# Oracle® Fusion Middleware Administering Node Manager for Oracle WebLogic Server





Oracle Fusion Middleware Administering Node Manager for Oracle WebLogic Server, 15c (15.1.1.0.0)

G31434-01

Copyright © 2007, 2025, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

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

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

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

## Contents

### Preface

Audience	
Documentation Accessibility	
Diversity and Inclusion	
Related Documentation	
Conventions	i
About Node Manager	
Introduction	1
Functionalities of Node Manager	1
Start, Shut Down, and Restart an Administration Server	1
Start, Shut Down, Suspend, and Restart Managed Servers	2
Restart Administration and Managed Servers Automatically	2
Monitor Servers and View Log Data	3
Node Manager Implementations	3
Java-Based Node Manager	3
Script-Based Node Manager	3
Determining Which Node Manager Implementation to Use	4
Accessing Node Manager	4
How Node Manager Works in the Oracle WebLogic Server Environment	4
Diagram of Node Manager and Servers	5
How Node Manager Starts an Administration Server	6
How Node Manager Starts a Managed Server	8
How Node Manager Restarts an Administration Server	g
How Node Manager Restarts a Managed Server	10
How Node Manager Shuts Down a Server Instance	12
Node Manager and System Crash Recovery	13
Node Manager Configuration and Log Files	14
Configuration Files	15
nodemanager.properties	15
nodemanager.domains	15
nm_password.properties	15
boot.properties	15

	startup.properties	15
	server_name.addr	16
	server_name.lck	16
	server_name.pid	16
	server_name.state	16
	Log Files	16
	nodemanager.log	17
	Log File Rotation	17
	server_name.out	20
	Configuring Log File Rotation	21
2	Configuring Java Node Manager	
	Overview	1
	Default Node Manager Configuration	1
	Configuring Per Host Node Manager	2
	Configuring Node Manager on Multiple Machines	4
	Controlling and Configuring Node Manager Using WLST	5
	Configuring Node Manager Using WLST Offline	5
	Configuring Java-Based Node Manager Security	6
	Specifying Node Manager User Name and Password	6
	Remote Server Start Security for Java-Based Node Manager	8
	Using SSL With Java-Based Node Manager	8
	Advanced Node Manager Configuration	9
	Defining the Administration Server Address	9
	Configuring Node Manager to Use Start and Stop Scripts	9
	Script Location	10
	Using Start Scripts	10
	Best Practices When Using Stop Script	11
	Using Stop Scripts	11
	Configuring nodemanager.domains File	11
	Reviewing nodemanager.properties	12
	Node Manager Properties	13
	Machine-Level Node Manager Settings for a Group of Server Instances	24
	Configuring Remote Startup Arguments Setting Server Startup Properties	25 26
		26
	startup.properties	
	Setting Startup Properties Using WLST	26 27
	Server Startup Properties Setting the Node Manager Environment Variables	28
	Configuring Node Manager as an xinetd Service	29
	Configuring Node Manager as a systemd Service	30
	Comiganing Node Manager as a systema Service	30

Configuring Per Domain Node Manager as a systemd Service	30
Configuring Per Host Node Manager as a systemd Service	31
Configuring Script-Based Node Manager	
Overview	1
Step 1: Create User Accounts	1
Step 2: Configure Node Manager Security	2
Step 3: Install Oracle WebLogic Server	3
Step 4: Create a WebLogic Domain	3
Step 5: Configure nodemanager.domains File	4
Step 6: Start the Administration Server	4
Step 7: Configure Node Manager on the Managed Servers	5
Step 8: Test Node Manager Setup and Configuration by Starting Managed Servers	6
Step 9: Configure UNIX Machines	7
Step 10: Assign Servers to Machines	8
Step 11: Start Managed Servers	8
Configuring Script-Based Node Manager Security	8
Overriding the Default SSH Port	8
Configuring Security for Oracle WebLogic Server Scripts	9
Configuring Remote Server Start Security for Script-Based Node Manager	9
Generating and Distributing Key Value Pairs	9
Shared Key Value Pair	10
Individual Key Value Pairs	10
Using Node Manager	
Starting and Stopping Node Manager	1
Running Node Manager as a Startup Service	1
Starting Java-Based Node Manager Using Scripts	2
Command Syntax for Starting Java-Based Node Manager	3
Running Script-Based Node Manager	4
Stopping Node Manager	6
Stopping Java-Based Node Manager Using Scripts	7
Using Node Manager to Control Servers	7
Starting the Administration Server Using Node Manager	7
Starting Managed Servers Using WLST	8
Starting Managed Servers Using the Remote Console	9
Configuring a Machine to Use Node Manager	9
Assigning Server Instances to a Machine	10
Starting Managed Servers without an Administration Server	10

## 5 Node Manager Tutorial

Creating Node Manager in a New Domain	1
Starting Node Manager	2
Using Node Manager to Start a Managed Server	2



## **Preface**

This document describes how to configure and use Node Manager to control and manage server instances within an Oracle WebLogic Server environment.

### **Audience**

This document is a resource for system administrators and operators responsible for using Node Manager. It is relevant to all phases of a software project, from development through test and production phases.

It is assumed that the reader is familiar with Jakarta Platform, Enterprise Edition and web technologies, object-oriented programming techniques, and the Java programming language.

## **Documentation Accessibility**

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <a href="http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc">http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc</a>.

#### **Access to Oracle Support**

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <a href="http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info">http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info</a> or visit <a href="http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs">http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs</a> if you are hearing impaired.

## **Diversity and Inclusion**

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

### Related Documentation

The list of related documents provides easy access to the associated/related content.

- Creating WebLogic Domains Using the Configuration Wizard
- Understanding Domain Configuration for Oracle WebLogic Server
- Oracle WebLogic Remote Console Online Help
- Administering Server Startup and Shutdown for Oracle WebLogic Server



#### **New and Changed WebLogic Server Features**

Several new features have been introduced in this release and features have been changed or modified. For a comprehensive listing of the new Oracle WebLogic Server features, see *What's New in Oracle WebLogic Server*.

### 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.

## **About Node Manager**

Node Manager is an Oracle WebLogic Server utility. It is used to control Administration Servers and Managed Servers.

This chapter includes the following topics:

### Introduction

Node Manager is an Oracle WebLogic Server utility that enables you to start, shut down, and restart Administration Server and Managed Server instances from a remote location.

Server instances in an Oracle WebLogic Server production environment are often distributed across multiple domains, machines, and geographic locations. Although Node Manager is optional, Oracle recommends using it if your Oracle WebLogic Server environment hosts applications with high availability requirements because Node Manager allows you to control the running state of distributed server instances from a centralized location.

Node Manager must run on each computer that hosts Oracle WebLogic Server instances—whether Administration Server or Managed Server—that you want to control with Node Manager. See <u>Running Node Manager as a Startup Service</u>.

For a basic tutorial that demonstrates how to create a default per domain Node Manager instance in a single-machine domain to start and stop Managed Server, see <a href="Node Manager">Node Manager</a> Tutorial.

## Functionalities of Node Manager

Node Manager controls the Administration and Managed Servers. Using Node Manager, you can start, shut down, restart the servers, and monitor the log data.

The following sections describe the basic functionalities of Node Manager:

### Start, Shut Down, and Restart an Administration Server

You can use the WebLogic Scripting Tool (or SSH client for script-based Node Manager only) to connect to a Node Manager process on the machine that hosts the Administration Server and issue the commands to start, shut down, or restart an Administration Server. The relationship between Administration Server and Node Manager varies for different scenarios.

- An Administration Server can be under Node Manager control You can start it, monitor it, and restart it using Node Manager.
- An Administration Server can be a Node Manager client When you start or stop Managed Servers from the WebLogic Remote Console, you are accessing Node Manager using the Administration Server.
- An Administration Server supports the process of starting up a Managed Server with Node Manager - When you start a Managed Server with Node Manager or a start script, the Managed Server contacts the Administration Server to obtain the pending configuration updates.



### Start, Shut Down, Suspend, and Restart Managed Servers

You can issue commands to Node Manager to start, shut down, suspend, and restart Managed Server instances and clusters by using one of the following options:

- WebLogic Server Scripting Tool (WLST)
- WebLogic Remote Console
- Scripts

Node Manager can restart a Managed Server after failure even when the Administration Server is unavailable if Managed Server Independence (MSI) mode is enabled for that Managed Server instance. This mode is enabled by default.

#### (i) Note

- If using the pack and unpack commands, Node Manager can start a Managed Server for the first time in the MSI mode. However, if using the nmEnroll command, Node Manager cannot start a Managed Server for the first time in MSI mode because the Administration Server for the domain must be available so the Managed Server can obtain its configuration settings.
- Node Manager uses the same command arguments that you provide when starting a Managed Server with a script or at the command line. For information about startup arguments, see weblogic. Server Command-Line Reference in Command Reference for Oracle WebLogic Server.

### Restart Administration and Managed Servers Automatically

If a server instance that was started using Node Manager fails, Node Manager automatically restarts it.



#### (i) Note

Node Manager can restart a server instance that was started using only Node Manager.

The restart feature is configurable. Node Manager's default behavior is to:

- Automatically restart server instances under its control that fail. You can disable this feature.
- Restart failed server instances no more than a specific number of times. You define the number of restarts by setting the RestartMax property in the Node Manager startup.properties file.

If Node Manager fails or is explicitly shut down, then upon restart, it determines the server instances that were under its control when it exited. Node Manager can restart any failed server instances as needed.





#### (i) Note

It is advisable to run Node Manager as an operating system service so that it restarts automatically when its host machine restarts.

### Monitor Servers and View Log Data

Node Manager creates a log file for a Node Manager process and a log file of server output for each server instance it controls. See Log Files.

## **Node Manager Implementations**

Oracle WebLogic Server provides two implementations of Node Manager, Java-based and script-based, with similar functionality. Each implementation has different configuration and security considerations.

This section includes the following topics:

### Java-Based Node Manager

Java-based Node Manager runs within a Java Virtual Machine (JVM) process. Oracle recommends that you run it as a Windows service on Windows platforms and as an operating system service on UNIX platforms, allowing it to restart automatically when the system is rebooted. You can configure Java-based Node Manager using the Configuration Wizard or WLST offline.

This implementation of Node Manager determines its configuration from the nodemanager.properties file. See Reviewing nodemanager.properties.

Java-based Node Manager provides a more fine-grained security model, and the administrator credentials for accessing Node Manager are separate from those of the domain administrator. See Configuring Java-based Node Manager Security.

### Script-Based Node Manager

For UNIX and Linux systems, Oracle WebLogic Server provides a script-based implementation of Node Manager. This script is based on UNIX shell scripts.

For information on configuring the script implementation of Node Manager, see Configuring Script-Based Node Manager.

Script-based Node Manager is not recommended for production environments. However, depending on the security requirements for the environment in which you are using Node Manager, the script-based implementation may be acceptable. The advantage of the scriptbased Node Manager is that it can remotely manage server instances over a network that has been configured to use SSH. No additional server installation is required. The scripts merely have to be copied to the remote machine.



#### Note

Oracle recommends that you run script-based Node Manager as an operating system service, which allows it to restart automatically when the system is rebooted.



### Determining Which Node Manager Implementation to Use

The implementation of Node Manager you should use depends on the requirements of your Oracle WebLogic Server environment. The following considerations can help you decide which implementation is ideal for your environment:

- If you are installing Oracle WebLogic Server on a Windows system, you must use the Java implementation of Node Manager. The scripted implementation of Node Manager is not supported on Windows.
- Java-based Node Manager can be configured at the time you create the domain. Script-based Node Manager is configured after the domain has been created.
- To use consensus leasing, you may see faster performance when using the Java implementation of Node Manager.
- The script-based Node Manager requires a much simpler security configuration than the Java implementation. RSH and SSH are generally easier to configure than SSL, which is the only way to secure Java-based Node Manager. The script implementation of Node Manager also requires a smaller footprint than the Java implementation.
- The Java implementation of Node Manager can be used in conjunction with inetd on supported UNIX systems. The inetd daemon allows Node Manager to be automatically restarted upon receiving a request on the configured port. You can also install the Java implementation of Node Manager as a Windows service.

## **Accessing Node Manager**

Use the WebLogic Remote Console or WLST to access the Java-based and the script-based implementations of Node Manager.

In addition, you can access script-based Node Manager from a shell command template provided to you. You can also use JMX to communicate with the Administration Server.

- WebLogic Remote Console: In the Edit Tree, go to Environment, then Machines. Select myMachine and then select the Node Manager tab.
- WLST commands and scripts: WLST offline serves as a Node Manager command-line
  interface that can run in the absence of a running Administration Server. You can use
  WLST commands to start, stop, and monitor a server instance without connecting to an
  Administration Server. For more information about using WLST and Node Manager to
  control server instances, see <u>Using Node Manager to Control Servers</u>.

## How Node Manager Works in the Oracle WebLogic Server Environment

Node Manager uses the startup properties and remote startup arguments to start an Administration Server and Managed Server, respectively. The nmConnect command provides a Node Manager user name and password that are used to authenticate the user with Node Manager. The nmStart command identifies the server instance and creates the Administration Server process. In the WebLogic Remote Console, you can specify the startup arguments that Node Manager uses to start a Managed Server.

The following sections provide a big picture diagram of Node Manager's role in the Oracle WebLogic Server environment, as well as illustrations and descriptions of the processes Node Manager uses to communicate with server instances:



### Diagram of Node Manager and Servers

<u>Figure 1-1</u> illustrates the relationship between Node Manager, its clients, and the server instances it controls.

In this diagram, Machine A hosts the Administration Server, and Machine B and Machine C host Managed Servers. Each machine contains a Node Manager instance. Using Node Manager, you can start, monitor, and restart the Administration Server in Machine A. By using the Administration Server as a Node Manager client, you can start or stop Managed Servers in Machine B and Machine C.

You can use the WebLogic Remote Console, WLST, or a JMX client to access Node Manager. You can use WLST commands to start, stop, and monitor a server instance when WLST is connected directly to the Node Manager instance or when WLST is connected to the Administration Server. When using the WebLogic Remote Console, you access Node Manager using the Administration Server. You can also use a JMX client to communicate with the Administration Server.



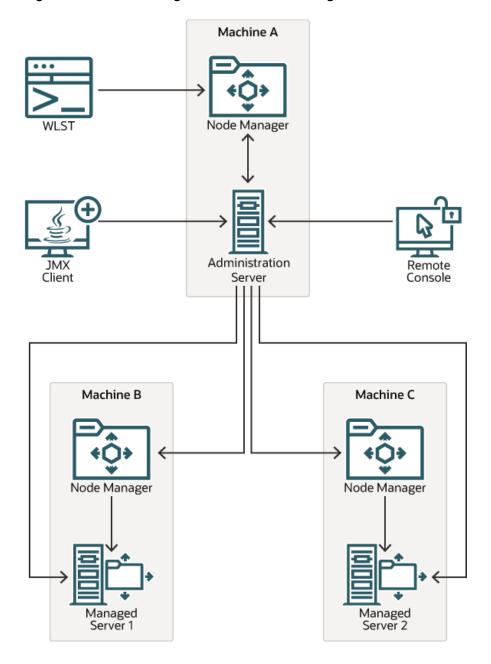


Figure 1-1 Node Manager in the Oracle WebLogic Server Environment

## How Node Manager Starts an Administration Server

Figure 1-2 illustrates the process of starting an Administration Server with Node Manager.

This section assumes that you have installed the Administration Server and created its domain directory using the Configuration Wizard.

Node Manager is running on Machine A, which hosts the Administration Server. The standalone Node Manager client is remote.



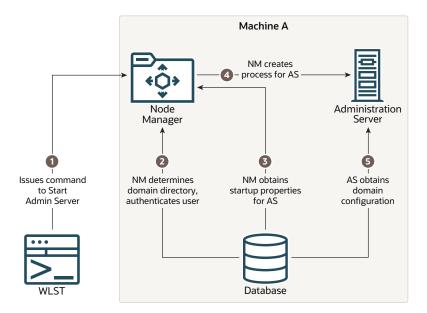


Figure 1-2 Starting an Administration Server

1. An authorized user issues the WLST offline command nmConnect to connect to a Node Manager (NM) process on the machine that hosts the Administration Server (AS). The nmConnect command provides a Node Manager user name and password that are used to authenticate the user with Node Manager.

#### (i) Note

If a Node Manager instance is the script-based implementation, the user can connect using the SSH client.

Then, the user issues the nmStart command and provides the credentials for starting the Administration Server. For example:

prps = makePropertiesObject(Username=username;Password=password") nmStart("AdminServer",props=prps)



#### (i) Note

A boot.properties file is generated if the user has already started the Administration Server and provided credentials.

The nmStart command identifies the server instance to start.

- 2. Node Manager looks up the domain directory in nodemanager.domains, and authenticates the user credentials using a local file that contains the encrypted user name and password.
- 3. Node Manager obtains the startup properties for the Administration Server. The nmStart command can optionally pass properties that are used to start the Managed Server. When these properties are passed, they overwrite any previously stored properties that Node Manager may have used. If no properties are passed with the nmStart command, then Node Manager uses the values in the startup.properties file that have



been persisted from a previous startup or from using the nmGenBootStartupProps WLST command.

- 4. Node Manager creates the Administration Server process.
- 5. The Administration Server obtains the domain configuration from its config directory.

#### Note

After the Administration Server starts, you can update the user credentials and startup properties by using the WLST nmGenBootStartupProps online command.

Alternatively, when the Administration Server and Node Manager are running, you can update the user credentials and startup properties in the WebLogic Remote Console. To update, in the **Edit Tree**, go to **Environment**, then **AdminServer**. On the **Advanced** tab, select the **Node Manager** subtab. The Administration Server then pushes the updates to the running Node Manager and Node Manager writes the information to the disk.

### How Node Manager Starts a Managed Server

<u>Figure 1-3</u> illustrates the process of starting a Managed Server with Node Manager by using the WebLogic Remote Console. You can also use WLST or a JMX client to connect to the Administration Server.

Node Manager is running on Machine B, which hosts Managed Server 1. The Administration Server for the domain is running on Machine A.



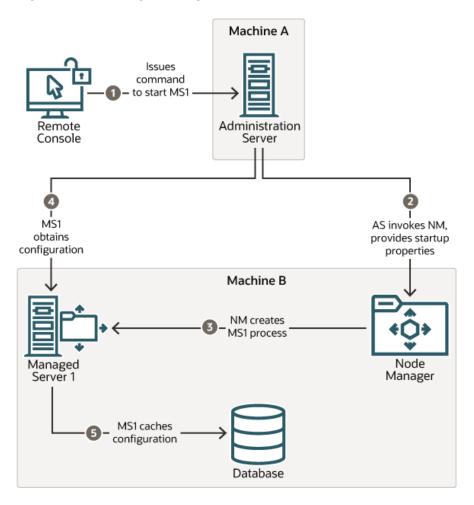


Figure 1-3 Starting a Managed Server

- From the WebLogic Remote Console, the user issues a start command for Managed Server 1 (MS1).
- The Administration Server (AS) issues a start command for Managed Server 1 to Node Manager (NM) on the Machine B, providing the remote start properties configured for Managed Server 1. See <u>Configuring Remote Startup Arguments</u>.
- 3. Node Manager starts the Managed Server 1 in the domain directory.
- Managed Server 1 contacts the Administration Server to check for updates to its configuration information.
- 5. If there are outstanding changes to the domain configuration, Managed Server 1 updates its local cache of configuration data.

## How Node Manager Restarts an Administration Server

Figure 1-4 illustrates the process of restarting an Administration Server with Node Manager.

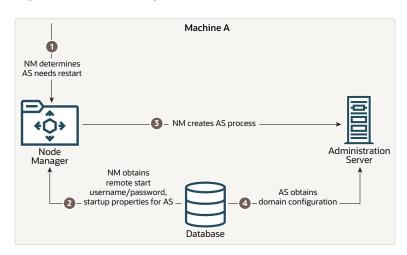
Node Manager is running on the machine that hosts the Administration Server. The Administration Server, which was initially started with Node Manager, has exited. The Administration Server's AutoRestart attribute is set to true.



#### (i) Note

If a server instance's AutoRestart attribute is set to false, Node Manager will not restart it. However, the CrashRecoveryEnabled property takes precedence over the AutoRestart property in a crash recovery scenario. For example, if a server instance has AutoRestart=false but CrashRecoveryEnabled=true, when Node Manager restarts, it tries to recover the server instance if the server instance failed when Node Manager was not running.

Figure 1-4 Restarting an Administration Server



- 1. Node Manager (NM) determines from the Administration Server (AS) process exit code that it requires a restart.
- Node Manager obtains the user name and password for starting the Administration Server from the boot.properties file, and the server startup properties from the server\_name/ data/nodemanager/startup.properties file.
- Node Manager starts the Administration Server.
- 4. The Administration Server reads its configuration data and starts up.

### How Node Manager Restarts a Managed Server

Figure 1-5 illustrates the process of restarting a Managed Server with Node Manager.

Node Manager is running on Machine B, which hosts Managed Server 1. Managed Server 1, which was initially started with Node Manager, has exited. Managed Server 1's AutoRestart attribute is set to true.

#### Note

If a server instance's AutoRestart attribute is set to false, Node Manager will not restart it. However, the CrashRecoveryEnabled property takes precedence over the AutoRestart property in a crash recovery scenario. For example, if a server instance has AutoRestart=false but CrashRecoveryEnabled=true, when Node Manager restarts, Node Manager tries to recover the server instance if the server instance failed when Node Manager was not running.



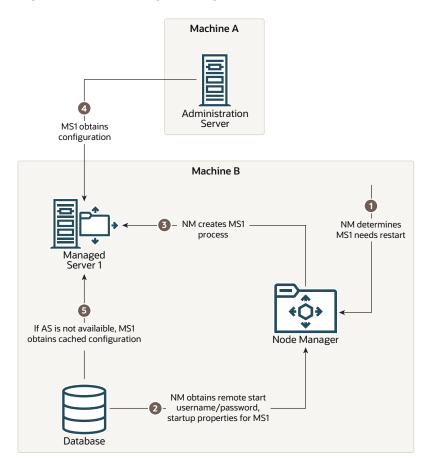


Figure 1-5 Restarting a Managed Server

- Node Manager (NM) determines from the Managed Server 1 (MS1) process exit code that it requires restart.
- 2. Node Manager obtains the user name and password for starting Managed Server 1 from the boot.properties file, and the server startup properties from the startup.properties file. These server-specific files are located in the server\_name/data/nodemanager/ directory for Managed Server 1.
- 3. Node Manager starts Managed Server 1.

#### Note

Node Manager waits RestartDelaySeconds after a server instance fails, before attempting to restart it.

- 4. Managed Server 1 attempts to contact the Administration Server (AS) to check for updates to its configuration data. If it contacts the Administration Server and obtains the updated configuration data, Managed Server 1 updates its local cache of the config directory.
- 5. If Managed Server 1 fails to contact the Administration Server, and if the Managed Server Independence mode (MSI) is enabled, Managed Server 1 uses its locally cached configuration data.



#### ① Note

The Managed Server Independence (MSI) mode is enabled by default. The MSI mode is also enabled when:

- an admin\_url is not provided, where admin\_url specifies the listen address (host name, IP address, or DNS name) and port number of the domain's Administration Server.
- an admin\_url is provided, but the Administration Server cannot be reached for any reason.

The MSI mode can be temporary as the Managed Server intermittently attempts to reconnect to the Administration Server.

### How Node Manager Shuts Down a Server Instance

<u>Figure 1-6</u> illustrates the communications involved in shutting down a Managed Server that is under Node Manager control. Depending on the state and availability of the Managed Server, Node Manager might need to forcefully shut down the Managed Server. Node Manager cannot gracefully shut down a Managed Server.

Node Manager is running on Machine B, which hosts Managed Server 1.

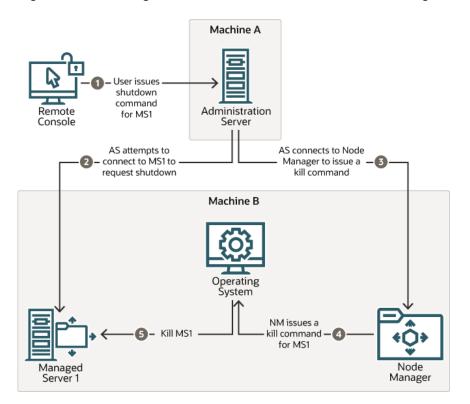


Figure 1-6 Shutting Down a Server Instance Under Node Manager Control

1. Through the WebLogic Remote Console, an authorized user issues a shutdown command for Managed Server 1 (MS1).



- 2. The Administration Server (AS) attempts to connect to Managed Server 1 and issues the shutdown command directly to Managed Server 1. If the Administration Server successfully contacts Managed Server 1, Managed Server 1 performs the shutdown sequence described in Shutting Down Instances of Oracle WebLogic Server in Administering Server Startup and Shutdown for Oracle WebLogic Server. For the Managed Server to gracefully shut down itself, it must be connected to the Administration Server.
- If, in the previous step, the Administration Server fails to contact Managed Server 1, the Administration Server connects to Node Manager (NM) to issue a kill command for Managed Server 1.
- Node Manager issues a request to the operating system to kill Managed Server 1.
- 5. The operating system ends the Managed Server 1 process.

## Node Manager and System Crash Recovery

To ensure that Node Manager properly restarts server instances after a system crash, set the crash recovery properties and perform the other required tasks for Java-based or script-based Node Manager.

To ensure that Node Manager properly restarts server instances, you must perform the following:

• For Java-based Node Manager, ensure that CrashRecoveryEnabled is set to true.

The CrashRecoveryEnabled configuration property allows Node Manager to restart server instances after a system crash. The property is not enabled by default.

#### (i) Note

If a server instance's AutoRestart attribute is set to false, Node Manager will not restart it. However, the CrashRecoveryEnabled configuration property takes precedence over the AutoRestart server startup property in a crash recovery scenario. For example, if a server instance has AutoRestart=false but CrashRecoveryEnabled=true, when Node Manager restarts, Node Manager tries to recover the server instance if the server instance failed when Node Manager was not running.

• For script-based Node Manager, place this line in machine start scripts or, if desired, run periodically on a given schedule:

wlscontrol.sh -d domain\_name CRASHRECOVERY

- You should start the Administration Server using Node Manager.
- All Managed Servers should be started using the Administration Server. You can accomplish this using WLST or the WebLogic Remote Console.

After the system is restarted, Node Manager checks each managed domain specified in the nodemanager.domains file to determine if there are any server instances that were not cleanly shutdown. This is determined by the presence of any lock files which are created by Node Manager when an Oracle WebLogic Server process is created. This lock file contains the process identifier for Oracle WebLogic Server startup script. If the lock file exists, but the process ID is not running, Node Manager will attempt to automatically restart the server instance.



If the process is running, Node Manager performs an additional check to access the management servlet running in the process to verify that the process corresponding to the process ID is an Oracle WebLogic Server instance.

#### (i) Note

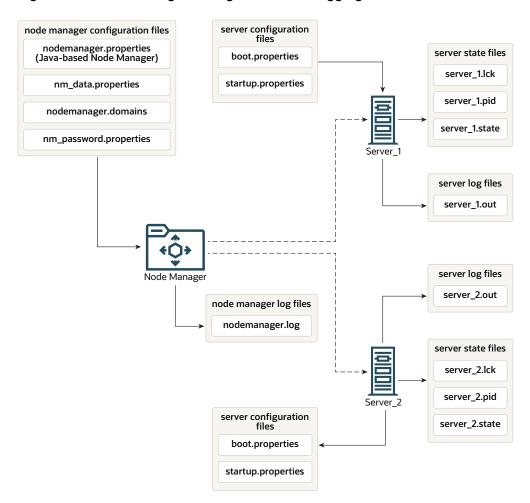
When Node Manager performs a check to access the management servlet, an alert may appear in the server log regarding improper credentials.

## Node Manager Configuration and Log Files

In managing multiple server instances, Node Manager uses multiple configuration files and outputs log files to multiple directories.

These files are shown in Figure 1-7.

Figure 1-7 Node Manager Configuration and Logging Environment



The following sections describe Node Manager configuration and log files:



### **Configuration Files**

Except where noted, configuration files apply to both Java-based and script-based Node Manager.

Node Manager includes the following configuration files:

### nodemanager.properties

This is the configuration file used by the Java-based implementation of Node Manager. See Reviewing nodemanager.properties.

By default, this file is located in NodeManagerHome, typically, ORACLE\_HOME\user\_projects\domains\domain\_name\nodemanager, where ORACLE\_HOME is the location you specified as the Oracle Home when you installed Oracle WebLogic Server.

### nodemanager.domains

This file contains mappings between the names of domains managed by Node Manager and their corresponding directories. See Configuring nodemanager.domains File.

For the Java-based Node Manager, this file is located in NodeManagerHome, typically, ORACLE\_HOME\user\_projects\domains\domain\_name\nodemanager.

For the script-based Node Manager, this file's default NodeManagerHome location is WL HOME/ common/nodemanager, where WL\_HOME is the location in which you installed Oracle WebLogic Server, for example, ORACLE HOME/wlserver.

### nm password.properties

This file stores the Node Manager user name and password. See Specifying Node Manager User Name and Password.

This file is located in DOMAIN HOME/config/nodemanager, where DOMAIN HOME is the location of the WebLogic domain, typically, ORACLE\_HOME\user\_projects\domains\domain\_name.

### boot.properties

Node Manager uses this file to specify user credentials when starting a server instance.

This file is located in DOMAIN\_HOME/servers/server\_name/data/nodemanager.

### startup.properties

Each Managed Server instance has its own startup properties file with properties that control how Node Manager starts up and controls the server instance. Node Manager automatically creates this file by using the properties passed to Node Manager when the Administration Server was last used to start the server instance. This allows a Node Manager client or startup scripts to restart a Managed Server by using the same properties that was last used by the Administration Server.



#### Note

For script-based Node Manager, startup properties file is not automatically created.



See Setting Server Startup Properties. These properties correspond to the server startup attributes contained in ServerStartMBean and the health monitoring attributes in ServerStartMBean.

This file is located in DOMAIN HOME/servers/server name/data/nodemanager.

### *server name*.addr

server\_name.addr stores the IP address added when a server instance starts or is migrated. This file is generated after the server IP address is successfully brought online during migration. server name.addr is deleted when the IP address is brought offline. The server IP address is used to validate remove requests to prevent addresses being erroneously removed while shutting down the server instance.

This file is located in DOMAIN HOME/servers/server name/data/nodemanager.

### server name.lck

server\_name.1ck is generated by each server instance and contains an internally used lock ID.

This file is located in DOMAIN\_HOME/servers/server\_name/data/nodemanager.

### server\_name.pid

server\_name.pid is generated by each server instance and contains the process ID of the server instance. Node Manager checks the process ID generated by the server instance during crash recovery.

This file is located in DOMAIN\_HOME/servers/server\_name/data/nodemanager.

### server name.state

server\_name.state is generated by the server instance and contains the server instance's current state. Node Manager monitors the contents of this file to determine the current state of the server instance.



### (i) Note

Do not delete or alter this file. Without this file Node Manager cannot determine the current state of the server instance.

This file is located in DOMAIN\_HOME/servers/server\_name/data/nodemanager.

### Log Files

Use Node Manager and Oracle WebLogic Server log files to help troubleshoot problems in starting or stopping individual Managed Servers.



Table 1-1 Node Manager Log File Locations

Log File	Location
Node Manager Log File	For Java-based Node Manager only, NodeManagerHome/ nodemanager.log, where NodeManagerHome is ORACLE_HOME\user_projects\domains\domain_name\nodemanager.
Node Manager Server Instance Log Files	DOMAIN_HOME/servers/server_name/logs/server_name.out, where DOMAIN_HOME is the location in which you installed your WebLogic domain, such as ORACLE_HOME\user_projects\domains\domain_name.
Oracle WebLogic Server Log Files	DOMAIN_HOME/servers/server_name/logs/server_name.log

This section includes the following topics:

### nodemanager.log

nodemanager.log is created for the Java-based Node Manager only; it is not created for the script-based Node Manager. This log file is generated by Node Manager and contains data for a given WebLogic domain that is controlled by Node Manager. The file typically is located in <code>ORACLE\_HOME\user\_projects\domains\domain\_name\nodemanager</code>.

Log output is appended to the current nodemanager.log file. Log rotation is disabled by default, but can be enabled by setting LogCount in nodemanager.properties.

You can view a Node Manager log file by:

- In the Monitoring Tree, navigate to Environment, Node Manager Logs. In the Node Manager Logs table, click Download.
- Using the WLST nmLog command

### Log File Rotation

In versions of Oracle WebLogic Server earlier than 12.1.3.0, when NativeVersionEnabled is set to false, you can configure the Node Manager log file rotation properties that are listed in Table 1-2. As of Oracle WebLogic Server 12.1.3.0, Node Manager uses the LogFileMBean properties for log file rotation. However, NativeVersionEnabled must still be set to false.

#### (i) Note

It is possible to rotate the log files when NativeVersionEnabled is set to true. To do this, add the following property to the startup arguments for your Managed Server:

-Dweblogic.nmservice.RotationEnabled=true

Node Manager checks the log file size every five minutes. The file starts rotating only after Node Manager verifies that the log file has reached or exceeded the maximum threshold value.

See Rotate Log Files in the Oracle WebLogic Remote Console Online Help.



For Coherence and other system components that have implemented a plug-in, Node Manager uses the log file rotation properties listed in <u>Table 1-3</u>.

Table 1-2 Node Manager Log File Rotation Properties

Property	Description	Default
FileSizeKB	This property is deprecated in Oracle WebLogic Server 12.1.3.0 and may be removed in a future release.	500
	Use the server <u>LogFileMBean</u> to adjust log file rotation values or process.FileSizeKB in <u>Table 1-3</u> for Coherence and system components.	
FileTimeSpan	This property is deprecated in Oracle WebLogic Server 12.1.3.0 and may be removed in a future release.	24
	Use the server <u>LogFileMBean</u> to adjust log file rotation values or process.FileTimeSpan in <u>Table 1-3</u> for Coherence and system components.	
FileTimeSpanFactor	This property is deprecated in Oracle WebLogic Server 12.1.3.0 and may be removed in a future release.	360000
	Use the server <u>LogFileMBean</u> to adjust log file rotation values or process.FileTimeSpanFactor in <u>Table 1-3</u> for Coherence and system components.	
NumberOfFilesLimited	This property is deprecated in Oracle WebLogic Server 12.1.3.0 and may be removed in a future release.	true
	Use the server <u>LogFileMBean</u> to adjust log file rotation values or process.NumberOfFilesLimited in <u>Table 1-3</u> for Coherence and system components.	
RotatedFileCount	This property is deprecated in Oracle WebLogic Server 12.1.3.0 and may be removed in a future release.	7
	Use the server <u>LogFileMBean</u> to adjust log file rotation values or process.RotatedFileCount in <u>Table 1-3</u> for Coherence and system components.	
RotationTimeStart	This property is deprecated in Oracle WebLogic Server 12.1.3.0 and may be removed in a future release.	00:00
	Use the server <u>LogFileMBean</u> to adjust log file rotation values or process.RotationTimeStart in <u>Table 1-3</u> for Coherence and system components.	
RotationType	This property is deprecated in Oracle WebLogic Server 12.1.3.0 and may be removed in a future release.	SIZE
	Use the server <u>LogFileMBean</u> to adjust log file rotation values or process.RotationType in <u>Table 1-3</u> for Coherence and system components.	



Table 1-3 Node Manager Log File Rotation Properties for Coherence and System Components

Property	Description	Default
process.FileSizeKB	Specifies the maximum size of the file before it is rotated.	500
process.FileTimeSpan	Indicates the interval (in hours) at which the server instance saves old log messages to another file. Requires that you specify a process.RotationType of TIME.	24
process.FileTimeSpanFactor	Allows log rotation to be tested at a different frequency. <b>Note:</b> This property rarely needs to be	360000
process.NumberOfFilesLimited	changed or configured.  Indicates whether to limit the number of log files that this server instance creates to store old messages. Requires that you specify a process.RotationType of SIZE or TIME. After the server instance reaches this limit, it deletes the oldest log file and creates a new log file with the latest suffix. If you do not enable this option, the server instance creates new files indefinitely and you must clean up these files as you require.	true
process.RotatedFileCount	Specifies the maximum number of rotated files to keep when process.NumberOfFilesLimited=true.	7
process.RotationTimeStart	Determines the start time (hour and minute) for a time-based rotation sequence. At the time that this value specifies, the server instance renames the current log file. Thereafter, the server instance renames the log file at an interval that you specify in process.FileTimeSpan. Note that Oracle WebLogic Server sets a threshold size limit of 500 MB before it forces a hard rotation to prevent excessive log file growth. Use the following format: H:mm, where H is the hour in a day (0-23) and mm is the minute in an hour.	00:00



Table 1-3 (Cont.) Node Manager Log File Rotation Properties for Coherence and **System Components** 

Property	Description	Default
process.RotationType	Specifies the criteria for moving old log messages to a separate file.	SIZE
	<ul> <li>NONE: Messages accumulate in a single file. You must erase the contents of the file when the size is too large. Note that Oracle WebLogic Server sets a threshold size limit of 500 MB before it forces a hard rotation to prevent excessive log file growth.</li> <li>SIZE: When the log file reaches the size that you specify in FileMinSize, the server instance renames the file as SERVER_NAME.lognnnnn.</li> <li>TIME: At each time interval that you specify in FileTimeSpan, the server instance renames the file as SERVER_NAME.lognnnnn.</li> </ul>	

### server name.out

For each server instance that it controls, Node Manager maintains a log file that contains stdout and stderr messages generated by the server instance. If the remote start debug property is enabled as a remote start property for the server instance, or if the Node Manager debug property is enabled. Node Manager will include additional debug information in the server output log information. By default, this file is located in DOMAIN\_HOME/servers/ server name/logs, where server name is the name of the server instance.

However, if you add -Dweblogic. Stdout = in the Arguments field of the ServerStartMBean, then this value overrides the default location, and Node Manager uses the supplied path as the file for the output. You cannot override the server\_name.out default location using the weblogic.startup.Arguments.prepend property in the nodemanager.properties file, as this property applies to multiple server instances.

Node Manager creates the server output log for a server instance in the server instance's logs directory, with the name:

server\_name.out



#### (i) Note

The rotation of the server output file is configured by the RotateLogOnStartup attribute on the LogFileMBean.

On Windows, the server output file is rotated during shutdown. In addition, if a server is restarted using Node Manager, the server output file will be rotated at server shutdown even if the RotateLogOnStartup attribute is set to false. This is specific to Windows only.

You can view a Node Manager log file for a particular server instance by:



- Selecting Diagnostics, and then Log Files.
- Using the WLST nmServerLog command.

There is no limit to the number of server output logs that Node Manager can create.

### Configuring Log File Rotation

To configure the Log File Rotation properties using Remote Console, perform the following steps:

### Note

It is possible to configure Log File Rotation properties using WLST, REST, or JMX. For more information, see Rotating Log Files in *Configuring Log Files and Filtering Log Messages for Oracle WebLogic Server*.

- 1. Log in to the Remote Console using the administrator user name and password.
- In the Edit Tree, go to Environment, then Servers.
- 3. Select the server name.
- 4. On the Logging tab, from the Rotation Type drop-down list, select the criteria that triggers the server to move log messages to a separate file.
- 5. Depending on your choice, configure the appropriate settings.
- Click Save.

The properties are saved for the server.

## Configuring Java Node Manager

Learn how to configure the Java implementation of Oracle WebLogic Servers Node Manager. This chapter includes the following topics:

### Overview

The Java implementation of Node Manager is configured by default to control all server instances belonging to the same domain, a *per domain* Node Manager.

The server instances need not reside on the same machine. If a machine has multiple domains, using per domain Node Manager results in multiple Node Manager process instances. See <u>Default Node Manager Configuration</u>.

In previous versions of Oracle WebLogic Server, Node Manager was not associated with a specific WebLogic domain but only with a host machine. You used the same Node Manager process to control server instances in any WebLogic domain, as long as the server instances resided on the same machine, a machine-scoped, a *per host* Node Manager. While you can still use per host Node Manager, additional configuration is required. Typically, per host Node Manager is used to manage multiple domains with a single Node Manager instance. See Configuring Per Host Node Manager.

If you are using per domain Node Manager, Oracle WebLogic Server provides the option to have the Node Manager configuration in a custom location. Using the Configuration Wizard or WLST offline, you can select a PerDomain or CustomLocation Java-based Node Manager configuration. For any type of Node Manager configuration, you can provide unique Node Manager credentials, but NodeManagerHome is the default location, as described in Default Node Manager Configuration. If you want a unique location for NodeManagerHome, select CustomLocation and specify an empty directory or select to create one.

## **Default Node Manager Configuration**

For each Oracle WebLogic Server domain you create, a domain-specific Node Manager instance is created by default. If you choose to use the default per domain Node Manager configuration, no additional steps are necessary to use Node Manager to start and stop server instances in your Oracle WebLogic Server domain.

Using the security credentials provided for the Administration Server, nm\_password.properties is created in <code>DOMAIN\_HOME</code>\config\nodemanager, where <code>DOMAIN\_HOME</code> is typically located at <code>ORACLE\_HOME\user\_projects\domains\domain\_name</code>. The nodemanager.properties and nodemanager.domains files are created for you under <code>DOMAIN\_HOME\nodemanager</code>. With the default Node Manager configuration, you cannot edit the <code>NodeManagerHome</code> location, <code>DOMAIN\_HOME\nodemanager</code>.

Domain-specific scripts to start, stop, install, and uninstall Node Manager as a Windows service, are located under <code>DOMAIN\_HOME\bin</code>. To install Node Manager as a Windows service, you may need to edit the <code>installNodeMgrSvc.cmd</code> script to specify appropriate listen address and listen port values:

1. Navigate to your DOMAIN\_HOME\bin directory.



- 2. Edit installNodeMgrSvc.cmd to specify Node Manager's listen address and listen port.
  - Make the same edits to uninstallNodeMgrSvc.cmd as you make to installNodeMgrSvc.cmd so that you can successfully uninstall the service in the future, as desired.
- 3. Run installNodeMgrSvc.cmd to reinstall Node Manager as a service, listening on the updated address and port.

#### (i) Note

When configuring multiple per domain Node Manager instances on the same machine, you must use a unique Node Manager address (hostname:port) for each domain, either by using unique ports or unique host names. For example, if you have three per domain Node Manager instances running on the machine, use address localhost:5556 for Domain 1, address localhost:5557 for Domain 2, and localhost:5558 for Domain 3.

## Configuring Per Host Node Manager

If you want to use per host Node Manager, for which scripts are located in <code>WL\_HOME\server\bin</code>, you must first perform certain prerequisite configuration steps.

#### (i) Note

For per host Node Manager configurations, do not set weblogic.RootDirectory in JAVA\_OPTIONS to the domain home. If weblogic.RootDirectory points to an existing domain, then the default location for the security subsystem is the domain instead of the Node Manager specific location. Node Manager may then use the domain-specific security settings by default, which could cause the SSL handshake to fail if a second domain attempts to communicate with Node Manager.

Additionally, if you want to use SSL for a per host Node Manager configuration, you must build your own certificate files.

- 1. Create a nodemanager.domains file that specifies the domains that you want this Node Manager instance to control, under <code>ORACLE\_HOME\oracle\_common\common\nodemanager</code>, the per host <code>NodeManagerHome</code> location.
  - You can manually create or copy this file. See Configuring nodemanager.domains File.
  - Alternatively, you can register WebLogic domains with Node Manager using the WLST command, nmEnroll.

By specifying multiple domains in the nodemanager.domains file, you can configure a single, machine-scoped Node Manager process which manages server instances belonging to multiple WebLogic domains, similar to Node Manager functionality from prior Oracle WebLogic Server releases.

- 2. Configure a machine definition for each machine that runs a Node Manager process. See Configuring a Machine to Use Node Manager.
- 3. If you want to use the demonstration Identity and Trust keystores for development or testing purposes *only*, you can create them using the DemoCertGen utility.





#### (i) Note

If your domain was upgraded from WebLogic Server 14.1.1.0.0 or earlier, you cannot use DemoCertGen to configure demo keystores. You must follow the process outlined in step 4 instead.

To properly set up the PATH and CLASSPATH variables, from a command prompt, run WL\_HOME\server\bin\setWLSEnv.cmd.



#### (i) Note

On UNIX operating systems, the setWLSEnv.sh command does not set the environment variables in all command shells. Oracle recommends that you execute this command using the Korn shell or bash shell.

**b.** Run the DemoCertGen utility to configure the demonstration keystores.

```
java utils.DemoCertGen -domain < DOMAIN HOME >
```

The demo identity and trust keystores are placed in the <code>DOMAIN\_HOME\security</code> directory.

- Copy the DemoIdentity.p12 keystore to the NodeManagerHome\security directory.
- For domains upgraded from WebLogic Server 14.1.1.0.0 or earlier only: If you want to use the demonstration Identity and Trust keystores, for development or testing purposes only, you can create them using the CertGen and ImportPrivateKey Java utilities as shown in the following examples:
  - To properly set up the PATH and CLASSPATH variables, from a command prompt run WL HOME\server\bin\setWLSEnv.cmd.



#### (i) Note

On UNIX operating systems, the setWLSEnv.sh command does not set the environment variables in all command shells. Oracle recommends that you execute this command using the Korn shell or bash shell.

**b.** Generate a certificate and private key.

java utils.CertGen -keyfilepass DemoIdentityPassPhrase -certfile democert keyfile demokey

By default utils. CertGen will use the short host name as the owner CN value in the generated certificate. To use the fully-qualified DN host name, add the -cn option to the above command. For example:

java utils.CertGen -keyfilepass DemoIdentityPassPhrase -certfile democert keyfile demokey -cn abc.oracle.com

c. Import the private key and certificate.

java utils.ImportPrivateKey -keystore DemoIdentity.jks -storepass DemoIdentityKeyStorePassPhrase -keyfile demokey -keyfilepass DemoIdentityPassPhrase -certfile democert.pem -keyfile demokey.pem -alias demoidentity -storetype jks



The DemoIdentity.jks keystore now contains one private key and certificate entry. The other files can be deleted.

d. Copy the DemoIdentity.jks keystore to the NodeManagerHome\security directory.

For information about configuring SSL for Node Manager in production environments, see <u>Using SSL With Java-based Node Manager</u>.

#### Note

By default, using SSL with Node Manager is enabled. If not needed, you can disable it by changing to SecureListener=false in the nodemanager.properties file. To review the SSL-related properties in nodemanager.properties, see Table 2-1.

## Configuring Node Manager on Multiple Machines

If you have a domain that has Managed Servers on multiple physical machines, you can use any type of Node Manager (per domain, per host, or custom) to configure Node Manager on multiple machines. However, you should use the same Node Manager type for all machines.

1. Ensure Node Manager is installed and configured on each machine.

Create the NodeManagerHome location and nodemanager.properties file. By default, NodeManagerHome is DOMAIN\_HOME\nodemanager. In a production environment, you might want to customize the location of the Node Manager root directory. The nodemanager.properties file is created in the directory specified in NodeManagerHome and specifies Node Manager properties. See Reviewing nodemanager.properties.

### ① Note

The  ${\it NodeManagerHome}$  location should not be shared by multiple Node Manager instances.

Therefore, in a scenario when you are using a shared drive, the <code>NodeManagerHome</code> location on each machine should be unique. For example, create a host name subdirectory under <code>NodeManagerHome</code>, and copy all files to the host name subdirectory.

- 2. To enroll a domain with Node Manager, Oracle recommends using the pack and unpack commands to copy all of the required domain and configuration information from one machine to another. See Node Manager Configuration in *Creating Templates and Domains Using the Pack and Unpack Commands*. You can also use the WLST command nmEnroll to perform this action.
- 3. Create the corresponding MachineMBean in the Administration Server that points to each Node Manager instance, so that you can start Managed Servers using Node Manager from any machine.

When the Administration Server is involved in starting a Managed Server, it uses the information in the targeted MachineMBean to create a connection with the correct Node Manager instance. This occurs when using either the WebLogic Remote Console or WLST after running the nmConnect command. Creating a corresponding MachineMBean for each physical machine helps point to each Node Manager instance and associates your server instances with the appropriate MachineMBean.



To create a MachineMBean in the Administration Server, perform one of the following actions:

- In the WebLogic Remote Console, in the **Edit Tree**, go to **Environment**, then **Machines**. Click **New** to create a new machine. Configure the information in the MachineMBean and its NodeManagerMBean to point to the Node Manager listen address and listen port. For more information, see Configure Machines in the *Oracle WebLogic Remote Console Online Help*.
- Using WLST, create the MachineMBean with the create command. Then, change the settings of the MachineMBean to point to the Node Manager listen address and listen port.
- 4. Optionally, repeat any configuration steps you performed if you created Node Manager locally.

See <u>Accessing Node Manager</u> and nmEnroll in WLST Command Reference for Oracle WebLogic Server.

## Controlling and Configuring Node Manager Using WLST

The WebLogic Scripting Tool (WLST) is a command-line scripting interface that system administrators and operators use to monitor and manage Oracle WebLogic Server instances and domains. You can start, stop, and restart server instances remotely or locally, using WLST as a Node Manager client. In addition, WLST can obtain server status and retrieve the contents of the server output log and Node Manager log.

See WLST Command and Variable Reference in WLST Command Reference for Oracle WebLogic Server.

## Configuring Node Manager Using WLST Offline

You can use WLST offline to configure Node Manager by performing the tasks such as setting the Node Manager user name and password, Node Manager properties, Node Manager type, and keystore properties.

<u>Example 2-1</u> shows how to set a domain's Node Manager listen address and listen port, the Node Manager user name and password, and the Node Manager type.



If the Node Manager type is Manual NodeManager Setup, you cannot use WLST offline to edit Node Manager properties.

### **Example 2-1 Configuring Node Manager**

```
# Set the Node Manager listen address and listen port.
cd('/')
cd('NMProperties')
set('ListenAddress','localhost')
set('ListenPort',9001)

# Set the Node Manager user name and password.
cd('/')
cd('SecurityConfiguration/domain_name')
set('NodeManagerUsername','username')
set('NodeManagerPasswordEncrypted','password')
```



```
# Set the Node Manager type to custom location type and set the custom location Node
Manager home.
setOption('NodeManagerType','CustomLocationNodeManager')
setOption('NodeManagerHome','C:/mydomains/nodemanager')
# Set the keystore properties
cd('/')
cd('NMProperties')
set('CustomIdentityKeyStoreFileName','identitykeystore.jks')
set('KeyStores','CustomIdentityAndCustomTrust')
set('CustomIdentityPrivateKeyPassPhrase','password')
set('CustomIdentityAlias','identitykeystore')
# The following property must be explicitly set for AIX systems; otherwise, it will
default to java.security.
set('CustomIdentityKeyStoreType','PKCS12')
```

See setOption in WLST Command Reference for Oracle WebLogic Server.

## Configuring Java-Based Node Manager Security

Java-based Node Manager security uses SSL by default and authenticates incoming connections against a set of credentials specific to each domain.

If you are establishing a command-line connection to the Java Node Manager using the Oracle WebLogic Server Scripting Tool (WLST) nmConnect command, you provide the Node Manager user name and password. Node Manager verifies the user name and password against the domain nm password.properties file.

Node Manager credentials are located in Advanced Options in the WebLogic Remote Console. To access this page, in the **Edit Tree**, go to **Environment**, then **Domain**, and select **Security**. Under the General tab, click Show Advanced Fields.

WebLogic Remote Console users do not need to explicitly provide credentials to connect to Node Manager—the Node Manager user name and password are available in the domain configuration and are provided automatically.

This section includes the following topics:

### Specifying Node Manager User Name and Password

The nm\_password.properties file contains the encrypted Node Manager user name and password. These are used to authenticate connection between a client (for example, the Administration Server) and Node Manager.



#### Note

This user name and password are only used to authenticate connections between Node Manager and clients. They are independent from the server administration ID and password.

This file is created for you when you use nmEnroll to copy the necessary configurations files from one machine to another when creating a domain or when using the Configuration Wizard.



The file is located in <code>DOMAIN\_HOME/config/nodemanager</code>, where <code>DOMAIN\_HOME</code> is the location of your WebLogic domain, typically, <code>ORACLE HOME\user projects\domains\domain name</code>.

The Configuration Wizard prompts for a Node Manager user name and password for the initial configuration. This value is populated in the required file locally. However, to get it distributed remotely, you must use the nmEnroll command.

After nm\_password.properties is created, you can change the values for the Node Manager password and properties using the WebLogic Remote Console. Changes are propagated to the nm\_password.properties file and are picked up by Node Manager.

You can use the following steps to alter Node Manager credentials:

- 1. Start the Administration Server.
- Using the WebLogic Remote Console, update the Node Manager credentials using the Advanced Options. To access this page, in the Edit Tree, go to Environment, then Domain, and select Security. Under the General tab, click Show Advanced Fields.
- Invoke WLST and connect to an Administration Server using the connect command. See Understanding the WebLogic Scripting Tool.
- 4. Run nmEnroll using the following syntax:

```
nmEnroll([domainDir], [nmHome])
```

#### For example,

```
nmEnroll('C:/oracle/user_projects/domains/prod_domain',
'C:/oracle/user_projects/domains/prod_domain/nodemanager')
```

Running nmEnroll ensures that the correct Node Manager user and password token are supplied to each Managed Server.

#### Note

If it does not already exist, you must manually copy the <code>SerializedSystemIni.dat</code> file from the Administration Server domain directory or an existing server in the domain, to this location, <code>\${DOMAIN\_HOME}/security/SerializedSystemIni.dat</code>, before running the <code>nmEnroll</code> command.

You must run nmEnroll on each machine that is running a Managed Server. Also run nmEnroll for each domain directory on each machine.

#### (i) Note

If you edit nm\_password.properties manually (not recommended), you must restart Node Manager in order for the changes to take effect, whereas a restart is not required if you modify the values using the WebLogic Remote Console with Node Manager running.

The nm\_password.properties file must exist in the domain directory for each physical machine that runs Node Manager. If you change the domain's Node Manager user name and password, you should run nmEnroll on each machine to synchronize the nm\_password.properties file. If you configure multiple domains on a machine, each domain can use a different Node Manager user name and password.



In a typical development environment, you may not be prompted to specify the Node Manager user name and password when you create your domain. The Node Manager user name and password default to the administrator credentials, which you can change from the WebLogic Remote Console or WLST. However, in a production environment, you must explicitly set the Node Manager user name and password.

### Remote Server Start Security for Java-Based Node Manager

A remote start user name and password is required to start a server instance with Node Manager. These credentials are provided differently for Administration Servers and Managed Servers.

- Credentials for Managed Servers When you invoke Node Manager to start a Managed Server it obtains its remote start user name and password from the Administration Server.
- Credentials for Administration Servers When you invoke Node Manager to start an Administration Server, the remote start user name and password can be provided in the following ways:
  - On the command line. See How Node Manager Starts an Administration Server.
  - From the Administration Server boot.properties file.
     The Configuration Wizard initializes the boot.properties file and the startup.properties file for an Administration Server when you create the domain.
  - Generated for you in a secure, encrypted way with the following steps:
    - \* Start the Administration Server with the following setting:
      - -Dweblogic.nodemanager.ServiceEnabled=true
    - \* Create the DOMAIN HOME\servers\AdminServer\data\nodemanager directory.
    - \* Update any startup properties or the server's credentials while both the Administration Server and Node Manager are running.

Any server instance started by Node Manager encrypts and saves the credentials with which it started in a server-specific boot.properties file, for use in automatic restarts.

### Using SSL With Java-Based Node Manager

Administration Servers and Managed Servers communicate with Java-based Node Manager using one-way SSL.

The default Oracle WebLogic Server installation includes demonstration Identity and Trust keystores that allow you to use SSL out of the box. DemoIdentity.pl2 and DemoTrust.pl2 are installed in the <code>DOMAIN\_HOME\security</code> directory. For testing and development purposes, the keystore configuration is complete.

If you want to use custom certificates, then configure the <code>CustomIdentityKeyStoreFileName</code> properties in <code>nodemanager.properties</code> to set up a certificate for Node Manager. Node Manager can have its own certificate or it can share a certificate with another aspect of the domain. However, in order to communicate with the Node Manager instance, clients must trust the Identity in the Node Manager certificate. You do not need a separate certificate for each Node Manager instance, but you can configure this option if desired. In production environments, Node Manager can use the same public certificate used for all server instances.

If you want to use the Certificate Management Service and Domain Keystores, then set the KeyStores property to DomainKeystores.



For per-host configurations, you must also:

- Set DomainKeystoresDomain to specify the domain that hosts the machine keystores for the Node Manager instance.
- Start a server on each host where Node Manager will run before starting Node Manager.
   Each host (other than the Administration Server's host) must have a corresponding
   MachineMBean configured. Users must refresh the machine trust and certificates, and then start Node Manager.

#### (i) Note

The Certificate Management Service is available as a Technical Preview in this release of WebLogic Server for testing purposes only. Do not use it in production environments. For more information, see Using the Certificate Management Service in What's New in Oracle WebLogic Server.

Configuring SSL for a production environment involves obtaining identity for Node Manager and then configuring both identity and trust for each Administration Server and Managed Server with which Node Manager will be communicating. In addition, the use of host name verification and the Administration port must be taken into consideration. To review the SSL-related properties in nodemanager.properties, see <u>Table 2-1</u>. For more information, see Configuring SSL in *Administering Security for Oracle WebLogic Server*.

If a Managed Server disables the non-SSL port, Node Manager needs to trust the certificate received from the server. To configure the trust store for Node Manager, add - Djavax.net.ssl.trustStore to the Node Manager start scripts.

## **Advanced Node Manager Configuration**

Node Manager supports several advanced settings that you can configure, such as the use of start and stop scripts, the ability to control server instances in multiple domains, the properties for a Java-based Node Manager, the advanced startup properties for Managed Server instances, and more.

This section includes the following topics:

## Defining the Administration Server Address

Ensure that a listen address is defined for each Administration Server that will connect to a Node Manager process. If the listen address for an Administration Server is not defined, when Node Manager starts a Managed Server it will direct the Managed Server to contact localhost for its configuration information.

Set the Listen Address using the General page. To access this page, in the **Edit Tree**, go to **Environment**, then **Servers**, then *myServer* in the WebLogic Remote Console. For more information, see Specify Listen Addresses in the *Oracle WebLogic Remote Console Online Help*.

## Configuring Node Manager to Use Start and Stop Scripts

You can configure Node Manager to use a script to start a Managed Server or to execute a script after server shutdown has completed. These scripts can be used to perform tasks that



need to be performed before a server instance is started or after it is shutdown. Mounting and unmounting remote disks is one example of a task that can be performed using scripts.



#### (i) Note

Node Manager uses startup scripts to perform any required configuration, then start the server instance. In contrast, stop scripts are executed after the server instance has shutdown.

This section includes the following topics:

### Script Location

Both the start and stop scripts should be placed in the following directory:

DOMAIN HOME\bin

Script execution should occur relative to this directory.

### **Using Start Scripts**

You can use a start script to specify required startup properties and perform any other work you need performed at start up. To define a start script:

- In the nodemanager.properties file, set the weblogic.StartScriptEnabled property to true. (The default is true.) If your start script is named startWebLogic.sh or startWebLogic.cmd, Node Manager uses one of those scripts as the default.
- 2. If you want to specify a custom start script, set the weblogic.StartScriptName property to the name or path of your script in the nodemanager.properties file.

Node Manager sets the JAVA\_VENDOR, JAVA\_HOME, JAVA\_OPTIONS, SECURITY\_POLICY, CLASSPATH, and ADMIN URL. It retrieves these values from the ServerMBean, ServerStartMBean, and SSLMBean when you use the WebLogic Server Administration Console to start the server instance, or WLST connected to the WebLogic Server Administration Server. When you use WLST connected directly to Node Manager, you can specify the values; otherwise, they are left empty.

Node Manager combines all of the command line startup options (-D flags) that are specified in the ServerStartMBean Arguments attribute, as well as the SSLArguments into a single environmental variable called JAVA OPTIONS. SSLArguments are retrieved from the values in the SSLMBean. The SSLMBean is inspected for ignoreHostnameVerification, HostnameVerifier, and ReverseDNSAllowed values, then those values are appended to the -D flags. All of those flags comprise the SSLArguments parameter. All of the values for SSLArguments as well as Arguments in the ServerStartMBean comprise the JAVA OPTIONS environment variable that is defined for the start script. In addition, the script will append any of its own defined values onto this environment variable.

If there are resulting overlaps in this set of values, it will appear to the Java command line like this:

java -Dflag1=value1 -Dflag1=value2 weblogic.Server

The Java invocation will resolve the duplicate values.



### Best Practices When Using Stop Script

When using stop script to control server behavior, Oracle recommends that you edit only the top line of the script that is provided. This ensures that all of the necessary environment variables are used during script execution.

### **Using Stop Scripts**

You can use a stop script to perform any tasks that are required after the server instance has failed or shut down.

#### To define a stop script:

- In the nodemanager.properties file, set the weblogic.StopScriptEnabled property to
- Set the weblogic.StopScriptName property to the name of your script in the nodemanager.properties file.

The following example shows a stop script that can be used to unmount a disk on UNIX systems:

```
#!/bin/sh
FS=/cluster/d2
if grep $FS /etc/mnttab > /dev/null 2>&1 ; then
   PIDS=`/usr/local/bin/lsof $FS | awk
   '\{if (\$2 \sim /[0-9]+/) \{ print \$2\} \}' | sort -u`
kill -9 $PIDS
sleep 1
   /usr/sbin/umount -f $FS
```

## Configuring nodemanager.domains File

The nodemanager.domains file specifies the domains that a Node Manager instance controls. Thus, standalone clients do not need to specify the domain directory explicitly.



#### (i) Note

If using per domain Node Manager, you should not modify the nodemanager.domains file. However, if using per host Node Manager, or a custom Node Manager instance, you may need to edit the nodemanager.domains file to specify your domains.

This file must contain an entry specifying the domain directory for each domain a Node Manager instance controls, in this form:

```
domain-name=domain-directory
```

When a user issues a command for a domain, Node Manager looks up the domain directory from nodemanager.domains.

This file provides additional security by restricting Node Manager client access to the domains listed in this file. The client can execute commands only for the domains listed in nodemanager.domains.



For the Java-based Node Manager, this file is typically located under

ORACLE\_HOME\user\_projects\domains\domain\_name\nodemanager. If you created your domain with the Configuration Wizard, the nodemanager.domains file was created for you. If configuring a per host Node Manager instance, you must manually create or copy a nodemanager.domains file under ORACLE HOME\oracle common\oommon\nodemanager, the per host NodeManagerHome location. See Configuring Per Host Node Manager.

If necessary, you can manually edit nodemanager.domains to add domains or register multiple domain locations under a single domain name.

To configure multiple domain registration, manually enter the alternate paths in the nodemanager.domains file, in this form:

domainName=primaryDomainPath;alternateDomainPath1;alternateDomainPath2

The primaryDomainPath is the path to the domain location where Managed Servers exist and from where they will run, as the Administration Server does not typically pass a path to Node Manager to access a domain. The domain is accessible only by name.

An alternateDomainPath is only accessible by name and path and is typically the location of the Administration Server. Clients connecting directly to Node Manager can access the alternate domain path with both a domain name value and a domain path value.



#### (i) Note

If you use the backslash character () in nodemanager.domains, you must escape it as

#### Example 2-2 nodemanager.domains File

#Domains and directories created by Configuration Wizard #Mon Jan 07 10:57:18 EST 2013 base\_domain=C\:\\Oracle\\Middleware\\Oracle\_Home\\user\_projects\\domains\\base\_domain prod\_domain=C\:\\Oracle\\Middleware\\Oracle\_Home\\user\_projects\\domains\\prod\_domain

### Reviewing nodemanager.properties

Node Manager properties define a variety of configuration settings for a Java-based Node Manager process. You can specify Node Manager properties on the command line or define them in the nodemanager.properties file. Values supplied on the command line override the values in nodemanager.properties.

nodemanager.properties is created in the directory specified in NodeManagerHome, where NodeManagerHome typically is

ORACLE\_HOME\user\_projects\domains\domain\_name\nodemanager. If NodeManagerHome is not defined, nodemanager properties is created in the current directory.

Each time you start Node Manager, it looks for nodemanager.properties in the current directory, and creates the file if it does not exist in that directory. You cannot access the file until Node Manager has started up once.

This section includes the following topics:



### **Node Manager Properties**

Node Manager properties define various configuration settings for a Java-based Node Manager process depending on the server environment. You can specify these properties on the command line or define them in the nodemanager.properties file.

In many environments, the SSL-related properties in nodemanager.properties may be the only Node Manager properties that you must explicitly define. However, nodemanager.properties also contains non-SSL properties that you might need to specify, depending on your environment and preferences. For example:

- For a non-Windows installation, it might be appropriate to specify the NativeVersionEnabled property. For detailed information, see NativeVersionEnabled in Table 2-1.
- If Node Manager runs on a multihomed system, and you want to control which address and port it uses, define ListenAddress and ListenPort.

Table 2-1 describes Node Manager properties.

**Table 2-1** Node Manager Properties

Node Manager Property	Description	Default
AuthenticationEnabled	If set to true, Node Manager authenticates the credentials against the domain.	true
certificateFile	Specifies the path to the certificate file used for SSL authentication.	none
	<b>Note:</b> This property is used only in the process of upgrading from Oracle WebLogic Server, version 7.x to version 9.x.	
CipherSuite	The name of the cipher suite to use with the SSL listener.	The default value is JDK and platform dependent.
	This property is deprecated in Oracle WebLogic Server 12.1.3.0 but remains fully supported in Oracle WebLogic Server 15.1.1.0.0. See CipherSuites for more information and current support limitations for the replacement property.	
CipherSuites	The name of the cipher suites to use with the SSL listener. You can specify multiple cipher suite values separated by a comma, for example:	The default value is JDK and platform dependent.
	<pre>CipherSuites=SUITE_A, SUITE_B, SUITE_C</pre>	
	This property introduced in Oracle WebLogic Server 12.1.3 is not currently supported by WLST offline or by the pack and unpack commands. When using those utilities, use CipherSuite instead.	



Table 2-1 (Cont.) Node Manager Properties

Node Manager Property	Description	Default
coherence.startup.JavaHome	Applies when Node Manager invokes Java directly. It is the Java home directory that Node Manager uses to start Managed Servers on this machine, if the Managed Server does not have a Java home configured in its Remote Start page. If not specified in either place, Node Manager uses the Java home defined for a Node Manager uses the Java home defined for a Node Manager process.  This property introduced in Oracle WebLogic Server 12.1.3 is not currently supported by WLST offline or by the pack and unpack commands. When using those utilities, use JavaHome instead.	none
CrashRecoveryEnabled	Enables system crash recovery.	false
	Note: The CrashRecoveryEnabled property takes precedence over the AutoRestart server startup property in a crash recovery scenario. For example, if a server instance has AutoRestart=false but CrashRecoveryEnabled=true, when Node Manager restarts, Node Manager tries to recover the server instance if the server failed when Node Manager was not running.	
CustomIdentityAlias	Specifies the alias when loading the private key into the keystore. This property is required when the Keystores property is set as CustomIdentityandCustomTrust or CustomIdentityAndJavaStandardTrust.	none
CustomIdentityKeyStoreFileNa me	Specifies the file location and name of the Identity keystore (meaning the keystore that contains the private key for a Node Manager). This property is required when the Keystores property is set as CustomIdentity and CustomTrust or CustomIdentityAndJavaStandardTrust.	none
CustomIdentityKeyStorePassPh rase	Specifies the password defined when creating the Identity keystore. This field is optional or required depending on the type of keystore. All keystores require the passphrase to write to the keystore. However, some keystores do not require the passphrase to read from the keystore. Oracle WebLogic Server reads only from the keystore. So, whether or not you define this property depends on the requirements of the keystore.	none
CustomIdentityKeyStoreType	Specifies the type of the Identity keystore.  Generally, this is JKS or PKCS12. This property is optional. If not specified, the JDK default keystore type in the java.security file is used.	The default keystore type from java.security



Table 2-1 (Cont.) Node Manager Properties

Node Manager Property	Description	Default
CustomIdentityPrivateKeyPass Phrase	Specifies the password used to retrieve the private key for Oracle WebLogic Server from the Identity keystore. This property is required when the Keystores property is set as CustomIdentityandCustomTrust or CustomIdentityAndJavaStandardTrust.	none
DomainsDirRemoteSharingEnabl ed	Specifies whether Node Manager is monitoring a shared domain directory. As such, more than one Node Manager may be monitoring the shared directory from different machines.	false
	Set to true to indicate that you have a shared domain directory (mounted directory or Windows NFS) that multiple nodes will be sharing. Enabling this property allows multiple Node Managers to share the domain without affecting each other.	
DomainsFile	The name of the nodemanager.domains file.	NodeManagerHome\nodemanager. domains
DomainsFileEnabled	If set to true, use the file specified in DomainsFile. If false, assumes the domain of the current directory or of WL_HOME.	true
DomainRegistrationEnabled	This property is deprecated in Oracle WebLogic Server 12.1.3.0 and may be removed in a future release.  To register multiple domain locations in the nodemanager.domains file, see Configuring nodemanager.domains File.	false
DomainKeystoresDomain	Specifies the domain name of the domain that hosts the machine keystores for this Node Manager instance. The domain name must match one of the domains in the nodemanager.domains file.  Only set this property if KeyStores is set to DomainKeystores and this is a per-host (not per-domain) configuration.	none
ExcludedCipherSuites	Specifies a list of cipher suites not to be enabled. This setting may override the CipherSuites property.  The strings can be full cipher suite names or may follow these patterns:	The default value is TLS_RSA_*,*_CBC_* which is a list of one or more strings, separated by a comma.
	<ul> <li>Start and end with an asterisk. For example, *CBC* to match the cipher suites containing CBC</li> <li>End with an asterisk. For example, TLS_RSA_* to match the cipher suites that start with TLS_RSA_</li> <li>Start with an asterisk. For example, *_CBC_SHA to match the cipher suites that end with _CBC_SHA</li> </ul>	



Table 2-1 (Cont.) Node Manager Properties

Node Manager Property	Description	Default
IfConfigDir	This configuration property sets a different directory for the location of the wlsifconfig.sh/.cmd script.	By default, this location is set appropriately, but you can use this property to modify the script location.
	This property is deprecated in Oracle WebLogic Server 12.1.3.0 but remains fully supported in Oracle WebLogic Server 15.1.1.0.0. See weblogic.IfConfigDir for more information and current support limitations for the replacement property.	
Interface	The primary interface names used by migratable servers. For server migration, the primary interface names used by migratable servers must be the same.	none
	See the <interfacename> property for more flexibility specifying multiple interfaces and a corresponding range of IP addresses that should be bound to a specific interface.</interfacename>	
	This property is deprecated in Oracle WebLogic Server 12.1.3.0 but remains fully supported in Oracle WebLogic Server 15.1.1.0.0. See <interfacename> for more information.</interfacename>	



Table 2-1 (Cont.) Node Manager Properties

Node Manager Property	Description	Default
<interfacename></interfacename>	An interface name along with a corresponding range of IP addresses and optional netmask value that should be bound to this specific network interface when migratable servers are started.	none
	<pre>Syntax: <interfacename>=<ip_range_min>-</ip_range_min></interfacename></pre>	
	For example, the syntax for binding addresses 1 - 4 to interface eth0 and addresses 5 - 8 to bond0 is:	
	eth0=1-4,NetMask=255.255.255.0 bond0=5-8,NetMask=255.255.248.0	
	You can leave out the NetMask value, if desired, and simply enter:	
	eth0=200.10.10.1-200.10.10.255 bond0=199.0.0.1-199.0.0.255	
	The original NetMask and Interface properties are still supported and when specified, would apply to any address that is not already defined in an IP range.	
	For example, specifying these properties in the original format would be the same as specifying this in the new format, as shown below:	
	Original Format:	
	<pre>Interface=oldEth0 NetMask=255.255.25.0</pre>	
	New Format:	
	oldEth0=*,Netmask=255.255.255.0	
	An asterisk (*) can be used to represent all IPs.	



Table 2-1 (Cont.) Node Manager Properties

Node Manager Property	Description	Default
JavaHome	The Java home directory that Node Manager uses to start Managed Servers on this machine, if the Managed Server does not have a Java home configured in its Remote Start page. If not specified in either place, Node Manager uses the Java home defined for a Node Manager process.	none
	<b>Note:</b> Oracle recommends not setting this property with the WLST set command. If set through WLST, this property reverts to its default value in the nodemanager.properties file.	
	This property is deprecated in Oracle WebLogic Server 12.1.3.0 but remains fully supported in Oracle WebLogic Server 15.1.1.0.0. For more information and current support limitations for the replacement properties, see weblogic.startup.JavaHome for Oracle WebLogic Server processes and coherence.startup.JavaHome for Coherence processes.	
JavaStandardTrustKeyStorePas sPhrase	Specifies the password defined when creating the Trust keystore. This field 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. Oracle WebLogic Server reads only from the keystore, so whether or not you define this property depends on the requirements of the keystore. This property is required when the Keystores property is set as CustomIdentityandJavaStandard Trust or DemoIdentityAndDemoTrust.	none
keyFile	The path to the private key file to use for SSL communication with the Administration Server.  Note: This property is used only in the process of upgrading from Oracle WebLogic Server,	none
	Version 7.x to Version 9.x.	
keyPassword	The password used to access the encrypted private key in the key file.	none
	<b>Note:</b> This property is used only in the process of upgrading from Oracle WebLogic Server, version 7.x to version 9.x.	



Table 2-1 (Cont.) Node Manager Properties

Node Manager Property	Description	Default
KeyStores	Indicates the configuration that Node Manager uses to find its identity (private key and digital certificate) and trust (trusted CA certificates) keystores.  Possible values are:	DemoIdentityAndDemoTrust
	<ul> <li>DemoIdentityAndDemoTrust - Uses the demonstration Identity and Trust keystores located in the DOMAIN_HOME\security directories that are configured by default. The demonstration Trust keystore trusts all the certificate authorities in the Java Standard Trust keystore (JAVA_HOME\jre\lib\security\cacert s)</li> </ul>	
	<ul> <li>CustomIdentityAndJavaStandardTrust         <ul> <li>Uses a keystore you create, and the trusted CAs defined in the cacerts file in the</li> <li>JAVA_HOME\jre\lib\security\cacerts directory.</li> </ul> </li> </ul>	
	<ul> <li>CustomIdentityAndCustomTrust - Uses the Identity and Trust keystores you create.</li> <li>DomainKeystores - Uses the machine-identity and machine-trust keystores maintained by the Certificate Management Service in a domain. For per-domain configurations, these are the machine keystores maintained by the host domain. For per-host configurations, these are the machine keystores maintained by the domain specified by the DomainKeystoresDomain property.</li> </ul>	
ListenAddress	Any address upon which the machine running Node Manager can listen for connection requests. This argument deprecates weblogic.nodemanager.listenAddress.	null With this setting, Node Manager will listen on any IP address on the machine
ListenBacklog	Maximum number of Node Manager backlog requests that the listener will accept. Additional incoming requests will be dropped until the backlogged requests are handled. Typically, you need not adjust this property.	50
ListenPort	The TCP port number on which Node Manager listens for connection requests. This argument deprecates weblogic.nodemanager.listenPort.	5556
LogAppend	If set to true, then a new log file is not created when Node Manager restarts; the existing log is appended instead.	true
LogCount	Maximum number of log files to create when LogLimit exceeds.	1
LogFile	Location of the Node Manager log file.	NodeManagerHome\nodemanagerlog



Table 2-1 (Cont.) Node Manager Properties

Node Manager Property	Description	Default
LogFormatter	Name of formatter class to use for Node Manager log messages.	weblogic.nodemanager.server. LogFormatter
LogLevel	Severity level of logging used for the Node Manager log. Node Manager uses the standard logging levels from the java.util.logging.level package, <a href="http://docs.oracle.com/javase/8/docs/api/java/util/logging/Level.html">http://docs.oracle.com/javase/8/docs/api/java/util/logging/Level.html</a> .	INFO
LogLimit	Maximum size of the Node Manager log specified as an integer in bytes. When this limit is reached, a new log file is started.	unlimited
LogToStderr	If set to true, the log output is also sent to the standard error output.	false
NativeVersionEnabled	A value of true causes native libraries for the operating system to be used.  For UNIX systems other than Solaris or Linux, set this property to false to run Node Manager in non-native mode.  For MacOS, set  NativeVersionEnabled=false or Node Manager will fail to start.  Note: When NativeVersionEnabled=false:	true
	<ul> <li>Node Manager cannot query if a PID is alive nor kill a particular process.</li> <li>Node Manager does not have the ability to determine if there are existing processes that need to be monitored and crash recovery is not fully implemented.</li> <li>nmKill is not supported when NativeVersionEnabled=false.</li> </ul>	
NetMask	The subnet mask for your network. For server migration, each Managed Server must use the same subnet mask to enable unicast and multicast communication among servers.  See the <interfacename> property for more flexibility entering multiple interfaces and corresponding netmask values.</interfacename>	none
	This property is deprecated in Oracle WebLogic Server 12.1.3.0 but remains fully supported in Oracle WebLogic Server 15.1.1.0.0. See <interfacename> for more information.</interfacename>	



Table 2-1 (Cont.) Node Manager Properties

Node Manager Property	Description	Default
NodeManagerHome	Node Manager root directory which contains the following configuration and log files:	NodeManagerHome
	<ul> <li>nodemanager.domains</li> </ul>	
	<ul><li>nodemanager.log</li></ul>	
	<ul> <li>nodemanager.properties</li> </ul>	
	See Node Manager Configuration and Log Files.	
	<ul> <li>Note:</li> <li>By default, NodeManagerHome is         DOMAIN_HOME\nodemanager. In a             production environment, you might want to             customize the location of the Node Manager             root directory.     </li> </ul>	
	<ul> <li>Oracle recommends not setting this property with the WLST set command. If set through WLST, this property reverts to its default value in the nodemanager.properties file.</li> </ul>	
PropertiesVersion	Specifies the version of the nodemanager.properties file. Do not change this value.	none
ProcessDestroyTimeout	Specifies the time a NodeManager waits to check if the Managed Server that was started by the start script is stopped.	20 seconds
	If NativeVersionEnabled is set to false (default value is true), a start script starts the monitored Managed Server and that will be a child process. In this case, the ProcessDestroyTimeout value (by default 20 seconds) determines how long the Node Manager should wait to check if the Managed Server is stopped.	
	<ul> <li>If the child process does not stop within the specified timeout value:</li> <li>Connect to the Administration Server to request a shutdown for the Managed Server. When the shutdown fails, use nmKill to kill the specified server instance.</li> </ul>	
	<ul> <li>If there is no option to stop the Managed Server, use Operating System (OS) level process kill to terminate the server instance and related OS processes.</li> </ul>	
	This property was introduced in Oracle WebLogic Server 12.1.2 and backported to 10.3.6.x.	
QuitEnabled	If set to true, allows you to remotely stop Node Manager.	false
	WLST overrides the default value, false, when stopping Node Manager using the stopNodeManager() command.	
SecureListener	If set to true, use the SSL listener, otherwise use the plain socket.	true



Table 2-1 (Cont.) Node Manager Properties

Node Manager Property	Description	Default
StateCheckInterval	Specifies the interval Node Manager waits to perform a check of the server state.	500 milliseconds
StopScriptEnabled	If set to true, execute the stop script specified by StopScriptName after the server instance has shutdown. See Configuring Node Manager to Use Start and Stop Scripts.  This property is deprecated in Oracle WebLogic Server 12.1.3.0 but remains fully supported in Oracle WebLogic Server 15.1.1.0.0. See weblogic.StopScriptEnabled for more information and current support limitations for the replacement property.	false
StopScriptName	The name of the script to be executed after server shutdown.  This property is deprecated in Oracle WebLogic Server 12.1.3.0 but remains fully supported in Oracle WebLogic Server 15.1.1.0.0. See weblogic.StopScriptName for more information and current support limitations for the replacement property.	none
UseMACBroadcast	Specifies whether or not to use a node's MAC address when sending ARP packets, that is, whether or not to use the -b flag in the arping command. When Node Manager binds IP addresses for automatic server migration, it also uses the arping command to attempt to notify nodes in the network that the machine is now using the virtual IP address. When weblogic.UseMACBroadcast=true, Node Manager uses the arping command with the -b flag.  This property is deprecated in Oracle WebLogic Server 12.1.3.0 but remains fully supported in Oracle WebLogic Server 15.1.1.0.0. See weblogic.UseMACBroadcast for more information and current support limitations for the replacement property.	false
WebLogicHome	Root directory of the Oracle WebLogic Server installation. This is used as the default value of - Dweblogic.RootDirectory for a Managed Server that does not have a root directory configured in its Remote Start page. If not specified in either place, Node Manager starts the Managed Server in the directory where Node Manager runs.	none



Table 2-1 (Cont.) Node Manager Properties

Node Manager Property	Description	Default
weblogic.IfConfigDir	This configuration property sets a different directory for the location of the wlsifconfig.sh/.cmd script.	By default, this location is set appropriately, but you can use this property to modify the script location.
	This property introduced in Oracle WebLogic Server 12.1.3 is not currently supported by WLST offline or by the pack and unpack commands. When using those utilities, use IfConfigDir instead.	
weblogic.startup.JavaHome	Applies when Node Manager invokes Java directly. Specifies the Java home directory that Node Manager uses to start Managed Servers on this machine, if the Managed Server does not have a Java home configured in its Remote Start page. If not specified in either place, Node Manager uses the Java home defined for a Node Manager process.  This property introduced in Oracle WebLogic	none
	Server 12.1.3 is not currently supported by WLST offline or by the pack and unpack commands. When using those utilities, use JavaHome instead.	
weblogic.StopScriptEnabled	If set to true, execute the stop script specified by weblogic.StopScriptName after the server instance has shutdown. See Configuring Node Manager to Use Start and Stop Scripts.  This property introduced in Oracle WebLogic Server 12.1.3 is not currently supported by WLST offline or by the pack and unpack commands. When using those utilities, use StopScriptEnabled instead.	false
weblogic.StopScriptName	The name of the script to be executed after server shutdown.  This property introduced in Oracle WebLogic Server 12.1.3 is not currently supported by WLST offline or by the pack and unpack commands. When using those utilities, use StopScriptName instead.	none



Table 2-1 (Cont.) Node Manager Properties

Node Manager Property	Description	Default
weblogic.UseMACBroadcast	Specifies whether or not to use a node's MAC address when sending ARP packets, that is, whether or not to use the -b flag in the arping command. When Node Manager binds IP addresses for automatic server migration, it also uses the arping command to attempt to notify nodes in the network that the machine is now using the virtual IP address. When weblogic.UseMACBroadcast=true, Node Manager uses the arping command with the -b flag.	false
	This property introduced in Oracle WebLogic Server 12.1.3 is not currently supported by WLST offline or by the pack and unpack commands. When using those utilities, use UseMACBroadcast instead.	

### Machine-Level Node Manager Settings for a Group of Server Instances

If you have a group of server instances started by the same Node Manager instance, you can configure certain machine-level settings once in the nodemanager.properties file and these settings will apply to all server instances in that group. Setting machine-level attributes simplifies configuration, as you do not have to configure common settings in the <a href="ServerStartMBean">ServerStartMBean</a> for each server instance.

If you configure specific settings for a server instance directly in the ServerStartMBean, that value takes precedence over any values configured in the nodemanager.properties file.

<u>Table 2-2</u> describes the machine-level settings you can configure in nodemanager.properties for a group of server instances started by the same Node Manager instance.

Table 2-2 Machine-Level Node Manager Properties for a Group of Server Instances

Property Name	Description
coherence.startup.Arguments	The Java arguments to use when starting the server instance.
	These are the first arguments appended immediately after java portion of the startup command. For example, you can set Java heap memory or specify any weblogic.nodemanager.server.provider.WeblogicCacheServer option.
	Separate arguments with a space.
coherence.startup.MW_Home	The MW_HOME directory (the path on the machine running Node Manager) to use when starting this server instance.
	Specifies the directory on the Node Manager machine under which all Oracle Middleware products are installed. For example, C:\Oracle\Middleware.



Table 2-2 (Cont.) Machine-Level Node Manager Properties for a Group of Server Instances

Property Name	Description
weblogic.startup.Arguments	The Java arguments to use when starting the server instance.
	Node Manager passes this value to a start script using the JAVA_OPTIONS environment variable. When issuing a Java command line to start the server instance, Node Manager passes the arguments as options.
weblogic.startup.Arguments.prepend	Prepends flags to any arguments configured for the server instances.
weblogic.startup.ClassPath	The classpath (the path on the machine running Node Manager) to use when starting this server instance.
	Node Manager passes this value to a start script using the CLASSPATH environment variable. When issuing a Java command line to start the server instance, Node Manager defines the classpath with – Djava.class.path=.
weblogic.startup.JavaVendor	The Java vendor value to use when starting the server instance.
	Node Manager does not pass this value when invoking a Java command line to start the server instance. It does pass this value in the environment variable JAVA_VENDOR to the start script.
weblogic.startup.MW_Home	The MW_HOME directory (the path on the machine running Node Manager) to use when starting this server instance.
	Node Manager does not pass this value to start scripts. When issuing a Java command line to start the server instance, Node Manager specifies MW_HOME with - Dbea.home=.
weblogic.startup.SecurityPolicyFile	The security policy file (the directory and file name on the machine running Node Manager) to use when starting this server instance.
	When Node Manager uses a start script, the security policy file is defined in an environment variable, SECURITY_POLICY. When issuing a Java command line to start the server instance, Node Manager defines the security policy with -Djava.security.policy=.
weblogic.startup.ServerGID	The group ID for the server instance.
weblogic.startup.ServerUID	The user ID for the server instance.

# **Configuring Remote Startup Arguments**

In the WebLogic Remote Console, use the Server Start page for the Managed Server to specify the startup arguments that Node Manager will use to start a Managed Server. To access this page, in the **Edit Tree**, go to **Environment**, then **Servers**. Select the Managed Server for which you want to configure startup arguments. On the **Advanced** tab, select the **Node Manager** subtab. For more information, see Configure Startup Arguments for a Managed Server in the *Oracle WebLogic Remote Console Online Help*.



If you do not specify startup arguments for a Managed Server, default values are used, as appropriate for the Managed Server. If using the Java-based implementation, Node Manager uses its own properties as defaults to start the Managed Server. If using the script-based implementation, the start script sets the default values. See <a href="Reviewing">Reviewing</a> <a href="mailto:nodemanager.properties">nodemanager.properties</a>.

Although these defaults are sufficient to boot a Managed Server, to ensure a consistent and reliable boot process, configure startup arguments for each Managed Server instance. The specified startup arguments are used for starting Managed Servers only. They will not be used by an Administration Server instance that is started by Node Manager.

If you run Node Manager as a Windows service, as described in <u>Running Node Manager as a Startup Service</u>, you must configure the -Xrs JVM property for each Managed Server that will be under the Node Manager control.

If you do not set this option, Node Manager will not be able to restart a Managed Server after a system reboot, due to this sequence of events:

- A reboot causes a running Managed Server to be killed before Node Manager and Administration Server operating system services are shut down.
- During the interval between the Managed Server being killed, and a Node Manager service being shut down, Node Manager continues to monitor the Managed Server, detects that it was killed, and attempts to restart it.
- 3. The operating system does not allow restart of the Managed Server because the machine is shutting down.
- 4. Node Manager marks the Managed Server as failed, and it will not start this server when the machine comes up again.

Starting a Managed Server with the -Xrs or -Xnohup option avoids this sequence of events by preventing the immediate shutdown of the Managed Server during machine shutdown.

### **Setting Server Startup Properties**

You can use Node Manager to set the startup properties for a server instance. These properties can be defined in startup.properties or passed as an object using administrative utilities such as WLST. The methods of setting startup properties and their valid values are outlined in the sections below.

This section includes the following topics:

### startup.properties

Node Manager uses the startup.properties file to determine the startup configuration when starting a server instance. This file is defined for each server instance and is located in <code>DOMAIN\_HOME/servers/server\_name/data/nodemanager/startup.properties</code>.

The contents of startup.properties are derived from the Server MBean, or the Cluster MBean if the server instance is part of a cluster. See the MBean Reference for Oracle WebLogic Server.

### Setting Startup Properties Using WLST

When using the WLST nmStart command, the server configuration cannot be determined directly. Therefore, you must pass the server start properties as a WLST properties object to the nmStart command.



## Server Startup Properties

The following server startup properties can be passed to a server instance when started using Node Manager.

**Table 2-3 Server Startup Properties** 

Property	Description
AdminURL	The URL of the Administration Server.
	<b>Note:</b> This value should only be specified in the startup.properties file for a Managed Server.
Arguments	The arguments used when starting the server instance.
AutoKillIfFailed	When a server instance is started by Node Manager, this attribute signals that the OverloadProtectionMBean settings will be changed to FailureAction == FORCE_SHUTDOWN and PanicAction == SYSTEM_EXIT. For more information about FailureAction and PanicAction, see OverloadProtectionMBean in MBean Reference for Oracle WebLogic Server.
AutoRestart	Specifies whether Node Manager can automatically restart this server instance if it fails.
	Note: The CrashRecoveryEnabled configuration property takes precedence over the AutoRestart property in a crash recovery scenario. For example, if a server instance has AutoRestart=false but CrashRecoveryEnabled=true, when Node Manager restarts, Node Manager tries to recover the server instance if the server failed when Node Manager was not running.
ClassPath	The classpath to use when starting a server instance.
JavaHome	Defines the Java home directory used when starting the server instance.
OracleHome	The Oracle home directory to use when starting a server instance.
RestartDelaySeconds	The number of seconds Node Manager should wait before attempting to restart the server instance.
RestartInterval	The amount of time Node Manager will spend attempting to restart a failed server instance. Within this period of time Node Manager will attempt to restart the failed server up to the number defined by RestartMax. By default, Node Manager will attempt to restart a server instance indefinitely until the FAILED_NOT_RESTARTABLE state is reached.
RestartMax	The number of times Node Manager will attempt to restart a failed server within the interval defined by RestartInterval. RestarMax is recognized only if RestartInterval is defined.
SecurityPolicyFile	Specifies the security policy file to use when starting this server.
ServerIP	The IP address of the server.
SSLArguments	These arguments are used when you have enabled the domain-wide administration port.



## Setting the Node Manager Environment Variables

By default, you need not set any additional environment variables before starting Node Manager. The sample Node Manager start scripts and install service scripts provided with Oracle WebLogic Server set the required variables and start Node Manager listening on the default address, localhost.

To start Node Manager listening on a non-default address, you can use one of the following methods:

Edit the nodemanager.properties file.

Set the LISTEN ADDRESS variable to <host> and the LISTEN PORT variable to <port> before calling the startNodeManager script. See Reviewing nodemanager.properties.

Set the values when executing the WL\_HOME\server\bin\startNodeManager for a per host Node Manager.



#### (i) Note

You cannot change the listen address or listen port using the startNodeManager script for the per domain Node Manager (DOMAIN HOME/bin/startNodeManager). This method works only for the per host Node Manager.

The startNodeManager scripts will set the first two positional parameters to LISTEN\_ADDRESS and LISTEN\_PORT when entered on the command line.

For example, enter this command to start Node Manager on host 11ama and port 7777:

```
startNodeManager.cmd llama 7777 (Windows)
sh startNodeManager.sh llama 7777 (UNIX)
```

Enter this command to start Node Manager on host 11ama:

```
startNodeManager.cmd llama (Windows)
sh startNodeManager.sh llama (UNIX)
```

Configuring a non-default listening address for Node Manager is most useful in production environments so that traffic from other machines can potentially reach it. Also, if you have a multihomed machine or a machine with multiple network interface cards, Node Manager can be listening on any one of the addresses on the machine.

**Table 2-4 Node Manager Environment Variables** 

<b>Environment Variable</b>	Description
CLASSPATH	You can set the Node Manager CLASSPATH either as an option on the java command line used to start Node Manager, or as an environment variable.
	Windows NT example:
	<pre>set CLASSPATH=.;%WL_HOME%\server\lib\weblogic_sp.jar;%WL_HOME%\server\lib\ weblogic.jar</pre>
JAVA_HOME	JDK root directory used by Node Manager. For example:
	set JAVA_HOME=c:\jdk1.7.0_06  Node Manager has the same JDK version requirements as Oracle WebLogic Server.



Table 2-4 (Cont.) Node Manager Environment Variables

Environment Variable	Description
LD_LIBRARY_PATH	For Solaris systems, you must include the path to the native Node Manager libraries.
(UNIX and Linux)	Solaris example:
	LD_LIBRARY_PATH: \$WL_HOME/server/lib/solaris: \$WL_HOME/server/lib/solaris/oci816_8 Linux example:
	LD_LIBRARY_PATH: \$WL_HOME/server/native/linux: \$WL_HOME/server/native/linux/i686  Note: Linux can be i686, ia64, or x86_64 architecture. The path would change to correspond with the appropriate architecture.
PATH	Must include the Oracle WebLogic Server bin directory and path to your Java executable. For example:
	set PATH=%WL_HOME%\server\bin;%JAVA_HOME%\bin;%PATH%
WL_HOME	Oracle WebLogic Server installation directory. For example:
	set WL_HOME=c:\Oracle\Middleware\Oracle_Home\wlserver

## Configuring Node Manager as an xinetd Service

When configuring Node Manager to run as an inetd or xinetd service, the following considerations apply:

- Ensure that NodeManagerHome and other system properties are defined.
- If xinetd is configured with libwrap, you should add the NOLIBWRAP flag.
- Ensure that the hosts.deny and hosts.allow files are configured correctly.
- Depending on your network environment, perform additional configurations if necessary.

The following example shows how Node Manager can be configured within xinetd:

```
# Create the $domaindir/bin/startNMService.sh script or the
# $WL_HOME/server/bin/startNMService.sh script to produce output to a file.
#!/bin/sh
$domaindir/bin/startNodeManager.sh >> $NM_HOME/nmservice.out 2>&1
# The service can now take advantage of the startNMService script.
# default: off
# description:nodemanager as a service
service nodemanager-svc
                  = UNLISTED
   type
  disable
                  = yes
  socket_type
                  = stream
  protocol
                  = tcp
  wait
                  = yes
  user
                  = <username>
```



```
port = 5556
flags = NOLIBWRAP
log_on_success += DURATION HOST USERID
server = /scratch/jdorr/dom1213/bin/startNMService.sh
env = MW_HOME=/Oracle/Middleware/Oracle_Home_WLS_12.2.1 JAVA_HOME=/Java/
jdk1.7.0_51LD_LIBRARY_PATH=/Oracle/Middleware/Oracle_Home_WLS_12.2.1/bea/wlserver/server/
native/linux/x86_64:/usr/lib:/lib:/usr/X11R6/lib
}
```

## Configuring Node Manager as a systemd Service

You can configure Node Manager as a systemd service. This section includes the following topics:

### Configuring Per Domain Node Manager as a systemd Service

To install and configure a per domain Node Manager as a systemd service, complete the following steps. Run all actions as the root user.

- Determine the service name, referred to as <service name> in the following steps.
   Oracle recommends using the naming pattern DOMAIN\_NAME-nodemanager.service, for example, base domain-nodemanager.service.
- 2. Copy the systemd unit file in Example 2-3 to the /etc/systemd/system directory.
- **3.** Edit /etc/systemd/system/<service name> and change the following variables according to your environment and installation:
  - WorkingDirctory Path to the domain home directory.
  - ExecStart Path to the startNodeManager.sh script in the domain.
  - ExecStop Path to the stopNodeManager.sh script in the domain.
  - User The user that the Node Manager service should be running.
  - Group The group that the Node Manager service should be running.
  - Environment:
    - MW\_HOME Path to the Oracle WebLogic Server installation.
    - The environment variable can be specified multiple times to provide environment variable name/values for the process started by systemd. For example:
      - \* Environment="MW\_HOME=/scratch/wls/Oracle/Middleware/Oracle\_Home"
      - \* Environment="NODEMANAGER\_DIR=/scratch/wls/Oracle/Middleware/ Oracle\_Home/oracle\_common/common/nodemanager"
- 4. The following table shows some commands for managing your Node Manager service.

Task	Command
To reload sytemd with the newly added or modified service	sudo systemctl daemon-reload
To start the service	sudo systemctl start <service name=""></service>
To stop the service	sudo systemctl stop <service name=""></service>
To show the status of the service	sudo systemctl status <service name=""></service>



Task	Command
To enable the service to start at system boot time	sudo systemctl enable <service name=""></service>
To disable the service from starting at system boot time	sudo systemctl disable <service name=""></service>
To view the console log of the Node Manager service	sudo journalctl -flu <service name=""></service>

For details and additional commands, see the **systemct** documentation.

#### Example 2-3 Configuring a Per Domain Node Manager as a systemd Service

[IInit]

Description=WebLogic NodeManager Service for mydomain

[Service]

Type=simple

WorkingDirectory=/scratch/wls/Oracle/Middleware/Oracle\_Home/user\_projects/domains/mydomain

Environment="MW\_HOME=/scratch/wls/Oracle/Middleware/Oracle\_Home"

ExecStart=/scratch/wls/Oracle/Middleware/Oracle\_Home/user\_projects/domains/mydomain/bin/ startNodeManager.sh

ExecStop=/scratch/wls/Oracle/Middleware/Oracle\_Home/user\_projects/domains/mydomain/bin/ stopNodeManager.sh

User=oracle

Group=oracle

[Install]

WantedBy=multi-user.target

### Configuring Per Host Node Manager as a systemd Service

To install and configure a per host Node Manager as a systemd service, complete the following steps. Run all actions as the root user.

Before you begin, ensure the per host Node Manager is correctly configured in your domain before using this script. See <u>Configuring Per Host Node Manager</u>.

- Determine the service name, referred to as <service name> in the following steps.
  - Oracle recommends using the naming pattern <a brief name for MW\_HOME>-nodemanager.service, for example, wls15.1.1-nodemanager.service.
- 2. Copy the sample systemd unit file in Example 2-4 to the /etc/systemd/system/ directory.
- 3. Edit /etc/systemd/system/<service name> and change the following variables according to your environment and installation:
  - WorkingDirctory Path to the Node Manager directory set up for the per host Node Manager.
  - ExecStart Path to the startNodeManager.sh script in the Oracle WebLogic Server installation.
  - ExecStop Path to the stopNodeManager.sh script in the Oracle WebLogic Server installation.
  - User The user that the Node Manager service should be running.
  - · Group The group that the Node Manager service should be running.
  - Environment:



- MW\_HOME Path to the Oracle WebLogic installation.
- NODEMANAGER\_DIR Path to the Node Manager directory set up for the per host Node Manager.
- The environment variable can be specified multiple times to provide environment variable name/values for the process started by systemd. For example:
  - \* Environment="MW\_HOME=/scratch/wls/Oracle/Middleware/Oracle\_Home"
  - \* Environment="NODEMANAGER\_DIR=/scratch/wls/Oracle/Middleware/ Oracle\_Home/oracle\_common/common/nodemanager"
- 4. The following table shows some commands for managing your Node Manager service.

Task	Command
To reload sytemd with the newly added or modified service	sudo systemctl daemon-reload
To start the service	sudo systemctl start <service name=""></service>
To stop the service	<pre>sudo systemctl stop <service name=""></service></pre>
To show the status of the service	<pre>sudo systemctl status <service name=""></service></pre>
To enable the service to start at system boot time	<pre>sudo systemctl enable <service name=""></service></pre>
To disable the service from starting at system boot time	sudo systemctl disable <service name=""></service>
To view the console log of the Node Manager service	sudo journalctl -flu <service name=""></service>

For details and additional commands, see the systemctl documentation.

#### Example 2-4 Configuring a Per Host Node Manager as a systemd Service

```
[Unit]
```

Description=WebLogic Per Host NodeManager Service

[Service]

Type=simple

WorkingDirectory=/scratch/wls/Oracle/Middleware/Oracle\_Home/oracle\_common/nodemanager

Environment="MW\_HOME=/scratch/wls/Oracle/Middleware/Oracle\_Home"

Environment="NODEMANAGER\_DIR=/scratch/wls/Oracle/Middleware/Oracle\_Home/oracle\_common/
common/nodemanager"

ExecStart=/scratch/wls/Oracle/Middleware/Oracle\_Home/wlserver/server/bin/

startNodeManager.sh

ExecStop=/scratch/wls/Oracle/Middleware/Oracle\_Home/wlserver/server/bin/

stopNodeManager.sh

User=joes

Group=sdsales

[Install]

WantedBy=multi-user.target

# Configuring Script-Based Node Manager

This chapter describes how to configure script-based version of Oracle WebLogic Servers Node Manager.

This chapter includes the following topics:

### Overview

Script-based Node Manager is a shell script, wlscontrol.sh, located in WL\_HOME/common/bin/. The wlscontrol.sh file must exist on each machine that hosts server instances that you want to control with Node Manager. This script can be customized to meet site-specific requirements.

You must have an SSH client executable on each machine where Node Manager or a Node Manager client runs. This script must also be in the path of the user ID running it. Typically, an SSH client is a standard part of a UNIX or Linux installation.

## Step 1: Create User Accounts

Before running Node Manager, create a dedicated UNIX user account for performing Node Manager functions. Add this user to all machines that host the script-based Node Manager and to all machines that host a Node Manager client, including the Administration Server.



On UNIX platforms, Oracle does not recommend running Node Manager as the root user. However, to achieve Post-Bind GID, you must start Node Manager as the root user. Post-Bind GID enables a server running on your machine to bind to a UNIX group ID (GID) after it finishes all privileged startup actions.

#### For example:

- 1. On each host machine, as the root user, create two new operating system (OS) users: **bea** and **ndmgr**, both associated with a new group called **bea**.
  - Use bea for installing Oracle WebLogic Server only.
  - Use ndmgr to create a WebLogic domain and start the Administration Server and remote Managed Servers using Node Manager.
- 2. Both OS users should have the same OS group (**bea**) to ensure that the correct permissions are in place for **ndmgr** to run WebLogic scripts and executables.

#### For example:

- > groupadd bea
- > useradd -g bea -m bea
- > passwd bea



- > useradd -q bea -m ndmqr
- > passwd ndmgr

# Step 2: Configure Node Manager Security

The Node Manager SSH shell script relies on SSH user-based security to provide a secure trust relationship between users on different machines. You create a UNIX user account typically one per domain, for running Node Manager commands and scripts. A user logged in as this user can issue Node Manager commands without providing a user name and password.

#### (i) Note

You must also ensure that the Node Manager and Oracle WebLogic Server commands are available in the path of the UNIX user ID used to run them. Change the environment file of the user to contain the path to WL HOME/common/bin/or DOMAIN HOME/bin/server migration.

#### For example:

PATH=/usr/bin:/bin:ORACLE\_HOME/user\_projects/domains/domain\_name/bin/ server\_migration

Configure SSH trust between the ndmgr user on each machine that will run an Oracle WebLogic Server instance and the same ndmgr user on the same machine, plus the corresponding **ndmgr** user on every other machine.

In other words, any ndmgr user on one machine must be able to establish an SSH session without being prompted for security credentials, with a **ndmgr** user of the same name on the same or a different machine. This is necessary because any Managed Server can become the cluster master for migratable servers and issue commands to start other remote Managed Servers in the cluster using SSH. For Managed Server migration to work, the ndmgr user needs only to be able to run the wlscontrol.sh script using SSH. See Configuring Security for WebLogic Server Scripts.

For example, to configure one instance of a user to trust another instance of a user for SSH version2:

From a terminal logged in as **ndmgr** user:

```
> ssh-keygen -t dsa
```

- 2. When prompted, accept the default locations and press **Enter** for passphrase so that no passphrase is specified.
- 3. Copy the **ndmgr** user's public key to the **ndmgr** user's home on the same machine and all other machines.

```
> scp .ssh/id_dsa.pub ndmgr@192.168.1.101:./
```



Establish an SSH session with the target machine as the ndmgr user and set up trust for the remote ndmgr user.

```
> ssh -l ndmgr 192.168.1.101 (you should be prompted for password)
> mkdir .ssh
> chmod 700 .ssh
> touch .ssh/authorized_keys
> chmod 700 .ssh/authorized_keys
> cat id_dsa.pub >> .ssh/authorized_keys
> rm id_dsa.pub
> exit
```

Test that you can establish an SSH session with the ndmgr user on the remote machine without requiring a password.

```
> ssh -l ndmgr 192.168.1.101
```

6. Repeat this process for all combinations of machines.

Alternatively, you can achieve the same result by generating a key value pair on each machine, concatenating all of the public keys into an authorized\_keys file, and copying (scp) that file to all machines. Try establishing SSH sessions between all combinations of machines to ensure that the ~/.ssh/known\_hosts files are correctly configured. See <a href="Generating and Distributing">Generating and Distributing</a> Key Value Pairs.

# Step 3: Install Oracle WebLogic Server

Install an Oracle WebLogic Server instance in the base directory, <code>ORACLE\_HOME/wlserver</code>, on all the machines that will run Oracle WebLogic Server, where <code>ORACLE\_HOME</code> represents the directory you specify as the Oracle Home when you install Oracle WebLogic Server, for <code>example</code>, <code>C:\Oracle\Middleware\Oracle\_Home</code>.

For example:

```
> java -jar wls_jrf_generic.jar
```

See Starting the Installation Program in *Installing and Configuring Oracle WebLogic Server and Coherence*.

# Step 4: Create a WebLogic Domain

Create a WebLogic domain using the Configuration Wizard on the machine which hosts the Administration Server only.

Subsequently, when you start the Administration Server, it will use the configuration in the config subdirectory of this domain directory to determine the settings for the Administration Server and the domain.

It is likely that most Managed Server instances will be run remotely with respect to the Administration Server. Therefore, these Managed Servers will not have direct access to the domain configuration directory of the Administration Server. Instead, they will run from a skeleton domain directory in their respective machine's **ndmgr** home directory and will obtain their configuration over the network on startup from the remotely running Administration Server.



As the **ndmgr** user, create the WebLogic domain.

#### For example:

- 1. Run the Configuration Wizard:
  - > ORACLE\_HOME/wlserver/common/bin/config.sh
- Create a new WebLogic domain based on the default Oracle WebLogic Server template.
- 3. For the Administration Server, specify a fixed IP address (for example, 192.168.1.100).
- 4. In Customize Environment and Service Settings, select Yes.
- In Configure Managed Servers, add two Managed Servers, MS1 and MS2.
   For the Managed Servers, specify floating IP addresses (for example, 192.168.1.201 and 192.168.1.202).
- 6. In Configure Clusters, add a cluster, CLUST, and then assign MS1 and MS2 to it.

  Do not specify any Machines or UNIX Machines; you will do this manually in a subsequent step.
- 7. Name the domain 'clustdomain' and save it to ORACLE\_HOME/clustdomain.

# Step 5: Configure nodemanager.domains File

The nodemanager.domains file specifies the domains that a Node Manager instance controls. The standalone clients do not need to specify the domain directory explicitly. The file nodemanager.domains provides additional security by restricting Node Manager client access to the domains listed in this file. The client can only execute commands for the domains listed in nodemanager.domains.

This file must contain an entry specifying the domain directory for each domain a Node Manager instance controls, in this form:

domain-name=domain-directory

When a user issues a command for a domain, Node Manager looks up the domain directory from nodemanager.domains.

For the script-based Node Manager, this file's default location is <code>WL\_HOME/common/nodemanager</code>, where <code>WL\_HOME</code> is the location in which you installed Oracle WebLogic Server, for example, <code>ORACLE\_HOME/wlserver</code>.

If you are using the script-based implementation of Node Manager, you must create or copy into <code>NodeManagerHome</code>, a nodemanager.domains file that specifies the domains that you want a Node Manager instance to control. Alternatively, you can register WebLogic domains with the script-based Node Manager using the WLST command, <code>nmEnroll</code>. See <a href="Configuring">Configuring</a> nodemanager.domains File.

# Step 6: Start the Administration Server

Aside from the standard method of starting the Administration Server by using the provided startWebLogic scripts, use the script-based implementation of Node Manager to monitor the Administration Server to ensure high availability. Run the wlscontrol.sh Node Manager script to start the Administration Server locally.



If using a production domain, create <code>boot.properties</code> for the Administration Server for successful server startup. As the <code>ndmgr</code> user, start the Administration Server using the <code>wlscontrol.sh</code> Node Manager script.

#### For example:

```
> ORACLE_HOME/wlserver/common/bin/wlscontrol.sh -d clustdomain -r ORACLE_HOME/clustdomain -c -f startWebLogic.sh -s AdminServer START
```

For verbose logging to standard out, add the -v parameter to the command.

#### Note

Once the Administration Server has been successfully started, and the boot.properties file has been created, you can start the Administration Server remotely. Stop the Administration Server using the stopWebLogic.sh Node Manager script. Then, start the Administration Server remotely using SSH.

#### For example:

```
> ssh -l ndmgr -o PasswordAuthentication=no 192.168.1.100
ORACLE_HOME/wlserver/common/bin/wlscontrol.sh -d clustdomain -r /home/
ndmgr/clustdomain -c -f startWebLogic.sh -s AdminServer START
```

# Step 7: Configure Node Manager on the Managed Servers

Each machine that will host a Managed Server will have a skeleton domain created and configured.

- 1. From a local terminal, create a new empty directory (clustdomain) in the home directory for the ndmgr user for each of the Managed Server host machines and also a back-up machine. For example:
  - > mkdir clustdomain
- For each of the Managed Server host machines and the back-up machine, as the ndmgr user, use WLST to enroll the user's home directory as being the base directory for remotely run servers and for Node Manager.

#### For example:

```
> ORACLE_HOME/wlserver/common/bin/wlst.sh
> connect('weblogic','weblogic','t3://192.168.1.100:7001')
> nmEnroll('/home/ndmgr/clustdomain','/home/ndmgr')
> exit()
```

Be sure to run nmEnroll on each remote machine. This command creates a property file, / home/ndmgr/nodemanager.domains, which maps domain names to home directories, and creates the required domain configuration and security information so that Managed Servers can communicate with the Administration Server.





#### (i) Note

In previous releases, nmEnroll would download the domain encryption key (SerializedSystemIni.dat file) from the Administration Server. However, now, if it does not already exist, you must manually copy the file from the Administration Server domain directory or an existing server in the domain *before* running the nmEnroll command. The file location should be \${DOMAIN\_HOME}/security/ SerializedSystemIni.dat.

The nodemanager domains file removes the need to specify the domain home directory (with -r) when starting wlscontrol.sh. However, since you changed the Node Manager home directory, you must specify -n /home/ndmgr. The default Node Manager home directory is ORACLE HOME/wlserver/common/nodemanager; you might not want to use this directory as it is in the product installation directory and owned by another user.



#### (i) Note

By default, you can start a Node Manager from any directory. A warning will be issued if no nodemanager.domains file is found. You must create or copy into NodeManagerHome, a nodemanager.domains file that specifies the domains that you want a Node Manager instance to control or register WebLogic domains using the WLST command, nmEnroll.

# Step 8: Test Node Manager Setup and Configuration by Starting **Managed Servers**

Start the Managed Servers to test the Node Manager setup and configuration.

Copy the scripts from the Administration Server's domain bin directory to the corresponding domain bin directory on each Node Manager machine (for example, /home/ ndmgr/bin). For example:

```
> scp ndmgr@192.168.1.100:"~/clustdomain/bin/*"
ndmgr@192.168.1.101:~/clustdomain/bin
```

2. For each Node Manager machine (including the back-up machine), edit the shell scripts in the bin directory to reflect the proper path for the local domain home, and the remote Administration Server's IP address.

#### For example:

- Edit the DOMAIN\_HOME and LONG\_DOMAIN\_HOME variables in the setDomainEnv.sh script to correctly reflect this remote domain home directory:
  - DOMAIN HOME=/home/ndmgr/clustdomain
  - LONG\_DOMAIN\_HOME=/home/ndmgr/clustdomain
- b. Similarly, edit the DOMAIN\_HOME variable in startWebLogic.sh.
- c. Edit the DOMAIN\_HOME and ADMIN\_URL (for example, t3://192.168.1.100:7001) variables in startManagedWebLogic.sh.



- For each of the Managed Server host machines (including the back-up machine), as the **ndmgr** user, create a server/security subdirectory in the domain directory. For example, for the Managed Server MS1:
  - > mkdir -p ~/clustdomain/servers/MS1/security



#### (i) Note

For the back-up machine, create a server directory for every migratable Managed Server (for example, both MS1 and MS2).

Create a new boot.properties file with the appropriate user name and password variables specified in each Managed Server's security directory (for example, /home/ ndmgr/clustdomain/servers/MS1/security). For example:

username=weblogic password=password



#### (i) Note

When a Managed Server is first started using the script-based Node Manager, the values in this file will be encrypted.

For each of the Managed Server machines, as the **ndmgr** user, start the Managed Server locally from a terminal using the wlscontrol.sh Node Manager script. For example, to start the Managed Server, MS1:

```
> ORACLE HOME/wlserver/common/bin/wlscontrol.sh -d clustdomain -n /home/
ndmgr -c -f startManagedWebLogic.sh -s MS1 START
```

For verbose logging to standard out, add the -v parameter to the command.

Once successfully started, stop the Managed Servers and then, as the **ndmgr** user, attempt to start the Managed Servers remotely using SSH. For example to start MS1:

```
> ssh -l ndmgr -o PasswordAuthentication=no -p 22 192.168.1.101
ORACLE_HOME/wlserver/common/bin/wlscontrol.sh -d clustdomain -n /home/
ndmgr -c -f startManagedWebLogic.sh -s MS1 START
```

7. Once successfully started, stop the Managed Servers again and then repeat the process by trying to start each Managed Server (MS1) on the back-up machine instead. Again, stop this server once it successfully starts.

# Step 9: Configure UNIX Machines

Use the WebLogic Remote Console to configure a new UNIX Machine for each machine which hosts an Administration or Managed Server (including the back-up machine). By using a UNIX machine, you can configure the machine to run all WebLogic Server instances under a



privileged user ID (UID) or a group ID (GID). The processes are assigned (bound) to the selected UID or GID after the computer has carried out all the privileged startup actions.

Table 3-1 UNIX Machine Settings

Property	Value
O.S. Type	UNIX
Node Manager Type	SSH
Node Manager Listen Address	<pre><primary-ip-address> (not floating IP address)</primary-ip-address></pre>
Node Manager Listen Port	22
Node Manager Home	/home/ndmgr
Node Manager Shell Command	ssh -l ndmgr -o PasswordAuthentication=no -p %P %H ORACLE_HOME/wlserver/common/bin/wlscontrol.sh -d %D -n / home/ndmgr -c -f startManagedWebLogic.sh -s %S %C
Node Manager Debug Enabled	true
Servers	<pre><name administration="" hosted="" machine="" managed="" of="" on="" or="" servers="" the=""></name></pre>

# Step 10: Assign Servers to Machines

After all of the UNIX Machines are created, use the WebLogic Remote Console to set the Machine property for each server, to ensure each server is associated with its corresponding UNIX Machine.

See Assign a Server Instance to a Machine in the *Oracle WebLogic Remote Console Online Help*.

# Step 11: Start Managed Servers

Use the WebLogic Remote Console to start the Managed Servers.

See Start Managed Servers in the Oracle WebLogic Remote Console Online Help.

Check the server logs in the <code>/home/ndmgr/clustdomain/servers/managed\_server\_name/logs</code> directory of each Managed Server to ensure that the server has started with no errors.

# Configuring Script-Based Node Manager Security

To configure the script-based Node Manager security, perform the steps to override the default SSH port, configure the security for Oracle WebLogic Server scripts, configure remote server start security, and generate and distribute key value pairs.

This section includes the following topics:

# Overriding the Default SSH Port

The default SSH port used by Node Manager is 22. You can override that setting in the following ways:



- Set the Port= parameter in the ~/.ssh/config file to set the default port for an individual user.
- Set the Port= parameter in the /etc/ssh\_config file to set the default port across the entire system.
- Start the Administration Server using the following system property:

```
-Dweblogic.nodemanager.ShellCommand="ssh -o PasswordAuthentication=no -p %P %H wlscontrol.sh -d %D -r %R -s %S %C"
```

After starting the server, you can edit the SSH port in the Administration Server's configuration file.

# Configuring Security for Oracle WebLogic Server Scripts

To perform server migration and other tasks, the user ID executing scripts such as wlscontrol.sh must have sufficient security permissions. This includes being able to bring an IP address online or take an IP address offline using a network interface.

Server migration is performed by the cluster master when it detects that a server has failed. It then uses SSH to launch a script on the target machine to begin the migration. The script on the target machine runs as the same user ID running the server on the cluster master.

The commands required to perform server migration are wlsifconfig and arping. Since these scripts require elevated OS privileges, it is important to note that this can prevent a potential security hole. Using sudo, you can configure SSH to allow only wlsifconfig and arping to be run using elevated privileges.

The scripts are located in the WL\_HOME/common/bin/ directory or the DOMAIN\_HOME/bin/server\_migration directory. See <a href="Step 2">Step 2</a>: Configure Node Manager Security.

## Configuring Remote Server Start Security for Script-Based Node Manager

A remote start user name and password is required to start a server instance with Node Manager. These credentials are provided differently for Administration Servers and Managed Servers.

- Credentials for Managed Servers When you invoke Node Manager to start a Managed Server, it obtains its remote start name and password from the Administration Server.
- Credentials for Administration Servers When you invoke Node Manager to start an
  Administration Server, the remote start user name can be provided on the command line,
  or obtained from the Administration Server's boot.properties file. The Configuration
  Wizard initializes the boot.properties file and the startup.properties file for an
  Administration Server when you create the domain.

Any server instance started by Node Manager encrypts and saves the credentials with which it started in a server-specific boot.properties file, for use in automatic restarts.

### Generating and Distributing Key Value Pairs

The script-based Node Manager uses two types of key value pairs.

This section contains instructions for distributing key value pairs to the machines that will host a Node Manager client or server.



### Shared Key Value Pair

This option distributes the same key value pair to all machines that will host a Node Manager client or server.

The simplest way to accomplish this is to set up your LAN to mount the Node Manager user home directory on each of the machines. This makes the key value pair available to the machines. Otherwise:

1. Generate an RSA key value pair for the user with the ssh-keygen command provided with the SSH installation.

The default location for the private and public keys are  $\sim/.ssh/id\_rsa$  and  $\sim/.ssh/id\_rsa$ .pub, respectively.

If these keys are stored in a different location, modify the ShellCommand template, adding an option to the ssh command to specify the location of the keys.

2. Append the public key to the ~/.ssh/authorized\_keys file on the Node Manager machine. For example:

```
command="ORACLE_HOME/wlserver/common/nodemanager/nodemanager.sh" 1024 33
23...2323
```

In this command, you will substitute the public key that you generated, as stored in id\_rsa.pub, for the string shown in the example as:

```
1024 33 23...2323
```



The prefix command=command ensures that a user that establishes a session with the machine using the public key can run only the command specified—nodemanager.sh. This ensures that the user can perform only Node Manager functions, and thus prevents unauthorized access to data, system utilities, or other resources on the machine.

- 3. Manually distribute the key value pair to each machine that will host a Node Manager server instance or client.
- 4. Execute the following command on the client machine to check that the Node Manager client can access Node Manager:

```
$ ssh montgomery wlscontrol.sh VERSION
```

This response indicates that the client accessed Node Manager successfully:

```
NodeManager version 12.1
```

### Individual Key Value Pairs

On each machine that will host a Node Manager client:

 Generate a separate RSA key value pair for the Node Manager user as described in step one in the previous section. See <u>Shared Key Value Pair</u>.



2. Append the public key to the machine's ~/.ssh/authorized\_keys file user as described in step two in the previous section. See <a href="Shared Key Value Pair">Shared Key Value Pair</a>.

## **Using Node Manager**

Oracle recommends running WebLogic Servers Node Manager as a startup service. Use the recommended methods to start and stop the Java-based and script-based Node Manager. This chapter includes the following topics:

## Starting and Stopping Node Manager

Oracle WebLogic Server provides various methods for starting and stopping Node Manager.

This section includes the following topics:

### Running Node Manager as a Startup Service

Node Manager must run on each computer that hosts an Oracle WebLogic Server instance that you want to control with Node Manager. Ideally, Node Manager should run as an operating system service or daemon, so that it is automatically restarted in the event of a system failure or reboot.

By default, the operating system service starts up Node Manager to listen on localhost:5556. If you want Node Manager to accept commands from remote systems, you must edit the script to listen on a non-localhost listen address.

Oracle recommends that you install Node Manager to run as a startup service. This allows Node Manager to start up automatically each time the system is restarted. See <u>Configuring Node Manager as an xinetd Service</u> and <u>Configuring Node Manager as an init.d Service</u>.

#### (i) Note

On UNIX platforms, Oracle does not recommend running Node Manager as the root user. However, to achieve Post-Bind GID, you must start Node Manager as the root user. Post-Bind GID enables a server instance running on your machine to bind to a UNIX group ID (GID) after it finishes all privileged startup actions.

On Windows machines, use the following steps to install a per domain Node Manager Windows service:

- 1. Log in to the machine with Administrator privileges.
- Open a DOS command prompt window.
- Change to the DOMAIN\_HOME\bin directory.
- 4. Enter the following command:

installNodeMgrSvc.cmd

5. After a few seconds, the following message is displayed:

Oracle WebLogic <domain-name> NodeManager installed.



The service is installed using the default Node Manager listen port (5556). If this listen port is already in use, the program prompts you to enter a different listen port.



### (i) Note

If the Node Manager Windows service is already installed, the following message is displayed instead:

CreateService failed - The specified service already exists.

If you want to uninstall a per domain Node Manager Windows service, use the following steps:

- 1. Log in to the machine with Administrator privileges.
- Open a DOS command prompt window.
- Change to the DOMAIN\_HOME\bin directory.
- Enter the following command:

uninstallNodeMgrSvc.cmd

After a few seconds, the following message is displayed:

Oracle WebLogic <domain-name> NodeManager removed.

By default, NODEMGR\_HOST is set to localhost in installNodeMgrSvc.cmd, which means that Node Manager will listen only on the local host. If you do not want Node Manager listening on the local host, set NODEMGR\_HOST to a valid hostname or IP address before installing the Node Manager service.



#### (i) Note

If you select to run a per host Node Manager as a Windows service, using WL\_HOME\server\bin\installNodeMgrSvc.cmd, you must first perform the prerequisite configuration steps described in Configuring Per Host Node Manager.

### Starting Java-Based Node Manager Using Scripts

Although running Node Manager as an operating system service is recommended, you can also start Node Manager manually at the command prompt or with a script. The environment variables Node Manager requires are described in Set Node Manager Environment Variables.

Sample start scripts for Node Manager are installed in each DOMAIN\_HOME/bin and the WL HOME\server\bin directory, where WL HOME is the top-level installation directory for Oracle WebLogic Server. However, if you select to use the script in WL HOME\server\bin, you must first perform the prerequisite steps described in Configuring Per Host Node Manager.

Use startNodeManager.cmd on Windows systems and startNodeManager.sh on UNIX systems.

The scripts set the required environment variables and start Node Manager in the appropriate NodeManagerHome directory. Node Manager uses this directory as a working directory for output and log files. To specify a different working directory, edit the start script with a text editor and set the value of the NODEMGR HOME variable to the desired directory.



### Command Syntax for Starting Java-Based Node Manager

The syntax for starting Java-based Node Manager is:

java [java\_option=value ...] -D[nodemanager\_property=value] -D[server\_ property=value] weblogic.NodeManager

#### Where:

java\_option is a direct argument to the java executable, such as -ms or -mx.



### (i) Note

If you did not set the CLASSPATH environment variable, use the -classpath option to identify required Node Manager classes.

nodemanager\_property is a Node Manager property. Instead of supplying Node Manager property values on the command line, you can edit the nodemanager.properties file, which is created in the NodeManagerHome directory. See Reviewing nodemanager.properties.

Node Manager property values you supply on the command line override the values in nodemanager.properties.

- server\_property is a server-level property that Node Manager accepts on the command line, including:
  - bea.home: The BEA home directory that server instances on the current machine use.
  - java.security.policy: The path to the security policy file that server instances on the current machine use.



### Note

### For UNIX systems:

If you run Node Manager on a UNIX operating system other than Solaris, you cannot have any white space characters in any of the parameters that will be passed to the java command line when starting Node Manager. For example, this command fails due to the space character in the name "big iron".

-Dweblogic.Name="big iron"

For UNIX systems other than Solaris and Linux operating systems, you must disable the weblogic.nodemanager.nativeVersionEnabled option at the command line when starting Node Manager (or set the property in nodemanager.properties) to use the pure Java implementation. See Reviewing nodemanager.properties.

### For MacOS systems:

In nodemanager.properties, set NativeVersionEnabled=false or Node Manager will fail to start.



### Running Script-Based Node Manager



#### (i) Note

In this release of Oracle WebLogic Server, prior to running the script-based implementation of Node Manager, you must create or copy into NodeManagerHome, a nodemanager.domains file that specifies the domains that you want a Node Manager instance to control. See Configuring nodemanager.domains File. Alternatively, you can register WebLogic domains with Node Manager using the WLST command, nmEnroll.

If not specified, the default NodeManagerHome location is WL\_HOME/common/ nodemanager.

To use the SSH Node Manager Command Shell, start the Administration Server using the following command line option:

-Dweblogic.nodemanager.ShellCommand='ssh -o PasswordAuthentication=no %H wlscontrol.sh -d %D -r %R -s %S -x -c -f sample\_custom\_startscript.sh %C'



### (i) Note

**C** must be the last argument supplied to wlscontrol.sh.

The weblogic.nodemanager.ShellCommand attribute specifies the command template to use to communicate with a remote script-based Node Manager and execute Node Manager functions for server instances under its control.

The template assumes that wlscontrol.sh is in the default path on the remote machine hosting Node Manager.

The ShellCommand syntax is:

ssh -o PasswordAuthentication=no %H wlscontrol.sh -d %D -r %R -s %S %C'

The possible command line options are listed in Table 4-1. The possible parameter values are listed in Table 4-2.

For example, if you type the following command, the listen address and port of the SSH server default to the listen address and port used by Node Manager on the remote machine. The domain name and domain directory are assumed to be the root directory specified for the target server instance, myserver.

ssh -o PasswordAuthentication=no wlscontrol.sh myserver start



Th following command issues a START command to the server instance named ServerA, in the domain called ProductionDomain, located in the domains/ProductionDomain directory:

ssh -o PasswordAuthentication=no 172.11.111.11 wlscontrol.sh -d
ProductionDomain
-r ProductionDomain -s ServerA'

The ssh command must include the string:

-o PasswordAuthentication=no

This string passes the ssh PasswordAuthentication option. A value of yes causes the client to hang when it tries to read from the console.

Table 4-1 wiscontrol.sh Command Line Options

Description
Description
Specifies the Node Manager root directory.
Specifies the server name.
Specifies the domain name.
Specifies the domain directory.
Enables a server start script.
The name of the server start script.
The name of the server stop script.
Enables verbose output.
Prints the usage for wlscontrol.sh.

**Table 4-2 Shell Command Templates** 

Parameter	Description	Default
%H	The host name of the SSH server.	NodeManagerMBean.ListenAddress
%N	The Node Manager home directory.	NodeManagerMBean.NodeManagerHome
%P	The port number of the SSH server.	NodeManagerMBean.ListenPort 22
%S	The Oracle WebLogic Server name.	none
%D	The Oracle WebLogic domain name.	none
%R	The domain directory (server root).	ServerStartMBean.RootDirectory



Table 4-2 (Cont.) Shell Command Templates

Parameter	Description	Default
%C	The Node Manager script command:	none
	<ul> <li>START - Starts the server.</li> <li>KILL - Kills the server.</li> <li>STAT - Gets the server status.</li> <li>GETLOG - Retrieves the server output log.</li> <li>VERSION - The Return Node Manager version.</li> <li>Note: This command must be the last element in the command.</li> </ul>	

### ① Note

If you notice that it takes a long time to start Node Manager or an Oracle WebLogic Server instance, it might be because of low entropy on your machine.

To check entropy:

cat /proc/sys/kernel/random/entropy\_avail

Any result below 500 indicates that your system is at risk of running out of entropy. You can run rngd which replenishes random bytes to /dev/random using /dev/urandom as the source. Start rngd (as root). This command will ensure that your system does not run out of entropy:

\$ rngd -r /dev/urandom -o /dev/random -b

You can configure the process by editing <code>/etc/sysconfig/rngd</code> with the following command:

EXTRA OPTIONS="-i -r /dev/urandom -o /dev/random -b -t 60 -W 2048"

Every 60 seconds, this will add bits to the entropy pool until the size is 2048. You can change the interval and size using the -t 60 and -w 2048 parameters.

You can refer to these blogs that describe this topic in detail:

- WebLogic Server and Entrophy
- Why Does my WebLogic Server Take a Long Time to Start?

## Stopping Node Manager

To stop Node Manager, close the command shell in which it is running.



Alternatively, after having set the nodemanager.properties attribute QuitEnabled to true (the default is false), you can use WLST to connect to Node Manager and shut it down.

```
wls:/nm/mydomain> stopNodeManager()
Stopped Node Manager Process successfully
wls:/offline>
```

See stopNodeManager in the WLST Command Reference for Oracle WebLogic Server.

## Stopping Java-Based Node Manager Using Scripts

The shutdown scripts for Node Manager are installed in the  $DOMAIN\_HOME/bin$  and the  $WL\_HOME/server/bin$  directories, where  $WL\_HOME$  is the top-level installation directory for Oracle WebLogic Server.

These scripts require the  $NODEMGR\_HOME$  variable to be set before execution.  $DOMAIN\_HOME/bin/stopNodeManager.[cmd|sh]$  comes configured with the appropriate  $DOMAIN\_HOME$  and  $NODEMGR\_HOME$  values. However, before executing  $WL\_HOME/stopNodemanager[.cmd|sh]$ , you must define  $NODEMGR\_HOME$  correctly.

To stop Node Manager, use stopNodeManager.cmd on Windows systems and stopNodeManager.sh on UNIX systems.

## **Using Node Manager to Control Servers**

Oracle recommends to use the WebLogic Scripting Tool and Node Manager to start and stop the Administration Server and Managed Servers.

Use the recommended procedures to start and stop server instances. See Using WLST and Node Manager to Manage Servers in *Understanding the WebLogic Scripting Tool* and Node Manager Commands in *WLST Command Reference for Oracle WebLogic Server*.

For information about changing the credentials used for starting and stopping Oracle WebLogic Server instances, see Changing the Credentials Used for Starting a Server in *Administering Server Startup and Shutdown for Oracle WebLogic Server*.

This section includes the following topics:

## Starting the Administration Server Using Node Manager

The following general procedures are recommended for starting an Administration Server using WLST and Node Manager.

Establish startup information using the nmGenBootStartupProps command. This command generates the Node Manager property files boot.properties and startup.properties.

You must complete all of the following steps when starting the Administration Server for the first time in your domain, or if you make any configuration changes to the Administration Server.

For subsequent startups of the Administration Server, go to Step 7 to start the Administration Server directly with Node Manager.



- Start the Administration Server manually by performing one of the following actions:
  - Run the startWebLogic script from the DOMAIN\_HOME/bin directory, where DOMAIN\_HOME represents the directory in which your Oracle WebLogic Server domain is configured.
  - Invoke WLST and use the WLST startServer command to start the Administration Server.
- 2. (optional) Add any additional configuration information for the Administration Server.
- 3. Invoke WLST, if not already running.
  - On Windows, you can use a shortcut on the Start menu to set your environment variables and invoke WLST.
- 4. Connect WLST to the Administration Server instance using the connect command.
- **5.** Establish startup information using the nmGenBootStartupProps command. This command generates the Node Manager property files boot.properties and startup.properties.
- 6. Shut down the Administration Server, which also disconnects WLST.
- 7. Start Node Manager. See Starting and Stopping Node Manager.
- Invoke WLST. See Invoking WLST in Understanding the WebLogic Scripting Tool.
- 9. Connect WLST to Node Manager using the nmConnect command.
- 10. Start the Administration Server using the nmStart command.

The following example starts the Administration Server in the specified WebLogic domain using Node Manager. In this example, the prps variable stores the system property settings and is passed to the command using the props argument.

```
wls:/nm/mydomain> prps = makePropertiesObject("AdminURL=http://
listen_address;listen_port;Username=username;Password=password
;weblogic.ListenPort=8001")
wls:/nm/mydomain> nmStart("AdminServer",props=prps)
Starting server AdminServer...
Server AdminServer started successfully
wls:/nm/mydomain>
```

After the Administration Server has been started, you can use WLST to start the Managed Servers in your domain.

#### (i) Note

Starting the server instance using the nmStart command allows Node Manager to monitor the state of your Administration Server and restart it in case of failure. Node Manager can only restart server instances that were started in this way.

Using nmStart allows you to pass specific properties to a server instance, but should only be used for debugging. Server properties passed through nmStart are not preserved the next time the server instance is restarted.

## Starting Managed Servers Using WLST

The following general procedures are recommended for starting a Managed Server using WLST and Node Manager.



- Start Node Manager. See Starting and Stopping Node Manager.
- Start an Administration Server. See Starting an Administration Server with a Startup Script in Administering Server Startup and Shutdown for Oracle WebLogic Server.
- Invoke WLST and connect to an Administration Server using the connect command. See Invoke WLST.
- 4. Start the Managed Server using the WLST start command.

Using the start command causes WLST to contact the Administration Server to determine the Managed Servers startup properties. These are in turn passed to Node Manager and are used to start the Managed Server.

### Starting Managed Servers Using the Remote Console

The following general procedures are recommended for starting a Managed Server using the Oracle WebLogic Remote Console:

- 1. If you have not already done so, create a Managed Server.
- Node Manager must run on each computer that hosts Oracle WebLogic Server instances
  that you want to control with Node Manager. Configure each computer as a machine in
  Oracle WebLogic Server. See Configuring a Machine to Use Node Manager.
- 3. Assign each server instance that you will control with Node Manager to the machine on which it runs. See Assigning Server Instances to a Machine.
- 4. Start Node Manager on the computer that you want to host the Managed Server. The Oracle WebLogic Server custom installation process optionally installs and starts Node Manager as a Windows service on Windows systems. If it's not already running, you can start Node Manager manually at a command prompt or with a script.

For more information, see Start Managed Servers in the *Oracle WebLogic Remote Console Online Help*.

### Configuring a Machine to Use Node Manager

An Oracle WebLogic Server machine resource associates a particular machine with the server instances it hosts, and specifies the connection attributes for a Node Manager process on that system.

Configure a machine definition for each machine that runs a Node Manager process using the **Node Manager** page in the WebLogic Remote Console. To access this page, in the **Edit Tree**, go to **Environment**, then **Machines**, and then *myMachine*. For more information, see Configure Machines in the *Oracle WebLogic Remote Console Online Help*.

On the **Node Manager** page, configure the following properties:

- The DNS name or IP address upon which Node Manager listens in the Listen Address field.
- The port number in the Listen Port field. Note that specifying the port number is especially important if you have modified it from the default value.





The Node Manager **Listen Address** value reflects the client-side configuration. It tells the Administration Server or clients on the Administration Server how to connect to Node Manager. This attribute does not configure Node Manager.

The listen address you specify must match exactly the host name appearing in the CN component of the Node Manager SSL server digital certificate subject DN.

### Assigning Server Instances to a Machine

After configuring each computer as a machine resource, you must assign each server instance that you will control with Node Manager to the machine on which it runs.

- 1. In the WebLogic Remote Console, in the Edit Tree, go to Environment, then Servers.
- 2. Select the server instance that you want to assign to a machine.
- 3. In the **Machine** drop-down list, select the machine to which you want to assign the server instance.
- 4. Click Save.

### (i) Note

You cannot change the machine of the Administration Server using the WebLogic Remote Console. You cannot change the cluster or machine of a running server instance.

For more information, see Assign a Server Instance to a Machine in the *Oracle WebLogic Remote Console Online Help*.

### Starting Managed Servers without an Administration Server

The following general procedures are recommended for starting a Managed Server using WLST and Node Manager if you do not want to use the Administration Server to determine a Managed Server's startup properties:

- 1. Start Node Manager. See Starting and Stopping Node Manager.
- 2. Invoke WLST and connect to Node Manager using the nmConnect command.
- Start the Managed Server using the WLST nmStart command.

### (i) Note

If you use the default security providers, the first time you start a Managed Server instance, it must be able to contact the Administration Server.

To start a Managed Server in MSI mode when the Administration Server is down, you must use WLST to connect directly to Node Manager.



Using the nmStart command allows you to restart a Managed Server without the Administration Server and to specify the server startup properties you want. However, the following considerations apply:

- In order to start a server instance with nmStart, you must ensure that boot.properties and startup.properties are already defined.
- nmStart should not be used to permanently change the startup properties for a server instance. The next time a server instance is migrated or restarted from the Administration Server, these properties will not be used.
- When passing the server instance user name and password using nmStart, these values are not encrypted.

The following example starts the managed1 server in the current WebLogic domain using Node Manager:

wls:/nm/mydomain> nmStart("managed1")
Starting server managed1 ...
Server managed1 started successfully
wls:/nm/mydomain>

# **Node Manager Tutorial**

The Node Manager tutorial provides a basic example of using the default per domain Node Manager. It provides the steps to create and use Java-based Node Manager to start and stop Oracle WebLogic Server instances in a single-machine domain that hosts both an Administration Server and a Managed Server. See <a href="Default Node Manager Configuration">Default Node Manager Configuration</a>. This chapter includes the following sections:

## Creating Node Manager in a New Domain

Use the Fusion Middleware Configuration Wizard to create and configure Node Manager in a new domain.

To create an instance of per domain, Java-based Node Manager in a new domain, complete the following steps:

- 1. Start the Configuration Wizard:
  - a. From a command window, change to the <code>ORACLE\_HOME/oracle\_common/common/bin</code> directory, where <code>ORACLE\_HOME</code> represents the directory you specified as the Oracle Home when you installed Oracle WebLogic Server.
  - b. Execute the config.cmd command for Windows or the config.sh command for UNIX.
- 2. Use the Configuration Wizard to create a new, basic domain:
  - a. On the Create Domain screen, select **Create a new domain** and select a domain directory location. This is your <code>DOMAIN\_HOME</code> directory. Click **Next**.
  - On the Templates screen, accept the default Create Domain Using Product Templates. Click Next.
  - c. On the Administrator Account screen, enter the credentials of the Oracle WebLogic Server administrator. Click Next.
  - d. On the Domain Mode and JDK screen, accept the default Domain Mode and JDK. Click Next.
- 3. Create a Node Manager instance, a Managed Server, and a machine in your domain:
  - a. On the Advanced Configuration screen, select Node Manager and Topology options.
     Click Next.
  - b. On the Node Manager screen, accept Per Domain Default Location as the default Node Manager type and enter the credentials of the Node Manager administrator. Click Next.
  - **c.** On the Managed Servers screen, click **Add**. Accept the default server name, listen address, and listen port (7003). Click **Next**.
  - d. On the Clusters screen, click Next.
  - e. On the Machines screen, click Add. Accept the default machine name, Node Manager listen address, and Node Manager listen port. Click Next.
- 4. Assign the Administration Server and Managed Server to your machine:



- a. On the Assign Servers to Machines screen, select your Administration Server and Managed Server in the left column.
- b. Use the right arrow button to assign the Administration Server and Managed Server to your machine in the right column.
- c. Click Next.
- 5. Complete your domain configuration:
  - a. On the Configuration Summary screen, review your configuration and click Create.
  - **b.** When the Configuration Process screen shows that your domain has been created successfully, click **Next**.
  - c. On the Configuration Success screen, select Start Admin Server to start your domain. Click Finish.

## Starting Node Manager

To start Node Manager in a WebLogic domain environment, use the startNodeManager.cmd command for Windows and the startNodeManager.sh command for UNIX.

To start Node Manager in your domain, complete the following steps:

- **1.** From a command window, change to the <code>DOMAIN\_HOME/bin</code> directory, where <code>DOMAIN\_HOME</code> represents the directory in which your Oracle WebLogic Server domain is configured.
- 2. To set your environment, execute the setDomainEnv.cmd command for Windows or the setDomainEnv.sh command for UNIX.
- 3. Return to the DOMAIN\_HOME/bin directory.
- 4. To start Node Manager in your domain, execute the startNodeManager.cmd command for Windows or the startNodeManager.sh command for UNIX.

If Node Manager starts successfully, you will see output similar to the following:

```
<DATE TIME> <INFO> <15.1.1.0.0>
<DATE TIME> <INFO> <Secure socket listener started on port 5556, host localhost/
127.0.01>
```

#### (i) Note

If Node Manager is started by any means other than the startNodeManager script, the use of the nmRestart WLST command to restart Node Manager is not supported. If Node Manager is started from a custom script, or by the java weblogic.NodeManager command, using the nmRestart command may subsequently fail.

## Using Node Manager to Start a Managed Server

Use the WebLogic Remote Console to ensure that Node Manager is running and to start a Managed Server in your domain.

To start a Managed Server in your domain using Node Manager, complete the following steps:

- Start the WebLogic Remote Console:
  - a. Ensure that the Administration Server is running.



If you selected the **Start Admin Server** option in the Configuration Wizard, the Administration Server should be running.

If the Administration Server is not running, change to the <code>DOMAIN\_HOME/bin</code> directory and execute the <code>startWebLogic.cmd</code> command for Windows or the <code>startWebLogic.sh</code> command for UNIX.

- b. Connect to an Administration Server. For more information, see Connect to an Administration Server in the *Oracle WebLogic Remote Console Online Help*.
- 2. Ensure that Node Manager is running.
  - a. In the Remote Console, in the **Monitoring Tree**, go to **Environment**, then **Node Manager Logs**.
  - b. In the Machines table, select the name of your machine.
  - c. If Node Manager is running, Reachable will be true.

### (i) Note

To configure Node Manager settings in the WebLogic Remote Console, in the **Edit Tree**, go to **Environment**, then **Machines**. Select *myMachine*, then **Node Manager**.

- In the Remote Console, in the Monitoring Tree, go to Environment, then Servers.
- 4. In the Servers table, select the checkbox next to your Managed Server.
- 5. Click Start.

A message appears confirming that Node Manager will start the selected server instance. Refresh the WebLogic Remote Console page, and the Managed Server state changes from Shutdown to Running.

Your Managed Server is now running in your domain.