

Oracle® Fusion Middleware

Administering Node Manager for Oracle WebLogic Server

12c (12.1.2)

E28093-04

April 2014

This document describes how to configure and use Node Manager to control and manage servers within a WebLogic Server environment.

Oracle Fusion Middleware Administering Node Manager for Oracle WebLogic Server, 12c (12.1.2)

E28093-04

Copyright © 2007, 2014, Oracle and/or its affiliates. All rights reserved.

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 or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. 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 failsafe, 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 and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon 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, Opteron, the AMD logo, and the AMD Opteron 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 on 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. 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.

Contents

Preface	vii
Documentation Accessibility	vii
Conventions	vii
1 Introduction and Roadmap	
1.1 Document Scope and Audience.....	1-1
1.2 Guide to This Document.....	1-1
1.3 Related Documentation.....	1-2
1.4 New and Changed Features in This Release.....	1-2
2 Node Manager Overview	
2.1 Introduction	2-1
2.2 Node Manager Versions	2-2
2.2.1 Java-based Node Manager	2-2
2.2.2 Script-based Node Manager.....	2-2
2.2.3 Determining Which Node Manager Version to Use	2-2
2.3 Accessing Node Manager	2-3
2.4 What You Can Do with Node Manager	2-3
2.4.1 Start, Shut Down, and Restart an Administration Server.....	2-3
2.4.2 Start, Shut Down, Suspend, and Restart Managed Servers.....	2-4
2.4.3 Restart Administration and Managed Servers	2-4
2.4.4 Monitor Servers and View Log Data	2-5
2.5 How Node Manager Works in the WebLogic Server Environment.....	2-5
2.5.1 Diagram of Node Manager and Servers.....	2-5
2.5.2 How Node Manager Starts an Administration Server.....	2-6
2.5.3 How Node Manager Starts a Managed Server.....	2-7
2.5.4 How Node Manager Restarts an Administration Server	2-8
2.5.5 How Node Manager Restarts a Managed Server	2-8
2.5.6 How Node Manager Shuts Down a Server Instance.....	2-10
2.6 Node Manager and System Crash Recovery	2-10
2.7 Node Manager Configuration and Log Files.....	2-11
2.7.1 Configuration Files	2-12
2.7.1.1 nodemanager.properties	2-12
2.7.1.2 nodemanager.domains	2-12
2.7.1.3 nm_password.properties.....	2-13

2.7.1.4	boot.properties	2-13
2.7.1.5	startup.properties	2-13
2.7.1.6	server_name.addr	2-13
2.7.1.7	server_name.lck	2-13
2.7.1.8	server_name.pid.....	2-13
2.7.1.9	server_name.state	2-13
2.7.2	Log Files	2-14
2.7.2.1	nodemanager.log.....	2-14
2.7.2.2	server_name.out	2-14
2.7.2.3	Log File Rotation.....	2-15
2.7.2.4	WebLogic Server Log Files.....	2-16

3 General Node Manager Configuration

3.1	Overview of Node Manager Configuration.....	3-1
3.1.1	Controlling and Configuring Node Manager Using WLST	3-1
3.2	Step 1: Configure Your Computer to Run Node Manager	3-2
3.3	Step 2: Specify Node Manager User Name and Password.....	3-2
3.4	Step 3: Configure a Machine to Use Node Manager.....	3-3
3.5	Step 4: Configuring nodemanager.domains File.....	3-4
3.6	Step 5: Configuring Remote Startup Arguments	3-5
3.7	Step 6: Setting Server Startup Properties.....	3-5
3.7.1	startup.properties	3-5
3.7.2	Setting Startup Properties Using WLST	3-5
3.7.3	Server Startup Properties.....	3-6
3.8	Step 7: Define the Administration Server Address.....	3-6
3.9	Step 8: (Optional) Set Node Manager Environment Variables	3-7

4 Configuring Java Node Manager

4.1	Default Node Manager Configuration.....	4-1
4.2	Configuring Node Manager Using WLST Offline	4-2
4.3	Configuring Java-based Node Manager Security	4-3
4.3.1	Remote Server Start Security for Java-based Node Manager.....	4-3
4.4	Reviewing nodemanager.properties.....	4-4
4.5	Configuring Node Manager to Use Start and Stop Scripts	4-11
4.5.1	Script Location.....	4-11
4.5.2	Best Practices When Using Start and Stop Scripts	4-11
4.5.3	Using Start Scripts	4-11
4.5.4	Using Stop Scripts.....	4-12
4.6	Using SSL With Java-based Node Manager.....	4-12
4.7	Configuring Node Manager on Multiple Machines	4-13
4.8	Configuring Node Manager as an xinetd Service	4-13

5 Configuring Script Node Manager

5.1	Overview	5-1
5.2	Step 1: Create User Accounts	5-2
5.3	Step 2: Configure Node Manager Security.....	5-2

5.4	Step 3: Install WebLogic Server	5-4
5.5	Step 4: Create a WebLogic Domain	5-4
5.6	Step 5: Start the Administration Server	5-4
5.7	Step 6: Configure Node Manager on the Managed Servers	5-5
5.8	Step 7: Test Node Manager Setup and Configuration by Starting Managed Servers.....	5-5
5.9	Step 8: Configure UNIX Machines	5-7
5.10	Step 9: Assign Servers to Machines	5-7
5.11	Step 10: Start Managed Servers.....	5-7
5.12	Overriding the Default SSH Port.....	5-8
5.13	Configuring Security for WebLogic Server Scripts.....	5-8
5.14	Configuring Remote Server Start Security for Script-based Node Manager	5-8
5.15	Generating and Distributing Key Value Pairs	5-9
5.15.1	Shared Key Value Pair	5-9
5.15.2	Individual Key Value Pairs	5-10

6 Using Node Manager

6.1	Starting and Stopping Node Manager	6-1
6.1.1	Running Node Manager as a Startup Service.....	6-1
6.1.2	Starting Java-based Node Manager Using Scripts.....	6-2
6.1.2.1	Command Syntax for Starting Java-based Node Manager	6-3
6.1.3	Running Script-based Node Manager	6-4
6.1.4	Stopping Node Manager	6-5
6.2	Using Node Manager to Control Servers	6-6
6.2.1	Starting the Administration Server Using Node Manager	6-6
6.2.2	Starting Managed Servers.....	6-6
6.2.3	Starting Managed Servers without an Administration Server	6-7

Preface

This preface describes the document accessibility features and conventions used in this guide—*Administering Node Manager for Oracle WebLogic Server*.

Documentation Accessibility

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

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

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.
<i>italic</i>	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.

Introduction and Roadmap

The following sections describe the contents and organization of this guide—*Administering Node Manager for Oracle WebLogic Server*.

- [Section 1.1, "Document Scope and Audience"](#)
- [Section 1.2, "Guide to This Document"](#)
- [Section 1.3, "Related Documentation"](#)
- [Section 1.4, "New and Changed Features in This Release"](#)

1.1 Document Scope and Audience

This document describes how to configure and use Node Manager to control and manage servers within a WebLogic Server environment.

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 Java Platform, Enterprise Edition (Java EE) and Web technologies, object-oriented programming techniques, and the Java programming language.

1.2 Guide to This Document

The document is organized as follows:

- This chapter, [Chapter 1, "Introduction and Roadmap"](#) describes the scope of the guide and lists related documentation.
- [Chapter 2, "Node Manager Overview"](#) provides a general description of Node Manager and describes how it works within a WebLogic domain. It also provides detailed descriptions of the configuration and log files used by Node Manager.
- [Chapter 3, "General Node Manager Configuration"](#) describes configuration procedures that are applicable to the both the Java and scripted versions of Node Manager.
- [Chapter 4, "Configuring Java Node Manager"](#) describes the configuration procedures for the Java version of Node Manager.
- [Chapter 5, "Configuring Script Node Manager"](#) describes the configuration procedures for the scripted version of Node Manager.

- [Chapter 6, "Using Node Manager"](#) provides procedures for starting Node Manager and servers. This chapter also includes recommendations for starting servers to take advantage of WebLogic Server's failover and migration features.

1.3 Related Documentation

- *Creating WebLogic Domains Using the Configuration Wizard*
- *Understanding Domain Configuration for Oracle WebLogic Server*
- *Oracle WebLogic Server Administration Console Online Help*

1.4 New and Changed Features in This Release

In this release, the Java-based version of Node Manager has a simplified, out-of-the-box configuration. For each WebLogic domain you create, a domain-specific Node Manager configuration is created by default, complete with security credentials, properties file, domain registration, and start scripts, including pre-configured scripts to install and uninstall Node Manager as a Windows service. For more information, see [Section 4.1, "Default Node Manager Configuration"](#).

For a comprehensive listing of the new WebLogic Server features introduced in this release, see *What's New in Oracle WebLogic Server*.

Node Manager Overview

This chapter provides an introduction to Node Manager, a WebLogic Server utility. It also describes how Node Manager controls Administration Servers and Managed Servers.

This chapter includes the following sections:

- [Section 2.1, "Introduction"](#)
- [Section 2.2, "Node Manager Versions"](#)
- [Section 2.3, "Accessing Node Manager"](#)
- [Section 2.4, "What You Can Do with Node Manager"](#)
- [Section 2.5, "How Node Manager Works in the WebLogic Server Environment"](#)
- [Section 2.6, "Node Manager and System Crash Recovery"](#)
- [Section 2.7, "Node Manager Configuration and Log Files"](#)

2.1 Introduction

Server instances in a WebLogic Server production environment are often distributed across multiple domains, machines, and geographic locations. Node Manager is a WebLogic Server utility that enables you to start, shut down, and restart Administration Server and Managed Server instances from a remote location. Although Node Manager is optional, it is recommended if your WebLogic Server environment hosts applications with high availability requirements.

In previous releases, a Node Manager process was not associated with a specific WebLogic domain but 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, *per host* Node Manager. In this release of WebLogic Server, the Java version 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. Now you can run each domain-specific Java-based Node Manager with a different configuration. For more information, see [Section 4.1, "Default Node Manager Configuration."](#)

Node Manager must run on each computer that hosts WebLogic Server instances—whether Administration Server or Managed Server—that you want to control with Node Manager.

2.2 Node Manager Versions

WebLogic Server provides two versions of Node Manager, Java-based and script-based, with similar functionality. However, each version has different configuration and security considerations.

2.2.1 Java-based Node Manager

Java-based Node Manager runs within a Java Virtual Machine (JVM) process. It is recommended 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.

Oracle provides native Node Manager libraries for Windows, Solaris, Linux on Intel, Linux on Z-Series, and AIX operating systems.

Note: Node Manager is not supported on Open VMS, OS/390, AS400, UnixWare, or Tru64 UNIX.

This version of Node Manager determines its configuration from the `nodemanager.properties` file. See [Section 4.4, "Reviewing nodemanager.properties"](#).

Java-based Node Manager provides more security than the script-based version. See [Section 4.3, "Configuring Java-based Node Manager Security"](#).

2.2.2 Script-based Node Manager

For UNIX and Linux systems, WebLogic Server provides a script-based version of Node Manager. This script is based on UNIX shell scripts, but uses SSH for increased security. SSH uses user-id based security.

For information on configuring the script version of Node Manager, see [Section 5, "Configuring Script Node Manager."](#)

This version does not provide as much security as the Java-based version. However, the advantage of the script-based Node Manager is that it can remotely manage servers 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: It is recommended that you run script-based Node Manager as an operating system service, which allows it to restart automatically when the system is rebooted.

2.2.3 Determining Which Node Manager Version to Use

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

- If you are installing WebLogic Server on a Windows system, you must use the Java version of Node Manager. The scripted version of Node Manager is not supported on Windows.
- In order to use consensus leasing, you may see faster performance when using the Java version of Node Manager.

- The script-based Node Manager requires a much simpler security configuration than the Java version. RSH and SSH are generally easier to configure than SSL which is the security method used by the Java version of Node Manager. The script version of Node Manager also requires a smaller footprint than the Java version.
- The Java version of Node Manager can be used in conjunction with `inetd` on supported UNIX systems. `inetd` allows Node Manager to be automatically restarted upon receiving a request on the configured port.

2.3 Accessing Node Manager

A Node Manager client can be local or remote to Node Managers with which it communicates. You access either version of Node Manager—the Java version or the script-based (SSH) version—from the following clients: (In addition, an SSH client in the form of a shell command template is provided for use with the script-based Node Manager.)

- Administration Server
 - Administration Console, from the **Environments > Machines > Configuration > Node Manager** page.

For example, you can create JMX utilities that communicate with the Administration Server and perform operations on the `ServerLifecycleRuntimeMBean` which in turn uses Node Manager internally to perform operations. For more information about JMX, see *Developing Custom Management Utilities Using JMX for Oracle WebLogic Server*.

- 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. Starting the Administration Server is the main purpose of the standalone client. However, you can also use WLST to:
 - Stop a server instance that was started by Node Manager.
 - Start a Managed Server.
 - Access the contents of a Node Manager log file.
 - Obtain server status for a server that was started with Node Manager.
 - Retrieve the contents of the server output log.

For more information on using WLST and Node Manager to control servers, see [Section 6.2, "Using Node Manager to Control Servers"](#).

2.4 What You Can Do with Node Manager

The following sections describe basic Node Manager functionality.

2.4.1 Start, Shut Down, and Restart an Administration Server

Using the WebLogic Scripting Tool (or SSH client for script-based Node Manager only), you connect to a Node Manager process on the machine that hosts the Administration Server and issue commands to start, shut down, or restart an Administration Server. The relationship of an Administration Server to 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 Administration 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, the Managed Server contacts the Administration Server to obtain outstanding configuration updates.

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

From the WebLogic Server Scripting Tool (WLST) command line or scripts, you can issue commands to Node Manager to start, shut down, suspend, and restart Managed Server instances and clusters.

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 is enabled by default.

Note: 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.

Note: Node Manager uses the same command arguments that you supply 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*.

2.4.3 Restart Administration and Managed Servers

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

Note: Node Manager can only restart a server that was started using 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 a Node Manager `startup.properties` file.

If Node Manager fails or is explicitly shut down, 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.

Note: It is advisable to run Node Manager as an operating system service, so that it restarts automatically if its host machine is restarted.

2.4.4 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. You can view these log files, as well as log files for a server instance using the Administration Console or WLST commands. For more information, see [Section 2.7.2, "Log Files"](#).

2.5 How Node Manager Works in the WebLogic Server Environment

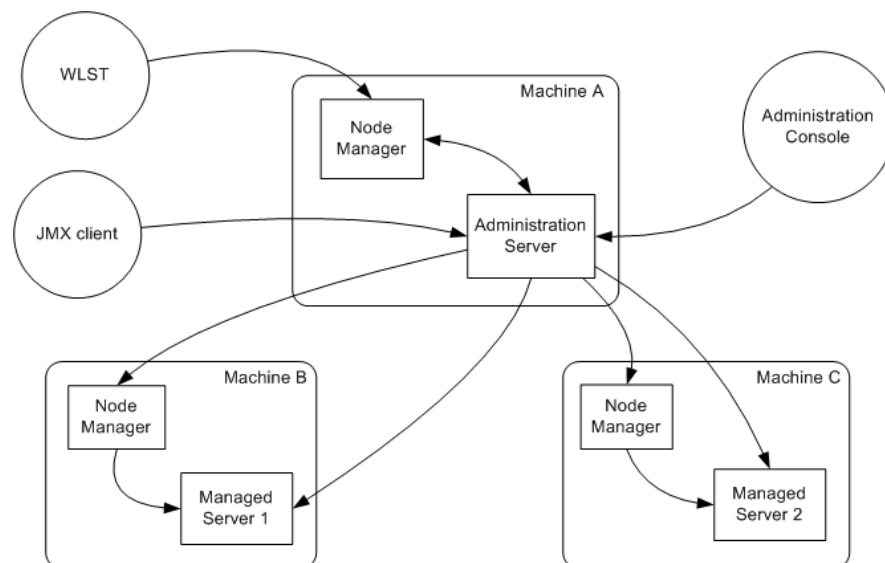
The following sections provide a "big picture" diagram of Node Manager's role in the WebLogic Server environment, as well as illustrations and descriptions of the processes Node Manager uses to communicate with servers:

- [Section 2.5.1, "Diagram of Node Manager and Servers"](#)
- [Section 2.5.2, "How Node Manager Starts an Administration Server"](#)
- [Section 2.5.3, "How Node Manager Starts a Managed Server"](#)
- [Section 2.5.4, "How Node Manager Restarts an Administration Server"](#)
- [Section 2.5.5, "How Node Manager Restarts a Managed Server"](#)
- [Section 2.5.6, "How Node Manager Shuts Down a Server Instance"](#)

2.5.1 Diagram of Node Manager and Servers

[Figure 2-1](#) illustrates the relationship between Node Manager, its clients, and the server instances it controls.

Figure 2-1 Node Manager in the WebLogic Server Environment



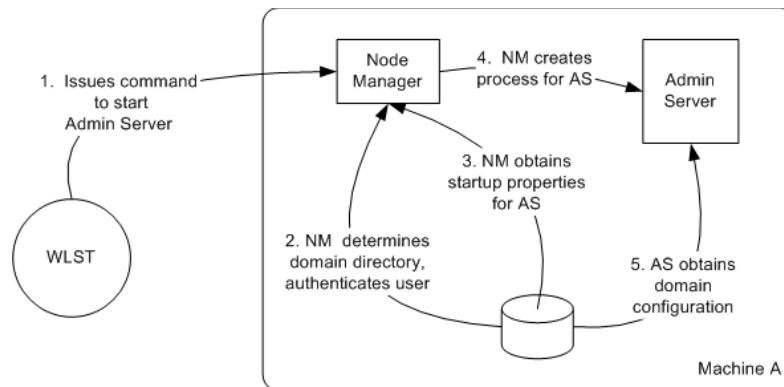
2.5.2 How Node Manager Starts an Administration Server

Figure 2–2 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 2–2 Starting an Administration Server



1. An authorized user issues the WLST offline command `nmConnect` to connect to a Node Manager process on the machine that hosts the Administration Server. (If a Node Manager instance is the SSH version, the user can connect using the SSH client). The `nmConnect` command provides a Node Manager user name and password that are used to authenticate the user with Node Manager.

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

```
prps = makePropertiesObject("AdminURL=http://listen_address:listen_
port;Username=username;Password=password")
nmStart("AdminServer", prps)
```

Note: If the user has previously connected to a Node Manager, a `boot.properties` file exists, and the user does not have to supply user name and password.

The `nmStart` command identifies the domain and 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.
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 is running, you can update the user credentials and startup properties using the WLST online command, `nmGenBootStartupProps`.

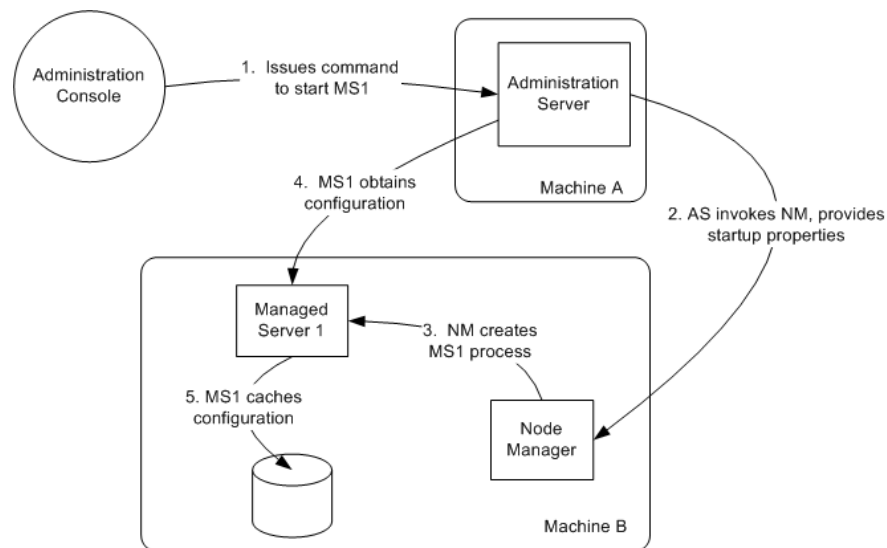
Alternatively, when the Administration Server and Node Manager are running, you can update the user credentials and startup properties in the Administration Console, on the *AdminServer* > **Configuration** > **Server Start** page. The Administration Server then pushes the updates to the running Node Manager and Node Manager writes the information to the disk.

2.5.3 How Node Manager Starts a Managed Server

Figure 2-3 illustrates the process of starting a Managed Server with Node Manager.

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

Figure 2-3 Starting a Managed Server



1. From the Administration Console, the user issues a start command for Managed Server 1.

Note: A standalone client can also issue a start command for a Managed Server.

2. The Administration Server issues a start command for Managed Server 1 to Node Manager on the Machine B, providing the remote start properties configured for Managed Server 1. For information about the arguments and how to specify them, see [Section 3.6, "Step 5: Configuring Remote Startup Arguments"](#).
3. Node Manager starts Managed Server 1.

Node Manager starts the Managed Server using the same root directory where a Node Manager process is running. To run the Managed Server in a different directory, set the `Root Directory` attribute in the **Server** > **Configuration** > **Server Start** Administration Console page.

4. 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.

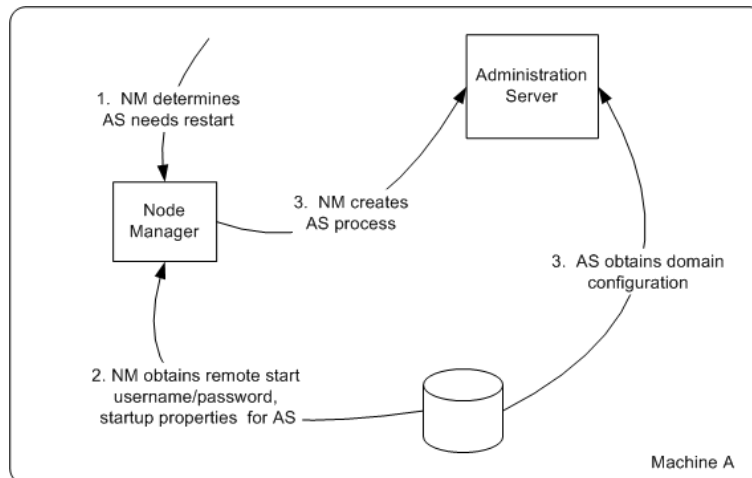
2.5.4 How Node Manager Restarts an Administration Server

Figure 2–4 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`.

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 failed when Node Manager was not running.

Figure 2–4 Restarting an Administration Server



1. Node Manager determines from the Administration Server process exit code that it requires restart.
2. 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.
3. Node Manager starts the Administration Server.
4. The Administration Server reads its configuration data and starts up.

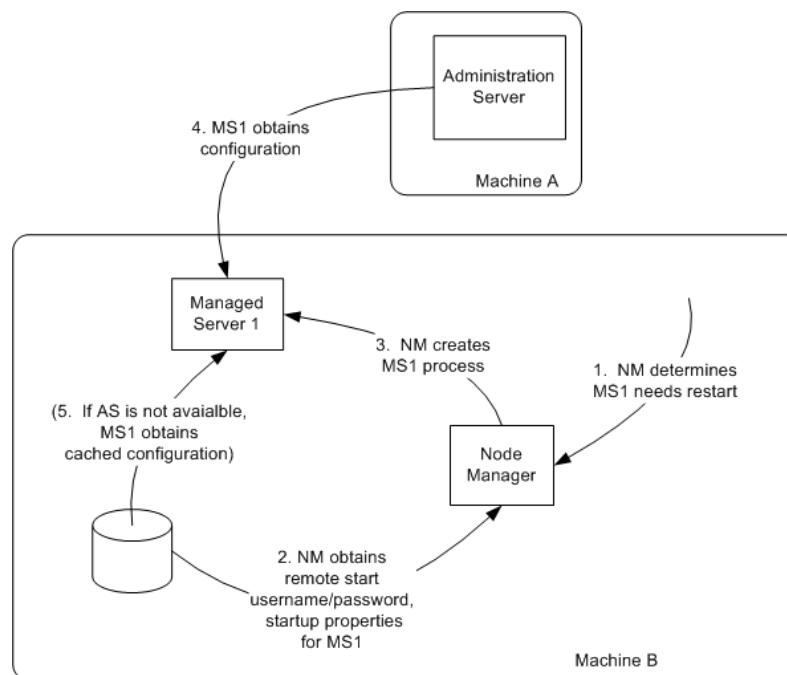
2.5.5 How Node Manager Restarts a Managed Server

Figure 2–5 illustrates 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 failed when Node Manager was not running.

Figure 2-5 Restarting a Managed Server



1. Node Manager determines from Managed Server 1's last known state that it requires restarting.
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 instances fails before attempting to restart it.

4. Managed Server 1 attempts to contact the Administration Server to check for updates to its configuration data. If it contacts the Administration Server and obtains updated configuration data, it updates its local cache of the `config` directory.

5. If Managed Server 1 fails to contact the Administration Server, and if Managed Server Independence mode (MSI) is enabled, Managed Server 1 uses its locally cached configuration data.

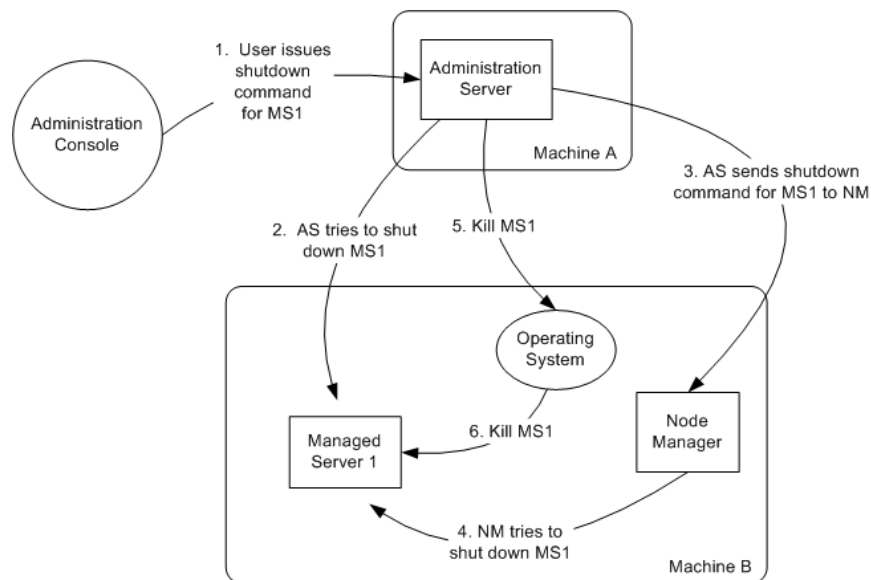
Note: Managed Server Independence mode is enabled by default.

2.5.6 How Node Manager Shuts Down a Server Instance

Figure 2–6 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 try alternative strategies to successfully initiate the shutdown.

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

Figure 2–6 Shutting Down a Server Instance Under Node Manager Control



1. Through the Administration Console, an authorized user issues a shutdown command for Managed Server 1.
2. The Administration Server issues the shutdown command directly to Managed Server 1. If it successfully contacts Managed Server 1, Managed Server 1 performs the shutdown sequence described in "Graceful Shutdown" in *Administering Server Startup and Shutdown for Oracle WebLogic Server*.
3. If, in the previous step, the Administration Server failed to contact Managed Server 1, it issues a shutdown command for Managed Server 1 to Node Manager on Machine B.
4. Node Manager issues a request to the operating system to kill Managed Server 1.
5. The operating system ends the Managed Server 1 process.

2.6 Node Manager and System Crash Recovery

To ensure that Node Manager properly restarts servers after a system crash, 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 servers after a system crash. The property is not enabled by default.

Note: 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 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 Administration 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 a WebLogic Server process is created. This lock file contains the process identifier for 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.

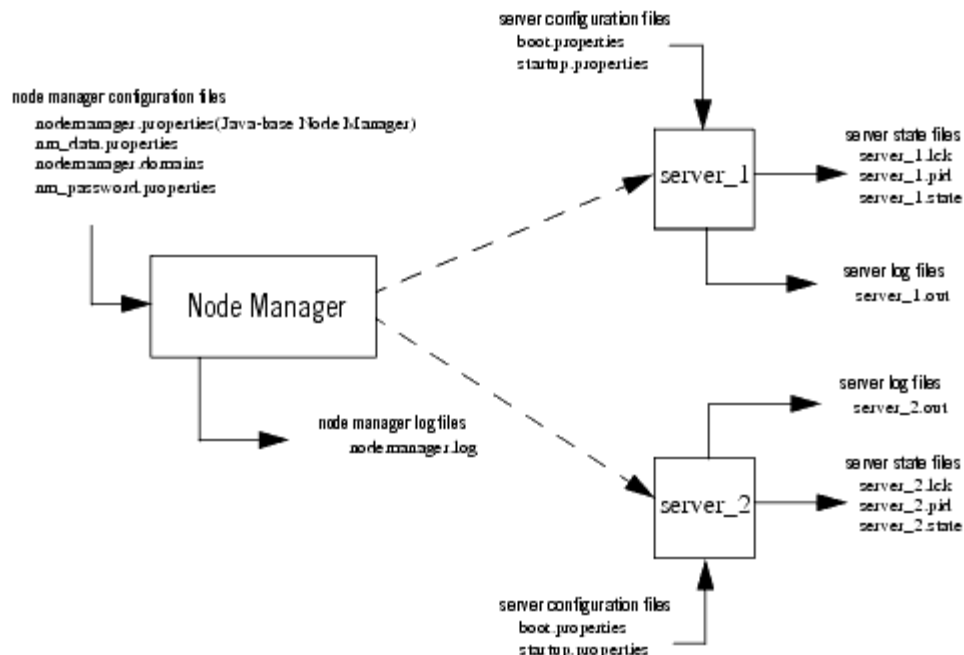
If the process is running, Node Manager performs an additional check to access the management servlet running in the process to verify that the process corresponding to the process ID is a WebLogic Server instance.

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

2.7 Node Manager Configuration and Log Files

In managing multiple servers, Node Manager uses multiple configuration files and outputs log files to multiple directories, as shown in [Figure 2-7](#).

Figure 2–7 Node Manager Configuration and Logging Environment



The following sections describe Node Manager configuration and log files:

- [Section 2.7.1, "Configuration Files"](#)
- [Section 2.7.2, "Log Files"](#)

2.7.1 Configuration Files

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

2.7.1.1 nodemanager.properties

This is the configuration file used by the Java-based version of Node Manager. See [Section 4.4, "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 Oracle Home when you installed WebLogic Server.

2.7.1.2 nodemanager.domains

This file contains mappings between the names of domains managed by Node Manager and their corresponding directories. See [Section 3.5, "Step 4: 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 WebLogic Server, for example, `ORACLE_HOME/wlserver`.

2.7.1.3 nm_password.properties

This file stores the Node Manager user name and password. See [Section 3.3, "Step 2: Specify Node Manager User Name and Password"](#).

This file is located in *DOMAIN_HOME/config/nodemanager*, where *DOMAIN_HOME* is the location of your WebLogic domain, typically, *ORACLE_HOME\user_projects\domains\domain_name*.

2.7.1.4 boot.properties

Node Manager uses this file to specify user credentials when starting a server. See [Section 3, "General Node Manager Configuration."](#)

This file is located in *DOMAIN_HOME/servers/server_name/data/nodemanager*.

2.7.1.5 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. Node Manager automatically creates this file by using properties passed to Node Manager when the Administration Server was last used to start the server. This allows a Node Manager client or startup scripts to restart a Managed Server using the same properties last used by the Administration Server.

For more information on *startup.properties*, see [Section 3.7, "Step 6: 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*.

2.7.1.6 server_name.addr

server_name.addr stores the IP address added when a server 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.

This file is located in *DOMAIN_HOME/servers/server_name/data/nodemanager*.

2.7.1.7 server_name.lck

server_name.lck is generated by each server and contains an internally used lock ID.

This file is located in *DOMAIN_HOME/servers/server_name/data/nodemanager*.

2.7.1.8 server_name.pid

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

This file is located in *DOMAIN_HOME/servers/server_name/data/nodemanager*.

2.7.1.9 server_name.state

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

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

This file is located in *DOMAIN_HOME/servers/server_name/data/nodemanager*.

2.7.2 Log Files

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

Table 2–1 Node Manager Log File Locations

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

2.7.2.1 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 *ORACLE_HOME\user_projects\domains\domain_name\nodemanager*.

Log output is appended to the current *nodemanager.log*. 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:

- Selecting **Machines > Monitoring > Node Manager Log** page in the Administration Console
- Using the WLST `nmLog` command

2.7.2.2 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.

Note: You cannot limit the size of the log files Node Manager creates. Logging to `stdout` is disabled by default.

This file is located in *DOMAIN_HOME/servers/server_name/logs*, where *server_name* is the name of the server instance.

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

```
server_name.out
```

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

- Selecting **Diagnostics > Log Files**.
- Using the WLST `nmServerLog` command.

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

2.7.2.3 Log File Rotation

You can configure Node Manager log file rotation when `NativeVersionEnabled=false`. However, see the limitations listed in [Table 4-1, "Node Manager Properties"](#).

Table 2-2 Node Manager Log File Rotation Properties

Property	Description	Default
<code>FileCount</code>	The maximum number of log files that the server creates when it rotates the log. This number does not include the file that the server uses to store current messages. Requires that you enable <code>NumberOfFilesLimited</code> .	7
<code>FileMinSize</code>	The size (1-2097150 KB) that triggers the server to move log messages to a separate file. The default is 500 KB. After the log file reaches the specified minimum size, the next time the server checks the file size, it will rename the current log file as <code>SERVER_NAME.lognnnnn</code> and create a new one to store subsequent messages. Requires that you specify a <code>RotationType</code> of <code>SIZE</code> .	500
<code>RotationType</code>	Specifies the criteria for moving old log messages to a separate file. <ul style="list-style-type: none"> ■ <code>NONE</code>: Messages accumulate in a single file. You must erase the contents of the file when the size is too large. Note that WebLogic Server sets a threshold size limit of 500 MB before it forces a hard rotation to prevent excessive log file growth. ■ <code>SIZE</code>: When the log file reaches the size that you specify in <code>FileMinSize</code>, the server renames the file as <code>SERVER_NAME.lognnnnn</code>. ■ <code>TIME</code>: At each time interval that you specify in <code>FileTimeSpan</code>, the server renames the file as <code>SERVER_NAME.lognnnnn</code>. 	<code>SIZE</code>
<code>FileTimeSpan</code>	The interval (in hours) at which the server saves old log messages to another file. Requires that you specify a <code>RotationType</code> of <code>TIME</code> .	24
<code>FileTimeSpanFactor</code>	Allows log rotation to be tested at a different frequency.	360000
<code>RotationTime</code>	Determines the start time (hour and minute) for a time-based rotation sequence. At the time that this value specifies, the server renames the current log file. Thereafter, the server renames the log file at an interval that you specify in <code>FileTimeSpan</code> . Note that 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: <code>H:mm</code> , where <code>H</code> is hour in day (0-23) and <code>mm</code> is the minute in hour.	00:00
<code>NumberOfFilesLimited</code>	Indicates whether to limit the number of log files that this server instance creates to store old messages. Requires that you specify a <code>RotationType</code> of <code>SIZE</code> or <code>TIME</code> . After the server 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 creates new files indefinitely and you must clean up these files as you require.	<code>true</code>

2.7.2.4 WebLogic Server Log Files

A server instance under Node Manager control has its own log file, in addition to the log file created by Node Manager.

You can view the log file for a server instance by selecting **Diagnostics > Log Files** selecting the server log file, and clicking **View**.

General Node Manager Configuration

This chapter describes a Node Manager configuration process and the general procedures that are applicable to both the Java and script-based version of Node Manager.

This chapter includes the following sections:

- [Section 3.1, "Overview of Node Manager Configuration"](#)
- [Section 3.2, "Step 1: Configure Your Computer to Run Node Manager"](#)
- [Section 3.3, "Step 2: Specify Node Manager User Name and Password"](#)
- [Section 3.4, "Step 3: Configure a Machine to Use Node Manager"](#)
- [Section 3.5, "Step 4: Configuring nodemanager.domains File"](#)
- [Section 3.6, "Step 5: Configuring Remote Startup Arguments"](#)
- [Section 3.7, "Step 6: Setting Server Startup Properties"](#)
- [Section 3.8, "Step 7: Define the Administration Server Address"](#)
- [Section 3.9, "Step 8: \(Optional\) Set Node Manager Environment Variables"](#)

3.1 Overview of Node Manager Configuration

This section describes general Node Manager configuration steps that apply to the Java and script version of Node Manager. After you have performed the general Node Manager configuration, you should perform the configuration procedures outlined in [Chapter 4, "Configuring Java Node Manager"](#) or [Chapter 5, "Configuring Script Node Manager"](#) depending on which version of Node Manager you are using.

In this release, the Java-based version of Node Manager has a simplified, out-of-the-box configuration. For more information, see [Section 4.1, "Default Node Manager Configuration."](#)

3.1.1 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 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. For more information on WLST Node Manager commands, see "WLST Command and Variable Reference" in *WLST Command Reference for WebLogic Server*.

3.2 Step 1: Configure Your Computer to Run Node Manager

Node Manager must run on each computer that hosts a 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 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. For more information, see [Section 6.1.1, "Running Node Manager as a Startup Service"](#).

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 running on your machine to bind to a UNIX group ID (GID) after it finishes all privileged startup actions.

3.3 Step 2: Specify Node Manager User Name and Password

The `nm_password.properties` file contains the 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 `DOMAIN_HOME/config/nodemanager`, where `DOMAIN_HOME` is the location of your WebLogic domain, typically, `ORACLE_HOME\user_projects\domains\domain_name`.

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, in order 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 Administration 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.
2. Using the Administration Console, update the Node Manager credentials using the Advanced options under `domain_name > Security > General`.
3. Invoke WLST and connect to an Administration Server using the `connect` command. See "Using the WebLogic Scripting Tool" in *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: You must run `nmEnroll` on each machine that is running a Managed Server. Additionally, you should run `nmEnroll` for each domain directory on each machine.

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 Administration 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.

3.4 Step 3: Configure a Machine to Use Node Manager

A 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 **Environment > Machines > *machine_name* > Node Manager** page in the Administration Console. Enter the following values:

1. The DNS name or IP address upon which Node Manager listens in the `Listen Address` field.
2. 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.

Note: 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.

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

1. In the Administration Console, select **Environment > Servers > *server_name* > Configuration > General**.
2. In the **Machine** field, select the machine to which you want to assign the server.

Note: You cannot change the machine of the Administration Server using the Administration Console. You cannot change the cluster or machine of a running server.

3.5 Step 4: 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.

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 only execute commands 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`. For the script-based Node Manager, this file's default location is `WL_HOME/common/nodemanager`, where `WL_HOME` is the location in which you installed WebLogic Server, for example, `ORACLE_HOME/wlserver`.

If you created your domain with the Configuration Wizard, the `nodemanager.domains` file was created for you. However, in this release of WebLogic Server, if you are using the script version 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. Alternatively, you can register WebLogic domains with the script-based Node Manager using the WLST command, `nmEnroll`.

If instead of a per domain Node Manager, you want to configure a per host Node Manager, you must manually create or copy a `nodemanager.domains` file under `ORACLE_HOME\oracle_common\common\nodemanager`, the per host `NodeManagerHome` location. For more information, see [Section 4.1, "Default Node Manager Configuration."](#)

If necessary, you can manually edit `nodemanager.domains` to add domains.

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

Example 3-1 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
```

3.6 Step 5: Configuring Remote Startup Arguments

In the Administration Console, on the **Server > Configuration > Server Start** page for the Managed Server, specify the startup arguments that Node Manager will use to start a Managed Server. If you do not specify startup arguments for a Managed Server, Node Manager uses its own properties as defaults to start the Managed Server. For more information, see [Section 4.4, "Reviewing nodemanager.properties."](#) 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 will run Node Manager as a Windows service, as described in [Section 6.1.1, "Running Node Manager as a Startup Service"](#), you must configure the `-Xrs` JVM property for each Managed Server that will be under 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:

1. A reboot causes a running Managed Server to be killed before Node Manager and Administration Server operating system services are shut down.
2. 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.

3.7 Step 6: Setting Server Startup Properties

You can use Node Manager to set the startup properties for a server. 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.

3.7.1 startup.properties

Node Manager uses the `startup.properties` file to determine the startup configuration when starting a server. This file is defined for each server instance and is located in `DOMAIN_HOME/servers/server_name/data/nodemanager/startup.properties`.

The contents of `startup.properties` are derived from the Server MBean, or the Cluster MBean if the server is part of a cluster. For more information, see the *MBean Reference for Oracle WebLogic Server*.

3.7.2 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.

3.7.3 Server Startup Properties

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

Table 3–1 Server Startup Properties

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

3.8 Step 7: Define the Administration Server Address

Make sure 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 **Servers > Configuration > General** page in the Administration Console.

3.9 Step 8: (Optional) Set 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 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 [Section 4.4, "Reviewing nodemanager.properties."](#)

- Set the values when executing the `startNodeManager` script.

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 `llama` and port `7777`:

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

Enter this command to start Node Manager on host `llama`:

```
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 3–2 Node Manager Environment Variables

Environment Variable	Description
<code>JAVA_HOME</code>	JDK root directory used by Node Manager. For example: <pre>set JAVA_HOME=c:\jdk1.7.0_06</pre> <p>Node Manager has the same JDK version requirements as WebLogic Server.</p>
<code>WL_HOME</code>	WebLogic Server installation directory. For example: <pre>set WL_HOME=c:\Oracle\Middleware\Oracle_Home\wlserver</pre>

Table 3–2 (Cont.) Node Manager Environment Variables

Environment Variable	Description
PATH	Must include the WebLogic Server bin directory and path to your Java executable. For example: <pre>set PATH=%WL_HOME%\server\bin;%JAVA_HOME%\bin;%PATH%</pre>
LD_LIBRARY_PATH (UNIX and Linux)	For Solaris systems, you must include the path to the native Node Manager libraries. Solaris example: <pre>LD_LIBRARY_PATH:\$WL_HOME/server/lib/solaris:\$WL_HOME/server/lib/solaris/oci816_8</pre> Linux example: <pre>LD_LIBRARY_PATH:\$WL_HOME/server/native/linux:\$WL_HOME/server/native/linux/i686</pre> Note: Linux can be i686, ia64, or x86_64 architecture. The path would change to correspond with the appropriate architecture.
CLASSPATH	You can set the Node Manager CLASSPATH either as an option on the <code>java</code> 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>

Configuring Java Node Manager

This chapter describes how to configure the Java version of Node Manager.

This chapter includes the following sections:

- [Section 4.1, "Default Node Manager Configuration"](#)
- [Section 4.3, "Configuring Java-based Node Manager Security"](#)
- [Section 4.4, "Reviewing nodemanager.properties"](#)
- [Section 4.5, "Configuring Node Manager to Use Start and Stop Scripts"](#)
- [Section 4.6, "Using SSL With Java-based Node Manager"](#)
- [Section 4.7, "Configuring Node Manager on Multiple Machines"](#)
- [Section 4.8, "Configuring Node Manager as an xinetd Service"](#)

4.1 Default Node Manager Configuration

In this release, WebLogic Server is moving to a per domain model for Node Manager. For each domain you create, a domain-specific version of Node Manager is created by default. Using the security credentials supplied for the Administration Server, `nm_password.properties` is created in `DOMAIN_HOME\config\nodemanager`, where `DOMAIN_HOME` is typically located at `ORACLE_HOME\user_projects\domains\domain_name`. The `nodemanager.properties` and `nodemanager.domains` files are created for you under `DOMAIN_HOME\nodemanager` and domain-specific scripts to start, stop, install and uninstall Node Manager as a Windows service, are located under `DOMAIN_HOME\bin`. With the default Node Manager configuration, you cannot edit the `NodeManagerHome` location, `DOMAIN_HOME\nodemanager`.

Using the Configuration Wizard or WLST offline, you can select a `PerDomain` or `CustomLocation` Java-based Node Manager configuration. The `PerDomain` configuration is exactly as described for the default configuration, except that it allows you to provide unique Node Manager credentials. If you want a unique location for `NodeManagerHome`, select `CustomLocation` and specify an empty directory or select to create one. The resulting Node Manager also runs as a per domain process.

If you want to continue with a *per host* Node Manager, for which scripts are located in `WL_HOME\server\bin`, you must first perform the following prerequisite configuration steps:

1. Create a `nodemanager.domains` file that specifies the domains that you want this Node Manager instance to control, under `ORACLE_HOME\oracle_common\common\nodemanager`, the per host `NodeManagerHome` location.

- You can manually create or copy this file. See [Section 3.5, "Step 4: 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 WebLogic Server releases.

2. If you want to use the demonstration Identity and Trust keystores, recommended for development or testing purposes *only*, you can create them using the `CertGen` and `ImportPrivateKey` Java utilities as shown in the following examples:

1. To properly set up the `PATH` and `CLASSPATH` variables, from a command prompt run `WL_HOME\server\bin\setWLSEnv.cmd`.
2. 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
```

3. 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
```

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

4. Copy the `DemoIdentity.jks` keystore to the `NodeManagerHome\security` directory.

For information on configuring SSL for Node Manager in production environments, see [Section 4.6, "Using SSL With Java-based Node Manager."](#)

For domains that include Oracle JRF, you can configure Node Manager to use the Oracle Platform Security Services Keystore Service (OPSS). See "Configuring Node Manager to Use the OPSS Keystore Service" in *Administering Oracle Fusion Middleware*.

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 4-1](#).

4.2 Configuring Node Manager Using WLST Offline

If needed, you can use WLST offline to perform the following Node Manager configuration tasks:

- Set the Node Manager user name and password
- Set Node Manager properties
- Set the Node Manager type

[Example 4–1](#) 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.

Note: If the Node Manager type is `ManualNodeManagerSetup`, you cannot use WLST offline to edit Node Manager properties.

Example 4–1 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')
```

For more information see "setOption" in *WLST Command Reference for WebLogic Server*.

4.3 Configuring Java-based Node Manager Security

Node Manager security relies on a one-way SSL connection between the client and server.

If you are establishing a command-line connection to the Java Node Manager using the 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. For more information on `nm_password.properties`, see [Section 3.3, "Step 2: Specify Node Manager User Name and Password"](#).

Node Manager credentials are located on the *domain_name* > **Security** > **General** > **Advanced Options** page in the Administration Console.

Administration 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.

4.3.1 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 [Section 2.5.2, "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 flag
`-Dweblogic.nodemanager.ServiceEnabled=true`.
 - * Create the `DOMAIN_HOME\servers\AdminServer\data\nodemanager` directory.
 - * Update any startup properties or the server's credentials while the 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.

4.4 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.

[Table 4–1](#) describes Node Manager properties.

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 in 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 `StartScriptEnabled` and `NativeVersionEnabled` properties.
- If Node Manager runs on a multihomed system, and you want to control which address and port it uses, define `ListenAddress` and `ListenPort`.

Table 4–1 Node Manager Properties

Node Manager Property	Description	Default
PropertiesVersion	Specifies the version of the <code>nodemanager.properties</code> file. Do not change this value.	none
AuthenticationEnabled	If set to <code>true</code> , Node Manager authenticates the credentials against the domain.	<code>true</code>
LogFile	Location of the Node Manager log file.	<code>NodeManagerHome\nodemanager.log</code>
LogLimit	Maximum size of the Node Manager Log specified as an integer. When this limit is reached, a new log file is started.	unlimited
LogCount	Maximum number of log files to create when <code>LogLimit</code> is exceeded.	1
LogAppend	If set to <code>true</code> , then a new log file is not created when Node Manager restarts; the existing log is appended instead.	<code>true</code>
LogToStderr	If set to <code>true</code> , the log output is also sent to the standard error output.	<code>false</code>
LogLevel	Severity level of logging used for the Node Manager log. Node Manager uses the standard logging levels from the <code>java.util.logging.level</code> package, http://docs.oracle.com/javase/7/docs/api/java/util/logging/Level.html .	INFO
LogFormatter	Name of formatter class to use for NM log messages.	<code>weblogic.nodemanager.server.LogFormatter</code>
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
CrashRecoveryEnabled	Enables system crash recovery. Note: The <code>CrashRecoveryEnabled</code> property takes precedence over the <code>AutoRestart</code> server startup property in a crash recovery scenario. For example, if a server instance has <code>AutoRestart=false</code> but <code>CrashRecoveryEnabled=true</code> , when Node Manager restarts, Node Manager tries to recover the server instance if the server failed when Node Manager was not running.	<code>false</code>
SecureListener	If set to <code>true</code> , use the SSL listener, otherwise use the plain socket.	<code>true</code>
CipherSuite	The name of the cipher suite to use with the SSL listener.	The default value is JDK and platform dependent.
StartScriptEnabled	If true, use the start script specified by <code>StartScriptName</code> to start a server. For more information, see Section 4.5, "Configuring Node Manager to Use Start and Stop Scripts" .	<code>true</code>

Table 4–1 (Cont.) Node Manager Properties

Node Manager Property	Description	Default
StartScriptName	The name of the start script, located in the domain directory.	startWebLogic.sh (UNIX) or startWebLogic.cmd (Windows)
StopScriptEnabled	If true, execute the stop script specified by StopScriptName after the server has shutdown. For more information, see Section 4.5, "Configuring Node Manager to Use Start and Stop Scripts" .	false
StopScriptName	The name of the script to be executed after server shutdown.	none
QuitEnabled	If set to true, allow the user to remotely stop Node Manager. WLST overrides the default value, false, when stopping Node Manager using the stopNodeManager () command.	false
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
StateCheckInterval	Specifies the interval Node Manager waits to perform a check of the server state.	500 milliseconds
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
CustomIdentityKeyStoreFileName	Specifies the file 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
CustomIdentityKeyStorePassPhrase	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 in order to write to the keystore. However, some keystores do not require the passphrase to read from the keystore. WebLogic Server only reads from the keystore, so whether or not you define this property depends on the requirements of the keystore.	none
CustomIdentityKeyStoreType	Specifies the type of the Identity keystore. Generally, this is JKS. This property is optional.	default keystore type from java.security
CustomIdentityPrivateKeyPassPhrase	Specifies the password used to retrieve the private key for WebLogic Server from the Identity keystore. This property is required when the Keystores property is set as CustomIdentityandCustomTrust or CustomIdentityAndJavaStandardTrust.	none

Table 4–1 (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
JavaStandardTrustKeyStorePassPhrase	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. WebLogic Server only reads 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 CustomIdentityandJavaStandardTrust or DemoIdentityandDemoTrust.	none
KeyStores	Indicates the keystore configuration Node Manager uses to find its identity (private key and digital certificate) and trust (trusted CA certificates). Possible values are: <ul style="list-style-type: none"> ■ DemoIdentityandDemoTrust Use the demonstration Identity and Trust keystores located in the <code>DOMAIN_HOME\security</code> and <code>WL_HOME\server\lib</code> directories that are configured by default. The demonstration Trust keystore trusts all the certificate authorities in the Java Standard Trust keystore (<code>JAVA_HOME\jre\lib\security\cacerts</code>) ■ CustomIdentityandJava ■ StandardTrust Uses a keystore you create, and the trusted CAs defined in the <code>cacerts</code> file in the <code>JAVA_HOME\jre\lib\security\cacerts</code> directory. ■ CustomIdentityandCustomTrust Uses Identity and Trust keystores you create. 	DemoIdentityandDemoTrust
ListenAddress	Any address upon which the machine running Node Manager can listen for connection requests. This argument deprecates <code>weblogic.nodemanager.listenAddress</code> .	null With this setting, Node Manager will listen on any IP address on the machine
ListenPort	The TCP port number on which Node Manager listens for connection requests. This argument deprecates <code>weblogic.nodemanager.listenPort</code> .	5556

Table 4–1 (Cont.) Node Manager Properties

Node Manager Property	Description	Default
NativeVersionEnabled	<p>A value of true causes native libraries for the operating system to be used.</p> <p>For UNIX systems other than Solaris or Linux, set this property to false to run Node Manager in non-native mode. This will cause Node Manager to use the start script specified by the StartScriptEnabled property to start Managed Servers.</p> <p>Note that when NativeVersionEnabled=false:</p> <ul style="list-style-type: none"> ■ Node Manager cannot query if a PID is alive nor kill a particular process ■ 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 ■ nmKill is not supported when NativeVersionEnabled=false and StartScriptEnabled=true 	true
NodeManagerHome	<p>Node Manager root directory which contains the following configuration and log files:</p> <ul style="list-style-type: none"> ■ nodemanager.domains ■ nodemanager.log ■ nodemanager.properties <p>For more information on these files, see Section 2.7, "Node Manager Configuration and Log Files".</p> <p>Note: By default, <i>NodeManagerHome</i> is <i>DOMAIN_HOME\nodemanager</i>. In a production environment, you might want to customize the location of the Node Manager root directory.</p>	<i>NodeManagerHome</i>
WebLogicHome	<p>Root directory of the WebLogic Server installation. This is used as the default value of <code>-Dweblogic.RootDirectory</code> 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.</p>	none
keyFile	<p>The path to the private key file to use for SSL communication with the Administration Server.</p> <p>Note: This property is used only in the process of upgrading from WebLogic Server, Version 7.x to Version 9.x.</p>	none
keyPassword	<p>The password used to access the encrypted private key in the key file.</p> <p>Note: This property is used only in the process of upgrading from WebLogic Server, Version 7.x to Version 9.x.</p>	none

Table 4–1 (Cont.) Node Manager Properties

Node Manager Property	Description	Default
certificateFile	Specifies the path to the certificate file used for SSL authentication. Note: This property is used only in the process of upgrading from WebLogic Server, Version 7.x to Version 9.x.	none
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 also, the <InterfaceName> property for more flexibility entering multiple interfaces and corresponding netmask values.	none
Interface	The primary interface names used by migratable servers. For server migration, the primary interface names used by migratable servers must be the same. See also, the <InterfaceName> property for more flexibility specifying multiple interfaces and a corresponding range of IP addresses that should be bound to a specific interface.	none

Table 4-1 (Cont.) Node Manager Properties

Node Manager Property	Description	Default
<InterfaceName>	<p>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.</p> <p>Syntax: <InterfaceName>=<IP_RANGE_MIN>-<IP_RANGE_MAX>, (optional) NetMask=<NETMASK_ADDRESS></p> <p>For example, the syntax for binding addresses 1 - 4 to interface eth0 and addresses 5 - 8 to bond0 is:</p> <pre>eth0=1-4,NetMask=255.255.255.0 bond0=5-8,NetMask=255.255.248.0</pre> <p>You can leave out the NetMask value, if desired, and simply enter:</p> <pre>eth0=200.10.10.1-200.10.10.255 bond0=199.0.0.1-199.0.0.255</pre> <p>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.</p> <p>For example, specifying these properties in the original format:</p> <pre>Interface=oldEth0 NetMask=255.255.255.0</pre> <p>Would be the same as specifying this in the new format:</p> <pre>oldEth0=*,Netmask=255.255.255.0</pre> <p>An asterisk (*) can be represent all IPs.</p>	none
DomainsDirRemoteSharingEnabled	<p>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.</p> <p>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.</p>	false
DomainRegistrationEnabled	<p>By default, clients cannot dynamically register a new domain; domains must be configured during the domain creation process or before starting Node Manager.</p> <p>If set to true, clients can automatically register new domains, however, this creates a security risk, so it is not advised.</p>	false

4.5 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 is started or after it is shutdown. Mounting and unmounting remote disks is one example of a task that can be performed using scripts.

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

4.5.1 Script Location

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

```
DOMAIN_HOME\bin\service_migration
```

Script execution should occur relative to this directory.

4.5.2 Best Practices When Using Start and Stop Scripts

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

4.5.3 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:

1. In the `nodemanager.properties` file, set the `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 `StartScriptName` property to the name of your script in the `nodemanager.properties` file.

Note: When creating a custom start script, start the server in place instead of running `startWebLogic.sh` in the background. This way, while the server is running, the custom script process is also running. For example:

```
# custom stuff
# custom stuff
startWebLogic.sh
```

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 Administration Console to start the server, or `WLST` connected to the 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.

4.5.4 Using Stop Scripts

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

Note: Stop scripts are used only to execute a script after a server fails and must be migrated.

To define a stop script:

1. In the `nodemanager.properties` file, set the `StopScriptEnabled` property to `true`.
2. Set the `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
sync
  PIDS=`/usr/local/bin/lsof $FS | awk
  '{if ($2 ~/[0-9]+)/} { print $2} }' | sort -u`
kill -9 $PIDS
sleep 1
sync
  /usr/sbin/umount -f $FS
fi
```

4.6 Using SSL With Java-based Node Manager

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

The default WebLogic Server installation includes demonstration Identity and Trust keystores that allow you to use SSL out of the box. `DemoIdentity.jks` is installed in the `DOMAIN_HOME\security` directory and `DemoTrust.jks` is in `WL_HOME\server\lib`. For testing and development purposes, the keystore configuration is complete.

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 [Table 4-1](#). To configure production SSL components, see "Configuring SSL" in *Administering Security for Oracle WebLogic Server*.

4.7 Configuring Node Manager on Multiple Machines

If you have a domain that has Managed Servers on multiple physical machines, you must ensure that Node Manager is installed and configured on each machine. You can use the WLST command `nmEnroll` to copy all of the required domain and configuration information from one machine to another. For more information, see [Section 2.3, "Accessing Node Manager"](#) and `nmEnroll` in *WLST Command Reference for WebLogic Server*.

4.8 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, additional configuration may be necessary.

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

```
# default: off
# description:nodemanager as a service
service nodemgrsvc
{
    type                = UNLISTED
    disable              = no
    socket_type          = stream
    protocol             = tcp
    wait                 = yes
    user                 = <username>
    port                 = 5556
    flags                = NOLIBWRAP
    log_on_success      += DURATION HOST USERID
    server               = <path-to-java>/java
    env                  = CLASSPATH=<cp> LD_LIBRARY_PATH=<ldpath>
    server_args          = -client -DNodeManagerHome=<NMHome> <java options>
                       <nodemanager options> weblogic.NodeManager -v
}
```

Configuring Script Node Manager

This chapter describes how to configure script-based Node Manager.

This chapter includes the following sections:

- [Section 5.1, "Overview"](#)
- [Section 5.2, "Step 1: Create User Accounts"](#)
- [Section 5.3, "Step 2: Configure Node Manager Security"](#)
- [Section 5.4, "Step 3: Install WebLogic Server"](#)
- [Section 5.5, "Step 4: Create a WebLogic Domain"](#)
- [Section 5.6, "Step 5: Start the Administration Server"](#)
- [Section 5.7, "Step 6: Configure Node Manager on the Managed Servers"](#)
- [Section 5.8, "Step 7: Test Node Manager Setup and Configuration by Starting Managed Servers"](#)
- [Section 5.9, "Step 8: Configure UNIX Machines"](#)
- [Section 5.10, "Step 9: Assign Servers to Machines"](#)
- [Section 5.11, "Step 10: Start Managed Servers"](#)
- [Section 5.12, "Overriding the Default SSH Port"](#)
- [Section 5.13, "Configuring Security for WebLogic Server Scripts"](#)
- [Section 5.14, "Configuring Remote Server Start Security for Script-based Node Manager"](#)
- [Section 5.15, "Generating and Distributing Key Value Pairs"](#)

5.1 Overview

The SSH 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.

5.2 Step 1: Create User Accounts

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

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 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 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 -g bea -m ndmgr
> passwd ndmgr
```

5.3 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. Authentication is not required. 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.

Note: You must also ensure that the Node Manager and 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:/home/username/bea/user_projects/domains/domain_
name/bin/server_migration
```

This file resides in *USER_HOME/.ssh/*.

Configure SSH trust between the **ndmgr** user on each machine that will run a 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. A **ndmgr** user must also be able to establish an SSH session with itself without being prompted for security credentials. 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. For more information, see [Section 5.13, "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:

1. 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:./
```

4. 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_keys2
```

```
> chmod 700 .ssh/authorized_keys2
```

```
> cat id_dsa.pub >> .ssh/authorized_keys2
```

```
> rm id_dsa.pub
```

```
> exit
```

5. 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_keys2` 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. For more information, see [Section 5.15, "Generating and Distributing Key Value Pairs"](#).

5.4 Step 3: Install WebLogic Server

As the **bea** user, install a WebLogic Server instance in the base directory, `/opt/bea/wlserver`, on all the machines that will run WebLogic Server.

For example:

```
> ./ wls_generic.jar
```

5.5 Step 4: Create a WebLogic Domain

In the **ndmgr** user's home directory, create a WebLogic domain on the machine which will host 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:

```
> /opt/bea/wlserver/common/bin/config.sh
```

2. Create a new WebLogic domain based on the default 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**.
5. 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 `/opt/bea/clustdomain`

5.6 Step 5: Start the Administration Server

As the **ndmgr** user, start the Administration Server locally from a terminal using the `wlscontrol.sh` Node Manager script.

For example:

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

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

Once successfully started, stop the Administration Server and then start it remotely using SSH.

For example:

```
> ssh -l ndmgr -o PasswordAuthentication=no %p 22 192.168.1.100
/opt/bean/wlserver_
103/common/bin/wlscontrol.sh -d clustdomain -r /home/ndmgr/clustdomain -c -f
startWebLogic.sh -s AdminServer START
```

5.7 Step 6: 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
```

2. 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:

```
> opt/bean/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.

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 `/opt/bean/wlserver/common/nodemanager`; you might not want to use this directory as it is in the product installation directory and owned by another user.

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`.

5.8 Step 7: Test Node Manager Setup and Configuration by Starting Managed Servers

1. Create a WebLogic script directory (`bin`) in Node Manager's new domain home.

For example:

```
> mkdir ~/clustdomain/bin
```

2. 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
```

3. 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:

- a. 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*
 - b. *LONG_DOMAIN_HOME=/home/ndmgr/clustdomain*
 - c. Similarly, edit the *DOMAIN_HOME* variable in *startWebLogic.sh*
 - d. Edit the *DOMAIN_HOME* and *ADMIN_URL* (for example, *t3://192.168.1.100:7001*) variables in *startManagedWebLogic.sh*.
4. 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
```

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

5. 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
```

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

6. 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:

```
> opt/bea/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 *-x* 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 /opt/bea/wlserver/common/bin/wlscontrol.sh -d
clustdomain -n /home/ndmgr -c -f startManagedWebLogic.sh -s
MS1 START
```

- 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.

5.9 Step 8: Configure UNIX Machines

Using the Administration Console, add a new UNIX Machine for each machine which will host an Administration or Managed Server (including the back-up machine) and include the following settings:

Table 5–1 UNIX Machine Settings

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

5.10 Step 9: Assign Servers to Machines

Once all of the UNIX Machines are created, use the Administration Console to set the Machine property for each server, to ensure each server is associated with its corresponding UNIX Machine. See "Assign server instances to machines" in the *Oracle WebLogic Server Administration Console Online Help*.

5.11 Step 10: Start Managed Servers

In the Administration Console, start each Managed Server. See "Start Managed Servers from the Administration Console" in the *Oracle WebLogic Server Administration Console Online Help*.

Check the server logs in the `/home/ndmgr/clustdomain/servers/managed_server_name/logs` directory of each Managed Server to ensure that the server has started with no errors.

5.12 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.

5.13 Configuring Security for 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 your SSH to only allow `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 [Step 2: Configure Node Manager Security](#).

5.14 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.

5.15 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.

- [Section 5.15.1, "Shared Key Value Pair"](#).
- [Section 5.15.2, "Individual Key Value Pairs"](#)

5.15.1 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 your SSH installation.

The default location for the private and public keys are `~/.ssh/id_rsa` and `~/.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="/home/bea/server90/common/nodemanager/nodemanager.sh
" 1024 33 23...2323
```

in which the you 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
```

Note: The prefix `command=command` ensures that a user that establishes a session with the machine using the public key can only run the command specified—`nodemanager.sh`. This ensures that the user can only perform Node Manager functions, and 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:

```
/home/bea$ ssh montgomery VERSION
```

This response indicates that the client accessed Node Manager successfully:

```
+OK NodeManager v9.1.0
```

5.15.2 Individual Key Value Pairs

On each machine that will host a Node Manager client:

1. Generate a separate RSA key value pair for the Node Manager user as described in step one in the previous section.
2. Append the public key to the machine's `~/.ssh/authorized_keys` file user as described in step two in the previous section.

Using Node Manager

This chapter describes how to start and stop the Java-based and script-based Node Manager. It also provides information on the recommended procedures for starting servers using Node Manager.

This chapter includes the following sections:

- [Section 6.1, "Starting and Stopping Node Manager"](#)
- [Section 6.2, "Using Node Manager to Control Servers"](#)

6.1 Starting and Stopping Node Manager

Use the following methods for starting and stopping Node Manager:

- [Section 6.1.1, "Running Node Manager as a Startup Service"](#)
- [Section 6.1.2, "Starting Java-based Node Manager Using Scripts"](#)
- [Section 6.1.3, "Running Script-based Node Manager"](#)
- [Section 6.1.4, "Stopping Node Manager"](#)

6.1.1 Running Node Manager as a Startup Service

It is recommended 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.

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 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.
2. Open a DOS command prompt window.
3. 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.

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.
2. Open a DOS command prompt window.
3. Change to the *DOMAIN_HOME*\bin directory.
4. Enter the following command:

```
uninstallNodeMgrSvc.cmd
```

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

```
Oracle WebLogic <domain-name> NodeManager removed.
```

By default, Node Manager listens only from the local host. If you want Node Manager to accept commands from remote systems, you must edit the script to listen on a non-localhost listen address.

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 [Section 4.1, "Default Node Manager Configuration"](#).

6.1.2 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 [Section 3.9, "Step 8: \(Optional\) 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 WebLogic Server. However, if you select to use the script in *WL_HOME*\server\bin, you must first perform the prerequisite steps described in [Section 4.1, "Default Node Manager Configuration"](#).

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.

6.1.2.1 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`.

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. For more information, see [Section 4.4, "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`— 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 version. For more information, see [Section 4.4, "Reviewing nodemanager.properties"](#).

6.1.3 Running Script-based Node Manager

Note: In this release of WebLogic Server, prior to running the script version 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 [Section 3.5, "Step 4: 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'
```

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 SSH 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 6-1](#). The possible parameter values are listed in [Table 6-2](#).

For example, if you type this command,

```
ssh -o PasswordAuthentication=no wlscontrol.sh myserver start
```

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`.

This command:

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

issues a `START` command to the server instance named `ServerA`, in the domain called `ProductionDomain`, located in the `domains/ProductionDomain` directory.

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 6–1 *wlscontrol.sh* Command Line Options

Parameter	Description
-n	Specifies the Node Manager root directory
-s	Specifies the server name
-d	Specifies the domain name
-r	Specifies the domain directory
-c	Enables a server start script
-f	The name of the server start script
-p	The name of the server stop script
-v	Enables verbose output
-h	Prints the usage for wlscontrol.sh

Table 6–2 *Shell Command Templates*

Parameter	Description	Default
%H	Host name of the SSH server	NodeManagerMBean.ListenAddress
%N	Node Manager home directory	NodeManagerMBean.NodeManagerHome
%P	Port number of SSH server	NodeManagerMBean.ListenAddress 22
%S	WebLogic Server name	none
%D	WebLogic domain name	none
%R	Domain directory (server root)	ServerStartMBean.RootDirectory
%C	Node Manager script command <ul style="list-style-type: none"> ■ START—Start server ■ KILL—Kill server ■ STAT—Get server status ■ GETLOG—Retrieve server output log. ■ VERSION—Return Node Manager version. 	none

Note: This must be the last element in the command.

6.1.4 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. For more information, see `stopNodeManager` in the *WLST Command Reference for WebLogic Server*.

6.2 Using Node Manager to Control Servers

In general, it is recommended that you use the WebLogic Scripting Tool and Node Manager to start and stop the Administration Server and Managed Servers. This section describes the recommended procedures for starting servers using Node Manager and WLST.

For more information, see "Using WLST and Node Manager to Manage Servers" and the "Node Manager Commands" in *WLST Command Reference for WebLogic Server*.

6.2.1 Starting the Administration Server Using Node Manager

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

1. Start Node Manager. See [Section 6.1, "Starting and Stopping Node Manager"](#).

2. Invoke WLST.

On Windows, you can use a shortcut on the Start menu to set your environment variables and invoke WLST.

3. Connect WLST to Node Manager using the `nmConnect` command.
4. Start the Administration Server using the `nmStart` command.

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

Note: Starting the server 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 servers that were started in this way.

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

6.2.2 Starting Managed Servers

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

1. Start Node Manager. See [Section 6.1, "Starting and Stopping Node Manager"](#).
2. Start an Administration Server. See "Starting Instances of WebLogic Server" in *Administering Server Startup and Shutdown for Oracle WebLogic Server*.
3. Invoke WLST and connect to an Administration Server using the `connect` command.
4. Start your 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.

6.2.3 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 [Section 6.1, "Starting and Stopping Node Manager"](#).
2. Invoke WLST and connect to Node Manager using the `nmConnect` command.
3. Start the Managed Server using the WLST `nmStart` command.

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.

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 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. The next time a server is migrated or restarted from the Administration Server, these properties will not be used.
- When passing the server user name and password using `nmStart`, these values are not encrypted.

