOBOX_HD_SPATIAL">
   <ComboBoxPlacement start="1,relative" width="1" height="1"
    weight="1,0"/>
   <ComboBoxBehavior
    data_source="DS_VIEWER_DEFAULT.SPATIAL_COMBOBOX">
      <Editable initial="false"/>
      <Keys>
         <Key value="None"/>
         <Key value="Oracle"/>
         <Key value="Oracle with Outage Summary"/>
   </Keys>
      <SelectPerform>
         <Command value="RefreshCommand"/>
      </SelectPerform>
   </ComboBoxBehavior>
</ComboBox>
```

8. If you are adding this optional feature after installing and configuring the web application server (described in the sections starting with Web Application Configuration below), install your new Java configurations using the following command:

```
$ nms-install-config --java
```

9. Re-deploy and restart the Web Application in WebLogic NMS.

Directory Structure

The Oracle Utilities Network Management System has three directory areas involved with product installation, project configuration, and runtime. This section describes how these directories are created and interact.

Directory Overview

The Oracle Utilities Network Management System Directory structure is comprised of three major areas:

- The Oracle Utilities Network Management System Installation directory
- The Oracle Utilities Network Management System Project Configuration directory
- The Oracle Utilities Network Management System Runtime directory

Installation Directory

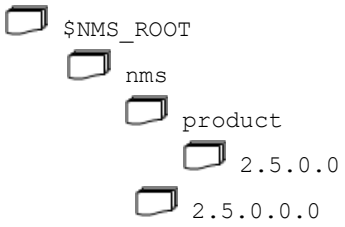
The Oracle Utilities Network Management System Installation directory is created as part of the installation process. There are two environment variables involved with this directory:

- **$NMS_ROOT:** Points to the top-level of the Oracle Utilities Network Management System installation directory. `$NMS_ROOT` is typically set to be the nmsadmin user home directory (for example, users/nmsadmin).

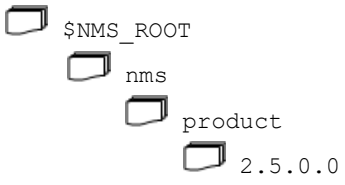
  The installation process will create a directory structure under `$NMS_ROOT`; the top level is named `nms`, and it has a sub-directory named `product`. The `product` directory contains directories for each installed Oracle Utilities Network Management System version, which will include the major release or service pack and any applied patches.

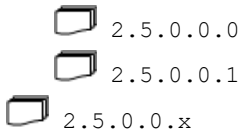
  For example:

  - **NMS 2.5.0.0.0 (Major Release)**

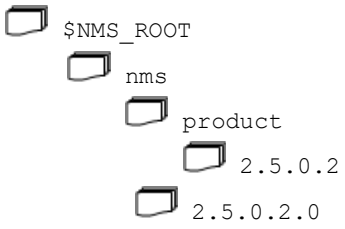


  As patches are applied, they are added to the product directory:





- **NMS 2.5.0 Service Pack 2**



As in the case of the major release example above, as patches are applied to the Service Pack, their directories are added to the product directory.

- **$NMS_BASE:** Points to the current major release or service pack version without the patch level number. For example, 2.5.0.0 (major release) or 2.5.0.1 (service pack).

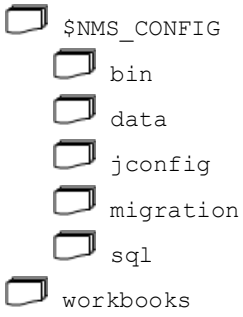
The installation directory will contain all content from the Oracle E-Delivery and should not be changed by the runtime system or user.

Project Configuration Directory

The Oracle Utilities Network Management System Project directory will contain all project-specific configurations, scripts, and programs required to run the Oracle Utilities Network Management System. This directory is identified by the `$NMS_CONFIG` environment variable and is typically set to `$NMS_HOME/project_name`; the Oracle Utilities Network Management System demonstration environment's project name is OPAL so $NMS_CONFIG is set to `$NMS_HOME/OPAL`.

> **Note:** Configuration files in the configuration directory need to be prefixed by the project name. For example, `OPAL_devices.sql`. For more information on creating custom configuration files, see the *Oracle Utilities Network Management System Configuration Guide.*

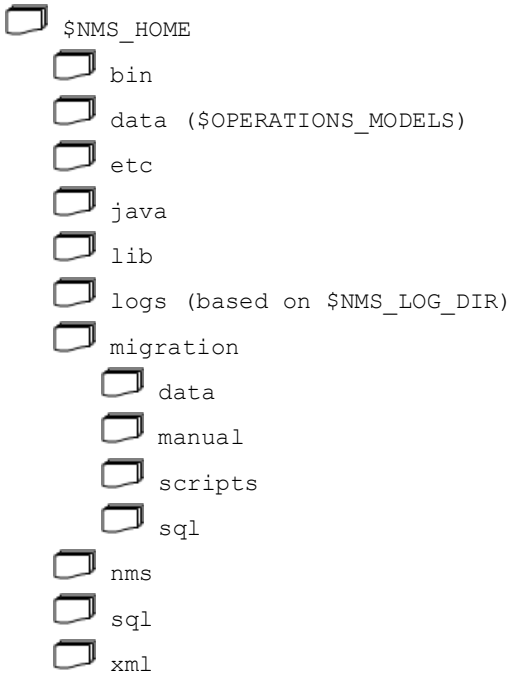
The top level directory structure for the project configuration directory is as follows.



Other files and directories can exist in the `$NMS_CONFIG` directory, but only those documented above will be utilized by the install script: nms-install-config.

Runtime Directory

The Oracle Utilities Network Management System runtime directory will contain all runtime specific configurations, scripts, and programs required to run the Oracle Utilities Network Management System. This directory is identified by the environment variable: `$NMS_HOME` and, for Oracle Utilities Network Management System Release 2.5.0.2.0, this must be set to `$HOME` (`/users/nmsadmin`).



The NMS executables and shared libraries are in the `$NMS_BASE/bin` (for example, `/users/nmsadmin/product/2.5.0.0/lib`) and `$NMS_BASE/lib` (for example, `/users/nmsadmin/product/2.5.0.0/lib`) directories.

Directory Administration

The install process establishes the Oracle Utilities Network Management System Installation Directory. The project implementation team creates the contents of the Oracle Utilities Network Management System Project Directory. The `nms-install-config` script installs the product and project configurations into the runtime directories. Project files will either be appended to or override the product configurations based on type.

Starting Services

Use the following procedure to start full Oracle Utilities Network Management System services:

1. Log in as the administrative user and stop any services that might be running, as follows:

   ```
   sms-stop -a
   ```

2. Run the sms-start script to start services, as follows:

   ```
   sms-start
   ```

3. To verify services are running, run **smsReport**, as follows:

   ```
   smsReport
   ```

Troubleshooting

Please refer to the "Troubleshooting" chapter of the *Oracle Utilities Network Management System Configuration Guide* for information on typical debugging strategies for various aspects of the system as well as locations of log files that contain pertinent information about the runtime applications.

Appendix A

Applying Migrations for a New Release

This appendix defines the steps required to migrate a system to a new release.

Disabling System Logins

When patches are applied to the system, it is important for all users to be out of the system. This can be accomplished with the following Action command from the user that is running the nms services:

```
Action any.pub* ejb enable_login false
```

This will force all users to log out and keep them from logging back in. After the system is updated, and you wish to restore logins, run this command:

```
Action any.pub* ejb enable_login true
```

Applying Product Migrations

The Apply Product Migrations process migrates the model of an older Oracle Utilities Network Management System release to that of a new software version. Based on a release level identifier, the migration process determines the differences between the current model and that of a new release. After the installation of a new release of software, and the loading of a copy of your existing production database, you will need to do the following:

- Execute the `$NMS_BASE/bin/nms-setup` script.

This script will call another script called `nms-apply-migrations`, which determines the differences between the release level of the software and the model database. This script then determines the required and optional migrations by accounting for differences in the release database requirements.

Manual Product Migrations

If a manual migration is required, the `nms-setup` script will stop at that point and alert the user of the required manual migration. When this occurs, please see the corresponding manual migration file in the `$NMS_HOME/migration/manual` directory

for details on what is required for this migration. The files in this directory are named <####>.txt, where <####> is the bug or problem report (PR) number.

The $NMS_CONFIG/migration/data/<project>_config_ready.dat file serves as a "sign-off" document for the Oracle Utilities Network Management System project team. As you determine that a manual migration has been completed (or is not needed for your system), you must add the corresponding bug numbers to the $NMS_CONFIG/migration/data/<project>_config_ready.dat file by entering one bug number per line. Once you have edited this file, you can run $NMS_BASE/bin/nms-install-config to copy it to the $NMS_HOME/migration/data directory or manually copy the file there if you prefer. This signals the migration script that this particular manual migration has been completed. Once the file has been properly copied to $NMS_HOME/migration/data, you need to rerun the nms-setup script. Continue this process until all manual and automated migrations are executed.

Please note that if the This appendix defines the steps required to migrate a system to a new release. process was followed, the configuration changes that are in $NMS_CONFIG/jconfig will have already been updated. While these changes are listed as requiring manual migrations, they just need to be added to the <project>_config_ready.dat. It is expected that in the future the manual migrations will include only those configuration changes needed to configure a new feature or option.

> **Note:** The bug numbers indicated in the manual migration may not be listed in the *Oracle Utilities Network Management System Release Notes* supplied with the release. The migrations always refer to an original bug, which is associated with a particular release; any other releases that receive the fix will have a separate bug number (a "copy-bug"). When resolving manual migration issues, always refer back to the text files placed in the $NMS_HOME/migration/manual directory and not the Release Notes HTML associated to that bug fix.

Command Line Options

The nms-apply-migrations script can be initiated directly from the command line in order to view some of the things that it will be doing when started from the nms-setup script. The following table describes all of the command line options for this script.

Option	Description
-debug	Displays debug information.
-showme	List all processes that would be executed, but do not actually execute any programs or SQL files.
-needConfig	Displays a list of migrations that are required by a project.
-listMigrations	Displays a list of migrations needed without applying them.

> **Note**: The nms-apply-migrations script should not be run without any command-line arguments since that would cause the migrations to actually be executed. The command-line arguments listed above are to be used with the script so that it can be run in a "show only" mode but won't actually do the migrations.

Installing Product Migration Files

The data files that are required for the migration process are installed in the `$NMS_HOME/migration/data` directory. After making changes to the project-specific `$NMS_CONFIG/migration/data/<project>_config_ready.dat` file and an optional special `$NMS_CONFIG/migration/data/<project>_migration.dat` file, run `nms-install-config` script to install them into the `$NMS_HOME/migration/data` directory.

The Product Migration Process

The `nms-apply-migrations` script determines the database differences by comparing the database release level in the `CES_PARAMETERS` table with the software release levels found in the software_release_id.dat and software_release_levels.dat files. Based on these differences, it will create a list containing all of the necessary migrations.

The migration process, or `nms-apply-migrations`, finds the necessary migrations in the `$NMS_HOME/migration/data/pr_migration.dat` file, which contains the list of PRs, releases, patch levels, and configuration types. If there are project-specific migrations, then a optional `<project>_pr_migration.dat` file is also used.

The pr_migration.dat files resemble the following example:

PR	Release	Patch	Required	Config Required	Script Exists	ConfigType
----	-------	-----	--------	------	------	----------
19254	5.5	3	Y	Y	Y	config_sql
19831	6.0	3	Y	N	Y	schema_sql

The following table describes the pr_migration.dat file columns.

Column	Description
PR	Bug number for the migration.
Release	Migration release level, two numbers not including the first digit. For example, release 1.8.1 would be just 8.1 in this field.
Patch	Migration patch level. If the release if 1.8.1.2, then the Patch would be 2.
Required	Whether or not this migration is required for the system to function properly. If set to Y, all projects would be forced to execute this migration when encountered. A value of N means that the migration is optional, and it would be skipped for any projects that do not list it within their <proj>_config_ready.dat file.

Column	Description
Config Required	Whether or not configuration is required by a project for the system to function properly. This value is set to Y whenever a change is made that requires configuration work. For instance, if a new required column is added to a configuration table, the population of this new column properly is the domain of the project engineer, not the developer. Setting this field to Y will flag to all project engineers that this migration requires their attention before the migration can be executed. The specific instructions for configuration migration will be documented in the bug's Migration section; the manual migration text file located in $NMS_HOME/migration/manual. Project engineers signify that the configuration has been examined and completed by adding this migration bug to the <proj>_config_ready.dat file.
Script Exists	Indicates whether a script exists for the migration. For example, if a script exists for bug 19254, then there is a script pr19254-migration that performs the migration. Not all migrations involve explicit scripts. As an example, a configuration table change would normally not require a migration. However, if it is important that a new configuration column be properly populated, this must be flagged for project engineers. This is done by adding the bug to pr_migration.dat, setting Config Required to Y and Script Exists to N. Even though there is no migration script, the migration process will not proceed until the project engineer has signified that the configuration is complete by adding the bug to the <proj>_config_ready.dat file.
Config Type	Describes the type of configuration change. Valid values are: • config_sql - A configuration SQL file has changed. • schema_sql - A schema SQL file has changed. • retain_sql - A retain SQL file has changed. • core_sql - A core (required) data SQL file has changed. • data - Model (facilities) data is being migrated. • app_defaults - New or obsolete application default options. • map_rebuild - The migration script will regenerate map files. • metafile_rebuild - The script will regenerate all map metafiles. • service_restart - Services must be restarted. • environment_restart - All user environments must be restarted. • post_setup - migration is run during nms-post-setup

Correcting Warnings and Errors

The table below shows the corrections for some possible errors you might receive when running the nms-apply-migrations script.

Warning	Remedy
WARNING THE FOLLOWING MIGRATIONS NEED CONFIGURATION PR_NUMBER RELEASE_PATCH	This warning is displayed when migrations requiring manual changes are found. To determine the necessary changes, refer to the corresponding file in the $NMS_HOME/migration/manual directory. After making the manual changes, add the PR number to the $NMS_CONFIG/migration/<project>_config_ready.dat file.
DATABASE RELEASE LEVEL IS GREATER THAN SOURCE RELEASE LEVEL MIGRATING BACKWARDS NOT SUPPORTED	This error indicates that the schema level of the database is greater than the runtime executables that are being used. You can return to a prior release if you execute the nms-setup script with the -clean command line option and perform a model build. You should not return to a prior release without running a nms-setup -clean and a model build, for there may be unresolved problems that could cause system instability.

Applying Custom Migrations for NMS Integrators

The custom migration process allows you to apply custom migrations to the database.

Process Overview

- Custom migrations are entered in an XML file (`utility_migrations.xml`) that is saved to `$NMS_CONFIG`.
- Migrations are applied by running `nms-setup` or when you run `nms-apply-migration` manually.
- `nms-setup` and `nms-apply-migrations` call a script (`utility-apply-migrations`) that launches `utility-migrations.jar`, which creates or updates the `UTILITY_MIGRATIONS` database table and then applies the migrations.

Adding Custom Migrations to the Utility Migrations XML File

The `utility_migrations.xml` file, which is based on the utility_migrations.xsd schema (found in `$NMS_BASE/xml/`), has the following structure:

```
    <utility_migration>
    <migration>
    <base_name>migration1</base_name>
    <before>...</before>
    </migration>
    <migration>
    <base_name>migration2</base_name>
    <before>...</before>
    </migration>
    .
    .
    .
</utility_migration>
```

- The **<base_name>** element contains the name of the migration file without any suffix. The migration files are either scripts or *.sql files found in `$NMS_HOME/migration/`.
- Any number of <migration> elements may be included in utility_migrations.xml.
- The **<before>** element must have one of the following values:

Value	Description
T	True, if the migration is to be run before product migrations.
F	False, if the migration is to be run after product migrations.

Running the Migration Scripts

nms-setup and nms-apply-migrations run a utility migration script that checks for the existence of the utility_migration.xml file. When the utility_migrations.xml file is read, the migration information is used to update the `UTILITY_MIGRATIONS` database table.

All migrations added to the table must actually exist or an error is generated. For any given `<base_name>` element, the migration system first looks in `$NMS_HOME/migrations/scripts` for a script with the given base_name found in the xml file. If such a base_name script is found, it will be run when nms-setup is run. Otherwise, the application will look in `$NMS_HOME/migrations/sql` for a `.sql` file with the base_name given in the xml file. If a migration is not found in either directory, an error is generated. All scripts must be executable or the migration will fail to run.

UTILITY_MIGRATIONS Database Table

The migrations found in the `$NMS_CONFIG/utility_migrations.xml` file are stored in the UTILITY_MIGRATIONS database table, which is defined by the following:

```
create table utility_migrations (
             stamp TIMESTAMP not null,
             base_name VARCHAR2(32) unique not null,
             before VARCHAR2(1) not null,
             applied VARCHAR2(1) not null,
             status VARCHAR2(32) not null)
```

where

- The **stamp** column is an Oracle TIMESTAMP datatype representing the time when the migration was added to the table. The base_name column should contain the simple file name of the migration without any suffix. The utility-apply-migrations script will look for this sql file in $NMS_HOME/migration/sql. It will also look in $NMS_HOME/migration/scripts for a corresponding file name script, if one exists. If a script exists, it will be executed; otherwise, the sql file is run in instead.
- The **before** column is set to "T" if a migration is to be applied before product migrations and to "F" if it is to be applied after product migrations.
- The **status** column will contain the most recent status for a migration. This will usually will be *Inserted*, *Applied*, or *Failed*.
- The **applied** column is "T" if the migration has been applied to the database. If the applied column is "F", the migration has not yet been applied to the database.

  Note: once a migration has been applied to the database, that migration can no longer be changed in the database.

Appendix B

Configuring a Translated User Interface

Available language translations for the Oracle Utilities Network Management System (NMS) user interface can be found in the directory $NMS_BASE/i18n in your Unix/Linux installation directory. These language translations are done in response to customer requests, and are released along with standard NMS patch bundles. If you do not see a subdirectory matching your desired language locale, please contact Oracle Support regarding availability of a translation.

To configure your system to use a language locale that is found in $NMS_BASE/i18n, complete the following steps:

1. Login as the administrative user (for example, nmsadmin).
2. Run the `nms-env-config --translation-config` script:
   - Respond with a 'y' when prompted "Should the NMS user interface be translated from English? (y/n)".
   - Enter the valid language locale code when prompted for the value of NMS_LANG (for example, tr_TR for Turkish)
   - Log out and login as the administrative user again to ensure that NMS_LANG is set correctly in your environment.
3. Edit $NMS_CONFIG/jconfig/build.properties and uncomment the following line

   ```
   dir.localization = ${env.NMS_HOME}/java/i18n
   ```

4. Run nms-setup. Check the log at the end to ensure that database translation was successful.
5. Create a translated symbols file:

   ```
   nms-make-symbols
   ```

6. Restart services

   ```
   sms-start
   ```

7. Apply the new Java configuration:

   ```
   nms-install-config -java
   ```

8. Redeploy the NMS applications by completing all steps in Deploying Oracle Utilities Network Management System in WebLogic Server.