

# **Oracle® Endeca Information Discovery Studio**

Studio Installation Guide

Version 3.2.0 • January 2016

# Copyright and disclaimer

Copyright © 2003, 2017, Oracle and/or its affiliates. All rights reserved.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners. UNIX is a registered trademark of The Open Group.

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 fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

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.

# Table of Contents

<b>Copyright and disclaimer</b> .....	<b>2</b>
<b>Preface</b> .....	<b>5</b>
About this guide .....	5
Who should use this guide .....	5
Conventions used in this document .....	5
Contacting Oracle Customer Support .....	6
<b>Part I: Before You Begin</b>	
<b>Chapter 1: Overview of Studio and the Installation Process</b> .....	<b>8</b>
About Oracle Endeca Information Discovery Studio .....	8
About the Provisioning Service .....	9
Other modules in Oracle Endeca Information Discovery .....	9
Recommended installation order for Studio and its associated modules .....	10
<b>Chapter 2: System Requirements</b> .....	<b>12</b>
Studio system requirements .....	12
Provisioning Service system requirements .....	14
Downloading Studio and the Provisioning Service .....	15
<b>Part II: Installing Studio</b>	
<b>Chapter 3: Installing Studio on Oracle WebLogic Server 12cR1 (12.1.3)</b> .....	<b>19</b>
Overview .....	19
Installing WebLogic Server .....	20
Creating a WebLogic domain for Studio .....	28
Deploying Studio to the WebLogic domain .....	38
Creating a Windows service for your Studio WebLogic domain .....	40
Changing the Studio home directory on WebLogic Server .....	43
Changing the context path for Studio on WebLogic Server .....	44
<b>Part III: Installing the Provisioning Service</b>	
<b>Chapter 4: Installing the Provisioning Service</b> .....	<b>48</b>
Overview .....	48
Running the Provisioning Service installation script .....	49
Silent installation .....	51
Verifying your installation .....	53
Provisioning Service data domain profile .....	53
Backend database configuration .....	54

Changing the communication security configurations . . . . .	55
Enabling SSL . . . . .	55
Disabling SSL . . . . .	56
Connecting the Provisioning Service to Studio . . . . .	56
<b>Chapter 5: Connecting the Provisioning Service to Studio . . . . .</b>	<b>59</b>
 <b>Part IV: After You Install</b>	
<b>Chapter 6: Getting Started with Studio . . . . .</b>	<b>63</b>
Logging in to Studio . . . . .	63
Displaying the Studio Control Panel . . . . .	63
<b>Chapter 7: Changing the Database Used to Store Studio Application Data . . . . .</b>	<b>65</b>
About using a different database . . . . .	65
Overview of the process for switching to a different database . . . . .	65
<b>Chapter 8: Using Studio with a Reverse Proxy . . . . .</b>	<b>67</b>
About reverse proxies . . . . .	67
What is a reverse proxy? . . . . .	67
Types of reverse proxies . . . . .	67
Example sequence for a reverse proxy request . . . . .	68
Recommendations for reverse proxy configuration . . . . .	68
Preserving HTTP 1.1 Host: headers . . . . .	69
Enabling the Apache ProxyPreserveHost directive . . . . .	69
Reverse proxy configuration options for Studio . . . . .	70
Simple Studio reverse proxy configuration . . . . .	70
Studio reverse proxy configuration without preserving Host: headers . . . . .	70
Configuring Studio to support an SSL-enabled reverse-proxy . . . . .	71
<b>Chapter 9: Creating a Studio Cluster . . . . .</b>	<b>72</b>
About Studio clustering . . . . .	72
Setting up the cluster . . . . .	73
Installing the Studio instances . . . . .	73
Configuring synchronized caching for the Studio instances . . . . .	74
About synchronized caching . . . . .	74
Updating portal-ext.properties to enable synchronized caching . . . . .	74
Customizing the clustered cache configuration files . . . . .	75
Clearing the cache for a cluster . . . . .	76
 <b>Part V: Uninstalling Oracle Endeca Information Discovery</b>	
<b>Chapter 10: Uninstalling Studio and the Provisioning Service . . . . .</b>	<b>79</b>
Uninstalling Studio . . . . .	79
Uninstalling the Provisioning Service . . . . .	79

## Preface

Endeca Information Discovery Studio is an industry-leading application composition environment and discovery experience that allows business users to easily upload and mash up multiple diverse data sources, and then quickly configure discovery applications - all within the context of an enterprise framework that maintains existing governance and enterprise definitions.

Studio includes world-class search, guided navigation, and filtering, as well as offering an array of powerful interactive visualizations, for rapid intuitive analysis that requires zero training.

## About this guide

This guide contains installation instructions for installing and setting up Oracle Endeca Information Discovery Studio on Windows and Linux.

Installing Oracle Endeca Information Discovery Studio consists of installing:

- Studio
- Provisioning Service

Oracle Endeca Information Discovery Studio requires the Oracle Endeca Server and Oracle Endeca Information Discovery Integrator ETL.

## Who should use this guide

This guide is intended for system administrators installing Oracle Endeca Information Discovery Studio on Windows or Linux.

## Conventions used in this document

The following conventions are used in this document.

### Typographic conventions

The following table describes the typographic conventions used in this document.

Typeface	Meaning
<b>User Interface Elements</b>	This formatting is used for graphical user interface elements such as pages, dialog boxes, buttons, and fields.
Code Sample	This formatting is used for sample code phrases within a paragraph.
<i>Variable</i>	This formatting is used for variable values. For variables within a code sample, the formatting is <i>Variable</i> .

Typeface	Meaning
File Path	This formatting is used for file names and paths.

## Symbol conventions

The following table describes symbol conventions used in this document.

Symbol	Description	Example	Meaning
>	The right angle bracket, or greater-than sign, indicates menu item selections in a graphic user interface.	File > New > Project	From the File menu, choose New, then from the New submenu, choose Project.

## Contacting Oracle Customer Support

Oracle Customer Support provides registered users with important information regarding Oracle software, implementation questions, product and solution help, as well as overall news and updates from Oracle.

You can contact Oracle Customer Support through Oracle's Support portal, My Oracle Support at <https://support.oracle.com>.

# **Part I**

## **Before You Begin**



## Chapter 1

---

# Overview of Studio and the Installation Process

Here is a brief overview of Oracle Endeca Information Discovery Studio and the installation process.

[About Oracle Endeca Information Discovery Studio](#)

[About the Provisioning Service](#)

[Other modules in Oracle Endeca Information Discovery](#)

[Recommended installation order for Studio and its associated modules](#)

## About Oracle Endeca Information Discovery Studio

Oracle Endeca Information Discovery Studio is a web-based application that allows business analysts to rapidly assemble dashboard applications. These applications enable analysts and other end users to explore a full range of structured and unstructured enterprise data from an Endeca Server.

Studio is easy to deploy and is ideal for the agile development of enterprise-quality applications. Studio provides a library of UI components that embody best practices in information discovery applications. Because Studio is component-based, Studio applications are simple to control, adapt, and extend.

### Supported locales in Studio

Studio supports the following languages:

- French
- German
- Italian
- Spanish
- Japanese
- Korean
- Simplified Chinese
- Traditional Chinese
- Portuguese-European

Note that this is a subset of the languages supported by Endeca Server.



## About the Studio SDK

The Studio SDK is a packaged development environment for components, page layout templates, and other Studio elements.

For information on installing and using the Studio SDK, see the *Studio Administration and Customization Guide*.

## About the Provisioning Service

Oracle Endeca Information Discovery Studio includes the Provisioning Service. The Provisioning Service allows users to create applications using data from uploaded files, or from connections to Oracle BI or JDBC servers.

The Provisioning Service profiles and creates data before sending it to Endeca Server for ingest.

The Provisioning Service runs as a web application in a WebLogic Server container.

## Other modules in Oracle Endeca Information Discovery

Studio includes the web-based Studio application and the Provisioning Service, used to upload data from spreadsheet files and other data sources. Studio and the Provisioning Service are contained in the same download.

Oracle Endeca Information Discovery 3.2.0 also includes the following separately-downloaded modules:

Module and Version	Description
<b>Oracle Endeca Server 7.7.0</b>	<p>Oracle Endeca Server provides the query engine that serves as the foundation for Studio applications. These applications can answer queries from and provide business analytics information to application end users.</p> <p>Oracle Endeca Server also allows you to administer Endeca data domains, which store the source data records that are loaded by Integrator ETL or the Provisioning Service.</p> <p>Each Studio application is connected to an Endeca data domain.</p>

Module and Version	Description
<b>Oracle Endeca Information Discovery Integrator 3.2.0</b>	<p>Oracle Endeca Information Discovery Integrator ETL is a high-performance data integration platform that allows you to extract source records from a different types of sources (from flat files to databases), then load that data into an Endeca Server data domain.</p> <p>The Integrator suite includes:</p> <ul style="list-style-type: none"> <li>• <b>Integrator ETL.</b> Used to build graphs to load source data into and configure the schema for an Endeca Server data domain.</li> <li>• <b>Information Discovery connectors.</b> Integrator ETL components used to perform various data ingest operations on Endeca data domains.</li> <li>• <b>Integrator ETL Server.</b> Used to run graphs in an enterprise-wide environment. In this environment, different users and user groups can access and run the graphs.</li> <li>• <b>Integrator Acquisition System.</b> Optional component. The Integrator Acquisition System, or IAS, is a set of components that crawl source data stored in a variety of formats including: file systems, delimited files, JDBC databases, Web servers, and custom data sources.</li> </ul> <p>IAS transforms the data, if necessary, and outputs the data to an XML file or a Record Store that can be accessed by Integrator ETL for use in the Endeca Server.</p> <p>Note that if you only use the Provisioning Service to create Studio applications, you do not need to install Integrator.</p>

Oracle Endeca Server and Oracle Endeca Information Discovery Integrator have separate documentation sets, including separate installation guides.

Note that Studio 3.2.0 is only compatible with the above versions of Oracle Endeca Server and Integrator. It is not compatible with earlier versions of these modules.

## Recommended installation order for Studio and its associated modules

Following the recommended order of installation helps you minimize dependencies among the different products.

Note that if you are installing all of the modules on the same WebLogic server, make sure that you create a separate WebLogic domain for each module, and that each domain has a unique port. The *Oracle Endeca Information Discovery Getting Started Guide* provides full instructions for installing all of the modules on a single WebLogic server.

Oracle recommends the following installation order:

1. Oracle Endeca Server.

For instructions on installing Oracle Endeca Server, including system requirements, see the *Oracle Endeca Server Installation Guide*.

2. Oracle Endeca Information Discovery Integrator ETL.

For instructions on installing Oracle Endeca Information Discovery Integrator ETL, including system requirements, see the *Integrator ETL Installation Guide*.

3. Studio (see [Installing Studio on Oracle WebLogic Server 12cR1 \(12.1.3\) on page 18](#)).

4. Provisioning Service (See [Installing the Provisioning Service on page 47](#)).

5. Integrator ETL Server.

For instructions about installing Integrator ETL Server, see the *Integrator Installation Guide*.

6. Integrator Acquisition System.

For instructions about installing IAS, see the *Integrator Acquisition System Installation Guide*.



## Chapter 2

# System Requirements

---

This section describes the requirements for Studio and the Provisioning Service.

For details on the Oracle Endeca Server system requirements, see the *Oracle Endeca Server Installation Guide*. For details on the Oracle Endeca Information Discovery Integrator ETL system requirements, see the *Integrator ETL Installation Guide*.

[Studio system requirements](#)

[Provisioning Service system requirements](#)

[Downloading Studio and the Provisioning Service](#)

## Studio system requirements

Studio's system requirements are listed below.

In addition to these, you must also have a running Oracle Endeca Server 7.7.x to point Studio to.

### Hardware requirements

Minimum hardware requirements:

- x64 2.0 GHz processor
- 4 GB RAM
- 4 GB disk space

Recommended hardware requirements:

- x64 3.0 GHz dual-core processor
- 8 GB RAM


### Supported operating systems

Platform	Description
OEL	Oracle Enterprise Linux 6 (version 6 for x64) running on x64 processors. Only the Red Hat Compatible Kernel is supported.

Platform	Description
RHEL	<ul style="list-style-type: none"> <li>Red Hat Enterprise Linux Server (version 6 for x64) running on x64 processors.</li> <li>Red Hat Enterprise Linux Advanced Platform (version 6 for x64) running on x64 processors.</li> </ul> <p>For best performance on Red Hat Linux version 6 (Server and Advanced), we recommend the latest version.</p>
Windows	Windows Server 2012 R2 Enterprise running on x64 processors.

## Software requirements

Studio is a Web-based application that runs in an application server. It supports the following software:

Software	Supported Versions
Application server	Oracle WebLogic Server 12cR1 (12.1.3)
Java	<ul style="list-style-type: none"> <li>Java SE JDK 7u67+</li> <li>Java SE JDK 8_66+</li> </ul> <p>If you're installing Studio on WebLogic Server, you must manually install the JDK beforehand. You can download it from the <a href="#">Oracle Technology Network</a>. When installing it, be sure to change the install location to one that doesn't contain spaces; for example, C:\Java\jdk1.8.0_66\.</p>
Database system	<ul style="list-style-type: none"> <li>MySQL 5.5.3+</li> <li>Oracle 11g</li> <li>Oracle 12c (12.1.0.1.0 or higher)</li> </ul> <p> <b>Note:</b> Hypersonic (HSQL) is also supported, but not for production environments.</p>
Browser	<ul style="list-style-type: none"> <li>Firefox ESR on Windows. This is the recommended browser for Studio.</li> <li>Internet Explorer 11.</li> <li>Safari on the iPad. iPad can only be used to log in to Studio and view applications. Users cannot create or configure Studio applications from an iPad.</li> </ul>

## Alternative database support

The Studio server uses a relational database to store configuration and state. Studio has been tested on the databases listed above. However, many other databases are expected to work.

Customers should feel free to use any database, including shared systems they may already have in place. As with application servers, customers who choose to deploy on un-tested databases will always be supported on any issue that can be traced back to core Studio code and can be reproduced on a supported database.

If you're installing in a production environment, you must change the database afterward. For instructions on doing this, see [Changing the Database Used to Store Studio Application Data on page 64](#).

## Provisioning Service system requirements

The Provisioning Service's system requirements are listed below.

In addition to these, the you must also have a running Oracle Endeca Server 7.7.x to connect the Provisioning Service to.

### Hardware requirements

Minimum hardware requirements:

- x64 1.8 GHz processor
- 4 GB of RAM

Recommended hardware specifications:

- x64 3.0+ GHz processors; we recommend Intel Xeon (including Nehalem) or AMD Opteron processors
- 8 GB
- Gigabit Ethernet




**Note:** In this guide, the term "x64" refers to any processor compatible with the AMD64/EM64T architecture. You might need to upgrade your hardware, depending on the data you are processing. All run-time code must fit entirely in RAM. Likewise, hard disk capacity must be sufficient based on the size of your data set. Please contact your Oracle representative if you need more information on sizing your hardware.

### Supported operating systems

Platform	Description
<b>OEL</b>	Oracle Enterprise Linux 6 (version 6 for x64) running on x64 processors. Only the Red Hat Compatible Kernel is supported.
<b>RHEL</b>	<ul style="list-style-type: none"> <li>• Red Hat Enterprise Linux Server (version 6 for x64) running on x64 processors.</li> <li>• Red Hat Enterprise Linux Advanced Platform (version 6 for x64) running on x64 processors.</li> </ul> <p>For best performance on Red Hat Linux version 6 (Server and Advanced), we recommend the latest version.</p>
<b>Windows</b>	Windows Server 2012 R2 Enterprise running on x64 processors.

## Software requirements

Provisioning Service is a Web-based application that runs in a WebLogic Server web application server. It supports the following software:

Software	Supported Versions
Application server	Oracle WebLogic Server 12cR1 (12.1.3)
Java	<ul style="list-style-type: none"> <li>• Java SE JDK 7u67+</li> <li>• Java SE JDK 8_66+</li> </ul> <p>If you're installing Studio on WebLogic Server, you must manually install the JDK beforehand. You can download it from the <a href="#">Oracle Technology Network</a>. When installing it, be sure to change the install location to one that doesn't contain spaces; for example, C:\Java\jdk1.8.0_66\.</p>
RDBMS	<ul style="list-style-type: none"> <li>• MySQL 5.5.3+</li> <li>• Oracle 11g</li> <li>• Oracle 12c (12.1.0.1.0 or higher)</li> </ul> <p> <b>Note:</b> Hypersonic (HSQL) is also supported, but not for production environments.</p>

## Downloading Studio and the Provisioning Service

You download Studio and the Provisioning Service from the Oracle Software Delivery Cloud.

To download Studio and the Provisioning Service:

1. Go to the [Oracle Software Delivery Cloud](#) and sign in.
2. Accept the Export Restrictions.
3. Check **Programs** if it isn't already.
4. In the **Product** text box, enter Oracle Endeca Information Discovery Studio.
5. Click **Select Platform**, check the appropriate platform for your environment, and click **Select**.  
**Oracle Endeca Information Discovery Studio** displays in the **Selected Products** table.
6. Click **Continue**.
7. Verify that **Available Release** and **Oracle Endeca Information Discovery Studio 3.2.0...** are both checked, then click **Continue**.  
Remember that EID Studio 3.2.0 is a component of OEID 3.2.0.
8. Accept the Oracle Standard Terms and Restrictions and click **Continue**.

9. In the **File Download** popup, click the ZIP file for **Oracle Endeca Information Discovery Integrator Acquisition System**.

You should also make a note of the part number: you will need this information to identify it.

10. From the **File Download** dialog, click the ZIP file for the appropriate version of Studio for your application server, and for the Provisioning Service.

The download .zip files are named using the part number.

For Windows, the available downloads are:

Title	Description
<b>Oracle Endeca Information Discovery Studio (3.2) for Microsoft Windows x64 (64-bit) and WebLogic</b>	Studio to install on WebLogic Server.
<b>Oracle Endeca Information Discovery Studio SDK (3.2) for Microsoft Windows x64 (64-bit)</b>	Studio Component SDK. Used for development of custom components.  For information on developing custom components, see the <i>Studio Administration and Customization Guide</i> .
<b>Oracle Endeca Information Discovery Provisioning Service (3.2) for Microsoft Windows x64 (64-bit)</b>	Provisioning Service installation files.
<b>Oracle Endeca Information Discovery Studio (3.2) Documentation</b>	Installation and migration documentation for Studio and the Provisioning Service.  The full documentation set is available from the <a href="#">Oracle Technology Network</a> .

For Linux, the available downloads are:

Title	Description
<b>Oracle Endeca Information Discovery Studio (3.2) for Linux x64 (64-bit) and WebLogic</b>	Studio to install on WebLogic Server.
<b>Oracle Endeca Information Discovery Studio SDK (3.2) for Linux x64 (64-bit)</b>	Studio Component SDK. Used for development of custom components.  For information on developing custom components, see the <i>Studio Administration and Customization Guide</i> .
<b>Oracle Endeca Information Discovery Provisioning Service (3.2) for Linux x64 (64-bit)</b>	Provisioning Service installation files.



Title	Description
<b>Oracle Endeca Information Discovery Studio (3.2) Documentation</b>	Installation and migration documentation for Studio and the Provisioning Service.  The full documentation set is available from the <a href="#">Oracle Technology Network</a> .

# **Part II**

## **Installing Studio**



## Chapter 3

---

# Installing Studio on Oracle WebLogic Server 12cR1 (12.1.3)

You can deploy Studio as a standalone application on Oracle WebLogic Server.

[Overview](#)

[Installing WebLogic Server](#)

[Creating a WebLogic domain for Studio](#)

[Deploying Studio to the WebLogic domain](#)

[Creating a Windows service for your Studio WebLogic domain](#)

[Changing the Studio home directory on WebLogic Server](#)

[Changing the context path for Studio on WebLogic Server](#)

## Overview

You install Studio as a standalone application on Oracle WebLogic Server. On WebLogic, you deploy Studio into its own WebLogic domain. These instructions assume you are installing Studio onto a new WebLogic Server instance, but for development purposes you may install Studio on the same WebLogic Server instance as Endeca Server.

These instructions for installing Studio on WebLogic Server include:

1. Downloading and installing WebLogic Server.
2. Creating and configuring the WebLogic domain for Studio.

Even if you are installing Studio on the same instance of WebLogic Server as Endeca Server, you must deploy it to its own domain.

3. Deploying Studio into the WebLogic domain.

The above instructions use the default configuration. Optionally, you can use a different directory for the Studio home directory (see [Changing the Studio home directory on WebLogic Server on page 43](#)), or deploy Studio to a different context path (see [Changing the context path for Studio on WebLogic Server on page 44](#)).

For full instructions on installing all of Oracle Endeca Information Discovery (Endeca Server, Integrator ETL, Studio, and Provisioning Service) on a single WebLogic Server in a Windows environment, see the *Oracle Endeca Information Discovery Getting Started Guide*.

## Installing WebLogic Server

If you're not installing Studio or the Provisioning Service on the same server as Endeca Server, then you need to install WebLogic Server.

Before performing this procedure, be sure that you installed the Java 7 or 8 JDK.

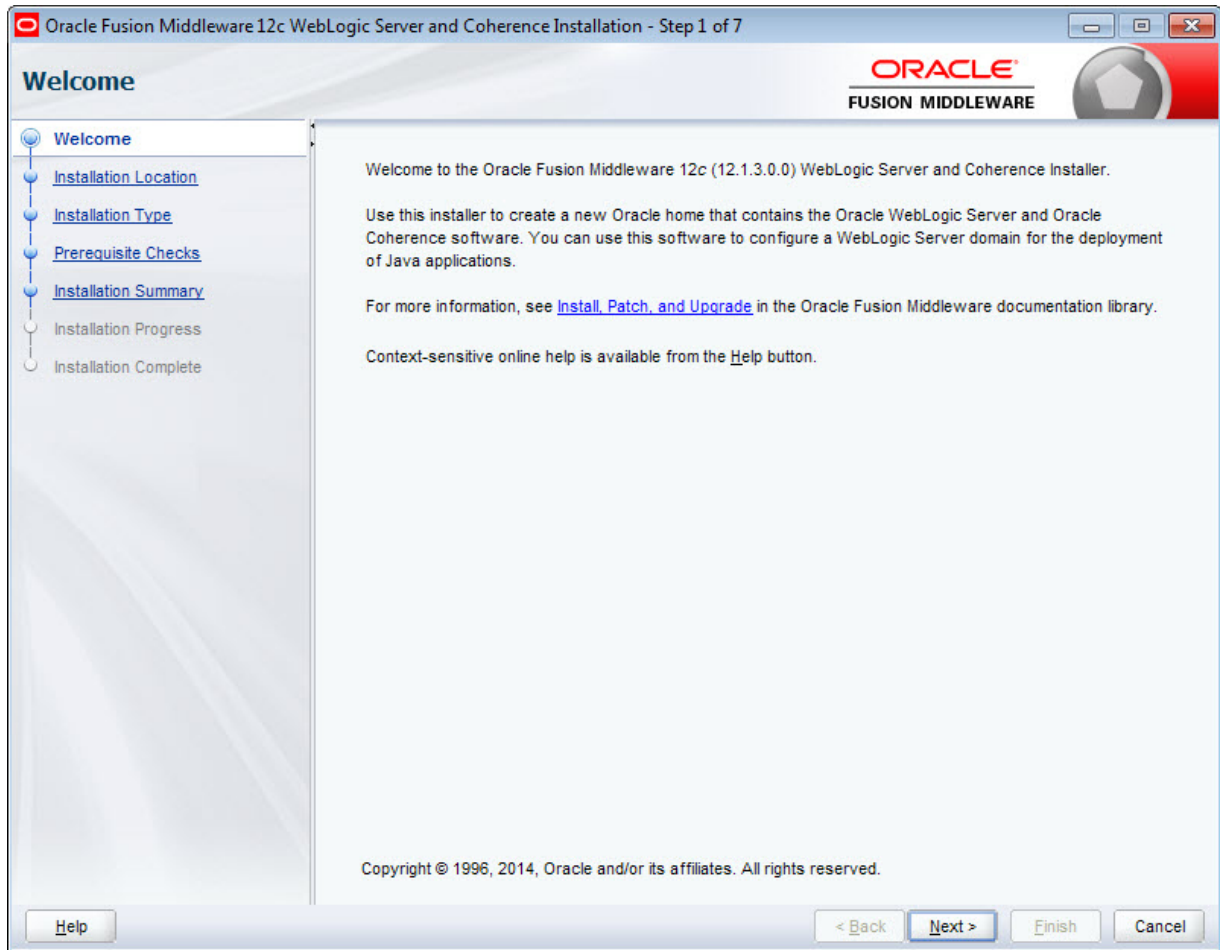
To install WebLogic Server:

1. Download the WebLogic Server installation package:
  - (a) Go to <http://www.oracle.com/technetwork/middleware/weblogic/downloads/wls-for-dev-1703574.html>.
  - (b) Select the **Accept License Agreement** radio button.
  - (c) Under **Oracle WebLogic Server Previous Releases**, locate the heading for Oracle WebLogic Server 12.1.3.
  - (d) Download the **Generic** version of the installer (`fmw_12.1.3.0.0_wls.jar`).
2. From a command prompt, navigate to the directory where you placed `fmw_12.1.3.0.0_wls.jar`.
3. Launch the installer by running:

```
java -jar fmw_12.1.3.0.0_wls.jar
```

When the installer runs, it displays the installation wizard's **Welcome** screen.

4. At the **Welcome** screen, click **Next**.

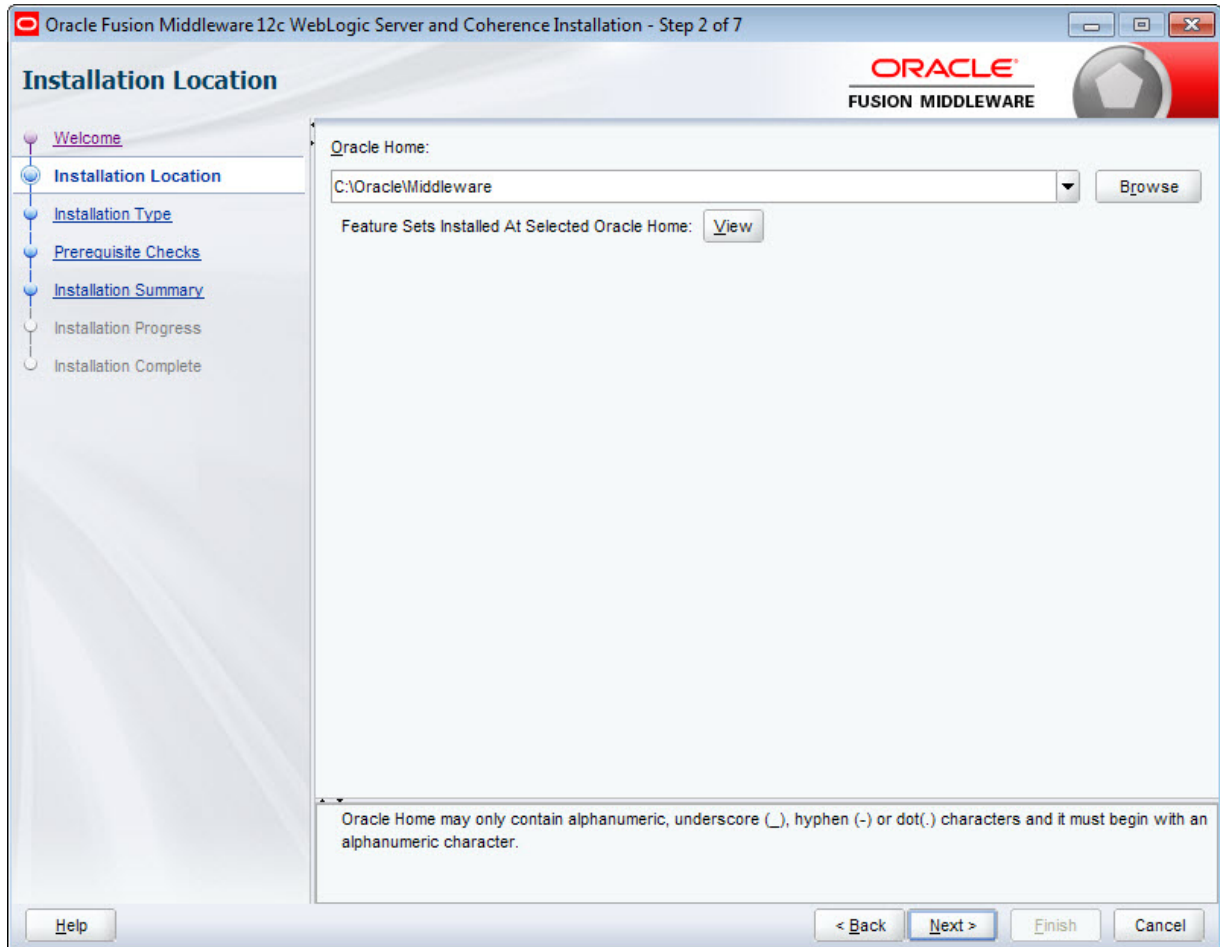


5. At the **Installation Location** screen, enter the name of the Middleware home directory in the **Oracle Home** field.

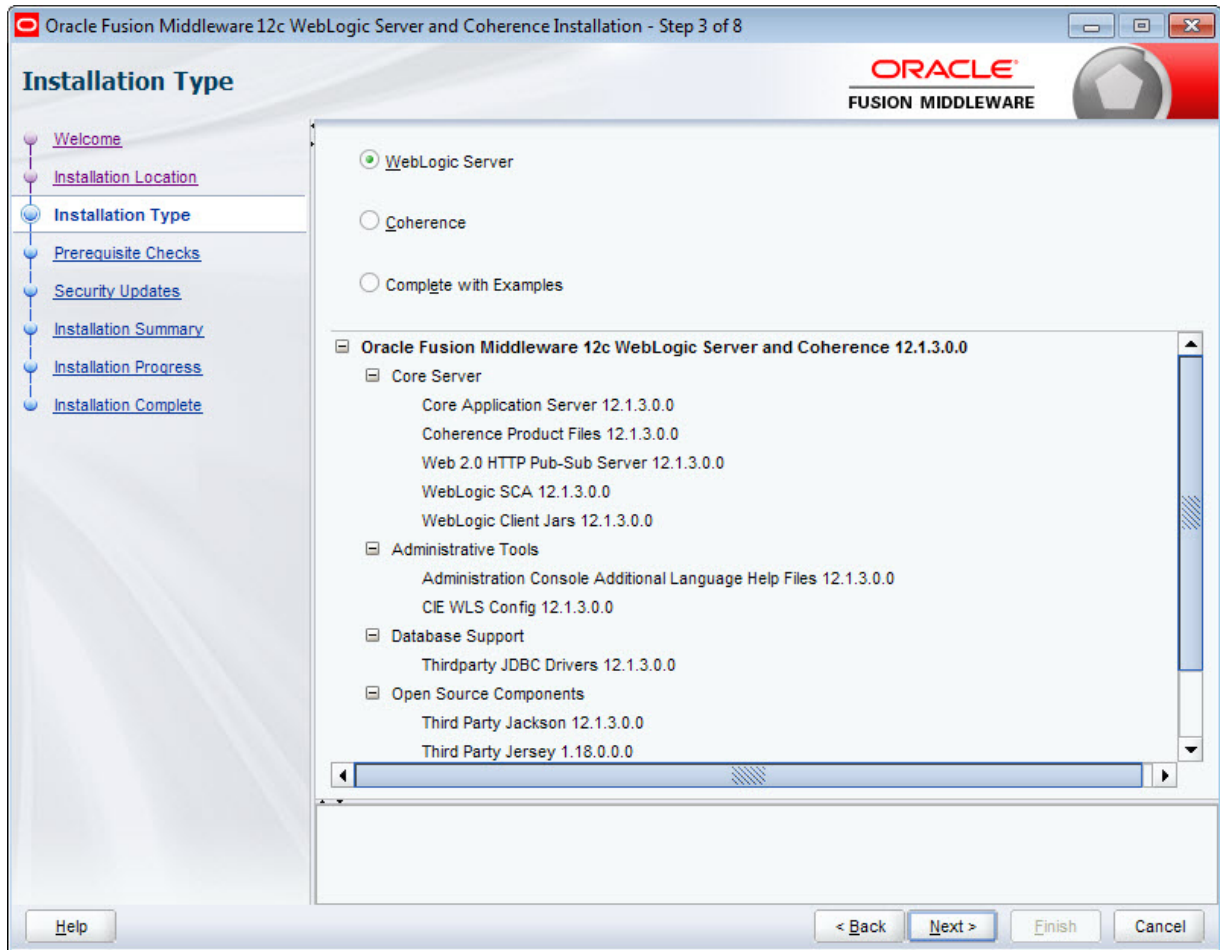
This will be the \$MW\_HOME directory of your installation. You can either accept the default location or enter a new one. The directory name may only contain alphanumeric, hyphen (-), dot (.), and underscore (\_) characters, and must begin with an alphanumeric character.

If the specified directory exists, it must be empty. If it doesn't exist, the installer will create it.

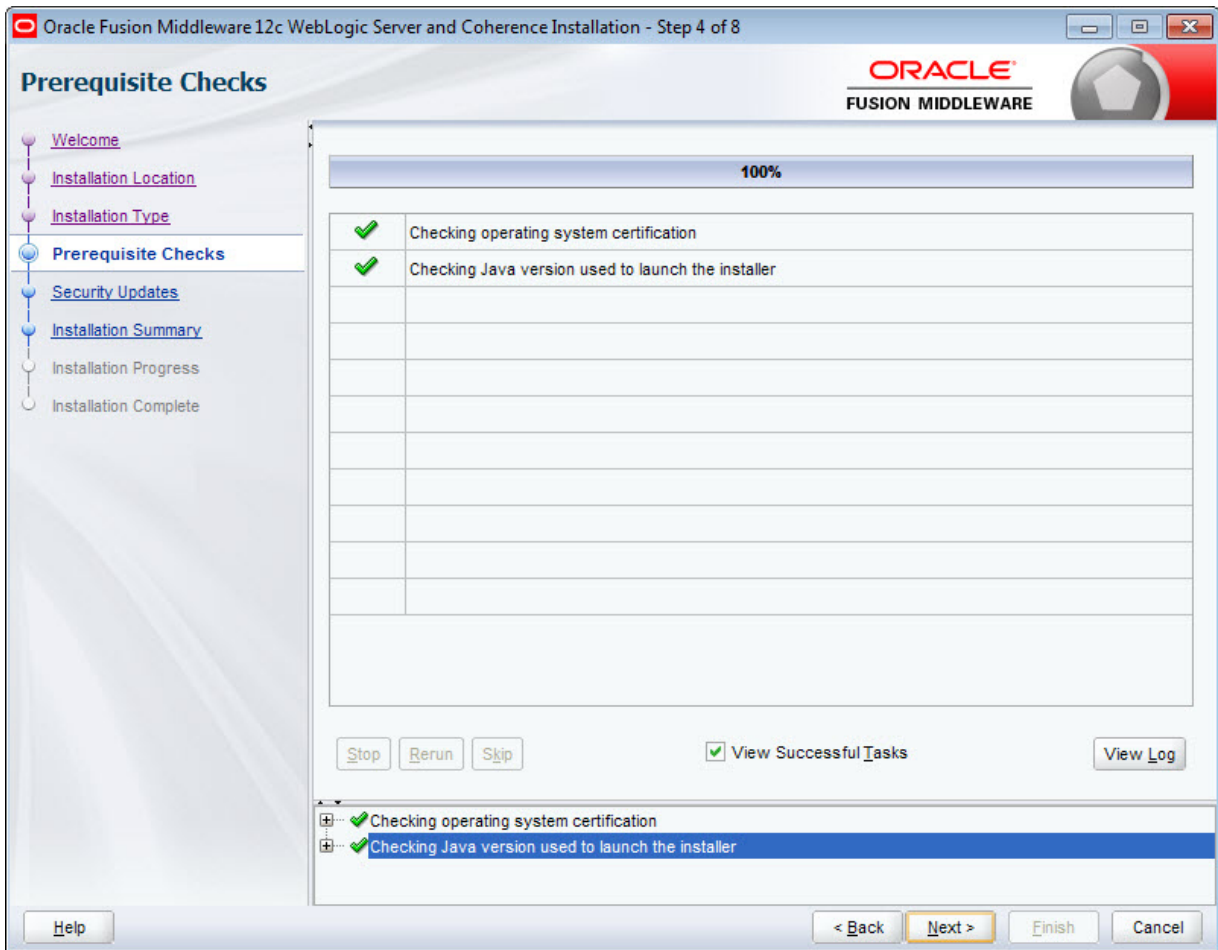
Click **Next** when you're done.



6. At the **Installation Type** screen, leave **WebLogic Server** selected and click **Next**.



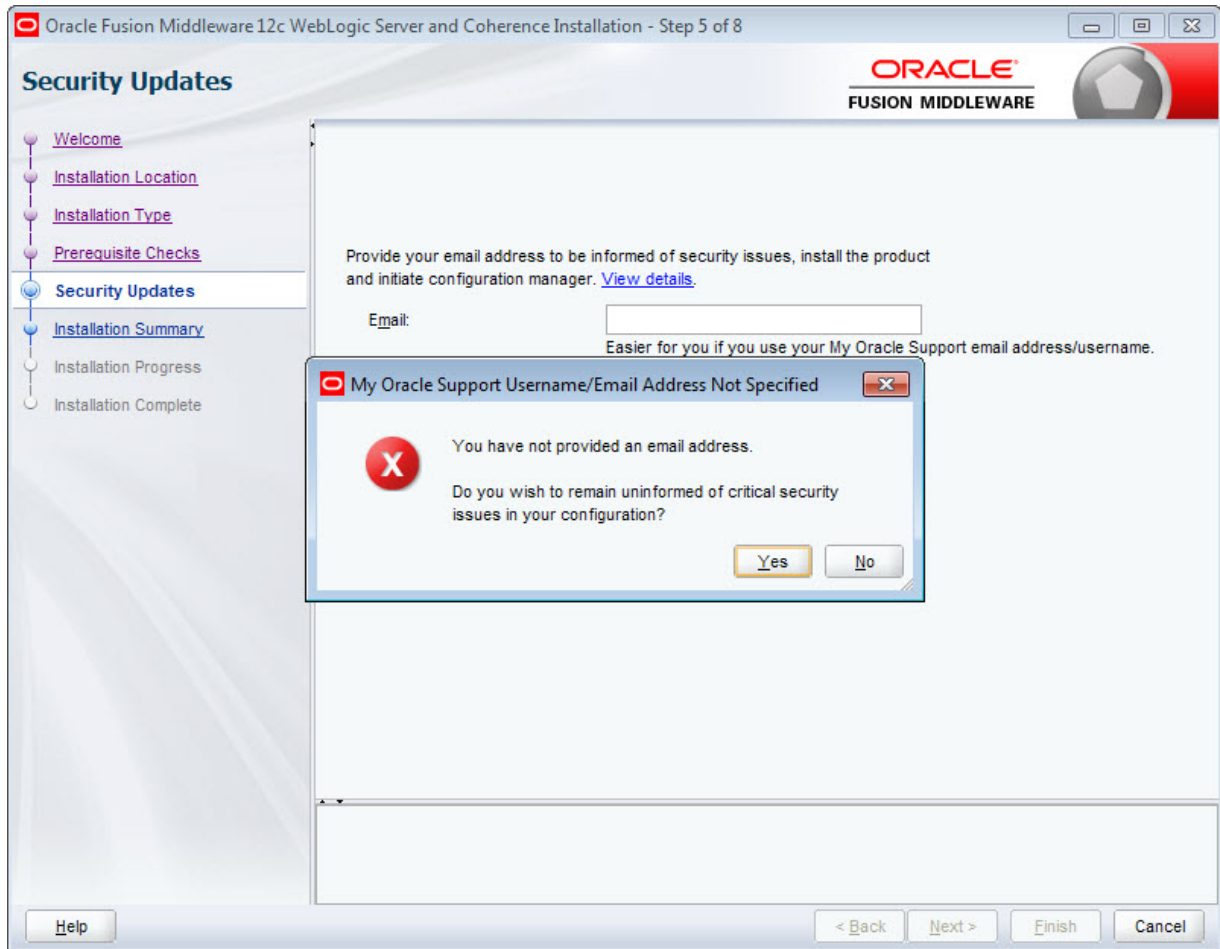
7. At the **Prerequisite Checks** screen, wait for the installer to verify your operating system and Java version, then click **Next**.



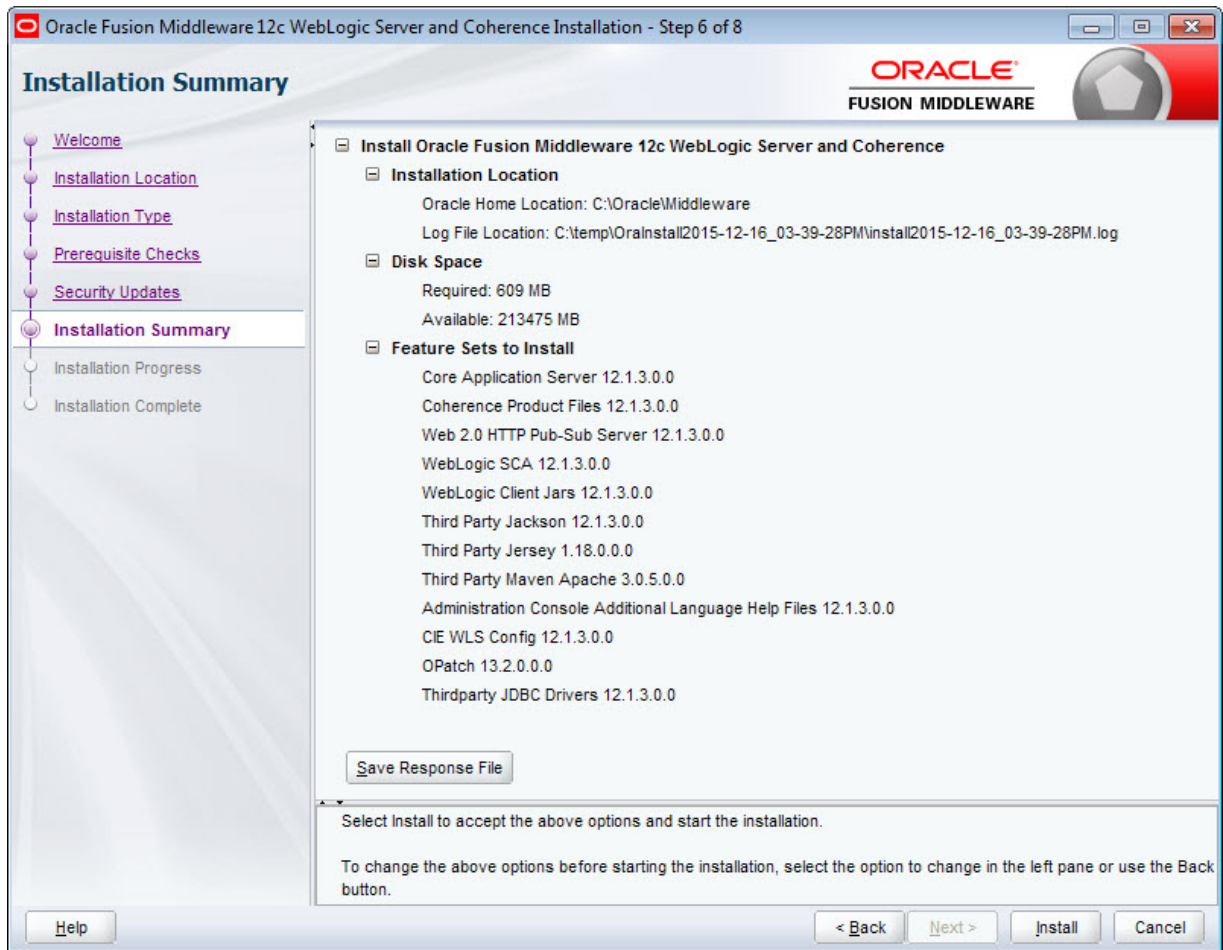


- At the **Security Updates** screen, if you want to receive security updates, enter your support details and click **Next**.

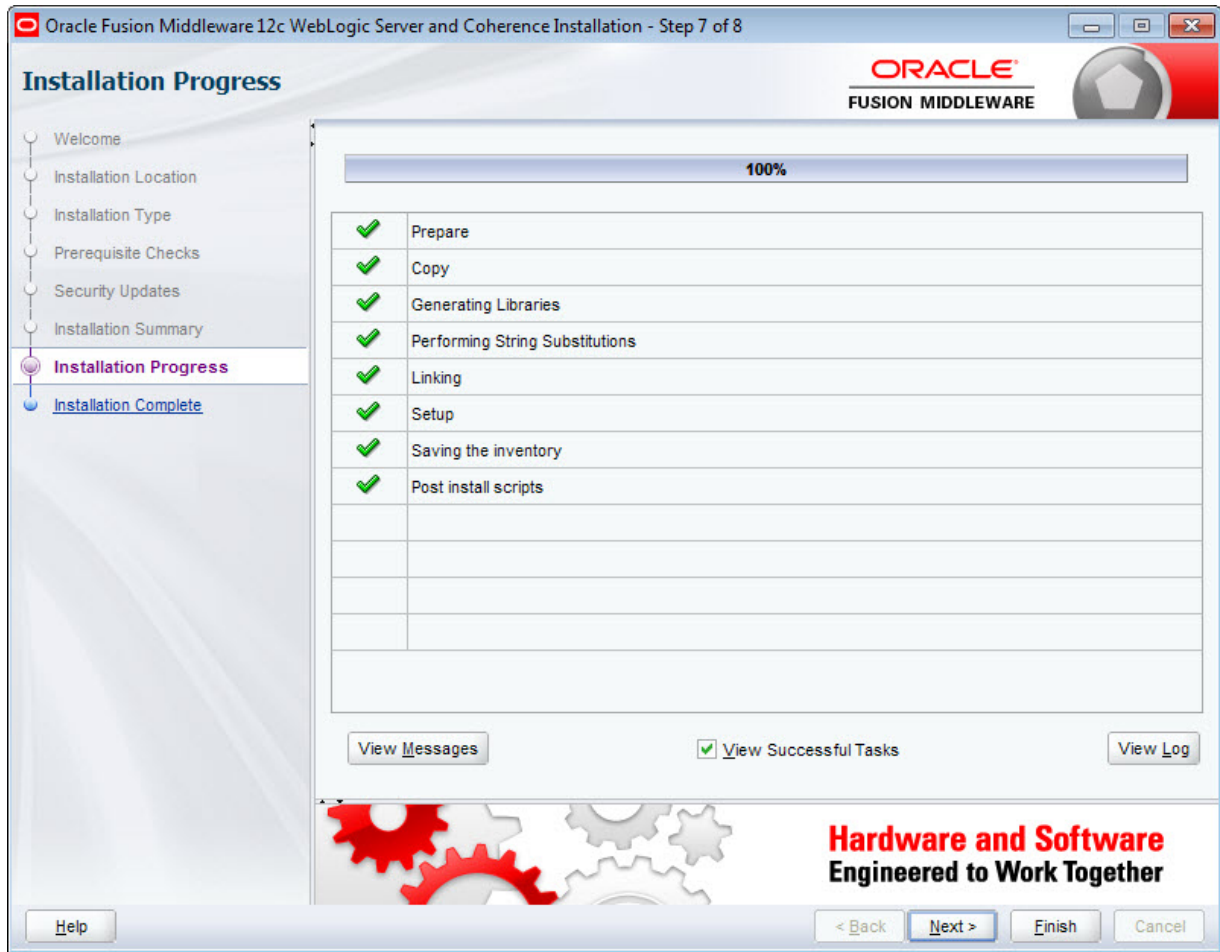
If you don't want to receive updates, uncheck **I wish to receive security updates via My Oracle Support**, click **Next**, then click **Yes** in the pop-up.



9. At the **Installation Summary** screen, verify that you're installing the correct products, then click **Install**.

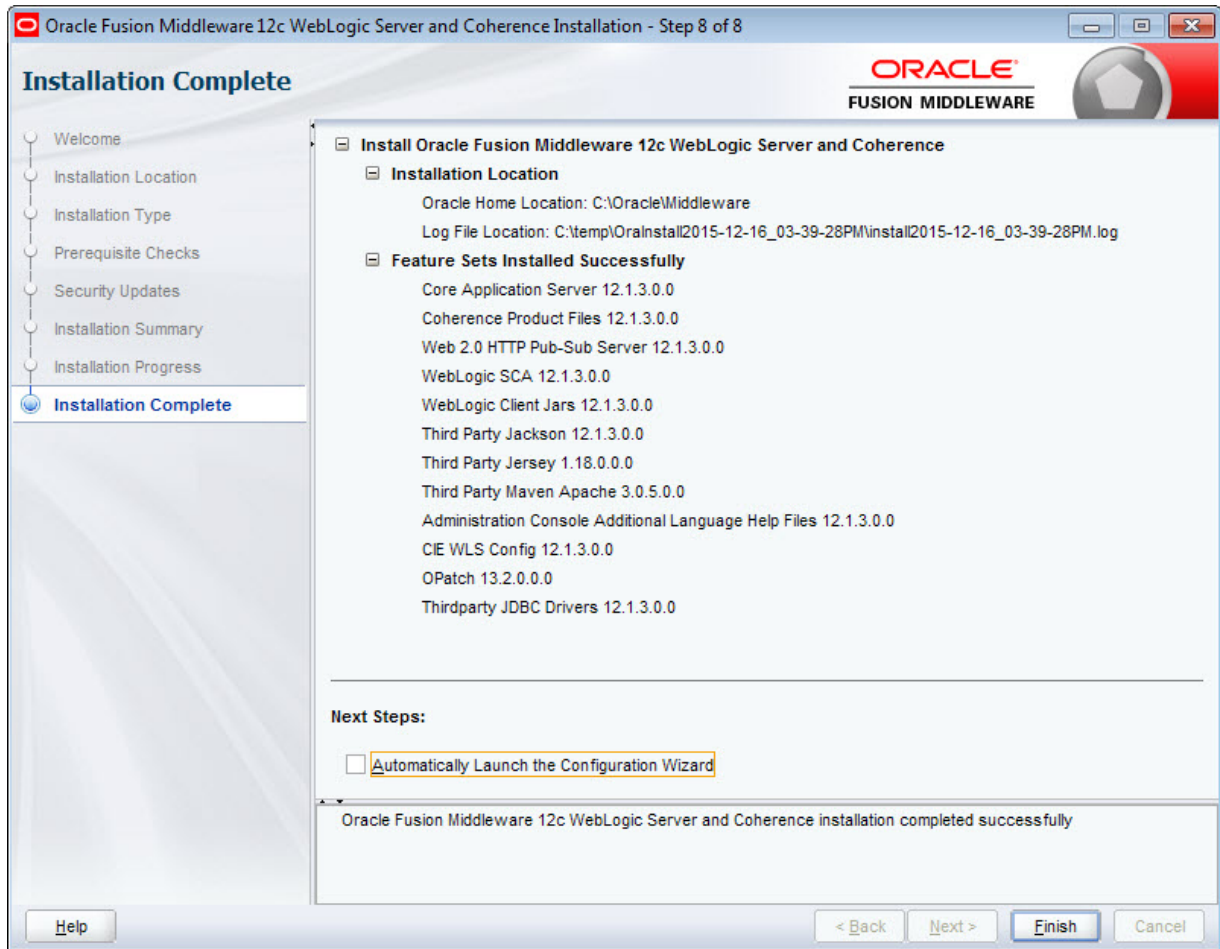


- At the **Installation Progress** screen, wait for the installation to complete, then click **Next**.



11. At the **Installation Complete** screen, uncheck **Automatically Launch the Configuration Wizard**, then click **Finish** to exit the installer.

You'll configure WebLogic Server later in the installation process.



Next, you must create a WebLogic domain for Studio.

## Creating a WebLogic domain for Studio

Studio must be installed in its own WebLogic domain. To avoid any conflicts, the domain cannot contain any other applications. Even if you are installing Studio on the same instance of WebLogic Server as Endeca Server, you must create a separate domain for Studio.

To create the WebLogic domain:

1. From a command prompt, change to the directory that contains the Configuration Wizard start-up program:
  - For Linux: `$MW_HOME/oracle_common/common/bin`
  - For Windows: `$MW_HOME\oracle_common\common\bin`

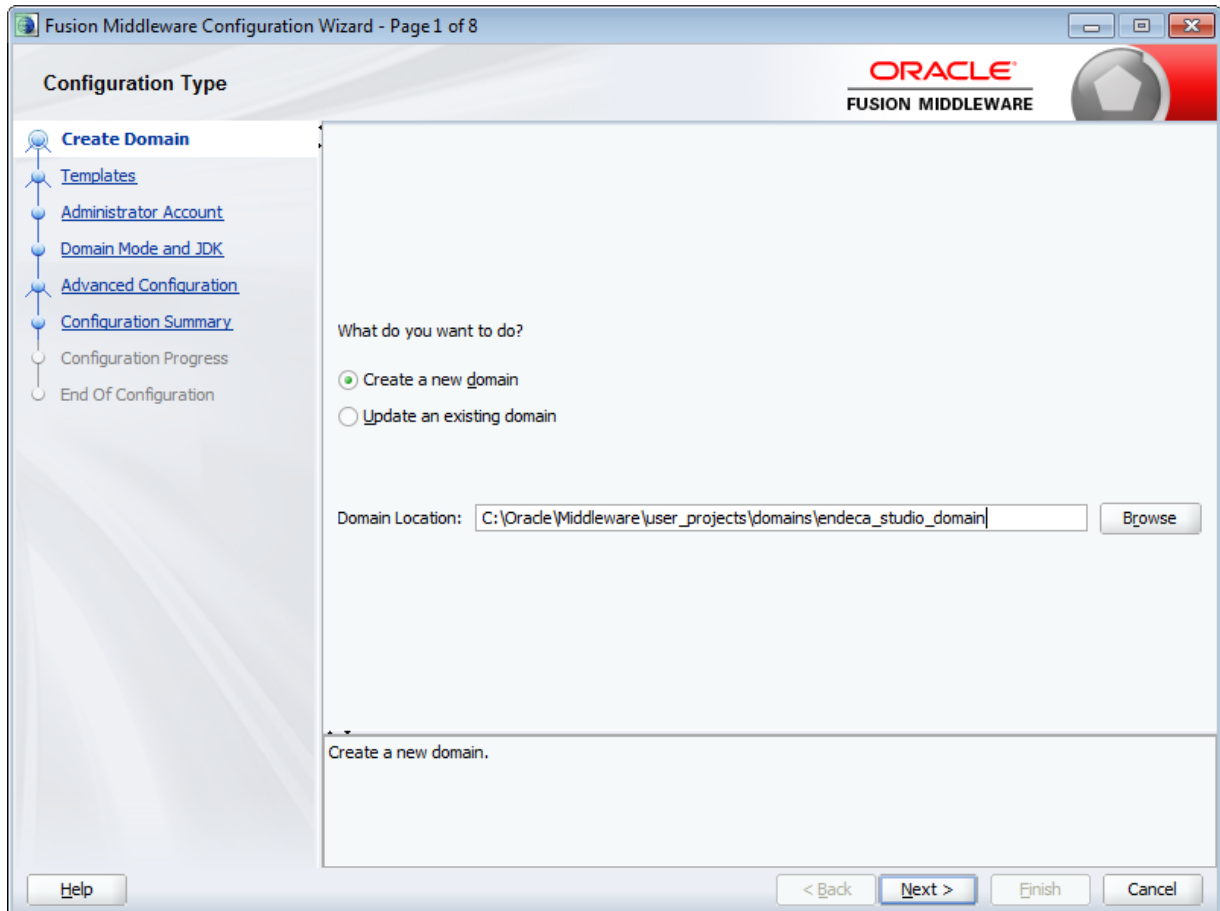
2. Run the command to start the Configuration Wizard:

- For Linux: `./config.sh`
- For Windows: `config.cmd`

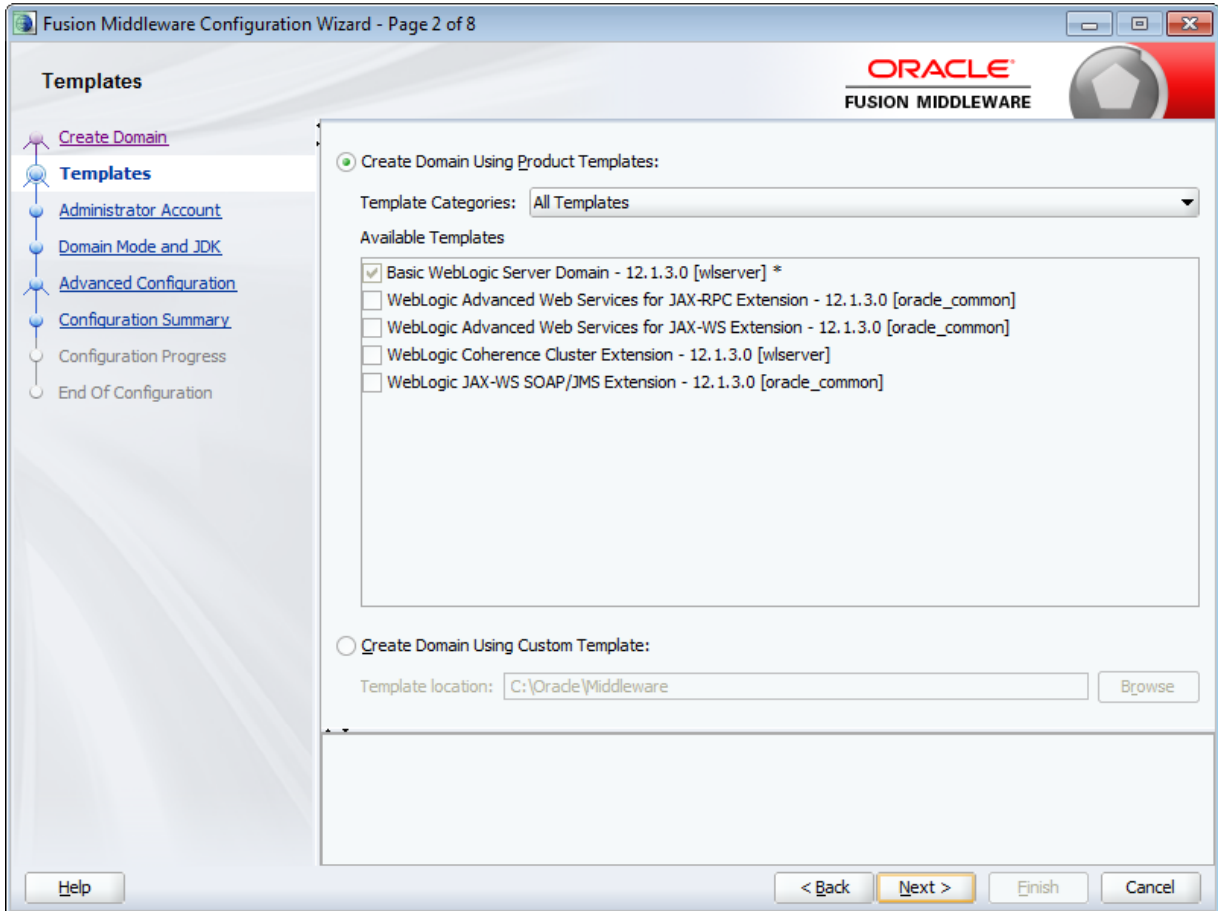
The Configuration Wizard Welcome screen is displayed.

3. At the Configuration Wizard's **Configuration Type** screen, select **Create a new domain** and either accept the default domain location or enter a new one.

Click **Next** when you're done.



4. At the **Templates** screen, leave the default template selected and click **Next**.



5. At the **Administrator Account** screen, enter a name and password for the domain administrator, then click **Next**.

The password must contain at least eight characters, including one special character or number.

Fusion Middleware Configuration Wizard - Page 3 of 8

**Administrator Account**

ORACLE  
FUSION MIDDLEWARE

Create Domain  
Templates  
**Administrator Account**  
Domain Mode and JDK  
Advanced Configuration  
Configuration Summary  
Configuration Progress  
End Of Configuration

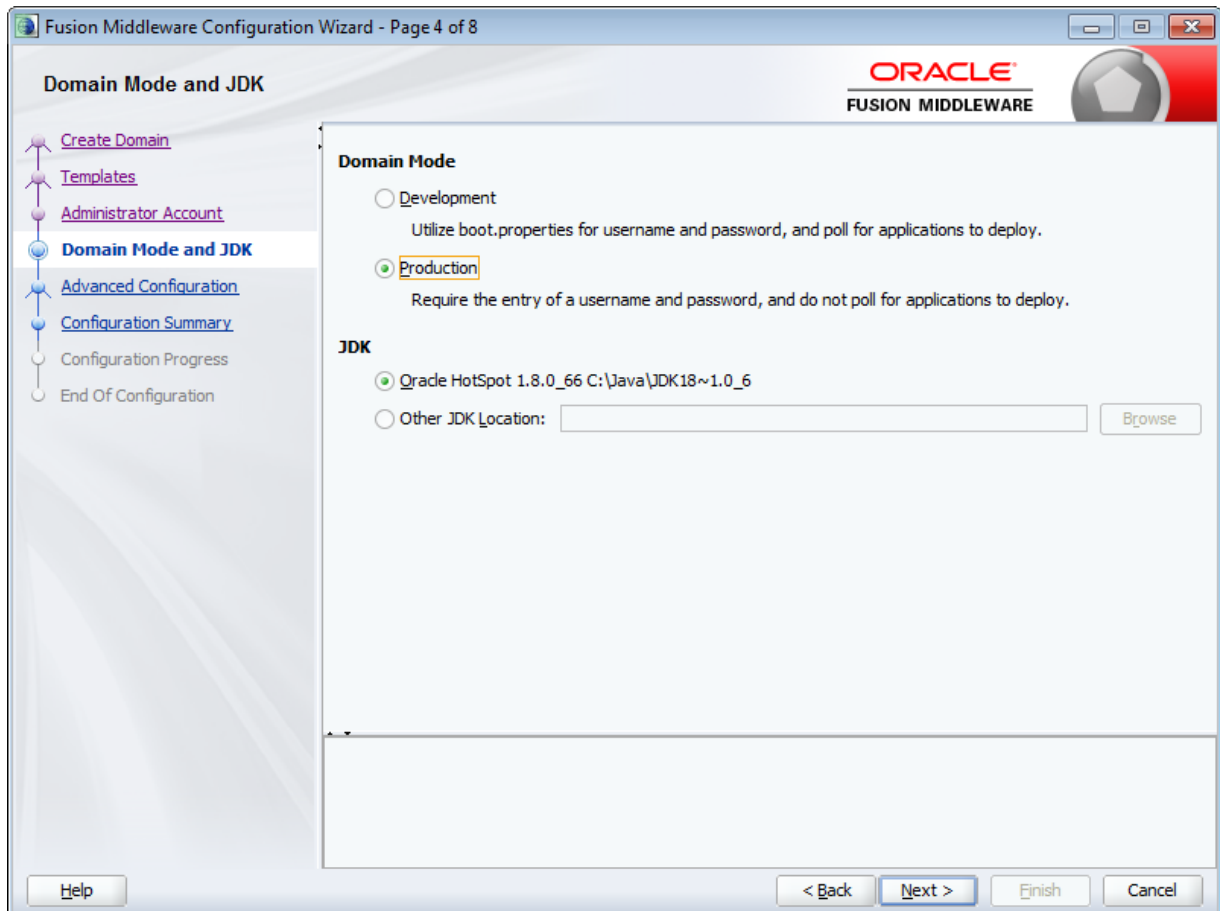
Name:

Password:

Confirm Password:

Help < Back Next > Finish Cancel

6. At the **Domain Mode and JDK** screen, select **Production** and make sure the correct JDK is selected. If the default JDK is incorrect, select **Other JDK Location** and browse to the correct one. Click **Next** when you're done.





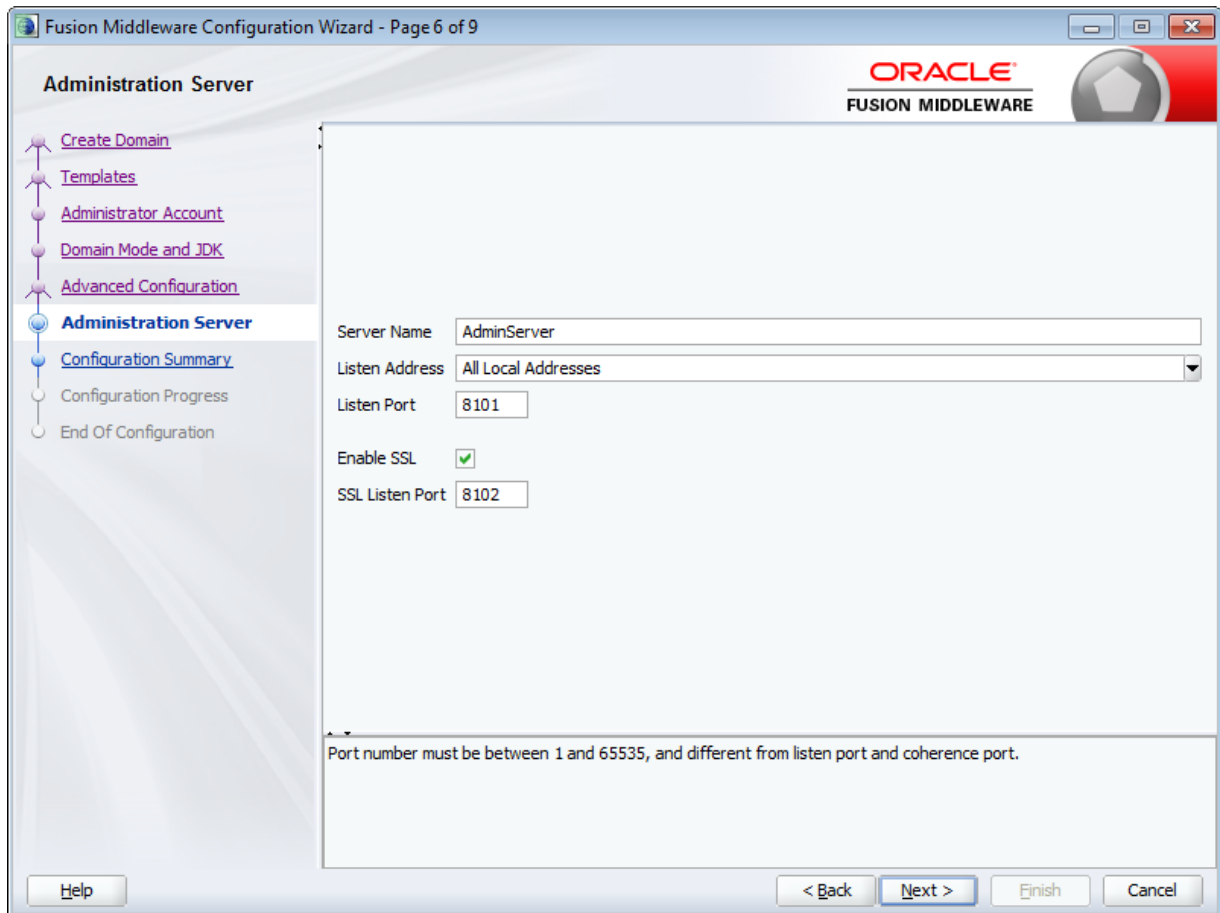
7. At the **Advanced Configuration** screen, select **Administration Server**, then click **Next**.



- At the **Advanced Configuration** screen, leave the defaults in the **Server Name** and **Listen Address** fields, and set the **Listen Port** to 8101.

If you want to enable SSL, check **Enable SSL** and set the **SSL Listen Port** to 8102.

Click **Next** when you're done.

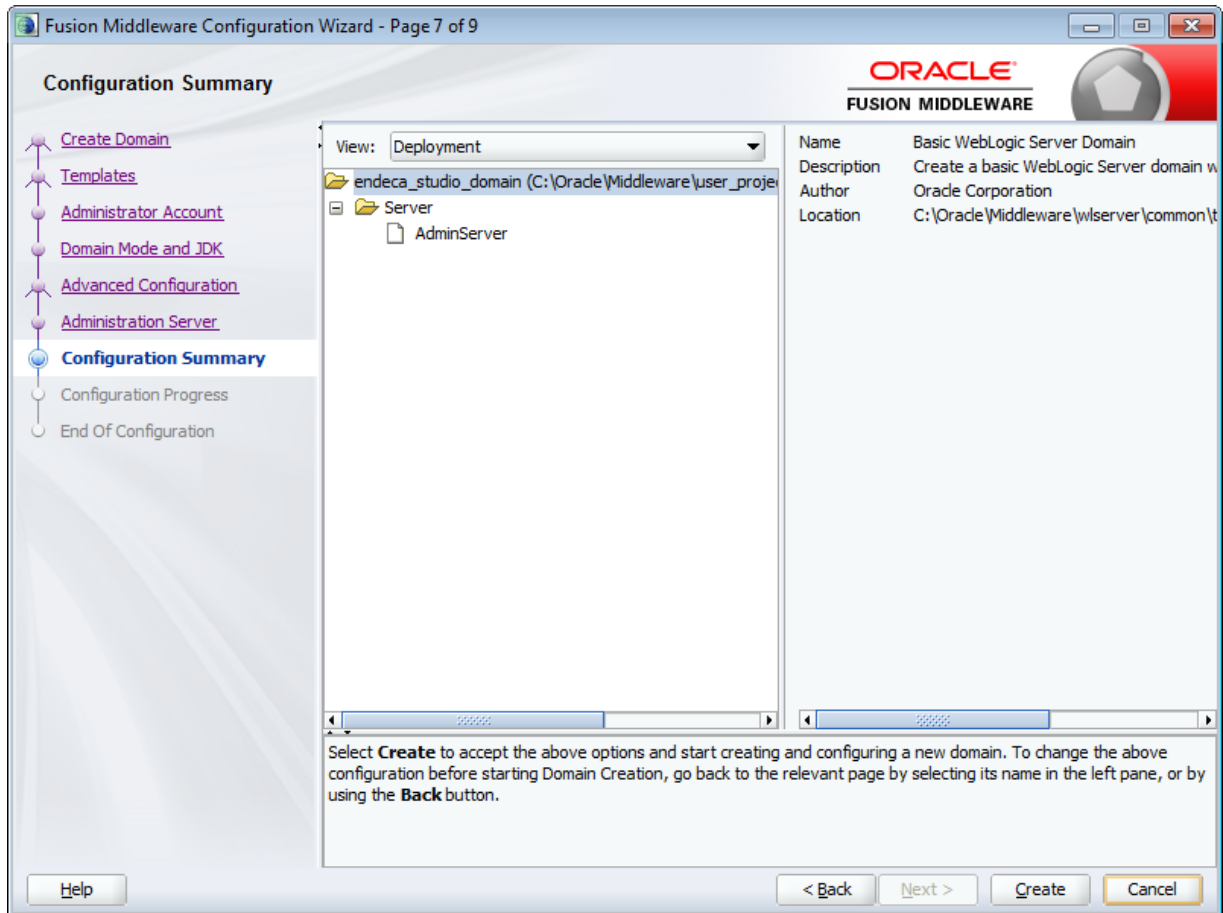


The screenshot shows the "Administration Server" configuration screen in the Fusion Middleware Configuration Wizard. The window title is "Fusion Middleware Configuration Wizard - Page 6 of 9". The Oracle Fusion Middleware logo is in the top right corner. On the left, a navigation pane lists the following steps: Create Domain, Templates, Administrator Account, Domain Mode and JDK, Advanced Configuration, Administration Server (selected), Configuration Summary, Configuration Progress, and End Of Configuration. The main area contains the following configuration fields:

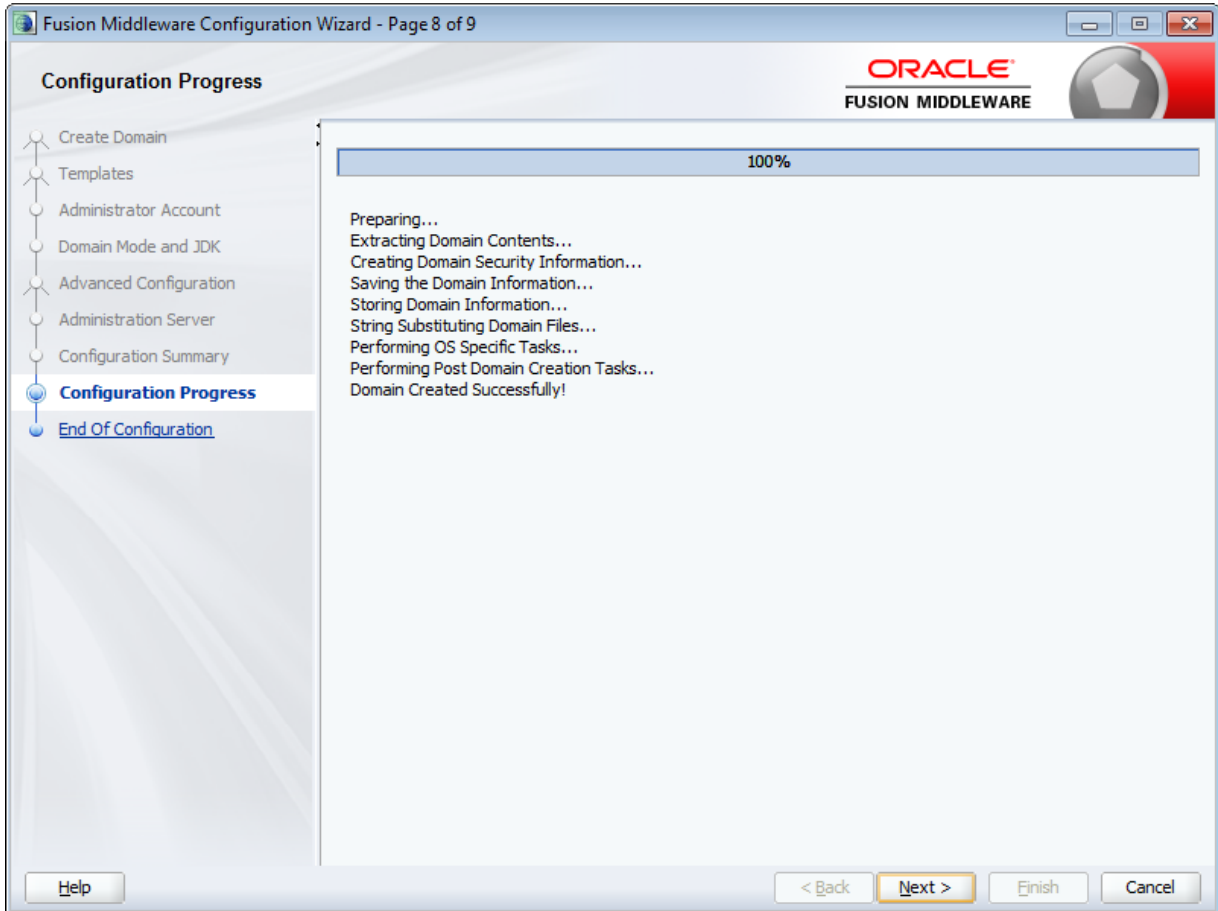
- Server Name: AdminServer
- Listen Address: All Local Addresses
- Listen Port: 8101
- Enable SSL:
- SSL Listen Port: 8102

At the bottom of the main area, a note states: "Port number must be between 1 and 65535, and different from listen port and coherence port." The bottom of the window features a "Help" button on the left and navigation buttons "< Back", "Next >", "Finish", and "Cancel" on the right.

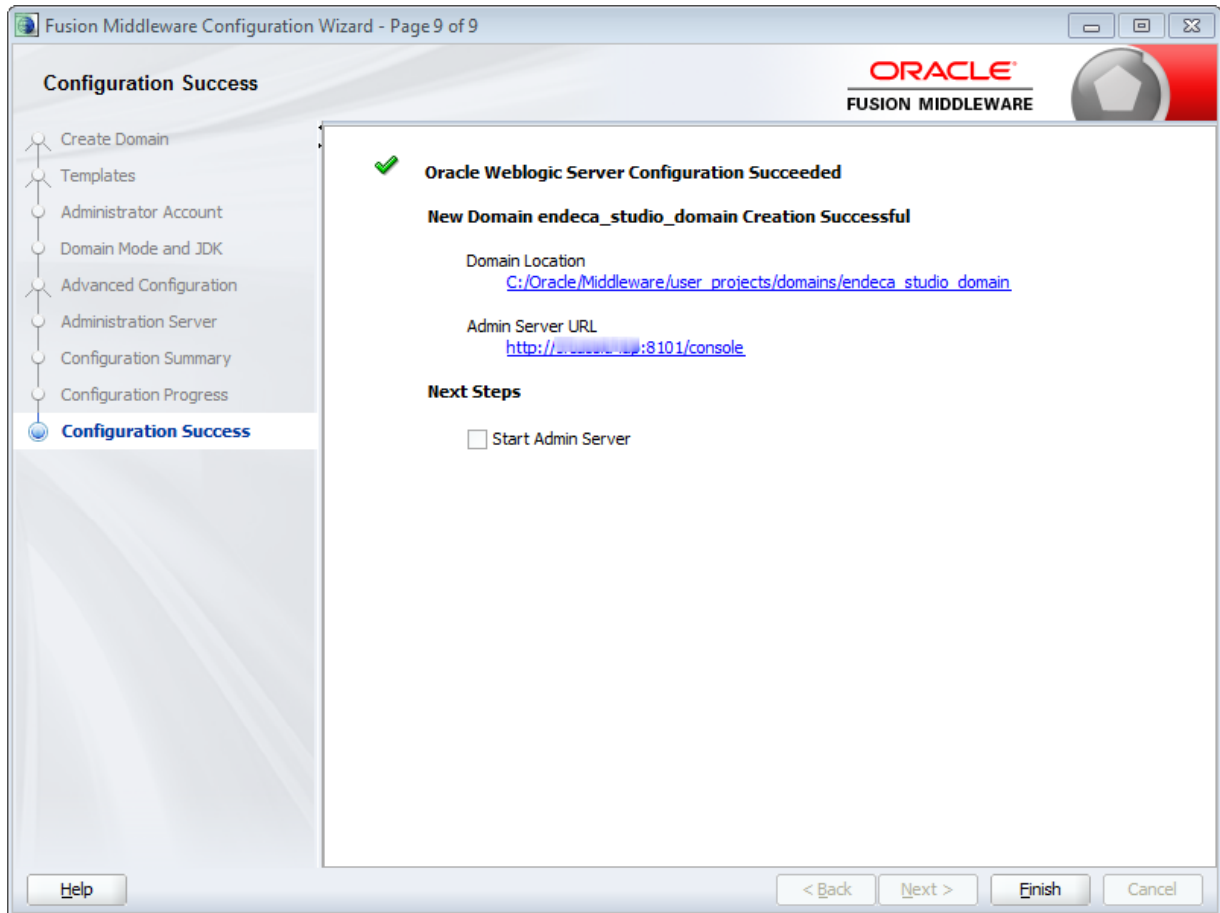
9. At the **Configuration Summary** screen, click **Create** to create the domain.



10. At the **Configuration Progress** screen, wait for the domain to be created, then click **Next**.



11. At the **Configuration Success** screen, review the domain information and click **Finish**.



12. Update the `setDomainEnv` script file (`setDomainEnv.cmd` for Windows and `setDomainEnv.sh` for Linux).

The file is located in the `bin` subdirectory of the domain directory (`$MW_HOME/user_projects/domains/endeca_studio_domain/bin/`).

- (a) Add the `JAVA_OPTIONS` argument close to the top of the file.

For `setDomainEnv.cmd` (Windows):

```
set JAVA_OPTIONS=-DUseSunHttpHandler=true -Djavax.xml.transform.TransformerFactory
=com.sun.org.apache.xalan.internal.xsltc.trax.TransformerFactoryImpl
-Djavax.xml.parsers.DocumentBuilderFactory
=com.sun.org.apache.xerces.internal.jaxp.DocumentBuilderFactoryImpl
-Djavax.xml.parsers.SAXParserFactory
=com.sun.org.apache.xerces.internal.jaxp.SAXParserFactoryImpl -Dfile.encoding=UTF8
```

For `setDomainEnv.sh` (Linux):

```
JAVA_OPTIONS="-DUseSunHttpHandler=true -Djavax.xml.transform.TransformerFactory
=com.sun.org.apache.xalan.internal.xsltc.trax.TransformerFactoryImpl
-Djavax.xml.parsers.DocumentBuilderFactory
=com.sun.org.apache.xerces.internal.jaxp.DocumentBuilderFactoryImpl
-Djavax.xml.parsers.SAXParserFactory
=com.sun.org.apache.xerces.internal.jaxp.SAXParserFactoryImpl -Dfile.encoding=UTF8"
export JAVA_OPTIONS
```

If you're copying and pasting out of this guide, make sure to remove any line breaks after pasting the text into the file.

- (b) Update all of the Java perm size arguments in the file to replace all occurrences of `-XX:MaxPermSize=128m` and `-XX:MaxPermSize=256m` with `-XX:MaxPermSize=512m`.
- (c) Update all of the memory arguments in the file to replace all occurrences of `-Xmx512m` with `-Xmx1024m`.

If you are going to change the Studio home directory from the default, then you can also add a JVM parameter to set the pointer to the custom directory. See [Changing the Studio home directory on WebLogic Server on page 43](#).

After you create the WebLogic domain for Studio, then you can deploy the Studio .ear file. See [Deploying Studio to the WebLogic domain on page 38](#).

## Deploying Studio to the WebLogic domain

To install Studio, you use the WebLogic Server Administration Console to deploy `endeca-portal-weblogic-3.2.x.ear` into the WebLogic domain.

Note that when you deploy the .ear file to a WebLogic Administration Server, and you use the default WebLogic staging mode for an Administration Server, which is "nostage", then WebLogic does not copy the file from the download directory to the server.

If the file is deleted from its current location, then Studio cannot be loaded. So with this mode, once you deploy Studio, the .ear file must remain in the same location for as long as Studio is deployed.

If you use the "stage" mode, then WebLogic does copy the file.

For more information on the WebLogic staging modes, see the [WebLogic documentation](#).

When you deploy using these instructions, the following default locations are used:

Location	Default Value
Studio home directory	<p>By default, the Studio home directory is <code>\$MW_HOME/user_projects/domains/&lt;StudioDomain&gt;/eid/studio</code>.</p> <p>If this directory already exists, then before installing, make sure that it does not already contain the following subdirectories:</p> <ul style="list-style-type: none"> <li>• <code>/data/endeca-data-sources</code></li> <li>• <code>deploy</code></li> <li>• <code>weblogic-deploy</code></li> </ul> <p>If these directories exist, then it means there is an instance of Studio installed. For the installation to work correctly, these existing directories must be removed. Before moving these directories, also make sure that the other instance has been removed.</p> <p>You can, if needed, change the location of the Studio home directory. To use a different directory for Studio home, then before you deploy Studio, you need to follow the instructions in <a href="#">Changing the Studio home directory on WebLogic Server on page 43</a>.</p>

Location	Default Value
Context path	<p>Studio is deployed to the <code>/eid/</code> context in the domain.</p> <p>When you created the domain, you should already have verified that there are no conflicts in that context.</p> <p>After you deploy Studio, you can change the context root. See <a href="#">Changing the context path for Studio on WebLogic Server on page 44</a>.</p>

To deploy `endeca-portal-weblogic-3.2.x.ear` into a production environment:

1. Set up the Studio home directory (the default is `$MW_HOME/user_projects/domains/<StudioDomain>/eid/studio`):
  - (a) In the Studio domain, create the Studio home directory.
  - (b) Extract the file `portal-ext.properties` from the top level of the Studio for WebLogic Server download .zip file.
  - (c) Place `portal-ext.properties` in the Studio home directory.
 

This file is basically a stub configuration file you use to override default Studio configuration, for example when:

    - Changing the database used for Studio
    - Overriding Studio framework settings
    - Configuring a Studio cluster
    - Configuring a reverse proxy

For the default Studio configuration, you do not need to make any changes to this file.
  - (d) In the Studio home directory, create the following subdirectories:
    - `/data/endeca-data-sources`
    - `deploy`
    - `weblogic-deploy`
2. If WebLogic Server isn't already started, then to start it, run the WebLogic Server startup script.
 

The startup script is in the domain directory.

For Windows, the script name is `startWebLogic.cmd`.

For Linux, the script name is `startWebLogic.sh`.
3. Extract `endeca-portal-weblogic-3.2.x.ear` from the Studio for WebLogic Server download .zip file.
 

To prevent the file from being inadvertently deleted after Studio is deployed, it is recommended that you place the file in a directory on the WebLogic Server. For example, you could create a directory such as `user_projects/applications/studio`.
4. Use the WebLogic Server Administration Console to deploy `endeca-portal-weblogic-3.2.x.ear` into the domain:
  - (a) Go to the Administration Console at `<hostname>:8101/console`.

- (b) Under **Helpful Tools**, click **Configure applications**.
- (c) If necessary, click **Lock & Edit** at the top left of the page.
- (d) Click **Install**.
- (e) Use the file browser to find the `endeca-portal-weblogic-3.2.x.ear` file, then click the radio button to the left of it.
- (f) Click **Next**.
- (g) On the next page, under **Choose targeting style**, make sure the **Install this deployment as an application** radio button is selected, then click **Next**.
- (h) On the next page, accept the default selections, then click **Finish**.
- (i) If necessary, click **Activate Changes**.

The deployment may take several minutes. Make sure you do not try to log in to Studio before the deployment is complete.

5. When the deployment process is completed, the status of the Studio deployment status is Prepared. To start Studio:
  - (a) In the **Deployments** list, check the checkbox for the Studio deployment.
  - (b) Click **Start > Servicing all requests**.
  - (c) Under **Start Deployments**, click **Yes**.
6. To verify that Studio has deployed successfully, in a Web browser, go to `http://<hostName>:<portNumber>/<context path>`, where `<hostName>` is the name of the server, `<portNumber>` is the port number, and `<context path>` is the context path.

For example, for a non-SSL implementation using the default context path:

```
http://localhost:8101/eid
```

You should see the Studio login page.

## Creating a Windows service for your Studio WebLogic domain

After you deploy Studio into the WebLogic domain, you can then create a Windows service for the domain.

To create and configure a Windows service for Studio:

1. In `$MW_HOME\user_projects\domains\<StudioDomain>\servers\AdminServer`, create the following files:
  - `install_studio_service.cmd`
  - `uninstall_studio_service.cmd`
2. Set the content of `install_studio_service.cmd` as follows:

```
echo off
SETLOCAL
set DOMAIN_NAME=<StudioDomain>
set SERVER_NAME=AdminServer
set USERDOMAIN_HOME=<MiddlewareHomeDirectory>\user_projects\domains\%DOMAIN_NAME%
set WL_HOME=<MiddlewareHomeDirectory>
```



```
set WLS_USER=<StudioDomainUserName>
set WLS_PW=<StudioDomainPassword>
call "%WL_HOME%\server\bin\installSvc-studio.cmd"
ENDLOCAL
```

Where:

Value	Definition
<StudioDomain>	The name of your Studio WebLogic domain.
<MiddlewareHomeDirectory>	The full path to the Middleware Home directory.
<StudioDomainUserName>	The user name for the Studio domain.
<StudioDomainPassword>	The password for the Studio domain.

The WLS\_USER and WLS\_PW parameters are used to provide the WebLogic credentials. Instead of providing these parameters, you can also use `boot.properties`.

For information on using `boot.properties` to provide the credentials, see <http://docs.oracle.com/middleware/1213/wls/START/winservice.htm#START271>.

- Set the content of `uninstall_studio_service.cmd` as follows:

```
echo off
SETLOCAL
set DOMAIN_NAME=<StudioDomain>
set SERVER_NAME=AdminServer
set WL_HOME=<MiddlewareHomeDirectory>
"%WL_HOME%\server\bin\beasvc" -remove -svcname:"beasvc %DOMAIN_NAME%_%SERVER_NAME%"
ENDLOCAL
```

Where:

Value	Definition
<StudioDomain>	The name of your Studio WebLogic domain.
<MiddlewareHomeDirectory>	The full path to the Middleware Home directory.

- Go to `%WL_HOME%\server\bin`.
- Make a copy of `installSvc.cmd`, and name it `installSvc-studio.cmd`.
- Go to `%WL_HOME%\common\bin`.
- Make a copy of `commEnv.cmd`, and name it `commEnv-studio.cmd`.
- Modify `commEnv-studio.cmd` as follows:
  - Modify the `:sun` and `:sun_prod_mode` sections of the script to set the memory parameters:

```
set JAVA_VM=-server
set MEM_ARGS=-Xms1G -Xmx1G -Xss1m -XX:MaxPermSize=512m -XX:+UseSpinning
goto continue
```

Set `-Xmx` to at least 1G.

Set `-Xms` to match `-Xmx`.

Increase `-XX:MaxPermSize` to 512m.

Set `-Xss` to 1m.

You may have to increase `-Xmx/-Xms` later on depending on your application's memory requirements.

- (b) Add the following `JAVA_OPTIONS` line to the top of `commEnv-studio.cmd`, after the comments section:

```
set
JAVA_OPTIONS=-Dweblogic.Stdout=%USERDOMAIN_HOME%\studio_service_stdout.txt "
-Dweblogic.Stderr=%USERDOMAIN_HOME%\studio_service_stderr.txt "
-Dfile.encoding=UTF8 -DUseSunHttpHandler=true
-Djavax.xml.transform.TransformerFactory
=com.sun.org.apache.xalan.internal.xsltc.trax.TransformerFactoryImpl
-Djavax.xml.parsers.DocumentBuilderFactory
=com.sun.org.apache.xerces.internal.jaxp.DocumentBuilderFactoryImpl
-Djavax.xml.parsers.SAXParserFactory
=com.sun.org.apache.xerces.internal.jaxp.SAXParserFactoryImpl
```

While the block above includes line breaks for readability, make sure to add this as a single line with the parameters separated by a space.

`-Dweblogic.Stdout` and `-Dweblogic.Stderr` redirect standard out/err for the Windows service to log files residing under `%USERDOMAIN_HOME%`.

Note that the value for `%USERDOMAIN_HOME%` is already set in `install_studio_service.cmd`.

9. Modify `installSvc-studio.cmd` to change the call statement at the top of the file to reference `commEnv-studio.cmd`:

```
call "%WL_HOME%\common\bin\commEnv-studio.cmd"
```

10. To install the service, from a command prompt launched in Run As Administrator mode in Windows, run the `install_studio_service.cmd` script.

This installs a new Windows service called `beasvc %DOMAIN_NAME%_%SERVER_NAME%` (for example, `beasvc endeca_studio_domain_AdminServer`):

Note that whenever you make changes to `commEnv-studio.cmd` or `installSvc-studio.cmd`, you must reinstall the Windows service:

- (a) Run `uninstall_studio_service.cmd`.  
 (b) Run `install_studio_service.cmd`.

11. Start the new Windows service.

If successful, the service starts up, and information is written to `%USERDOMAIN_HOME%\studio_service_stdout.txt` showing a successful startup message after a short wait.

12. Once you confirm that Studio works correctly when started using the Windows service, stop the Windows service.

Confirm that the service can successfully stop and terminate the WebLogic domain.

For additional details on creating a Windows service for WebLogic, including information on enabling graceful shutdowns for the Windows service, see the WebLogic documentation <https://docs.oracle.com/middleware/1213/wls/START/overview.htm#START112>.

## Changing the Studio home directory on WebLogic Server

For a Studio installation on WebLogic Server, the default location of the Studio home directory is `$MW_HOME/user_projects/domains/<StudioDomain>/eid/studio`, where `$MW_HOME` is the directory where you installed WebLogic Server, and `<StudioDomain>` is the domain you created for Studio. You can, if needed, use a different directory as the Studio home directory.

To provide a pointer to the new Studio home directory, you can either:

- Add a JVM parameter to `setDomainEnv.cmd` or `setDomainEnv.sh`
- Add a setting to a copy of `portal-ext.properties` in the default Studio home directory that redirects to your custom Studio home directory

To create and configure a different location for the Studio home directory:

1. Create the directory you want to use as the Studio home directory.
2. Place `portal-ext.properties` in the Studio home directory:
  - (a) Extract `portal-ext.properties` from the top level of the Studio for WebLogic Server download .zip file.
  - (b) Place `portal-ext.properties` in the Studio home directory you created.

This is the version of `portal-ext.properties` you use to override default Studio configuration, for example when:

- Changing the database used for Studio
  - Overriding Studio framework settings
  - Configuring a Studio cluster
  - Configuring a reverse proxy
3. To use the `setDomainEnv` script file (`setDomainEnv.cmd` for Windows and `setDomainEnv.sh` for Linux) to point to the new Studio home directory:
    - (a) Open the file.

The file is located in the `bin` subdirectory of the domain directory (`<MiddlewareHomeDirectory>/user_projects/domains/endeca_studio_domain/bin/`).

- (b) In the file, immediately before the `WL_HOME` setting, add the following setting:

```
EXTRA_JAVA_PROPERTIES=-Deid.studio.home=<pathToCustomStudioHome>
```

For example:

```
EXTRA_JAVA_PROPERTIES=-Deid.studio.home=/localdisk/user_projects/domains/endeca_studio_domain/eid/myhome
```

Remember that for Windows, you must escape the back slashes. For example:

```
EXTRA_JAVA_PROPERTIES=-Deid.studio.home
=C:\\Oracle\\Middleware\\user_projects\\domains\\endeca_studio_domain\\eid\\myhome
```

4. To use `portal-ext.properties` to redirect from the default Studio home directory to the custom Studio home directory:

- (a) Create a new empty text file.

This file will only contain the parameters needed to redirect to your custom Studio home directory.

- (b) Add `eid.studio.home` and `include-and-override` parameters to the file.

Set the `eid.studio.home` parameter to the full path to your custom Studio home directory.

Set the `include-and-override` parameter to the full path to the `portal-ext.properties` file in your custom Studio home directory.

For example:

```
eid.studio.home=/localdisk/user_projects/domains/endeca_studio_domain/eid/myhome
include-and-override=/localdisk/user_projects/domains/endeca_studio_domain/eid/myhome
/portal-ext.properties
```

Remember that for Windows, you must escape the back slashes. For example:

```
eid.studio.home
=C:\\Oracle\\Middleware\\user_projects\\domains\\endeca_studio_domain\\eid\\myhome
include-and-override
=C:\\Oracle\\Middleware\\user_projects\\domains\\endeca_studio_domain\\eid\\myhome\\portal
-ext.properties
```

- (c) Save the file as `portal-ext.properties` in the default Studio home directory (`<MiddlewareHomeDirectory>/user_projects/domains/<StudioDomain>/eid/studio`).
5. In your custom Studio home directory, create the following subdirectories:
    - `/data/endeca-data-sources`
    - `deploy`
    - `weblogic-deploy`
  6. You can then continue with step 2 of the Studio deployment process.

See [Deploying Studio to the WebLogic domain on page 38](#).

## Changing the context path for Studio on WebLogic Server

When you first install Studio, it uses the context path `/eid/`. After you deploy Studio, you can change to a different context path.

To change the context path, you use the deployment plan provided with Studio. By default, as an example, the deployment plan is configured to change the context path from `/eid/` to `/studio/`.

To deploy Studio to a different context path:

1. From the top level of the WebLogic .zip file, extract the file `change-context-roots-deployment-plan.xml`.

2. In the `<module-override>` section of the file:
  - (a) Verify that the value of `<module-name>` matches the actual name of the Studio WebLogic .ear file.

For example:

```
<module-override>
  <module-name>endeca-portal-weblogic-3.2.x.ear</module-name>
```

If necessary, update the value to match the file name.

- (b) Under `<variable-definition>`, edit the values of the `<variable>` elements to reflect the new context path.

Make sure to update all of the variables. There are variables for Studio as a whole as well as individual variables for each Studio component.

By default, as an example, the `<variable>` elements are configured to change the context path to `/studio/`, as shown in the following excerpt.

```
<variable>
  <name>endeca-portal-context-root</name>
  <value>/studio/</value>
</variable>
<variable>
  <name>endeca-navigation-portlet-context-root</name>
  <value>/studio/endeca-navigation-portlet/</value>
</variable>
```

Replace all instances of `/studio/` with the context path you want to use. For example, if you were changing the context path to `/newpath/`, in values of the `<variable>` elements, you would replace all instances of `/studio/` with `/newpath/`:

```
<variable>
  <name>endeca-portal-context-root</name>
  <value>/newpath/</value>
</variable>
<variable>
  <name>endeca-navigation-portlet-context-root</name>
  <value>/newpath/endeca-navigation-portlet/</value>
</variable>
```

- (c) Save the file.
3. Apply the file to your Studio deployment.
  - (a) Go to the **Administration Console** at `<hostname>:8101/console`.
  - (b) Click **Lock & Edit** at the top left of the page.
  - (c) Under **Domain Structure**, click **Deployments**.
  - (d) In the **Deployments** list, check the checkbox for your Studio deployment.
  - (e) Click the **Update** button.
  - (f) Under **Deployment plan path**, click the **Change Path** button.
  - (g) In the **Path** field, specify the full path to `change-context-roots-deployment-plan.xml`.
  - (h) Click the radio button next to the file name.
  - (i) Click **Finish**.

(j) Click **Activate Changes**.

# **Part III**

## **Installing the Provisioning Service**



## Chapter 4

# Installing the Provisioning Service

---

The Provisioning Service is installed as an application on Oracle WebLogic Server.

Separate Provisioning Service WebLogic installation packages are available for Windows and Linux. Both packages contain the following files and subdirectories:

- `eidProvisioningTemplate.jar`

This file is the domain template used in the basic installation process.

- `install.py`
- `doc`

This directory contains the release notes and *Licensing Guide* for the Provisioning Service.

Recommended practice is to install the Provisioning Service after installing Endeca Server and Studio.

In production environments, the Provisioning Service should be installed on a separate machine from other Oracle Endeca Information Discovery products, such as Endeca Server and Studio. Also, when installing the WebLogic Server for the Provisioning Service in production environments, install the additional database drivers, which includes the MySQL driver.

In other environments (evaluation, training, development, and demonstration), the Provisioning Service can be installed in the same machine and WebLogic Server as other Oracle Endeca Information Discovery components, although the Provisioning Service still requires a unique WebLogic domain.

[Overview](#)

[Running the Provisioning Service installation script](#)

[Provisioning Service data domain profile](#)

[Backend database configuration](#)

[Changing the communication security configurations](#)

## Overview

Install the Provisioning Service into its own WebLogic domain.

The process described below assumes you are installing the Provisioning Service into a new WebLogic instance for production. For evaluation, training, and development, you can install the Provisioning Service on the same WebLogic Server as Endeca Server and Studio. Note that in this case, the Provisioning Service still requires its own WebLogic domain.

To install the Provisioning Service:

1. Download and install WebLogic Server. For instructions, see [Installing WebLogic Server](#).
2. Run the Provisioning Service installation script, `install.py`.



This script creates a WebLogic domain and installs the Provisioning Service.

## Installing in silent mode

Normally, when the installation script runs, it prompts you for configuration information it requires to properly install the Provisioning Service. If you want to avoid this, however, you can add this information to the script itself before running it. The script will then execute silently.

For the full list of requires changes, see [Silent installation on page 51](#). The installation procedure below specifies when you should make them.

## Running the Provisioning Service installation script

The Provisioning Service installation script automates the process of creating a WebLogic domain and installing the Provisioning Service.

Before running the installation script, be sure you have installed the following:

- Java 7 or 8 JDK
- WebLogic Server
- Endeca Server

To run the Provisioning Service installation script:

1. Unzip the Provisioning Service installation package into a convenient directory.
2. Optionally, if you are using SSL, copy the identity and trust keystore files from the Endeca Server to the directory where you unzipped the installation package.  
The Endeca Server keystore files are stored in `$DOMAIN_HOME/config/ssl`. Copy the following files:
  - `endecaServerClientCert.ks`
  - `endecaServerTrustStore.ks`
3. If you want to run the installation script in silent mode, open it in a text editor and make the changes described in the following section.
4. Open a terminal window and change to the directory where you unzipped the installation package.
5. Enter the command `$MW_HOME\wlserver\common\bin\wlst.cmd install.py`.  
WebLogic Scripting Tool executes the installation script.
6. Either accept the default name of the Provisioning Service domain or enter a new one.
7. Enter the name of the WebLogic user for the Provisioning Service domain.
8. Enter the password for the WebLogic user for the Provisioning Service domain. This must include at least eight characters, and one or more of the characters must be a number or special character.
9. Either accept the default non-SSL port or enter a new one.

10. If you want to enable SSL for the Provisioning Service, enter `y` when prompted, then enter the following:



**Note:** All components of Oracle Endeca Information Discover must use the same SSL configuration. In other words, if you use SSL for Endeca Server, you must also use SSL for Studio and the Provisioning Service. If you do not use SSL for Endeca Server, you cannot use SSL for Studio and the Provisioning Service.

- The SSL port number
- The identity keystore password
- The trust keystore password

If you don't want to enable SSL, enter `n` at the prompt.

11. Enter the host of the Endeca Server. You can enter either the host name or the IP address.
12. The default port non-SSL port of Endeca Server is 7001. The default SSL port is 7002. Press the enter key to accept the default, or type a new port and press the enter key.
13. When prompted, specify the database configuration for the backend database.

The Provisioning Service supports three options for the backend database:

- Derby. This is a light-weight, in-memory database suitable for evaluation and test environments. This is recommended for your first installation of the Provisioning Service; however, it isn't recommended for production environments.
- Oracle
- MySQL
- To configure a Derby database, at the `Please select your database type.` prompt, enter `1` and press the return key.
- To configure an Oracle database:
  1. At the `Please select your database type.` prompt, enter `2` and press the return key.
  2. At the `Please enter your Oracle database url` prompt, enter the connect string for your database .  
  
The connect string format for an Oracle database is  
`jdbc:oracle:thin:@[HOST]:[PORT]:[SID]`.
  3. At the `Enter the JDBC database username` prompt, enter the user name of the Provisioning Service database user.
  4. At the `Enter the JDBC database password` prompt, enter the password of the Provisioning Service database user.
- To configure a MySQL database:
  1. At the `Please select your database type.` prompt, enter `3` and press the return key.
  2. At the `Please enter your MySQL database url` prompt, enter the connect string for your database .  
  
The connect string format for a MySQL database is  
`jdbc:mysql://[HOST]:[PORT]/[DATABASENAME]`.

3. At the `Enter the JDBC database username` prompt, enter the user name of the Provisioning Service database user.
4. At the `Enter the JDBC database password` prompt, enter the password of the Provisioning Service database user.

After you enter this data, the installation script runs.

The script creates a domain, installs the Provisioning Service to that domain, and configures the Provisioning Service. Finally, the script starts the Provisioning Service domain.

### *Silent installation*

### *Verifying your installation*

## Silent installation

Optionally, you can run the installation script in silent mode. This means that the installation script will execute without prompting you for configuration information.

To run a silent installation, open the installation script in any text editor and update the following properties. Be sure to save the file before closing. Afterwards, you can continue with the installation procedure described above.

Property	Description
<code>SILENT_INSTALL</code>	Determines whether the script will execute silently. Set this to <code>TRUE</code> .
<code>PROVISIONING_SERVICE_USERNAME</code>	The username for the Provisioning Service domain admin.
<code>PROVISIONING_SERVICE_PASSWORD</code>	The password for the Provisioning Service domain admin.

Property	Description
JDBC_DRIVER	<p>The driver used to connect to the backend database. The Provisioning Service includes the following drivers:</p> <ul style="list-style-type: none"> <li>• Derby: <code>org.apache.derby.jdbc.EmbeddedDriver</code></li> </ul> <p>If you specify this option, the Provisioning Service uses a light-weight, in-memory Derby database. This option is suitable for evaluation and test environments, and is the recommended option for your first installation of the Provisioning Service. The Derby database is not recommended for production environments.</p> <ul style="list-style-type: none"> <li>• Oracle: <code>oracle.jdbc.xa.client.OracleXADataSource</code></li> <li>• MySql: <code>com.mysql.jdbc.Driver</code></li> </ul>
JDBC_URL	<p>The JDBC URL used to connect to the backend database. The Provisioning Service supports the following:</p> <ul style="list-style-type: none"> <li>• <code>'jdbc:derby:memory:testDB;create=true'</code> for Derby databases. This is the default setting.</li> <li>• <code>jdbc:oracle:thin:@[HOST]:[PORT]:[SID]</code> for Oracle databases.</li> <li>• <code>jdbc:mysql://[HOST]:[PORT]/[DATABASE NAME]</code> for MySQL databases.</li> </ul>
JDBC_USER	The username the Provisioning Service connects to the backend database with.
JDBC_PASS	The password the Provisioning Service connects to the backend database with.
PROVISIONING_SERVICE_PORT	The port the Provisioning Service listens on for non-SSL requests. This defaults to 8201.
PROVISIONING_SERVICE_PORT_SSL	The port the Provisioning Service listens on for SSL requests. This defaults to 8202.
ENDECA_SERVER_ADDRESS	Identifies the machine running Endeca Server. This can either be a hostname or an IP address. The default is either <code>localhost</code> or an Endeca Server installed on the same machine as the Provisioning Service.

Property	Description
ENDECA_SERVER_PORT	The port the Endeca Server listens on. The default is 7002.
ENDECA_SERVER_SSL	Determines whether SSL is enabled on the Endeca Server. The default is <code>true</code> .
USE_SSL	Enables/disables SSL for the Provisioning Service. If you want to enable SSL, leave this set to <code>true</code> .
KEYSTORE_IDENTITY_PASS	The password you specified when creating the <code>EndecaServerClientCert.ks</code> file for the Endeca Server.
KEYSTORE_TRUST_OASS	The password you specified when creating the <code>EndecaServerTrustStore.ks</code> file for the Endeca Server.

## Verifying your installation

After you finish installing the Provisioning Service, you may want to confirm that it runs.

To confirm your installation, start a browser and in the address bar enter `http://host:port/eid-ps/status`, where `hostname` is the name or IP address of the machine where you installed the Provisioning Service and `port` is the port of the Provisioning Service.

If the Provisioning Service is functioning correctly, a page similar to the following will be returned:

### Oracle Endeca Information Discovery Provisioning Service 3.2.0.0 Revision 2689

**Endeca Server Communication**      SUCCESS

**Backend Database Access**      SUCCESS

If either check fails, the message returned is Error: check logs.

## Provisioning Service data domain profile

The installation creates an Endeca Server data domain profile for the Provisioning Service named `eid-provisioning-service`. This is a clone of the Endeca Server's default data domain profile.

If you need to change the configurations of the data domains created by the Provisioning Service, modify this data domain profile to match the configurations you need.

## Backend database configuration

The installation script automatically sets up the backend database for the Provisioning Service. If you want to use a different RDBMS, you need to change the configuration manually within WebLogic Server.

You must set up the schema for the database you want to use before using the Provisioning Service. You can do this with the following scripts:

- `oracleCreateDDL.sql`: Creates a schema for an Oracle database.
- `mysqlCreateDDL.sql`: Creates a schema for a MySQL database.

Both are located in the `eidProvisioningConfig` subdirectory of the Provisioning Service domain.

If you use a MySQL database, the value of the `max_allowed_packet` variable must be at least 32 MB.

If you need to configure WebLogic Server to use an Oracle or MySQL database:

1. Log in to WebLogic Administration Console as the user who installed the Provisioning Service. Lock and edit the Provisioning Service domain.
2. In the Domain Structure box, expand the Services node and click on **Data Sources**. Administration Console displays the Summary of JDBC Data Sources dialog.
3. Select the `jdbc/oracle.eid-ps` datasource and click **Delete**.
4. Create a new datasource with the JNDI name `jdbc/oracle.eid-ps`:
  - (a) Click **New** and choose `Generic Data Source` from the drop down menu.

The Administration Console displays the **JDBC Data Source** wizard.
  - (b) Enter a **Name** for the datasource. You can use any name you like.
  - (c) In the **JNDI Name** field, enter `jdbc/oracle.eid-ps`. You must use this name.
  - (d) In the **Database Type** drop list, select either `Oracle` or `MySQL`.
  - (e) Click **Next**.
  - (f) In the **Database Driver** drop list, choose the appropriate driver for your database.
    - If you use an Oracle database, you must use an XA-type driver.
    - If you use MySQL, choose `MySQL's Driver (Type 4) Versions: using com.mysql.jdbc.Driver`
  - (g) Click **Next**.
  - (h) If you specified MySQL, on the Transaction Options page check the **Supports Global Transactions** box and select the **Logging Last Resource** radio button. If you specified Oracle, take no actions on this page. Click **Next**.
  - (i) Consult your database administrator for the appropriate data for the rest of the wizard.
  - (j) Target the datasource to the Admin Server when asked.
  - (k) On the Summary of Data Sources page, click on the new `jdbc/oracle.eid-ps` data source. Administration Console displays the Settings for `jdbc/oracle.eid-ps2` page.
  - (l) Click the **Connection Pool** tab and expand the **Advanced** section.
  - (m) Check the **Check Connections On Reserve** box, then click **Save**.

- (n) If you specified Oracle, click the **Transactions** tab. Check the **Set XA Transaction Timeout** . In the **XA Transaction Timeout** field, enter 0. Click **Save**.
  - (o) Click **Activate Changes**.
5. Restart WebLogic Server.

## Changing the communication security configurations

When you run the installation script, you specify the communication security configuration for the Provisioning Service. You can update the configuration later if you want to change the communication security configuration.

Note that all components should use the same communication security configuration. In other words, if you use SSL for the Provisioning Service, you should also be using SSL for Endeca Service and Studio. If you choose to implement unsecured communication for the Provisioning Service, you should also implement unsecured communication for Endeca Server and Studio as well.

[Enabling SSL](#)

[Disabling SSL](#)

### Enabling SSL

The Provisioning Service supports SSL by default; however, additional implementation is required to enable SSL-secured communication between the Provisioning Service and Endeca Server and Studio.

To enable SSL for the Provisioning Service:

1. Copy the following keystore cert files from the home directory of the Endeca Server installation to the home directory of the Provisioning Service domain:
  - `endecaServerClientCert.ks`
  - `endecaServerTrustStore.ks`
2. Configure the identity and trust keystore files on the WebLogic Server:
  - (a) Start the Provisioning Service WebLogic domain.
  - (b) In a browser, go to the Administration Console for your server and log in.
  - (c) On the **Keystores** tab, change value in the **Keystores** field to `Custom Identity` and `Custom Trust`. Save this change.
  - (d) In the **Custom Identity Keystore** field, enter the path to the identity keystore.
  - (e) In the **Custom Identity Keystore Type** field, enter `JKS`.
  - (f) In the **Custom Identity Keystore Passphrase** and **Confirm Custom Identity Keystore Passphrase** fields, enter the keystore passphrase used to generate the key on the Endeca Server.
  - (g) In the **Custom Trust Keystore** field, enter the path to the trust keystore.
  - (h) In the **Custom Trust Keystore Type** field, enter `JKS`.
  - (i) In the **Custom Trust Keystore Passphrase** and **Confirm Custom Trust Keystore Passphrase** fields, enter the keystore passphrase used to generate the key on the Endeca Server.

3. Configure the WebLogic Server SSL parameters:
  - (a) On the Administration Console of your WebLogic Server, click the **SSL** tab.
  - (b) The value of the **Private Key Alias** must be `Oracle Endeca Server Client Certificate`. (This value should default from the keystore.)
  - (c) In the **Private Key Passphrase** and **Confirm Private Key Passphrase** fields, enter the passphrase you entered when creating the certificates on the Endeca Server installation.
  - (d) Click **Save**.

## Disabling SSL

The default configuration of the Provisioning Service assumes secure communication with Endeca Server and with Studio over SSL. You must modify this configuration if you want to permit non-SSL communication.

To configure non-secure communication, open `plan.xml` and make the following changes:

- Set `endeca-server-security-enabled` to `false`.
- Set `transport-guarantee` to `NONE`.
- Set `protected-url-pattern` to `/DISABLED`.

Also, confirm that `endeca-server-ws-port` is set to `7001` (the non-secure port); the value of this parameter defaults to `7002`, which is the secure port.

## Connecting the Provisioning Service to Studio

In order for users to be able to create data sets by uploading files or using the Data Source Library, you must configure the connection between the Provisioning Service and Studio.

If the connection configuration is invalid, then when users create an application, the file upload and Data Source Library options are not displayed. Users also cannot create available data sources in the Data Source Library.

The connection is configured using a JSON string. The general connection settings are:

Setting	Description
<code>server</code>	The name of the server on which the Provisioning Service is running.
<code>port</code>	The port on which the Provisioning Service is listening.

For example:

```
{
  "server": "ps.us.acme.com",
  "port": "7004"
}
```



If the Provisioning Service is installed on a context path other than `endeca-server`, then you need to add a setting to provide the context path:

Setting	Description
<code>contextPath</code>	<p>The context path for the Provisioning Service.</p> <p>If this setting is not present, then the context path defaults to <code>eid-ps</code>, which is the Provisioning Service's default context path.</p> <p>To specify a root context, set the value to either <code>""</code> or <code>"/</code>.</p>

For example:

```
{
  "server": "ps.us.acme.com",
  "port": "7004",
  "contextPath": "my-ps-path"
}
```

By default, the Provisioning Service has SSL enabled, and the configuration must include the `sslConfig` setting, which contains the following settings:

Setting	Description
<code>caFile</code>	<p>The name of the truststore file for the SSL connection to the Provisioning Service.</p> <p>This is the truststore file from the secured Endeca Server configuration. For the default configuration, the file is <code>endcaServerTrustStore.ks</code>.</p>
<code>caPassword</code>	<p>The password for the truststore file for the SSL connection to the Provisioning Service.</p> <p>This is the password generated during the Endeca Server installation.</p> <p>Note that once you save the Provisioning Service configuration, the value of <code>caPassword</code> is masked as <code>*****</code>. The value also is encrypted in the Studio database.</p> <p>When you edit the Provisioning Service connection, you must re-type the actual password value before saving. Otherwise, Studio uses the masking asterisks as the password value.</p>
<code>certFile</code>	<p>The name of the keystore file for the SSL connection to the Provisioning Service.</p> <p>This is the keystore file from the secured Endeca Server configuration. For the default configuration, the file is <code>endecaServerClientCert.ks</code>.</p>

Setting	Description
certPassword	<p>The password for the keystore file for the SSL connection to the Provisioning Service.</p> <p>This is the password generated during the Endeca Server installation.</p> <p>Note that once you save the Provisioning Service configuration, the value of certPassword is masked as *****. The value also is encrypted in the Studio database.</p> <p>When you edit the Provisioning Service connection, you must re-type the actual password value before saving. Otherwise, Studio uses the masking asterisks as the password value.</p>

For example:

```
{
  "server": "ps.us.acme.com",
  "port": "7004",
  "contextPath": "my-ps-path",
  "sslConfig": {
    "caFile": "endecaServerTrustStore.ks",
    "caPassword": "*****",
    "certFile": "endecaServerClientCert.ks",
    "certPassword": "*****"
  }
}
```

The **Control Panel** includes a **Provisioning Service** page you use to configure the connection.

To configure the Provisioning Service connection:

1. From the administrator menu, select **Control Panel**.
2. In the **Control Panel** menu, click **Provisioning Service**.
3. On the **Provisioning Service** page, update the placeholder configuration with the connection information for the Provisioning Service.
4. Click **Save**.



## Chapter 5

# Connecting the Provisioning Service to Studio

In order for users to be able to create data sets by uploading files or using the Data Source Library, you must configure the connection between the Provisioning Service and Studio.

If the connection configuration is invalid, then when users create an application, the file upload and Data Source Library options are not displayed. Users also cannot create available data sources in the Data Source Library.

The connection is configured using a JSON string. The general connection settings are:

Setting	Description
server	The name of the server on which the Provisioning Service is running.
port	The port on which the Provisioning Service is listening.

For example:

```
{
  "server": "ps.us.acme.com",
  "port": "7004"
}
```

If the Provisioning Service is installed on a context path other than `endeca-server`, then you need to add a setting to provide the context path:

Setting	Description
contextPath	The context path for the Provisioning Service.  If this setting is not present, then the context path defaults to <code>eid-ps</code> , which is the Provisioning Service's default context path.  To specify a root context, set the value to either "" or "/".

For example:

```
{
  "server": "ps.us.acme.com",
  "port": "7004",
  "contextPath": "my-ps-path"
}
```

By default, the Provisioning Service has SSL enabled, and the configuration must include the `sslConfig` setting, which contains the following settings:

Setting	Description
<code>caFile</code>	<p>The name of the truststore file for the SSL connection to the Provisioning Service.</p> <p>This is the truststore file from the secured Endeca Server configuration. For the default configuration, the file is <code>endecaServerTrustStore.ks</code>.</p>
<code>caPassword</code>	<p>The password for the truststore file for the SSL connection to the Provisioning Service.</p> <p>This is the password generated during the Endeca Server installation.</p> <p>Note that once you save the Provisioning Service configuration, the value of <code>caPassword</code> is masked as <code>*****</code>. The value also is encrypted in the Studio database.</p> <p>When you edit the Provisioning Service connection, you must re-type the actual password value before saving. Otherwise, Studio uses the masking asterisks as the password value.</p>
<code>certFile</code>	<p>The name of the keystore file for the SSL connection to the Provisioning Service.</p> <p>This is the keystore file from the secured Endeca Server configuration. For the default configuration, the file is <code>endecaServerClientCert.ks</code>.</p>
<code>certPassword</code>	<p>The password for the keystore file for the SSL connection to the Provisioning Service.</p> <p>This is the password generated during the Endeca Server installation.</p> <p>Note that once you save the Provisioning Service configuration, the value of <code>certPassword</code> is masked as <code>*****</code>. The value also is encrypted in the Studio database.</p> <p>When you edit the Provisioning Service connection, you must re-type the actual password value before saving. Otherwise, Studio uses the masking asterisks as the password value.</p>

For example:

```
{
  "server": "ps.us.acme.com",
  "port": "7004",
  "contextPath": "my-ps-path",
  "sslConfig": {
    "caFile": "endecaServerTrustStore.ks",
    "caPassword": "*****",
    "certFile": "endecaServerClientCert.ks",
    "certPassword": "*****"
  }
}
```

The **Control Panel** includes a **Provisioning Service** page you use to configure the connection.

To configure the Provisioning Service connection:

1. From the administrator menu, select **Control Panel**.
2. In the **Control Panel** menu, click **Provisioning Service**.
3. On the **Provisioning Service** page, update the placeholder configuration with the connection information for the Provisioning Service.
4. Click **Save**.

# **Part IV**

## **After You Install**



## Chapter 6

# Getting Started with Studio

---

After installing Studio, use the following steps to launch and configure Studio and begin to work with it.

[Logging in to Studio](#)

[Displaying the Studio Control Panel](#)

## Logging in to Studio

After you complete the Studio installation, you can start and log in to the application.

To log in to Studio:

1. Start your application server.
2. In your Web browser, go to the portal.
3. Log in using the default login and password:

Field	Value
<b>Login:</b>	admin@oracle.com
<b>Password:</b>	Welcome123

You are immediately prompted to change the password. The new password must contain:

- At least 6 characters
- At least one non-alphabetic character

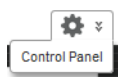
## Displaying the Studio Control Panel

The **Control Panel** provides access to Studio administrative functions, including functions to configure Studio settings, manage available data, manage users, and monitor performance.

For information on using the **Control Panel**, see the *Studio Administration and Customization Guide*.

To display the **Control Panel**:

1. Click the administrator menu icon at the top right of the Studio user interface.
2. From the administrator menu, select **Control Panel**.









## Chapter 7

---

# Changing the Database Used to Store Studio Application Data

You may choose to select a different database from the default settings.

[About using a different database](#)

[Overview of the process for switching to a different database](#)

## About using a different database

Studio uses a relational database to store configuration and state, such as component configuration, user permissions, and system settings.

Note that Studio has been tested on MySQL and Oracle 11g. Other databases are expected to work but have not been explicitly tested.

Also, when migrating to a new database, make sure that the current data does not conflict with the new database's schema. For example, different databases may have different constraints for column sizes.

## Overview of the process for switching to a different database

Here is a high-level overview of the steps involved in switching from the default database to the production RDBMS of your choice.

The details vary from database to database.

To switch to a different database:

1. Install and verify that your database is working.
2. Create a new empty database or schema for the application.  
When you create the new database, make sure to use UTF-8 encoding.
3. Create a database user for the application.
4. Grant that user access to the appropriate database/schema, with privileges to create tables, alter schemas, and so on in that database.  
Ensure that the user has remote access from the application servers.
5. Stop Studio if it is running.

6. Next, edit the JDBC section of the `portal-ext.properties` file to change the database connection to your database.

(a) The default version of the file has the following settings enabled. Comment out these settings.

```
# Hypersonic
#
jdbc.default.driverClassName=org.hsqldb.jdbcDriver
jdbc.default.url=jdbc:hsqldb:${eid.studio.home}/data/hsqldb/portal
jdbc.default.username=sa
jdbc.default.password=
```

(b) To configure the connection to your database, the recommended option is to uncomment and configure the `jdbc.default.jndi.name` parameter.

```
# Set the JNDI name to lookup the JDBC data source. If none is set,
# then the portal will attempt to create the JDBC data source based on the
# properties prefixed with "jdbc.default.".
#
#jdbc.default.jndi.name=jdbc/LiferayPool
```

Using the JNDI name is more secure than providing the connection settings, which include the user name and password, in `portal-ext.properties`.

If you do not use the JNDI name, Studio logs a warning.

Before you can use the JNDI name, you must first configure a JDBC data source within your application server. The steps to configure a JDBC data source will vary based on your application server. See your application server documentation for details.

Make sure that the value of `jdbc.default.jndi.name` matches exactly the JNDI name you assign to the data source.

7. For a MySQL database, if users will be uploading custom images to use for application components, then you need to increase the value of the `max-allowed-packet` variable.

The default is 1MB. However, Studio allows users to upload images up to 3MB. So you should change it to a value greater than 3MB, such as 4MB.

If you do not change the value of the variable, then when users try to upload an image greater than 1MB, a JDBC error, "Packet for query is too large", is returned.

8. Start Studio. Monitor the logs to check for error messages while Studio connects to the database and creates the tables.
9. After the tables have been created and you have verified that Studio is running, you may remove the user's alter table privileges.

Note that you may have to restore these later if you upgrade Studio or install components that require schema changes.



## Chapter 8

---

# Using Studio with a Reverse Proxy

Studio can be configured to use a reverse proxy.

[About reverse proxies](#)

[Example sequence for a reverse proxy request](#)

[Recommendations for reverse proxy configuration](#)

[Reverse proxy configuration options for Studio](#)

## About reverse proxies

A reverse proxy provides a more secure way for users to get access to application servers.

[What is a reverse proxy?](#)

[Types of reverse proxies](#)

## What is a reverse proxy?

A reverse proxy retrieves resources on behalf of a client from one or more servers, and then returns these resources to the client as though they came from the server itself.

A reverse proxy is located between the client and the proxied server(s). Clients access content through the proxy server. The reverse proxy server assumes the public hostname of the proxied server. The hostname(s) of the actual/proxied servers are often internal and unknown to the client browser.

Some common reasons for implementing a reverse proxy include:

- Security or firewalling
- SSL termination
- Load balancing and failover
- Resource caching/acceleration
- URL partitioning

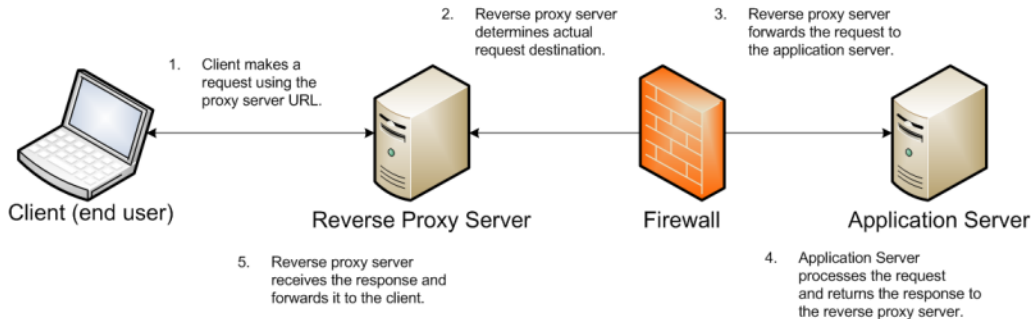
## Types of reverse proxies

Reverse proxies may be either be devices/appliances or specially configured web servers.

A very popular software-based reverse proxy is the Apache HTTP Server configured with the `mod_proxy` module. Many commercial web servers and reverse proxy solutions are built on top of Apache HTTP Server, including Oracle HTTP Server.

## Example sequence for a reverse proxy request

Here is an example of the typical sequence for a request processed using a reverse proxy server.



1. The client makes a request to the public URL.

For this example, for a Studio application, the request URL might be something like `http://mystudio/eid/web/myapp`, using the default port 80.

The hostname resolves to the address of the reverse proxy server. The reverse proxy is listening on this address and receives the request.

2. The reverse proxy server analyzes the URL to determine where the request needs to be proxied to.

A reverse proxy might use any part of the URL to route the request, such as the protocol, host, port, path, or query-string. Typically the path is the main data used for routing.

The reverse proxy configuration rules determine the outbound URL to send the request to. This destination is usually the end server responsible for serving the content. The reverse proxy server may also rewrite parts of the request. For example, it may change or make additions to path segments.

Reverse proxies can also add standard or custom headers to the request.

For example, the URL `http://mystudio/web/myapp` might be proxied to `http://studioserver1:8080/eid/web/myapp`. In this case:

- The hostname of the target server is `studioserver1`
- The port is changed to 8080
- The context path `/eid/` is added

3. The reverse proxy server sends the request to the target server.
4. The target server sends the response to the reverse proxy server.
5. The reverse proxy server reads the request and returns it to the client.

## Recommendations for reverse proxy configuration

Here are some general configuration recommendations for setting up a reverse proxy.

[Preserving HTTP 1.1 Host: headers](#)

[Enabling the Apache ProxyPreserveHost directive](#)

## Preserving HTTP 1.1 Host: headers

HTTP 1.1 requests often include a `Host:` header, which contains the hostname from the client request. This is because a server may use a single IP address or interface to accept requests for multiple DNS hostnames.

The `Host:` header identifies the server requested by the client. When a reverse proxy proxies an HTTP 1.1 request between a client and a target server, when it makes the request, it must add the `Host:` header to the outbound request. The `Host:` header it sends to the target server should be the same as the `Host:` header it received from the client. It should not be the `Host:` header that would be sent if accessing the target server directly.

When the application server needs to create an absolute, fully-qualified URL, such as for a redirect URL or an absolute path to an image or CSS file, it must provide the correct hostname to the client to use in a subsequent request.

For example, a Java application server sends a client-side redirect to a browser (HTTP 302 Moved). It uses the `ServletRequest.getServerName()` method to fetch the hostname in the request, then constructs a `Host:` header.

The URL sent by the client is `http://mystudio/web/myapp`. The actual internal target URL generated by the reverse proxy will be `http://studioserver1:8080/eid/web/myapp`.

If there is no specific configuration for the target server, then if the reverse proxy retains the `Host:` header, the header is:

```
Host: http://mystudio
```

If the reverse proxy does not retain the `Host:` header, the result is:

```
Host: http://studioserver1:8080
```

In the latter case, where the header uses the actual target server hostname, the client may not have access to `studioserver1`, or may not be able to resolve the hostname. It also will bypass the reverse proxy on the next request, which may cause security issues.

If the `Host:` header cannot be relied on as correct for the client, then it must be configured specifically for the web or application server, so that it can render correct absolute URLs.

Most reverse proxy solutions should have a configuration option to allow the `Host:` header to be preserved.

## Enabling the Apache ProxyPreserveHost directive

The `ProxyPreserveHost` directive is used to instruct Apache `mod_proxy`, when acting as a reverse proxy, to preserve and retain the original `Host:` header from the client browser when constructing the proxied request to send to the target server.

The default setting for this configuration directive is `Off`, indicating to not preserve the `Host:` header and instead generate a `Host:` header based on the target server's hostname.

Because this is often not what is wanted, you should add the `ProxyPreserveHost On` directive to the Apache HTTPD configuration, either in `httpd.conf` or related/equivalent configuration files.

## Reverse proxy configuration options for Studio

Here are some options for configuring reverse proxy for Studio.

[Simple Studio reverse proxy configuration](#)

[Studio reverse proxy configuration without preserving Host: headers](#)

[Configuring Studio to support an SSL-enabled reverse-proxy](#)

### Simple Studio reverse proxy configuration

Here is a brief overview of a simple reverse proxy configuration for Studio. The configuration preserves the `Host:` header, and does not use SSL or path remapping. Studio only supports matching context paths.

In this simple configuration:

- A reverse proxy server is in front of a single Studio application server.
- The reverse proxy server is configured to preserve the `Host:` header.
- The context paths match.
- Neither the reverse proxy nor the application server is configured for SSL.

With this setup, Studio should be able to be accessed correctly using the reverse proxy without additional configuration.

### Studio reverse proxy configuration without preserving Host: headers

If a reverse proxy fronting Studio does not preserve the `Host:` header, and instead makes a request with a `Host:` header referring to the target application server, Studio and its application server receive an incorrect hostname. This causes Studio to generate absolute URLs that refer to the proxied application server instead of to the reverse proxy server.

If the reverse proxy cannot be configured to preserve the `Host:` header, you must configure a fixed hostname and port. To do this, you can either:

- Configure the application server to have a fixed hostname and port
- Use `portal-ext.properties` to configure Studio with a fixed hostname and port

### Configuring a fixed hostname for the application server

For WebLogic, you set up a virtual host with the fixed hostname and port.

### Configuring Studio with a fixed hostname

To configure Studio with a fixed hostname and port, add the following properties to `portal-ext.properties`:

```
web.server.host=<hostname of reverse proxy server>
web.server.http.port=<port of reverse proxy server>
```

## Configuring Studio to support an SSL-enabled reverse-proxy

If Studio is installed behind a reverse proxy that has SSL capabilities, and the client SSL is terminated on the reverse proxy, you must configure Studio to set the preferred protocol to HTTPS.

To do this, in `portal-ext.properties`, set `web.server.protocol` to `https`.

```
web.server.protocol=https
```



## Chapter 9

---

# Creating a Studio Cluster

For a larger production environment, you may want to configure a cluster of Studio instances.

*[About Studio clustering](#)*

*[Setting up the cluster](#)*

## About Studio clustering

Studio allows you to create clusters of Studio instances. In a cluster, changes made to one instance are automatically made to the other instances. For a large production environment, using clustering provides redundancy and support for higher throughput, allowing for more concurrent users.

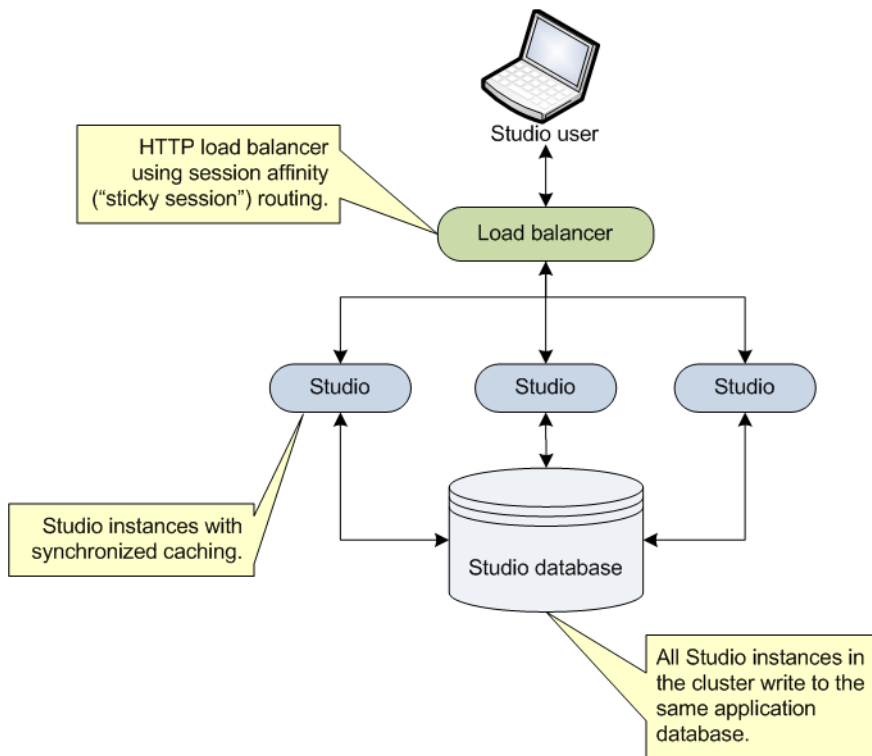
The cluster is made up of Studio instances configured to write to the same application database.

The Studio instances also must be configured to use synchronized caching, so that information cached on one instance is available to all of the other instances in the cluster. Studio uses Ehcache ([www.ehcache.org](http://www.ehcache.org)), which uses RMI (Remote Method Invocation) multicast to notify each member of the cluster when the cache has been updated.

While there are multiple ways to cluster an application, based on the application server, Studio supports using an HTTP load balancer in front of the Studio instances. The load balancer must use session affinity (also



known as "sticky session") load balancing. If a member of the cluster is down, the load balancer routes requests to another instance in the cluster.



## Setting up the cluster

To configure a cluster, you connect each instance to the same database, and then configure a clustered cache for those instances.

*Installing the Studio instances*

*Configuring synchronized caching for the Studio instances*

## Installing the Studio instances

Each instance in the cluster is first installed as a standalone instance.

## Connecting each instance to the same Studio database

Each instance in the Studio cluster must be connected to the same Studio application database. For details on changing the database system used to store the Studio application database, see [Changing the Database Used to Store Studio Application Data on page 64](#). Optionally, you could use a clustered database configuration. For clustering, Oracle 11g uses RAC and MySQL has MySQL Cluster. For details on setting up a clustered database configuration, see the documentation for your database system.

## Using the same configuration for each instance

In a clustered configuration, each instance should have the same configuration, to ensure that users have the same experience no matter which instance in the cluster they are connected to.

Most of the application settings are stored in the database. Because each instance writes to the same database, those settings remain constant among the cluster instances.

Also make sure that each instance has the same settings in `portal-ext.properties`. This includes any framework settings that you set in the file instead of from the **Control Panel** user interface.

## Configuring synchronized caching for the Studio instances

Studio instances in a cluster must use synchronized caching.

### About synchronized caching

Synchronized caching ensures that the information cached by one Studio instance is available to all of the instances in the cluster.

This reduces the number of times each instance needs to query the Studio database, which allows for faster response times and better performance. Studio uses Ehcache ([www.ehcache.org](http://www.ehcache.org)) for caching synchronization.

### Updating `portal-ext.properties` to enable synchronized caching

The `portal-ext.properties` file for each instance includes commented-out settings for synchronizing the caches.

For each instance in the cluster, uncomment the following clustering settings in `portal-ext.properties`. You should be able to use the default values provided.

```
##
## Cluster
##
# Uncomment the following properties to enable clustering
# Note: Clustering will not work with Hypersonic.  Configure a common database for all cluster nodes.

#net.sf.ehcache.configurationResourceName=/ehcache/hibernate-clustered.xml
#ehcache.multi.vm.config.location=/ehcache/liferay-multi-vm-clustered.xml
#org.quartz.jobStore.isClustered=true
```

The settings are:

Setting	Description
<code>net.sf.ehcache.configurationResourceName</code>	<p>The name and location of the XML configuration file for Hibernate caching. Hibernate is used by Studio to read from and write to the Studio application database.</p> <p>In the default <code>portal.properties</code> file, the configuration file is set to <code>hibernate.xml</code>, to implement caching in a non-clustered implementation.</p> <p>When you uncomment this property in <code>portal-ext.properties</code>, which changes the configuration file to <code>hibernate-clustered.xml</code>, then Hibernate synchronizes the cache with the other members of the cluster.</p>
<code>ehcache.multi.vm.config.location</code>	<p>The name and location of the XML configuration file for Ehcache.</p> <p>In the default <code>portal.properties</code> file, the file is set to <code>liferay-multi-vm.xml</code>, to implement caching in a non-clustered implementation.</p> <p>When you uncomment this property in <code>portal-ext.properties</code>, which changes the configuration file to <code>liferay-multi-vm-clustered.xml</code>, then the cache is synchronized with the other members of the cluster.</p>
<code>org.quartz.jobStore.isClustered</code>	<p>Enables clustering on the built-in Quartz job scheduling engine.</p>

Both configuration files are configured to automatically detect the other members of the cluster, and to use IP address 233.0.0.1 and port 4446 to send the updated cache information.

### Customizing the clustered cache configuration files

The default versions of the clustered cache configuration files should work in most cases. However, you can if needed create and deploy customized versions.

The most likely customization that might be needed would be to the IP address and port number configured near the top of each file:

```
<cacheManagerPeerProviderFactory
    class="net.sf.ehcache.distribution.RMICacheManagerPeerProviderFactory"
```

```

        properties="peerDiscovery=automatic,multicastGroupAddress=230.0.0.1,multicastGroupPort
=4446,timeToLive=1"
        propertySeparator=","
    />

```

If you make any changes to these configuration files, make sure to make the same changes for all of the instances in the cluster.

To customize the clustered cache configuration files:

1. Extract the default files from the ehcache directory in `portal-impl.jar`.

The file is in the `WEB-INF\lib` directory, which is located in `endeca-portal-versionNumber.war`, which is in `endeca-portal-weblogic-versionNumber.ear`

2. Make the necessary updates to the files.

To ensure that Studio uses the correct files, you may want to rename the customized files to something like:

- `hibernate-clustered-custom.xml`
- `liferay-multi-vm-clustered-custom.xml`

3. To deploy the customized files in Weblogic:

- (a) Undeploy `endeca-portal-weblogic-versionNumber.ear`.

Use the appropriate method to undeploy the file based on whether you auto-deployed the `.ear` file or installed it.

- (b) Update `endeca-portal-weblogic-versionNumber.ear` to add a subdirectory `APP-INF/classes/ehcache/` that contains the customized XML files.

- (c) Redeploy the updated `.ear` file.

4. If needed, update `portal-ext.properties` to reflect the customized file names:

```

net.sf.ehcache.configurationResourceName=/ehcache/hibernate-clustered-custom.xml
ehcache.multi.vm.config.location=/ehcache/liferay-multi-vm-clustered-custom.xml

```

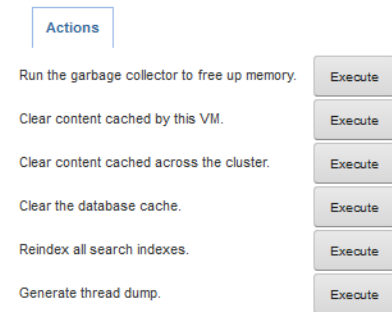
## Clearing the cache for a cluster

As part of troubleshooting issues with a clustered implementation, you can clear the cache. From the Studio **Control Panel**, you can clear the cache for either the current instance or for the entire cluster.

To clear the cache:

1. From the administrator menu, select **Control Panel**.
2. On the **Control Panel** menu, in the **Server** section, click **Server Administration**.

3. At the bottom of the page, on the **Actions** tab:



- To clear the cache for the current instance only, click the **Execute** button next to **Clear content cached by this VM.**
- To clear the cache for the entire cluster, click the **Execute** button next to **Clear content cached across the cluster.**

# **Part V**

## **Uninstalling Oracle Endeca Information Discovery**



## Chapter 10

---

# Uninstalling Studio and the Provisioning Service

This chapter describes how to uninstall the Studio and Provisioning Service modules of an Oracle Endeca Information Discovery Studio installation. For information on uninstalling Oracle Endeca Server, see the *Oracle Endeca Server Installation Guide*. For information on uninstalling Oracle Endeca Information Discovery Integrator ETL, see the *Integrator ETL Installation Guide*.

[Uninstalling Studio](#)

[Uninstalling the Provisioning Service](#)

## Uninstalling Studio

To uninstall Studio, remove the packages and directories that you installed.

## Uninstalling the Provisioning Service

To uninstall Provisioning Service, follow standard WebLogic procedures for uninstalling a Web application.

For details, see "Delete Web applications" in the *WebLogic Server Administration Console Online Help* .

# Index

## C

- clustering, Studio
  - about 72
  - clearing the cache 76
  - customizing the cache configuration 75
  - enabling synchronized caching 74
  - installing instances 73
- Control Panel, displaying in Studio 63

## D

- database
  - changing from default 65
  - configuring Provisioning Service for production 54
- data domain profile 53

## I

- installation
  - Provisioning Service 48
  - Studio on Oracle WebLogic Server 19

## L

- languages, supported 8

## M

- MySQL database 54

## O

- Oracle database 54

## P

- Provisioning Service 49
  - connecting Studio to 56, 59
  - installing 48
  - silent installation 51
  - SSL 55
  - system requirements 14
  - uninstalling 79

## R

- reverse proxy, using with Studio 67

## S

- SSL
  - Provisioning Service 55
- Studio
  - changing from the default database 65
  - clustering, about 72
  - clustering, about synchronized caching 74
  - clustering, clearing the cache 76
  - clustering, customizing the cache configuration 75
  - clustering, enabling synchronized caching 74
  - clustering, installing instances 73
  - connecting to the Provisioning Service 56, 59
  - Control Panel, displaying 63
  - downloading 15
  - installing WebLogic Server 20
  - Oracle WebLogic Server installation steps 19
  - running as Windows service (WebLogic) 40
  - starting 63
  - supported languages 8
  - system requirements 12
  - uninstalling 79
  - using a reverse proxy 67
- system requirements
  - Provisioning Service 14
  - Studio 12

## U

- uninstalling
  - Provisioning Service 79
  - Studio 79

## W

- WebLogic Server
  - context path, changing for Studio 44
  - creating the Studio domain 28
  - deploying Studio 38
  - installing 20
  - running Studio as a Windows Service 40
  - Studio home directory, changing 43
- Windows Service for Studio (WebLogic) 40