

Oracle® Endeca Information Discovery Studio

Studio Installation Guide

Version 3.1.0 Rev. B • December 2013

Copyright and disclaimer

Copyright © 2003, 2014, 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

Copyright and disclaimer	2
Preface	6
About this guide	6
Who should use this guide	6
Conventions used in this document	6
Contacting Oracle Customer Support	7
Part I: Before You Begin	
Chapter 1: Overview of Studio and the Installation Process	9
About Oracle Endeca Information Discovery Studio	9
About the Provisioning Service	10
Other modules in Oracle Endeca Information Discovery	10
Recommended installation order for Studio and its associated modules	11
Chapter 2: System Requirements	13
Studio system requirements	13
Provisioning Service system requirements	15
Chapter 3: Downloading the Studio and Provisioning Service Software	17
Part II: Installing Studio	
Chapter 4: About the Studio Installation Process	20
Chapter 5: Installing Studio Using the Tomcat Bundle for Windows	21
Installing the Windows Tomcat bundle	21
Changing the context root for the Windows Tomcat bundle	22
Chapter 6: Installing Studio Using the Tomcat Bundle for Linux	24
Installing the Linux Tomcat bundle	24
Changing the context root in the Linux Tomcat bundle	25
Chapter 7: Installing Studio on Tomcat 6.0	27
About installing on Tomcat 6.0	27
Installing Tomcat and deploying the dependency libraries	28
Modifying the Tomcat configuration to work with Studio	29
Deploying and starting Studio	30
Chapter 8: Running Studio Under Tomcat as a Windows Service	33
About running Studio as a Windows service	33
Obtaining the service installer files	33

Configuring the service	34
Installing and starting the service	35
Troubleshooting the service installation	36
Chapter 9: Installing Studio on Oracle WebLogic Server 11gR1 (10.3.6)	38
About installing Studio on WebLogic Server	38
Downloading and installing the Sun JDK	39
Downloading and installing WebLogic Server	42
Creating and configuring a WebLogic domain for Studio	48
Deploying Studio to the WebLogic domain	53
Creating a Windows service for your Studio WebLogic domain	58
Changing the Studio home directory on WebLogic Server	60
Changing the context path for Studio on WebLogic Server	62
Chapter 10: Troubleshooting Known Studio Issues	64
Changing the JavaScript timeout value on Internet Explorer 8	64
Increasing the Tomcat connector keepAliveTimeout for Internet Explorer 8	65
Chapter 11: Getting Started with Studio	66
Starting and logging in to Studio	66
Displaying the Studio Control Panel	66
Chapter 12: Changing the Database Used to Store Studio Application Data	68
About using a different database	68
Overview of the process for switching to a different database	68
Chapter 13: Using Studio with a Reverse Proxy	70
About reverse proxies	70
What is a reverse proxy?	70
Types of reverse proxies	70
Example sequence for a reverse proxy request	71
Recommendations for reverse proxy configuration	71
Preserving HTTP 1.1 Host: headers	72
Enabling the Apache ProxyPreserveHost directive	72
Reverse proxy configuration options for Studio	73
Simple Studio reverse proxy configuration	73
Studio reverse proxy configuration without preserving Host: headers	73
Configuring Studio to support an SSL-enabled reverse-proxy	74
Chapter 14: Creating a Studio Cluster	75
About Studio clustering	75
Setting up the cluster	76
Installing the Studio instances	76
Configuring synchronized caching for the Studio instances	77
About synchronized caching	77
Updating portal-ext.properties to enable synchronized caching	77
Customizing the clustered cache configuration files	78
Clearing the cache for a cluster	79

Part III: Installing the Provisioning Service

Chapter 15: Installing the Provisioning Service	82
Provisioning Service installation process	82
Downloading and installing Oracle ADF Runtime	83
Running the Provisioning Service installation script	89
Running the Provisioning Service installation script in silent mode	91
Checking the Provisioning Service installation	93
Provisioning Service data domain profile	94
Backend database configuration	94
Changing communication security configurations for the Provisioning Service	95
Implementing SSL communication with Endeca Server and Studio	95
Configuring the Provisioning Service for non-secure communication	96
Chapter 16: Configuring the Studio Connection to the Provisioning Service	97

Part IV: Uninstalling Oracle Endeca Information Discovery

Chapter 17: Uninstalling Oracle Endeca Information Discovery Studio	101
Uninstalling Studio	101
Uninstalling the Provisioning Service	101

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
User Interface Elements	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 3.1.0 also includes the following separately-downloaded modules:

Module and Version	Description
Oracle Endeca Server 7.6.0	<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 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
Oracle Endeca Information Discovery Integrator 3.1.0	<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"> • Integrator ETL. Used to build graphs to load source data into and configure the schema for an Endeca Server data domain. • Information Discovery connectors. Integrator ETL components used to perform various data ingest operations on Endeca data domains. • Integrator ETL Server. Used to run graphs in an enterprise-wide environment. In this environment, different users and user groups can access and run the graphs. • Integrator Acquisition System. 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. <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 3.1.0 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 [About the Studio Installation Process on page 19](#)).

4. Provisioning Service (See [Installing the Provisioning Service on page 81](#)).

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

The system requirements listed in this section are for Studio and the Provisioning Service only. 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](#)

Studio system requirements

In addition to these Studio system requirements, the Studio installation instructions also assume that you have a running Oracle Endeca Server at which to point Studio. This version of Studio is compatible with the 7.6.x version of Oracle Endeca Server.

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
Oracle Linux	Oracle Linux Release 6 for x64 While Oracle Linux 6 is recommended, Studio is also supported on Oracle Linux 5 (version 5 for x64), update 7 or higher, running on x64 processors. Only the Red Hat Compatible Kernel is supported.

Platform	Description
Linux RHEL	<ul style="list-style-type: none"> • Red Hat Enterprise Linux Server (version 6 for x64) running on x64 processors. • Red Hat Enterprise Linux Advanced Platform (version 6 for x64) running on x64 processors. <p>While Red Hat Linux 6 is recommended, Studio is also supported on Red Hat Linux version 5 for x64. For best performance on Red Hat Linux version 5 (Server and Advanced), we recommend the latest version of RHEL 5.</p>
Windows 2008	Windows Server 2008 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	<ul style="list-style-type: none"> • Tomcat 6. The Studio Tomcat bundle includes Tomcat 6.0.37. • Oracle WebLogic Server 11gR1 (10.3.6)
Java	<p>Sun Java 6, update 18 or greater. We recommend that you use the latest generally available Java 6 patch, 1.6.0_45.</p> <p>The Studio Tomcat bundle includes the 64-bit version 1.6.0_51.</p>
Database system	<ul style="list-style-type: none"> • MySQL 5.1 • Oracle 11g
Browser	<ul style="list-style-type: none"> • Firefox ESR on Windows. Firefox is the recommended browser for Studio. • Internet Explorer 8 (with compatibility mode disabled) on Windows. Running Internet Explorer 8 in compatibility mode is not supported. Also note that Internet Explorer 8 is usually slower than Firefox, and may throw script warnings on scripts that run a long time. • Internet Explorer 9 on Windows • 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.

Software	Supported Versions
Browser plugin	<p>For users on Internet Explorer 8, Adobe Flash 10.0 or greater, for displaying charts. For other browsers, charts are displayed using HTML5.</p> <p>We recommend using the most recent version of Flash supported by the browser.</p>

Alternative database support

The Studio server uses a relational database to store configuration and state. Hypersonic is used by default, but is not recommended for production use due to performance issues. Studio has been tested on MySQL and Oracle 11g. 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.

For information on changing the database used, see [Changing the Database Used to Store Studio Application Data on page 67](#).

Provisioning Service system requirements

Here are the system requirements for the Provisioning Service.

The installation instructions assume a running Endeca Server is available to which Provisioning Service can connect and which Provisioning Service will provision with data uploaded by the user. This version of Provisioning Service is compatible with Oracle Endeca Server Version 7.6.0.

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
Oracle Linux	<ul style="list-style-type: none"> Oracle Linux 6 (version 6 for x64) running on x64 processors. Oracle Linux 5 (version 5 for x64) running on x64 processors. Only the Red Hat Compatible Kernel is supported.
Linux RHEL	<ul style="list-style-type: none"> Red Hat Enterprise Linux Server (version 6 for x64) running on x64 processors. Red Hat Enterprise Linux Advanced Platform (version 6 for x64) running on x64 processors. While Red Hat Linux 6 is recommended, Studio is also supported on Red Hat Linux version 5 for x64. For best performance on Red Hat Linux version 5 (Server and Advanced), we recommend the latest version of RHEL 5.
Windows	Windows Server 2008 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 11gR1 (10.3.6)
Java	Sun Java 6, update 18 or greater, 64-bit version
RDBMS	Oracle 11g



Chapter 3

Downloading the Studio and Provisioning Service Software

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

For details on downloading the Oracle Endeca Server, see the *Oracle Endeca Server Installation Guide*.

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

To download the Studio software:

1. Log in to <https://edelivery.oracle.com>.
2. Accept the terms and restrictions.
3. On the **Media Pack Search** page:
 - (a) From the **Select a Product Pack** drop-down list, select **Oracle Endeca**.
 - (b) From the **Platform** drop-down list, select the platform you are installing on.
 - (c) Click **Go**.

The list of media packs for the selected product and platform is displayed.

For the Windows platform, the media pack is **Oracle Endeca Information Discovery Studio (3.1) Media Pack v1 for Microsoft Windows x64 (64-bit)**.

For the Linux platform, the media pack is **Oracle Endeca Information Discovery Studio (3.1) Media Pack v1 for Linux x86-64**.

4. In the media pack list, click the name of the media pack.
The **Download** page displays, listing the downloads for the selected media pack.
5. From the **Download** page, click the Download button 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
Oracle Endeca Information Discovery Studio (3.1) for Microsoft Windows x64 (64-bit) and Tomcat	Studio Tomcat bundle.
Oracle Endeca Information Discovery Studio (3.1) for Microsoft Windows x64 (64-bit) and WebLogic	Studio to install on WebLogic Server.

Title	Description
Oracle Endeca Information Discovery Studio Standalone (3.1) for Microsoft Windows x64 (64-bit)	Studio to install as a standalone application on Tomcat.
Oracle Endeca Information Discovery Provisioning Service (3.1) for Microsoft Windows x64 (64-bit)	Provisioning Service installation files.
Oracle Endeca Information Discovery Studio SDK (3.1) for Microsoft Windows x64 (64-bit)	<p>Studio Component SDK. Used for development of custom components.</p> <p>For information on developing custom components, see the <i>Studio Administration and Customization Guide</i>.</p>
Oracle Endeca Information Discovery Studio (3.1) Documentation	<p>Installation and migration documentation for Studio and the Provisioning Service.</p> <p>The full documentation set is available from the Oracle Technology Network.</p>

For Linux, the available downloads are:

Title	Description
Oracle Endeca Information Discovery Studio (3.1) for Linux x86-64 and Tomcat	Studio Tomcat bundle.
Oracle Endeca Information Discovery Studio (3.1) for Linux x86-64 and WebLogic	Studio to install on WebLogic Server.
Oracle Endeca Information Discovery Studio Standalone (3.1) for Linux x86-64	Studio to install as a standalone application on Tomcat.
Oracle Endeca Information Discovery Provisioning Service (3.1) for Linux x86-64	Provisioning Service installation files.
Oracle Endeca Information Discovery Studio SDK (3.1) for Linux x86-64	<p>Studio Component SDK. Used for development of custom components.</p> <p>For information on developing custom components, see the <i>Studio Administration and Customization Guide</i>.</p>
Oracle Endeca Information Discovery Studio (3.1) Documentation	<p>Installation and migration documentation for Studio and the Provisioning Service.</p> <p>The full documentation set is available from the Oracle Technology Network.</p>

Part II

Installing Studio



Chapter 4

About the Studio Installation Process

You install Studio using one of the available installation options. For production environments, you must change the default Studio database.

Available options for installing Studio

The available options for installing Studio are:

- Studio Tomcat bundle for Windows. This is based on Tomcat 6 and Java 1.6. See [Installing Studio Using the Tomcat Bundle for Windows on page 20](#).
- Studio Tomcat bundle for Linux Tomcat bundle. This is based on Tomcat 6 and Java 1.6. See [Installing Studio Using the Tomcat Bundle for Linux on page 23](#).
- Studio as a standalone application on a Tomcat application server. See [Installing Studio on Tomcat 6.0 on page 26](#).
- Studio as a standalone application on Oracle WebLogic Server 11g. WebLogic can be used on both Linux and Windows. See [Installing Studio on Oracle WebLogic Server 11gR1 \(10.3.6\) on page 37](#).

Note on changing the default Studio database

The installation instructions deploy Studio using the default embedded Hypersonic database, which is not intended for production use.

In production, you must deploy using an alternate database. For more information about this process, see [Changing the Database Used to Store Studio Application Data on page 67](#).



Chapter 5

Installing Studio Using the Tomcat Bundle for Windows

In the Tomcat bundle version of Studio, Tomcat 6 and the JVM 1.6 are embedded. You can, however, use your own version of Java 6, as long as it's update 18 or greater.

Installing the Windows Tomcat bundle

Changing the context root for the Windows Tomcat bundle

Installing the Windows Tomcat bundle

To install the Tomcat bundle, you extract the Studio application, then add the Studio components.

The Studio Tomcat bundle download for Windows contains the following files:

- `EID_3.1.x_Studio_portal.zip`
- `EID_3.1.x_Studio_components.zip`

To install the Studio Tomcat bundle:

1. Unzip `EID_3.1.x_Studio_portal.zip` to the directory of your choice.

Studio creates a directory called `endeca-portal` in that directory. For example, if you unzip into `C:`, Information Discovery Studio installs into `C:\endeca-portal`.

It is recommended that you create a directory `C:\Oracle\Endeca\Discovery\3.1.0\` to extract the file into.

2. Extract the `.war` files from `EID_3.1.x_Studio_components.zip` and place them into the `endeca-portal\deploy` directory.

The `.war` files go in the root of `endeca-portal\deploy`. There should be no subdirectories.



Note: This directory already contains hooks and layouts required by the portal. It is safe to overwrite these files with the versions in `EID_3.1.x_Studio_Components.zip`.

3. If the environment variables `CATALINA_HOME` or `JAVA_HOME` are already set, update them to point to your newly installed Tomcat directory and a valid 1.6 JRE.

For example, set `CATALINA_HOME` to `C:\<path_to_endeca-portal>\tomcat-6.0.37`, and set `JAVA_HOME` to `C:\<path_to_endeca-portal>\tomcat-6.0.37\jre1.6.0_21\win`.

If you do not have these environment variables set, you can leave them un-set.

4. To start the portal's Tomcat instance, run `endeca-portal\tomcat-6.0.37\bin\startup.bat`.



Note: Server startup can take several minutes. You can follow the log messages to ascertain when the process is complete. Do not shut down the Tomcat window while Studio is running.

- To test that the application is running, go to the portal (<http://localhost:8080/eid>) in your browser. Log in using the following default credentials:

Field	Value
Login:	admin@oracle.com
Password:	Welcome123

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

- At least 6 characters
 - At least one non-alphabetic character
- Optionally, you can set up [log4j](#) logging. `log4j` provides configurable, Java-based logging in an open-source utility.



Note: For more information about Studio logging, see the *Studio Administration and Customization Guide*.

Changing the context root for the Windows Tomcat bundle

Optionally, after installing the Windows Tomcat bundle, you can change the context root. Changing the context root affects both the Studio application and any automatically deployed components and other hooks.

When you first install the bundle, it uses the default context root of `/eid`, and you get access to Studio by going to <http://localhost:8080/eid>. If you change the context root to `sales`, then the URL would be <http://localhost:8080/sales>.

To change the context root:

- Stop the server.
- Rename `endeca-portal\tomcat-6.0.37\conf\Catalina\localhost\eid.xml` file to `<context root>.xml`.

For example, if the new context root is `sales`, the file name should be `sales.xml`.

For multi-level context paths, separate the name with `#`. For example, for a context path of `/sales/east`, the file name should be `sales#east.xml`.

- Rename the `endeca-portal\tomcat-6.0.37\webapps\eid` directory to `endeca-portal\tomcat-6.0.37\webapps\<context root>`.

For multi-level context paths, use a multi-level path such as: `endeca-portal\tomcat-6.0.37\webapps\sales#east`.

- Restart the server.

When you change the context root of the Studio web application, any automatically deployed components are deployed to their own contexts, which are nested under the context path of the Studio application.

For example, if Studio is deployed to `/mycompany/portal/`, the component is automatically deployed as `mycompany#portal#component-name-portlet.war`, resulting in a context path of `/mycompany/studio/component-name-portlet/`.

If you change the context root after the first startup of Studio, then you may want to undeploy and redeploy the components.



Chapter 6

Installing Studio Using the Tomcat Bundle for Linux

In the Linux Tomcat bundle version of Studio, Tomcat 6 is embedded.

[Installing the Linux Tomcat bundle](#)

[Changing the context root in the Linux Tomcat bundle](#)

Installing the Linux Tomcat bundle

To install the Tomcat bundle, you extract the Studio application, then add the Studio components.

The Studio Tomcat bundle download for Linux contains the following files:

- `EID_3.1.x_Studio_portal.tgz`
- `EID_3.1.x_Studio_components.zip`

To install the Studio Linux Tomcat bundle:

1. Extract `EID_3.1.x_Studio_portal.tgz` to the directory of your choice.

Studio creates a directory called `endeca-portal` in that directory.

It is recommended that you create a directory `Oracle/Endeca/Discovery/3.1.0/` to extract the file into.

2. Extract the `.war` files from `EID_3.1.x_Studio_components.zip` and place them into the `endeca-portal/deploy` directory. The `.war` files go in the root of `endeca-portal/deploy`. There should be no subdirectories.



Note: This directory already contains hooks and layouts required by the portal. It is safe to overwrite these files with the versions in `EID_3.1.x_Studio_components.zip`.

3. If the environment variable `CATALINA_HOME` is already set, update it to point to your newly installed Tomcat directory.
4. Make sure that the `JAVA_HOME` environment variable is set to point to a valid 1.6 JRE.
5. To start the portal's Tomcat instance, run `endeca-portal/tomcat-6.0.37/bin/startup.sh`.



Note: Server startup can take several minutes. You can follow the log messages to ascertain when the process is complete.

6. To test that the application is running, go to the portal (<http://localhost:8080/eid>) in your browser. Log in using the following default credentials:

Field	Value
Login:	admin@oracle.com
Password:	Welcome123

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

- At least 6 characters
 - At least one non-alphabetic character
7. Optionally, you can set up [log4j](#) logging. `log4j` provides configurable, Java-based logging in an open-source utility.



Note: For more information about Studio logging, see the *Studio Administration and Customization Guide*.

Changing the context root in the Linux Tomcat bundle

Optionally, you can change the context root used by your Studio application. Changing the context root affects both the Studio application and any automatically deployed components and other hooks.

When you first install the bundle, it uses the default context root of "/eid", and you get access to Studio by going to <http://localhost:8080/eid>. If you change the context root to "sales", then the URL would be <http://localhost:8080/sales>.

To change the context root:

1. Stop the server.
2. Rename `endeca-portal/tomcat-6.0.37/conf/Catalina/localhost/eid.xml` file to `<context root>.xml`.

For example, if your new context root is `sales`, the file name should be `sales.xml`.

For multi-level context paths, separate the name with `#`. For example, for a context path of `/sales/east`, the file name should be `sales#east.xml`.

3. Rename the `endeca-portal/tomcat-6.0.37/webapps/eid` directory to `endeca-portal/tomcat-6.0.37/webapps/<context root>`.

For multi-level context paths, use a multi-level path such as: `endeca-portal/tomcat-6.0.37/webapps/sales#east`.

4. Restart the server.

When you change the context root of the Studio web application, any automatically deployed components are deployed to their own contexts, which are nested under the context path of the Studio application.

For example, if Studio is deployed to `/mycompany/portal/`, the component is automatically deployed as `mycompany#portal#component-name-portlet.war`, resulting in a context path of `/mycompany/studio/component-name-portlet/`.

If you change the context root after the first startup of Studio, then you may want to undeploy and redeploy the components.



Chapter 7

Installing Studio on Tomcat 6.0

You can also install Studio on the most recent version of the Tomcat application server, without using the Studio Tomcat bundle.

[About installing on Tomcat 6.0](#)

[Installing Tomcat and deploying the dependency libraries](#)

[Modifying the Tomcat configuration to work with Studio](#)

[Deploying and starting Studio](#)

About installing on Tomcat 6.0

You can deploy Studio as a standalone application on Tomcat 6.0.

These instructions assume that you have obtained the `apache-tomcat-6.0.x.zip` or `tar.gz` file from the Apache Foundation, but that you have not yet installed it.

The rest of these instructions refer to the installation directory as `apache-tomcat-6.0.x`, leaving off the minor version number.

You also must have downloaded the Studio standalone package for Windows or Linux. The standalone package contains the following files:

- `EID_3.1.x_Studio_portal_dependencies.zip`
- `EID_3.1.x_Studio_portal_war.zip`
- `EID_3.1.x_Studio_components.zip`

At a high level, to install and deploy Studio on Tomcat 6.0:

1. Install Tomcat and deploy the Studio dependency libraries.
2. Modify the Tomcat configuration to work with Studio.
3. Deploy and start Studio.



Note: The examples in these instructions are based on a Windows server Tomcat deployment. If you install on Linux, the steps are similar, although you must substitute Linux binaries and paths. Significant differences are called out.

Installing Tomcat and deploying the dependency libraries

Studio requires the deployment of several Java libraries.

To install the Tomcat software and deploy the Studio dependency libraries:

1. Create an `endeca-portal` directory.
This is the home directory for your Studio installation.
2. Unzip `apache-tomcat-6.0.x.zip` into `endeca-portal/apache-tomcat-<version>`.
Unzipping this file creates much of the directory structure mentioned below.
3. Unzip `EID_3.1_Studio_portal_dependencies.zip` into a temporary directory.
This zip file contains a collection of `.jar` files and other dependency files.
4. Create the `endeca-portal/apache-tomcat-<version>/endorsed` directory.
5. From the temporary directory, copy the following `.jar` files into `endeca-portal/apache-tomcat-<version>/endorsed`:
 - `commons-logging.jar`
 - `log4j.jar`
 - `log4j.properties.jar`
6. Under the `endeca-portal/apache-tomcat-<version>/lib` directory, create an `ext` directory.
7. From the temporary directory, copy the following `.jar` files into `endeca-portal/apache-tomcat-<version>/lib/ext`:
 - `activation.jar`
 - `annotations.jar`
 - `ccpp.jar`
 - `commons-lang.jar`
 - `container.jar`
 - `cxfr-2.2.8.jar`
 - `cxfr-orawSDL-adapter.jar`
 - `endeca-images.jar`
 - `endeca-portal.jar`
 - `enrichmentWSStubs.jar`
 - `hsqldb.jar`
 - `jabsorb.jar`
 - `jackson-core-lgpl-1.7.2.jar`
 - `jackson-mapper-lgpl-1.7.2.jar`
 - `jms.jar`
 - `jta.jar`

- jutf7.jar
- mail.jar
- mdex_bindings.jar
- mdex_ingest.jar
- mysql-connector-java-commercial-5.1.17-bin.jar
- ojdbc6.jar
- oracle.http_client_11.1.1.1.jar
- orawsdl.jar
- portal-kernel.jar
- portal-service.jar
- portlet-container.jar
- portlet.jar
- provisioningServerWSStubs.jar
- saw-api.jar
- support-tomcat.jar
- XmlSchema-1.4.5.jar

Modifying the Tomcat configuration to work with Studio

Before you can deploy Studio, you must modify some of the Tomcat configuration files.

To complete the Tomcat configuration updates:

1. In the `endeca-portal/apache-tomcat-<version>/bin/` directory, modify `catalina.bat` (on Windows) or `catalina.sh` (on Linux) by setting the value of `JAVA_OPTS`, to increase the memory size for the server and establish security configuration for Studio.

Add the setting under the line `Execute The Requested Command`. It must be added as a single line (no line breaks), with the parameters separated by spaces.

For Windows, the line to add is:

```
set JAVA_OPTS=%JAVA_OPTS% -Xmx1024m -XX:MaxPermSize=256m -Dfile.encoding=UTF8
-Duser.timezone=GMT -Djava.security.auth.login.config="%CATALINA_HOME%/conf/jaas.config"
-Dorg.apache.catalina.loader.WebappClassLoader.ENABLE_CLEAR_REFERENCES=false
-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
```

For Linux, the line to add is:

```
JAVA_OPTS="$JAVA_OPTS -Xmx1024m -XX:MaxPermSize=256m
-Dfile.encoding=UTF8 -Duser.timezone=GMT
-Djava.security.auth.login.config="$CATALINA_HOME"/conf/jaas.config
-Dorg.apache.catalina.loader.WebappClassLoader.ENABLE_CLEAR_REFERENCES=false
```

```
-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"
```

2. Modify the file `endeca-portal/apache-tomcat-<version>/conf/catalina.properties` as follows to add the `ext` directory to the common class loader:

```
common.loader=
  ${catalina.base}/lib,\
  ..\
  ${catalina.home}/lib/ext/*.jar
```

3. To deploy Studio in the recommended context path of `/eid`, create a new file called `eid.xml` in `endeca-portal/apache-tomcat-<version>/conf/Catalina/localhost/`.

To deploy Studio into any other context, create a new file called `<context root>.xml` in `endeca-portal/apache-tomcat-<version>/conf/Catalina/localhost/`.

For multi-level context paths, separate the name with `#`. For example, for a context path of `/sales/east`, the file name should be `sales#east.xml`.

Note that the directory already contains a file called `ROOT.xml`, which provides access to the root context for Tomcat.

4. Modify the XML file created in the previous step to add:

```
<Context crossContext="true" />
```

5. To support UTF-8 URI encoding, edit the `server.xml` file, located in the `endeca-portal/apache-tomcat-<version>/conf` directory, as follows:

```
<!-- Define a non-SSL HTTP/1.1 Connector on port 8080 -->
  <Connector port="8080" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443"
    URIEncoding="UTF-8"
  />
```

Deploying and starting Studio

Once the Tomcat configuration is complete, you can deploy and start Studio.

To deploy and start Studio:

1. Unzip `EID_3.1.x_Studio_portal_war.zip` into a temporary directory.
This zip file contains the Studio `.war` file and the `copyright.txt` file.
2. Read the `copyright.txt` file and then save it to the location of your choice.
3. Assuming you are using the `/eid` context, unzip the contents of the `.war` file into `endeca-portal/apache-tomcat-<version>/webapps/eid`.
4. If using a different context for your deployment, unzip the `.war` file into `endeca-portal/apache-tomcat-<version>/webapps/<context root>`.

For example, if the context is `/sales`, unzip the file into `endeca-portal/apache-tomcat-<version>/webapps/sales`

For multi-level context paths, use the multi-level path. For example, if the context path is `/mycompany/sales`, unzip the file into `endeca-portal/apache-tomcat-<version>/webapps/mycompany/sales`.

- Copy the `portal-ext.properties` file from the temporary directory you created for the `EID_3.1.x_Studio_portal_dependencies.zip` to the `endeca-portal` 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.

- Under the `endeca-portal` directory, create a `data` directory, and then create an `endeca-data-sources` directory below that.
- In the `endeca-portal` directory, create a `deploy` directory.
- Extract the `.war` files from `EID_3.1.x_Studio_components.zip`. Place the files into the `endeca-portal/deploy` directory.

The `.war` files are placed at the root of `endeca-portal/deploy`. There should be no subdirectories.

- To start the portal's Tomcat instance, run `endeca-portal/tomcat<version>/bin/startup.bat`.



Note: Server startup can take several minutes. You can follow the log messages to ascertain when the process is complete. Do not shut down the Tomcat window while Studio is running.

- To test that the application is running, go to the portal (`http://localhost:8080/<context root>`) in your browser.

For example, if you are using the recommended context path of `/eid`, you would go to `http://localhost:8080/eid`

- Log in using the following default credentials:

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

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

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

12. Optionally, you can set up [log4j](#) logging. `log4j` provides configurable, Java-based logging in an open-source utility.



Note: For more information about Studio logging, see the *Studio Administration and Customization Guide*.



Chapter 8

Running Studio Under Tomcat as a Windows Service

If you have installed the Studio Tomcat Windows bundle or have installed Studio as a standalone application on a Windows version of Tomcat, then you can run Studio as a Windows service.

[About running Studio as a Windows service](#)

[Obtaining the service installer files](#)

[Configuring the service](#)

[Installing and starting the service](#)

[Troubleshooting the service installation](#)

About running Studio as a Windows service

Running Studio as a Windows service requires the Tomcat service installer files. You also should install the Tomcat service monitor, which is used to configure and monitor the Windows service, and is useful for troubleshooting.

If you have installed the Studio Tomcat Windows bundle, then the server installer and monitor executable files are provided for you. You just need to install the service.

If you have installed Studio as a standalone application on your own instance of Tomcat, then you need to obtain the files from the Tomcat download, which is available from the Apache web site. After you obtain the files, you then configure and install the service.

Obtaining the service installer files

If you have installed Studio as a standalone application on your own version of Tomcat, then you need to obtain the service installer and monitor files for that version. These files are part of the Tomcat download.

If you have installed the Studio Tomcat bundle for Windows, then the files are provided as part of the bundle. You can skip this step and go straight to *[Installing and starting the service on page 35](#)*. You must use the files for Tomcat version 6.0.37.

To obtain the files for your version of Tomcat and add them to Studio:

1. From the Apache Tomcat website (<http://tomcat.apache.org>), download the Tomcat file (apache-tomcat-6.0.x-windows-x86.zip).

A sample URL for the archive directory is: <http://archive.apache.org/dist/tomcat/tomcat-6/v6.0.37/bin/>

If you are using a 64-bit JVM, then you need to download the 64-bit version of the installer (apache-tomcat-6.0.x-windows-x64.zip).

2. Extract the file to a temporary directory.
3. In the `bin` subdirectory of the temporary download directory, locate the following files:
 - `service.bat`
 - `tomcat6.exe`
 - `tomcat6w.exe`. This is the Tomcat service monitor.
4. Copy these files to the `bin` directory of Information Discovery Studio:
`endeca-portal\tomcat-6.0.x\bin`

Configuring the service

If you downloaded the service installer files directly from the Apache web site, then before you install the service, you must make some configuration changes to the `service.bat` file.

For the Studio Tomcat bundle for Windows, the file is already configured. You can skip this step and go directly to [Installing and starting the service on page 35](#).

After you download the `service.bat` and copy it to the `bin` directory, you must update the file to:

- Edit the service name and descriptions to reflect your Studio installation
- Add additional JVM startup arguments
- Increase the memory allocation. Studio requires more memory than is set in the default values.

If you change the service name, then you also must change the name of the Tomcat monitor executable.

To update the configuration:

1. Open the file `service.bat`.
2. In the file, find the following lines:

```
set SERVICE_NAME=Tomcat6
set PR_DISPLAYNAME=Apache Tomcat 6
```

3. Change the name and display name to reflect your Studio installation. For example:

```
set SERVICE_NAME=Studio31
set PR_DISPLAYNAME=Studio 3.1
```

4. Next, find the following line:

```
set PR_DESCRIPTION=Apache Tomcat 6.0.x Server -
http://tomcat.apache.org/
```

5. Change the service description to reflect your Studio installation. For example:

```
set PR_DESCRIPTION=Studio server, version 3.1
```

6. Next, find the following line:

```
"%EXECUTABLE%" //US//%SERVICE_NAME% ++JvmOptions "-Djava.io.tmpdir=%CATALINA_BASE%\temp
;-Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager
;-Djava.util.logging.config.file
=%CATALINA_BASE%\conf\logging.properties" --JvmMs 128 --JvmMx 256
```

7. Replace the last part of the line:

```
-Djava.util.logging.config.file
=%CATALINA_BASE%\conf\logging.properties" --JvmMs 128 --JvmMx 256
```

with:

```
-Djava.util.logging.config.file=%CATALINA_BASE%\conf\logging.properties;-XX:MaxPermSize
=256m" --JvmMs 256 --JvmMx 1024
```

Make sure that there are no manual line breaks or extra spaces.

8. Immediately below that line, add the following line:

```
"%EXECUTABLE%" //US//%SERVICE_NAME% ++JvmOptions "-Dfile.encoding=UTF8
;-Djava.net.preferIPv4Stack=true;-Duser.timezone=GMT;-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"
```

Make sure there are no manual line breaks or extra spaces.

9. Save and close the file.
10. Create a copy of the Tomcat monitor file (tomcat6w.exe). Rename the copy to be <value of SERVICE_NAME>w.exe.

For example, if you set SERVICE_NAME=Studio31 in service.bat, then you must rename the copy of the tomcat6w.exe to Studio31w.exe.

Installing and starting the service

To install the service, you run the service.bat file. You also must update the Tomcat monitor to point to the JVM.

If you are not using the bundled JVM, then make sure that the JAVA_HOME environment variable is set to the location of your JDK. By default, service.bat looks for %JAVA_HOME%\jre\server\jvm.dll.

To install and start the Studio service:

1. From the command line, navigate to the Studio Tomcat bin directory.


```
endeca_portal\tomcat-6.0.x\bin\
```
2. Run the following command:


```
service.bat install
```
3. Configure the Tomcat monitor to point to the JVM:
 - (a) Double-click the monitor executable (for example, Studio23w.exe).
 - (b) On the properties dialog, click the **Java** tab.
 - (c) Uncheck **Use default**.
 - (d) In the **Java Virtual Machine** field, set the full path to jvm.dll.

For the bundled JVM, the file is endeca-portal\tomcat-6.0.x\jre1.6.0_x\win\bin\server\jvm.dll.

If you are not using the bundled JVM, then set the path to your JVM.
 - (e) Click **OK**.

4. When you install the service, it is set up to be started manually. To configure the service to start automatically:
 - (a) Display the **Services** list (**Control Panel**>**Administrative Tools**>**Services**).
 - (b) In the list, double-click the Studio service.
The properties dialog for the service is displayed.
 - (c) From the **Startup type** drop-down list, select **Automatic**.
 - (d) Click **OK**.
5. From the **Services** list, to start the service for the first time, right click the service, then click **Start**.

Troubleshooting the service installation

If the service installs properly, but fails to start, you can use the steps provided here to troubleshoot.

For additional details on using the Tomcat service and service monitor, see <http://tomcat.apache.org/tomcat-6.0-doc/windows-service-howto.html>.

As you are troubleshooting, check the log files (endeca-portal\tomcat-6.0.x\logs\jakarta_service*.log) for the relevant messages.

If the service will not start:

1. If you downloaded the service installer and monitor files manually, then make sure that you have used the correct version of the Tomcat download:
 - Tomcat version 6.0.x
 - For a 32-bit JVM (including the bundled JVM), apache-tomcat-6.0.x-windows-x86.zip
 - For a 64-bit JVM, apache-tomcat-6.0.x-windows-x64.zip
2. If you are not using the bundled JVM, make sure that the `JAVA_HOME` environment variable is set to the location of your JDK.
By default, `service.bat` looks for `%JAVA_HOME%\jre\server\jvm.dll`.
To change `JAVA_HOME` after the service is installed:
 - (a) Uninstall the service. To uninstall the service, run the following command:
`service.bat remove`
 - (b) Update `JAVA_HOME`.
 - (c) Reinstall and restart the service.
3. Make sure the Tomcat service monitor is configured to point to the location of your JVM.
 - (a) Double-click the monitor executable.
 - (b) On the properties dialog, click the **Java** tab.
 - (c) Uncheck **Use default**.
 - (d) In the **Java Virtual Machine** field, specify the path to `jvm.dll`.
For the bundled JVM, the file is `endeca-portal\tomcat-6.0.x\jre1.6.0_x\win\bin\client\jvm.dll`.
If you are not using the bundled JVM, then set the path to your JVM.
 - (e) Click **OK**.

4. Use the Tomcat service monitor to set the startup and shutdown modes to Java.
 - (a) Double-click the monitor executable.
 - (b) On the properties dialog, click the **Startup** tab.
 - (c) From the **Mode** drop-down list, select **Java**.
 - (d) Click the **Shutdown** tab.
 - (e) From the **Mode** drop-down list, select **Java**.
 - (f) Click **OK**.



Chapter 9

Installing Studio on Oracle WebLogic Server 11gR1 (10.3.6)

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

[About installing Studio on WebLogic Server](#)

[Downloading and installing the Sun JDK](#)

[Downloading and installing WebLogic Server](#)

[Creating and configuring 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](#)

About installing Studio on 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 the Sun JDK.
2. Downloading and installing WebLogic Server.
3. 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.

4. 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 60](#)), or deploy Studio to a different context path (see [Changing the context path for Studio on WebLogic Server on page 62](#)).

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*.

Downloading and installing the Sun JDK

Studio requires the Sun Java 6 JDK.

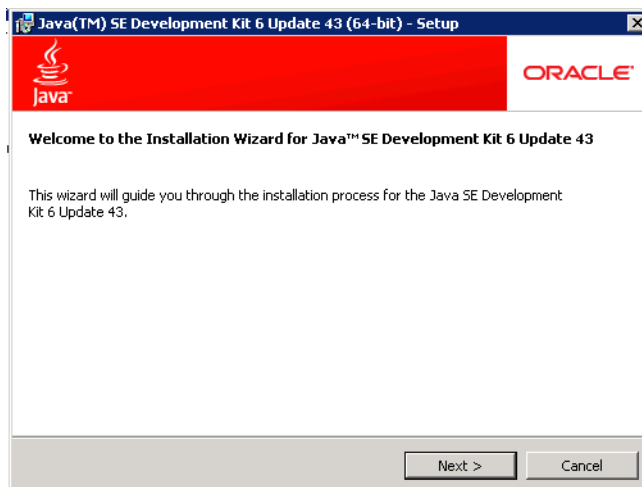
To download and install the JDK:

1. To download Sun Java 6:
 - (a) Go to <http://www.oracle.com/technetwork/java/javase/downloads/index.html>.
 - (b) On the **Downloads** tab, click **Previous Releases**.
 - (c) On the **Oracle Java Archive** page, click **Java SE 6**.
 - (d) On the **Java SE 6 Downloads** page, click the link for the most recent version of the Java SE Development Kit.

The page scrolls to the list of downloads for that version of the JDK.

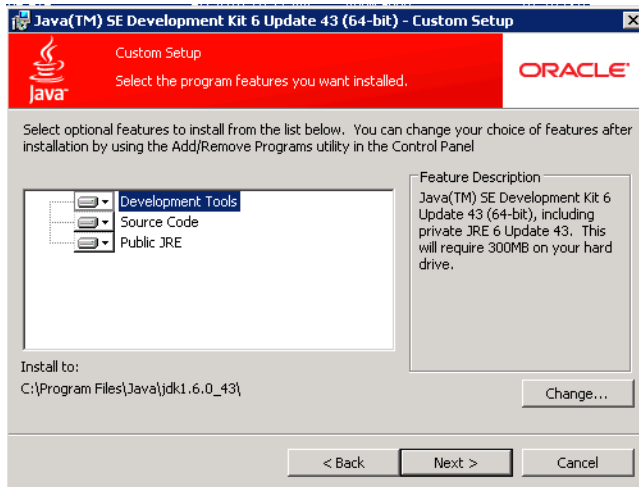
- (e) At the top of the list, click the **Accept License Agreement** radio button.
 - (f) In the list, click the **Download** link for Windows x64 or Linux x64.
 - (g) Save the file (`jdk-6u43-windows-x64.exe` or `jdk-6u43-linux-x64.bin`) to your machine.
2. To install the JDK on Windows:
 - (a) Double-click the JDK installer file.

The welcome page of the installation wizard is displayed.

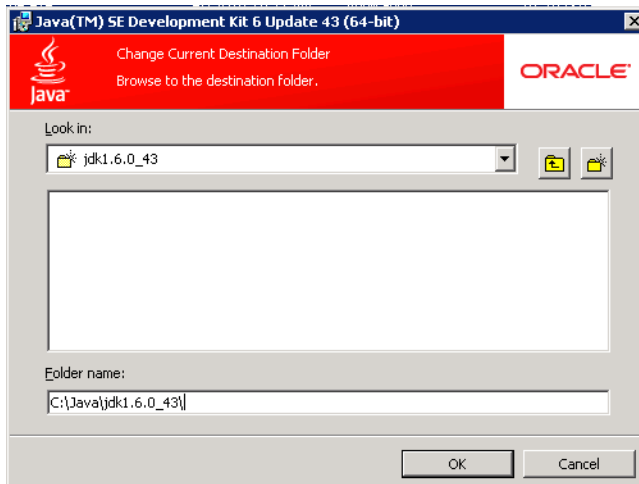


- (b) On the welcome page, click **Next**.

- (c) Because the default installation location (under `Program Files`) has a space, you must change it. On the **Custom Setup** page, click the **Change** button.

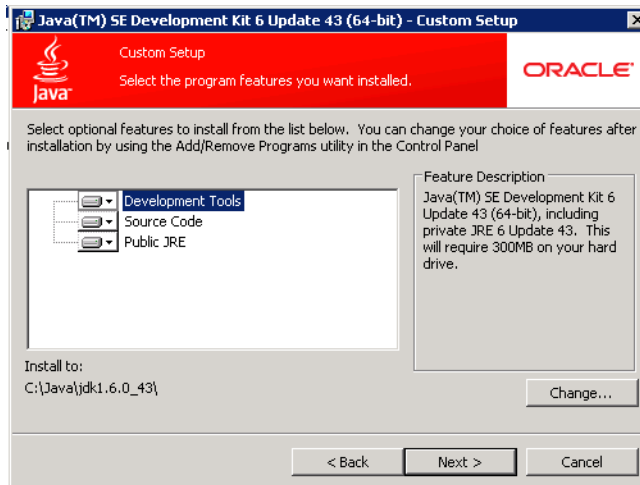


- (d) In the **Folder name** field, remove the `Program Files` directory, to change the location to:
`C:\Java\jdk1.6.0_43\`



- (e) Click **OK**.

- (f) On the **Custom Setup** page, verify that the installation location has been updated, then click **Next**.



- (g) The **Destination Folder** page sets the location of the public JRE. Because this JRE is not used by Studio, you can leave the default directory. Click **Next**.

The installation process begins.

- (h) When the installation is completed, click **Finish**.



3. To install the JDK on Linux:

- Select a location for installation. For example, `$HOME/Oracle/Endeca`.
- Place the JDK .bin file (`jdk-6u43-linux-x64.bin`) into `$HOME/Oracle/Endeca`, and ensure that it has execute permissions,
- Execute the file.

The JDK is extracted into the directory. For example, `$HOME/Oracle/Endeca/jdk1.6.0_43`.

Downloading and installing WebLogic Server

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

To download and install WebLogic Server:

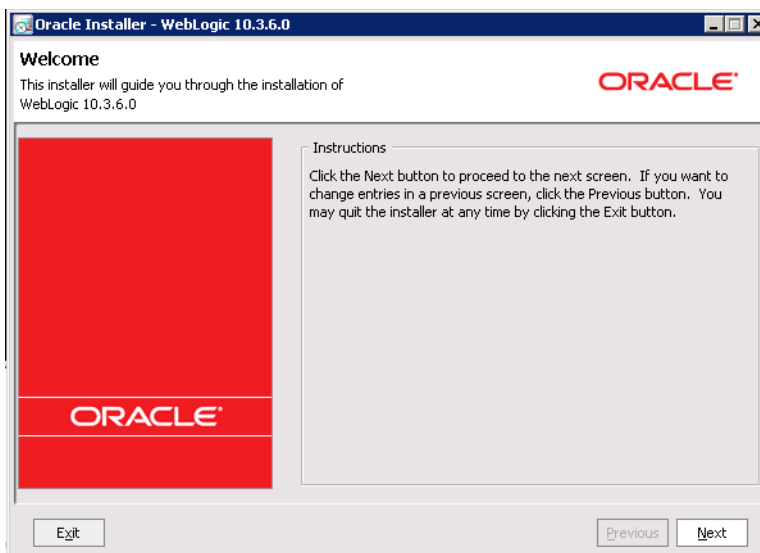
1. To download the WebLogic Server installation package:
 - (a) Go to <http://www.oracle.com/technetwork/middleware/weblogic/downloads/wls-for-dev-1703574.html>.
 - (b) Under **Oracle WebLogic Server Previous Releases**, expand the list for Oracle WebLogic Server 10.3.6.
 - (c) Download the **Generic** version of the installer (`wls1036_generic.jar`).

	Microsoft Windows (32-bit JVM)	Linux (32-bit JVM)	Solaris (32-bit JVM)	Additional Platforms (For 64-bit JVM Support, See Note Above)
Oracle WebLogic Server 10.3.6	See all files			
Oracle WebLogic Server 11gR1 (10.3.6) + Coherence + OEPE - Package Installer	x86: 1.5 GB File1	x86: 1.5 GB File1		Mac OS X: 1.7 GB File1
Oracle WebLogic Server 11gR1 (10.3.6) + Coherence - Package Installer	x86: 820 MB File1	x86: 831 MB File1	SPARC: 783 MB File1	Generic: 1 GB File1
Oracle WebLogic Server 11gR1 (10.3.6) 703 File Distribution	x86: 330 MB File1	x86: 330 MB File1		Mac OS X: 330 MB File1

2. From a command prompt, navigate to the directory where you placed `wls1036_generic.jar`.
3. To launch the installer, issue the following command:

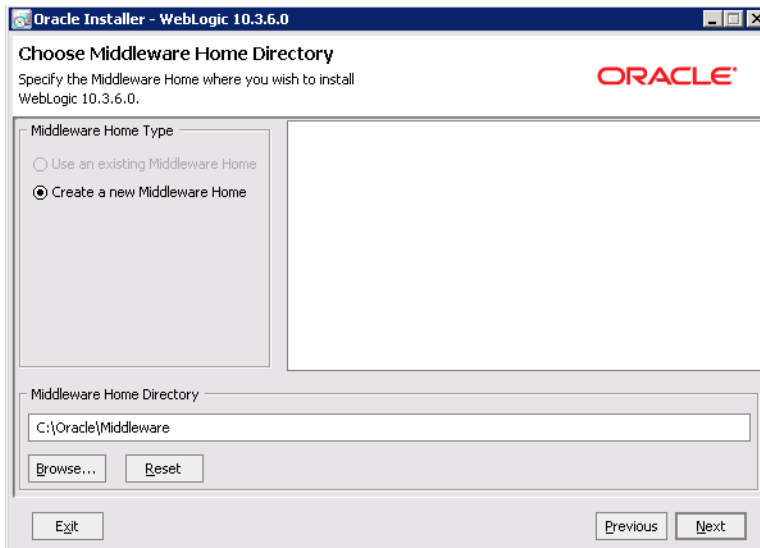
```
java -jar wls1036_generic.jar
```

The **Welcome** page of the installation wizard is displayed.

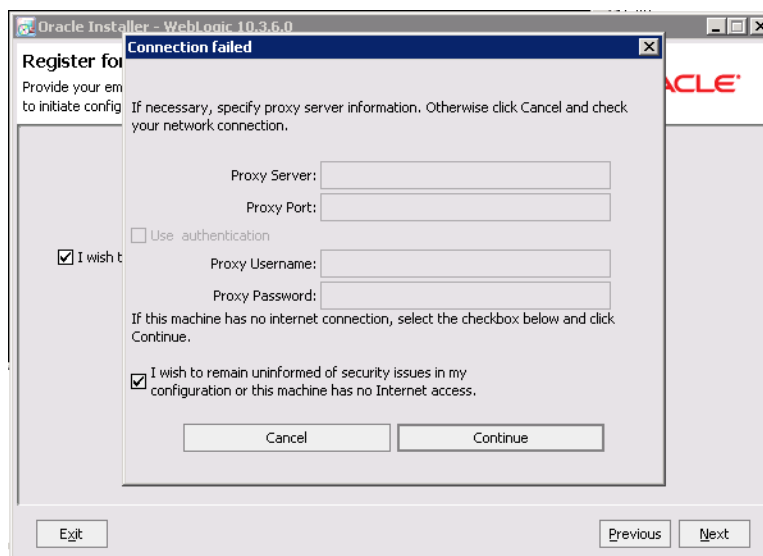


4. On the **Welcome** page, click **Next**.

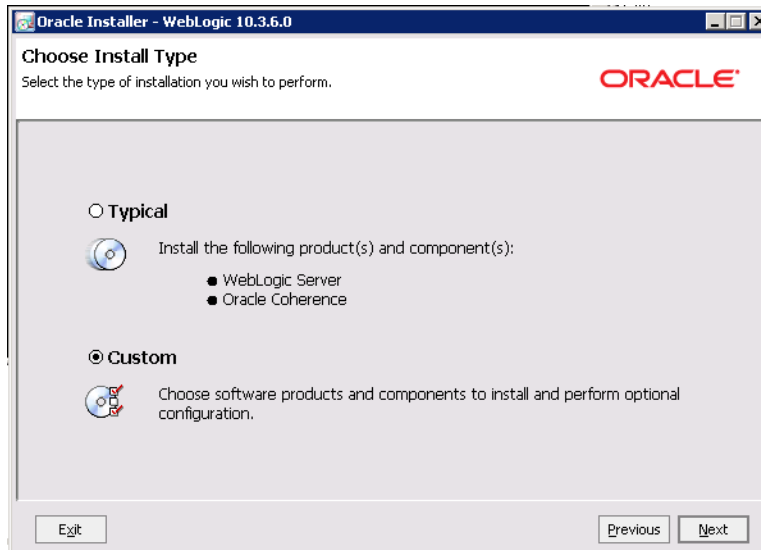
5. On the **Choose Middleware Home Directory** page:
 - (a) Click the **Create a new Middleware Home** radio button.
 - (b) Accept the default location for the Middleware Home Directory.
 - (c) Click **Next**.



6. To bypass the **Register for Security Updates** page:
 - (a) Click **Next** without entering an email address.
 - (b) On the **Email Address Not Specified** dialog, click **Yes**.
 - (c) On the **Are You Sure?** dialog, click **Yes**.
 - (d) On the **Connection failed** dialog, check the **I wish to remain uninformed of security issues in my configuration or the machine has no Internet access** checkbox, then click **Continue**.

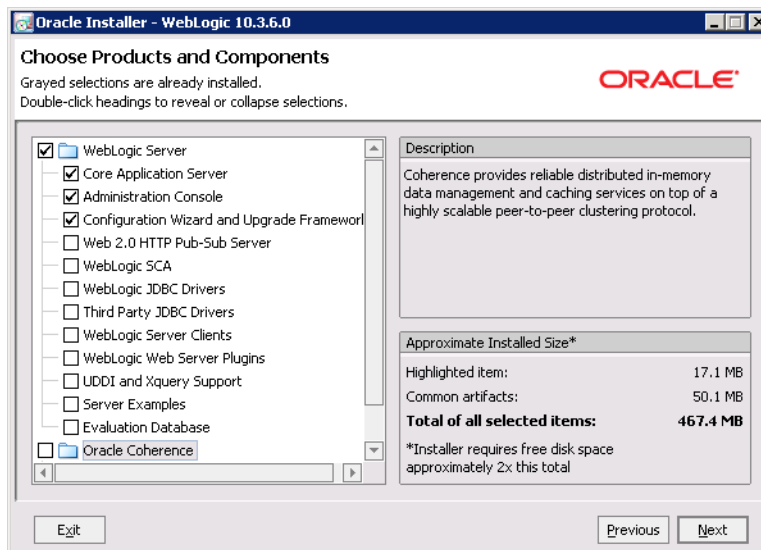


7. On the **Choose Install Type** page, click the **Custom** radio button, then click **Next**.



8. On the **Choose Products and Components** page, check the following products, then click **Next**:
- **Core Application Server**
 - **Administration Console**
 - **Configuration Wizard and Upgrade Framework**

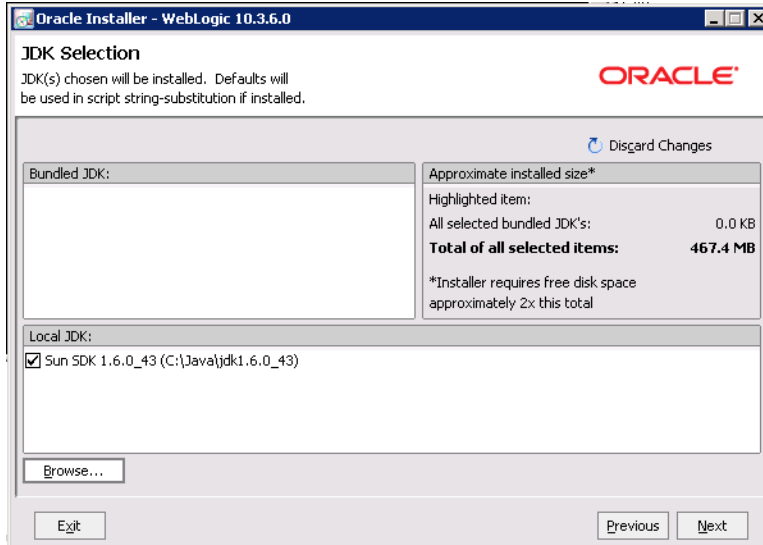
For a production instance, where you will need to update Studio to connect to a database other than Hypersonic, you should also install the JDBC drivers.



9. On the **JDK Selection** page, verify that the JDK you installed is selected. If it is, click **Next**. If that version of the JDK is not selected:
- (a) Under **Local JDK**, click the **Browse** button.

- (b) Browse to and select the directory that contains the JDK (C:\Java\jdk1.6.0_43\ or \$HOME/Oracle/Endeca/jdk1.6.0_43)

The **Local JDK** section is updated to display the selected JDK with a checked box next to it.

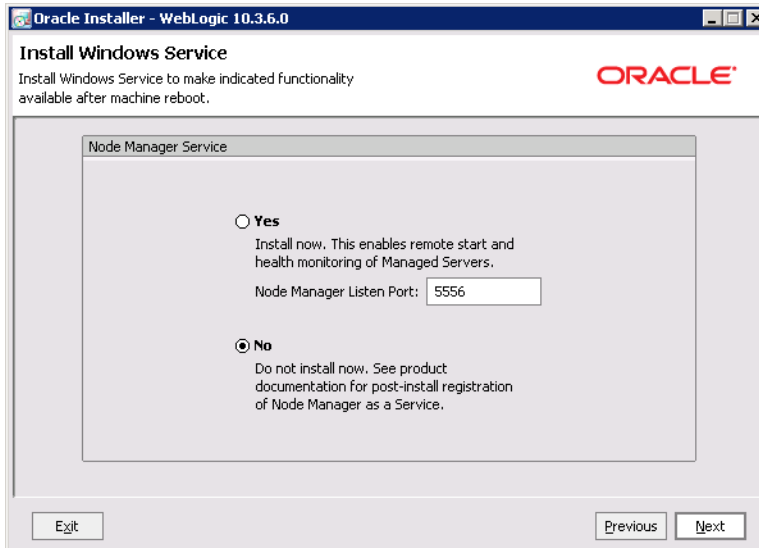


- (c) Click **Next**.

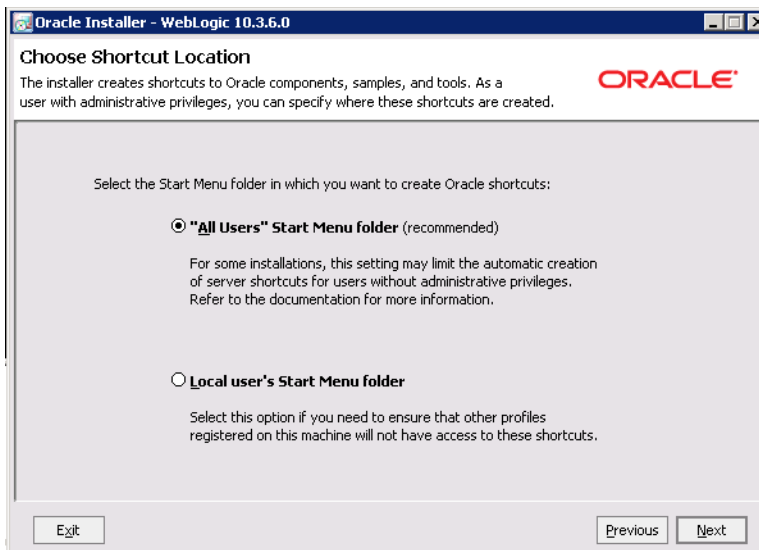
10. On the **Choose Product Installation Directories** page, accept the default installation directory, then click **Next**.



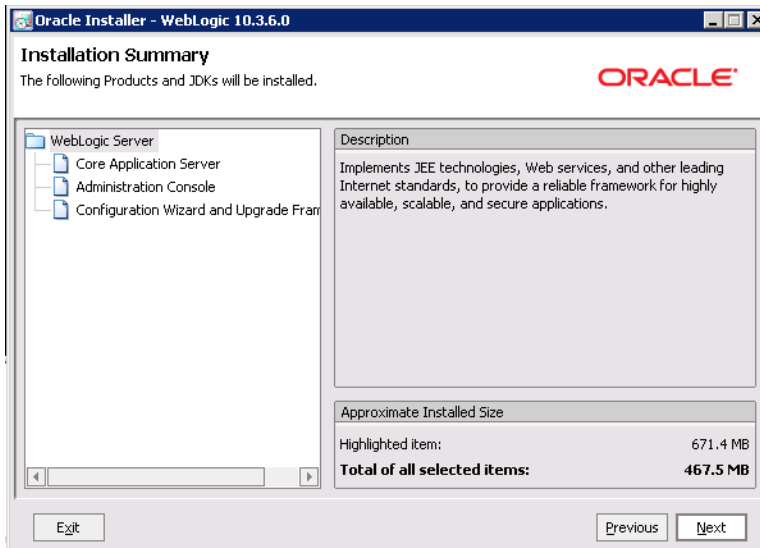
11. If the **Install Windows Service** page is displayed, click the **No** radio button, then click **Next**.



12. If the **Choose Shortcut Location** page is displayed, click a radio button to select the location of the Windows shortcut, then click **Next**.

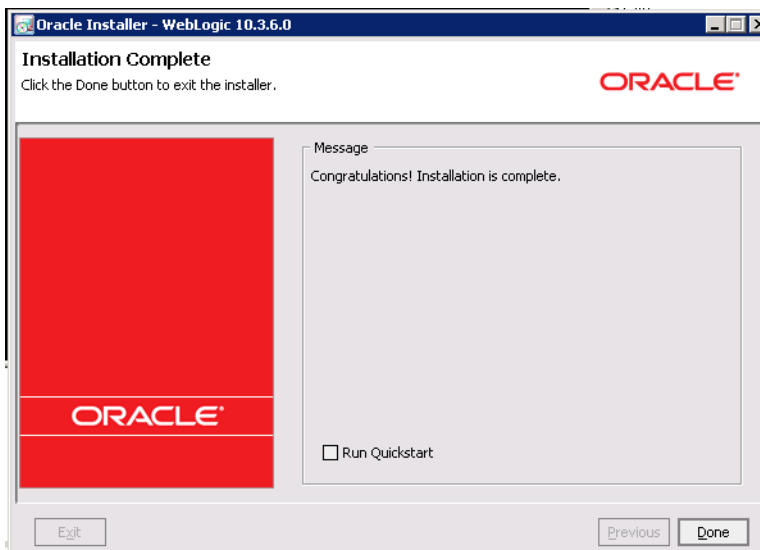


13. On the **Installation Summary** page, verify that you are installing the correct products, then click **Next**.



14. On the **Installation Complete** page, if you leave the **Run Quickstart** checked, then when you click **Done**, you immediately start the process of creating a new domain.

If you want to create the domain later, then uncheck the **Run Quickstart** checkbox before you click **Done**.



For Windows, an **Oracle WebLogic** shortcut is created in the **Start Menu**.

Creating and configuring 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 and configure the WebLogic domain, and verify that the domain is not already in use:

1. If you are not starting the domain creation using the **Run Quickstart** option of the WebLogic **Installation Complete** page, then to start the Configuration Wizard:

- (a) Change to the directory that contains the Configuration Wizard start-up program.

For example, for Windows, go to `C:\Oracle\Middleware\wlserver_10.3\common\bin`.

For Linux, from a command prompt, change to:

```
$MW_HOME/wlserver_10.3/common/bin
```

- (b) Run the start-up program.

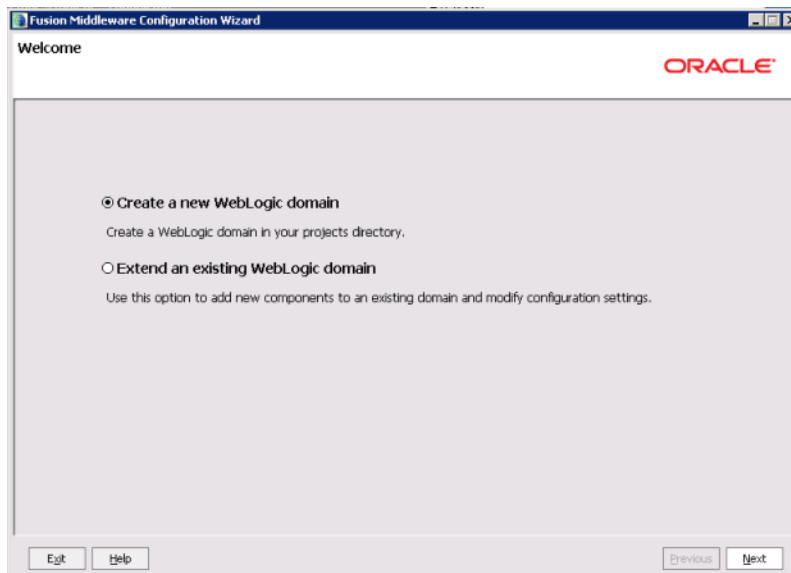
For Windows, to make sure the domain is included in the Windows start menu, you must run the program as an administrator. If you are logged in as an administrator, then to run the program, double-click **config.cmd**.

If you're not logged in as an administrator, then to run the command, right click the file, then click **Run as administrator**.

For Linux, run the following command:

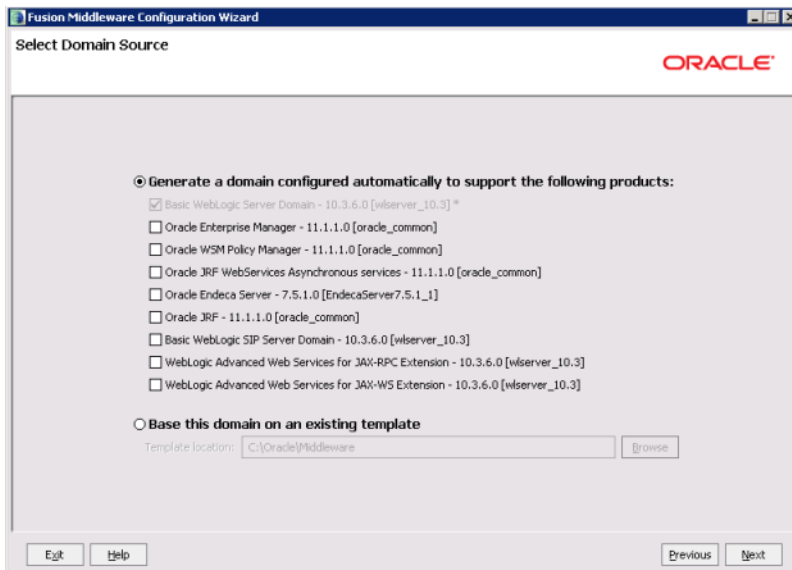
```
./config.sh
```

The Configuration Wizard **Welcome** page is displayed.

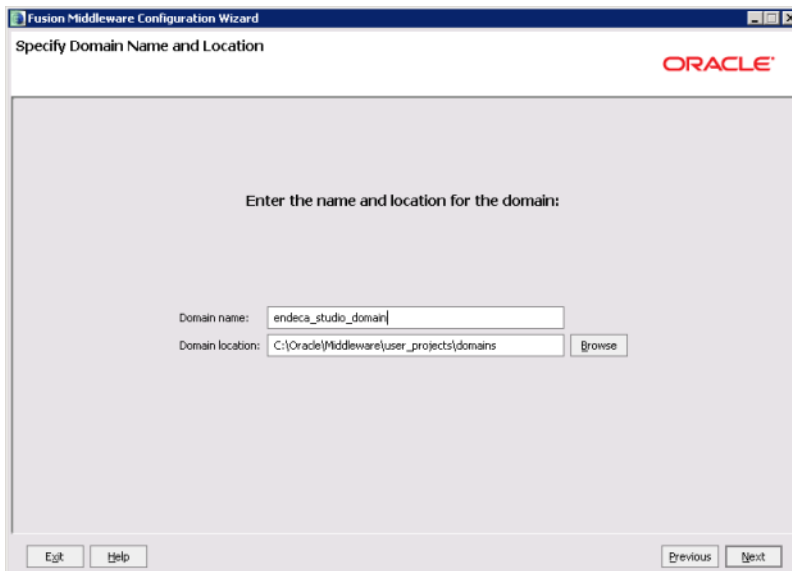


2. On the **Welcome** page, click the **Create a new WebLogic domain** radio button, then click **Next**.

3. On the **Select Domain Source** page, the **Basic WebLogic Server Domain** checkbox is checked by default. Click **Next**.



4. On the **Specify Domain Name and Location** page:
 - (a) Set the domain name to `endeca_studio_domain`
 - (b) Keep the default domain location.
 - (c) Click **Next**.



5. On the **Configure Administrator User Name and Password** page, to create the administrator credentials for starting the Studio domain:
 - (a) In the **Name** field, type the name for the administrator user.
 - (b) In the **User password** field, type the user password.

The password must be at least eight characters long.

- (c) In the **Confirm user password** field, retype the password.
- (d) Optionally, in the **Description** field, type a description for the user account.
- (e) Click **Next**.

The screenshot shows the 'Configure Administrator User Name and Password' window. It contains the following fields:

- *Name: studio_admin
- *User password: [masked]
- *Confirm user password: [masked]
- Description: This user is the default administrator.

Buttons at the bottom include 'Exit', 'Help', 'Previous', and 'Next'.

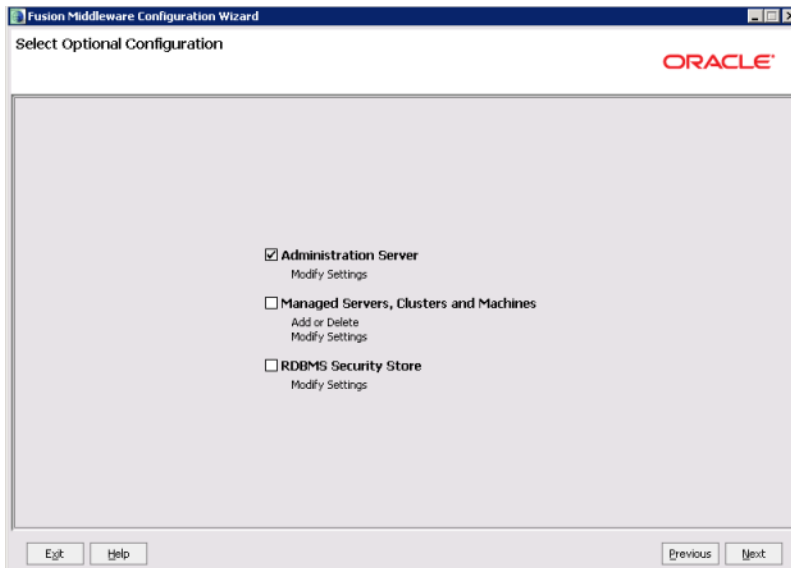
6. On the **Configure Server Start Mode and JDK** page:
 - (a) Under **WebLogic Domain Startup Mode**, click the **Production Mode** radio button.
 - (b) Under **JDK Selection**, click the **Available JDKs** radio button.
 - (c) In the JDK list, click the Sun Java JDK that you installed earlier. If the JDK is not displayed, click the **Other JDK** button, and then browse to it.
 - (d) Click **Next**.

The screenshot shows the 'Configure Server Start Mode and JDK' window. It contains the following sections:

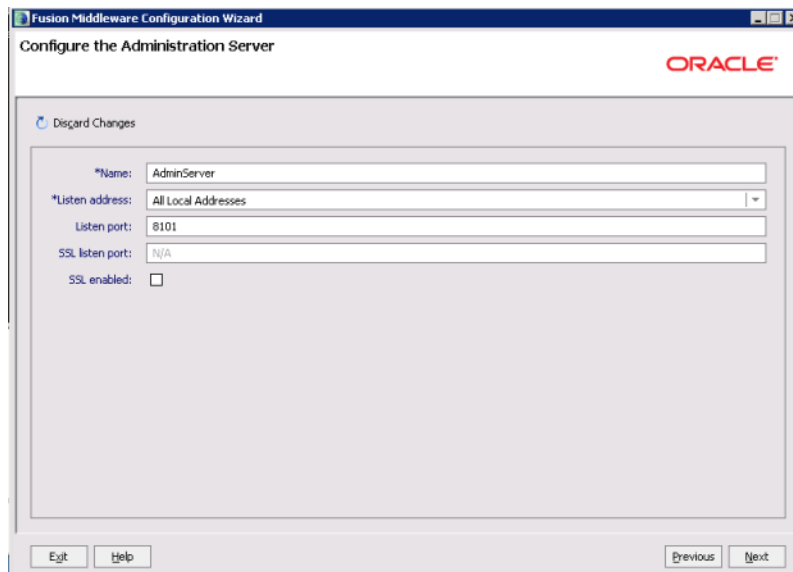
- WebLogic Domain Startup Mode:**
 - Development Mode
 - Production Mode
- JDK Selection:**
 - Available JDKs
 - Sun SDK 1.6.0_43 @ C:\Java\jdk1.6.0_43
 - Other JDK
 - Location: [text field] [Browse]

Buttons at the bottom include 'Exit', 'Help', 'Previous', and 'Next'.

7. On the **Select Optional Configuration** page, check the **Administration Server** checkbox, then click **Next**.

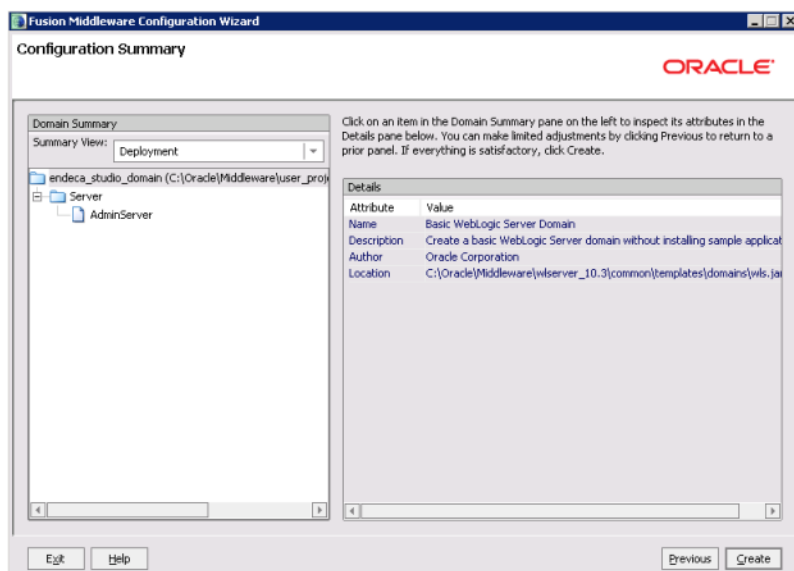


8. On the **Configure the Administration Server** page:
 - (a) Use the default values for the **Name** and **Listen address** fields.
 - (b) In the **Listen port** field, type 8101.

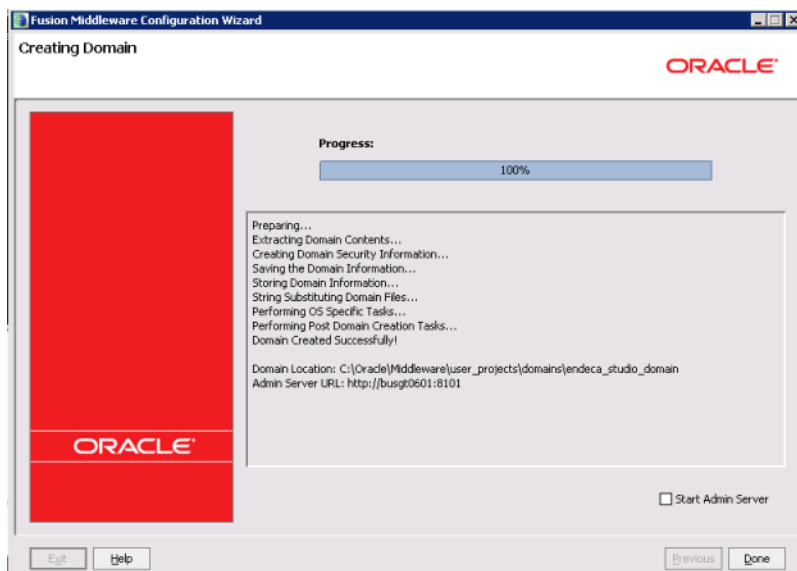


- (c) If you want to enable SSL, then check the **SSL enabled** checkbox, and in the **SSL listen port** field, type 8102.
 - (d) Click **Next**.

9. On the **Configuration Summary** page, to create the domain, click **Create**.



On the **Creating Domain** page, the message "Domain Created Successfully!" signifies the end of the domain creation process.



10. To exit the Configuration Wizard, click **Done**.
11. 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 (`<MiddlewareHomeDirectory>/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
```

Note that 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 60](#).

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 53](#).

Deploying Studio to the WebLogic domain

To install Studio, you use the WebLogic Server **Administration Console** to deploy endeca-portal-weblogic-3.1.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><MiddlewareHomeDirectory>/user_projects/domains/<StudioDomain>/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"> • <code>/data/endeca-data-sources</code> • <code>deploy</code> • <code>weblogic-deploy</code> <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 Changing the Studio home directory on WebLogic Server on page 60.</p>
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 Changing the context path for Studio on WebLogic Server on page 62.</p>

To deploy `endeca-portal-weblogic-3.1.x.ear` into a production environment:

1. Set up the Studio home directory (the default is `<MiddlewareHomeDirectory>/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`.

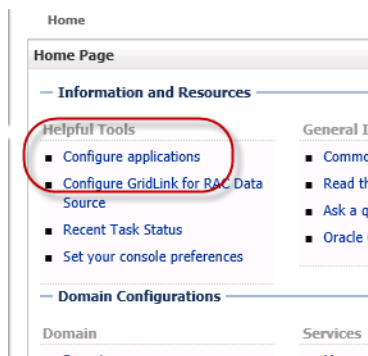
3. Extract `endeca-portal-weblogic-3.1.x.ear` from the Studio for WebLogic Server download .zip file.

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-3.1.x.ear` into the domain.

(a) Go to the **Administration Console** at `<hostname>:8101/console`.

(b) Under **Helpful Tools**, click **Configure applications**.

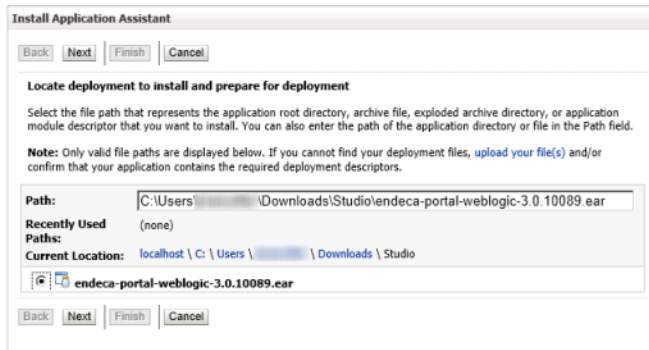


(c) If necessary, click **Lock & Edit** at the top left of the page.

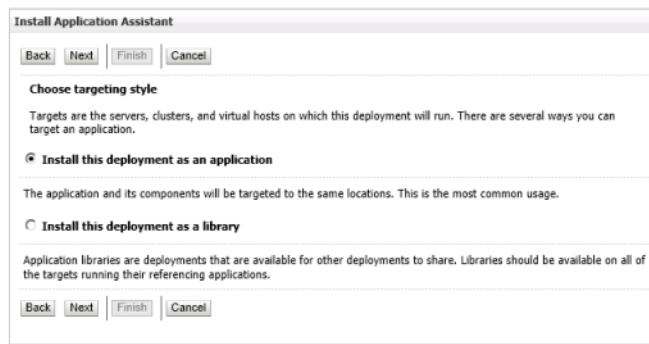
(d) Click the **Install** button.



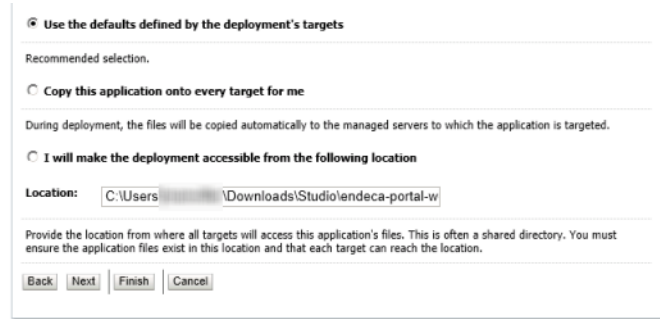
- (e) Use the file browser to find the `endeca-portal-weblogic-3.1.x.ear` file, then click the radio button to the left of it.



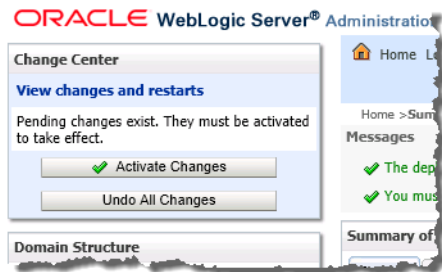
- (f) Click **Next**.
- (g) On the next page, under **Choose targeting style**, make sure the **Install this deployment as an application** radio button is selected, then click **Next**.



- (h) On the next page, accept the default selections, then click **Finish**.



- (i) If necessary, click **Activate Changes**.

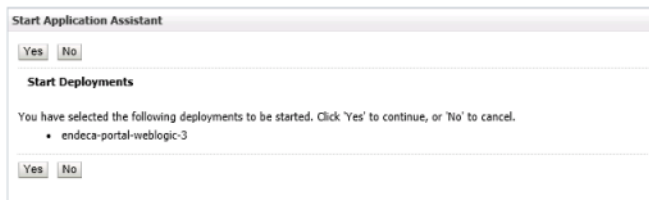


The deployment may take several minutes. Make sure you do not try to log in to Studio before the deployment is complete.

5. When the deployment process is completed, the deployment status is Prepared. To start the deployment:
- (a) In the **Deployments** list, check the checkbox for the Studio deployment.



- (b) Click the **Start** button.
- (c) From the drop-down list, select **Servicing all requests**.
- (d) 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:

- In `<MiddlewareHomeDirectory>\user_projects\domains\<StudioDomain>\servers\AdminServer`, create the following files:
 - `install_studio_service.cmd`
 - `uninstall_studio_service.cmd`
- 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
<code><StudioDomain></code>	The name of your Studio WebLogic domain.
<code><MiddlewareHomeDirectory></code>	The full path to the Middleware Home directory.
<code><StudioDomainUserName></code>	The user name for the Studio domain.
<code><StudioDomainPassword></code>	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/cd/E23943_01/web.1111/e13708/winservice.htm#1188258.

- 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.

4. Go to %WL_HOME%\server\bin.
5. Make a copy of installSvc.cmd, and name it installSvc-studio.cmd.
6. Go to %WL_HOME%\common\bin.
7. Make a copy of commEnv.cmd, and name it commEnv-studio.cmd.
8. Modify commEnv-studio.cmd as follows:
 - (a) 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 (http://docs.oracle.com/cd/E23943_01/web.1111/e13708/winservice.htm).

Changing the Studio home directory on WebLogic Server

For a Studio installation on WebLogic Server, the default location of the Studio home directory is `<MiddlewareHomeDirectory>/user_projects/domains/<StudioDomain>/eid/studio`, where `<MiddlewareHomeDirectory>` 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 53](#).

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-3.1.0.12345.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) Click the **Redeploy this application using the following deployment files** radio button.
 - (g) Under **Deployment plan path**, click the **Change Path** button.
 - (h) In the **Path** field, specify the full path to `change-context-roots-deployment-plan.xml`.
 - (i) Click the radio button next to the file name.
 - (j) Click **Finish**.
 - (k) Click **Activate Changes**.



Chapter 10

Troubleshooting Known Studio Issues

Here are some troubleshooting tips for some general issues that can occur in a Studio instance.

[Changing the JavaScript timeout value on Internet Explorer 8](#)

[Increasing the Tomcat connector keepAliveTimeout for Internet Explorer 8](#)

Changing the JavaScript timeout value on Internet Explorer 8

Internet Explorer 8 keeps track of the number of JavaScript lines executed. After a fixed value, the browser issues an error message, prompting the user to decide whether he or she would like to continue running the script. Because Studio is a rich Internet application that leverages JavaScript heavily in all components, it can trigger this error during normal usage.

Microsoft describes this issue in Knowledge Base Article 175500 and specifies a fix. More detail can be found in the Knowledge Base Article (<http://support.microsoft.com/kb/175500>). The following is a condensed version of Microsoft's fix for the Internet Explorer 8 issue.



Important: If you intend to run Studio on Internet Explorer 8, Endeca strongly recommends that you apply the fix outlined here.

To change the script timeout value:

1. Using a registry editor such as `Regedt32.exe`, open this key:

```
HKEY_CURRENT_USER\Software\Microsoft\Internet Explorer\Styles
```



Note: If the `Styles` key is not present, create a new key that is called `Styles`.

2. Create a new `DWORD` value called "MaxScriptStatements" under this key, and set the value to the desired number of script statements. You will have to try different values for your application environment, but the suggested starting point is a `DWORD` value of `0x1CFFFFF`.



Note: You can turn off this Internet Explorer 8 feature using a `DWORD` value of `0xFFFFFFFF`.

Increasing the Tomcat connector `keepAliveTimeout` for Internet Explorer 8

If your users are using Internet Explorer 8 on a slower network, and experience intermittent timing out of requests to the Endeca Server, you may need to increase the value of `keepAliveTimeout` for Tomcat's HTTP 1.1 connector.

Note that increasing this setting can introduce the risk of lower throughput, or require configuration of additional server connection threads.



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

[Starting and logging in to Studio](#)

[Displaying the Studio Control Panel](#)

Starting and logging in to Studio

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

To start 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
Login:	admin@oracle.com
Password:	Welcome123

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

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

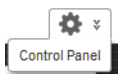
Displaying the Studio Control Panel

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 12

Changing the Database Used to Store Studio Application Data

For production instances, you need to switch from the default Hypersonic (HSQL) database.

[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.

By default, Studio uses Hypersonic (HSQL), which is an embedded database running inside the Java virtual machine. HSQL is useful for standing up a Studio instance very quickly, but must NOT be used in production due to performance issues and its inability to support clustered instances.

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 Hypersonic 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 from Hypersonic to your database.

- (a) The default version of the file has the Hypersonic 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 13

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

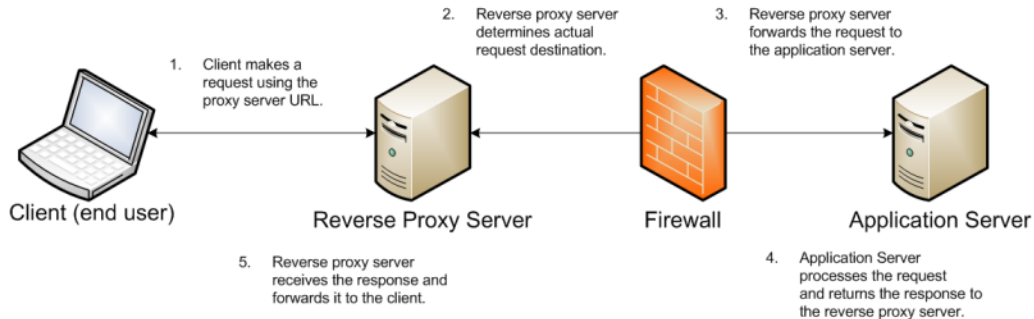
Types of reverse proxies

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.

For the bundled Tomcat configuration (running on port 8080), a sample `mod_proxy` would be:

```
ProxyRequests Off
ProxyPreserveHost On
ProxyPass /eid http://studioserver1:8080/eid/
ProxyPassReverse /eid http://studioserver1:8080/eid/
```

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 Tomcat, in `server.xml`, in the `<Connector>` element, set the `scheme`, `proxyName` and `proxyPort` attributes.

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
```



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

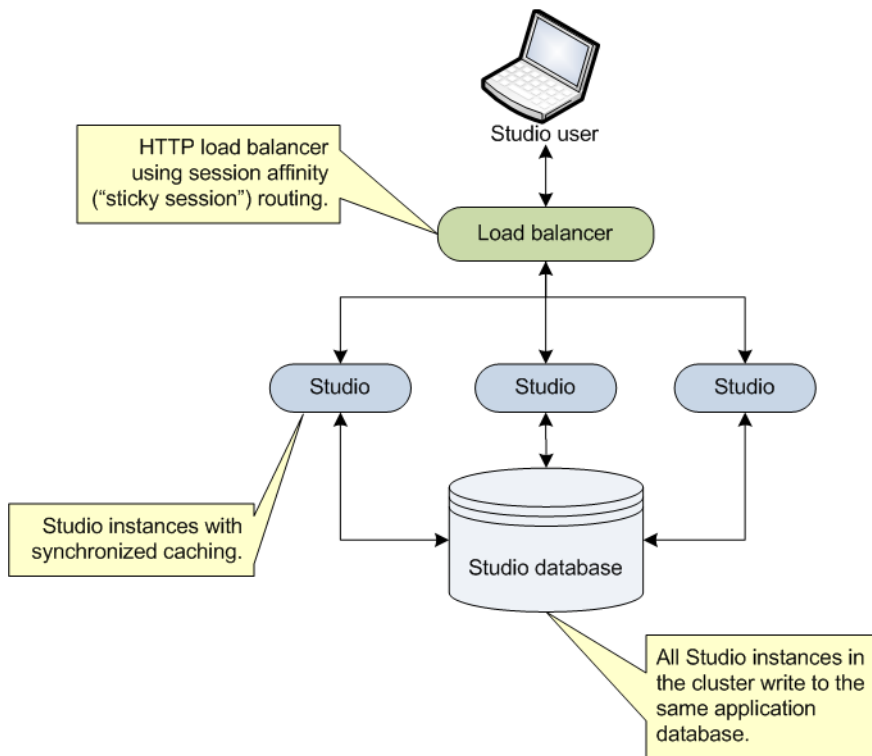
Studio allows you to create clusters of Studio instances. In a cluster, changes made to one instance are automatically made to the other instances. For a large production environment, using clustering provides redundancy and support for higher throughput, allowing for more concurrent users.

The cluster is made up of Studio instances configured to write to the same application database. For a clustered implementation, you must use a database other than the default Hypersonic 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), 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 a clustered configuration, you must use a database system other than the default Hypersonic 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 67](#). 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) 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:

- For Tomcat, in `webapps\ROOT`
- For WebLogic, 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. For Tomcat, copy the customized files into `webapps\ROOT\WEB-INF\classes\ehcache`.

4. 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.

5. 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

```

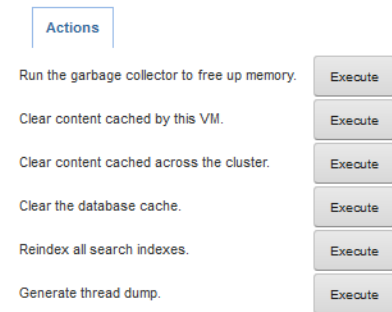
Clearing the cache for a cluster

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 III

Installing the Provisioning Service



Chapter 15

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.

[Provisioning Service installation process](#)

[Downloading and installing Oracle ADF Runtime](#)

[Running the Provisioning Service installation script](#)

[Checking the Provisioning Service installation](#)

[Provisioning Service data domain profile](#)

[Backend database configuration](#)

[Changing communication security configurations for the Provisioning Service](#)

Provisioning Service installation process

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 the Oracle Sun JDK.

For details about installing the Oracle Java JDK, see [Downloading and installing the Sun JDK on page 39](#).

2. Download and install WebLogic Server.

For details about installing the WebLogic Server, see [Downloading and installing WebLogic Server on page 42](#).

3. Download and install the Application Development Framework (ADF) Runtime package.

For details about installing the ADF Runtime package, see [Downloading and installing the ADF Runtime package on page 83](#).

4. Run the installation script, install.py.

This script creates a WebLogic domain and installs the Provisioning Service. For details, see [Running the Provisioning Service installation script on page 89](#)

You can also run this script in silent mode. For details, see [Running the Provisioning Service installation script in silent mode on page 91](#)

Downloading and installing Oracle ADF Runtime

After you install WebLogic Server, install ADF Runtime 11.1.1.6, which contains the JRF needed by the Provisioning Service.

This topic assumes that you have installed Oracle WebLogic Server 10.3.6.

You will need an Oracle account to download the Application Development Framework installer.



Important: The installation instructions in this topic are a distillation of the complete instructions in the *Oracle Fusion Middleware Installation Guide for Application Developer*. The complete instructions provide detailed information about installing the software, such as system requirements. The complete instructions are available online at: http://docs.oracle.com/cd/E23943_01/doc.1111/e14827/toc.htm



Note: Before installing Oracle ADF Runtime on Linux, ensure you:

- Do not run the installation program as the root user.
- Have an X-Windows (X11) environment. The installer requires that your monitor must be configured to display at least 256 colors.



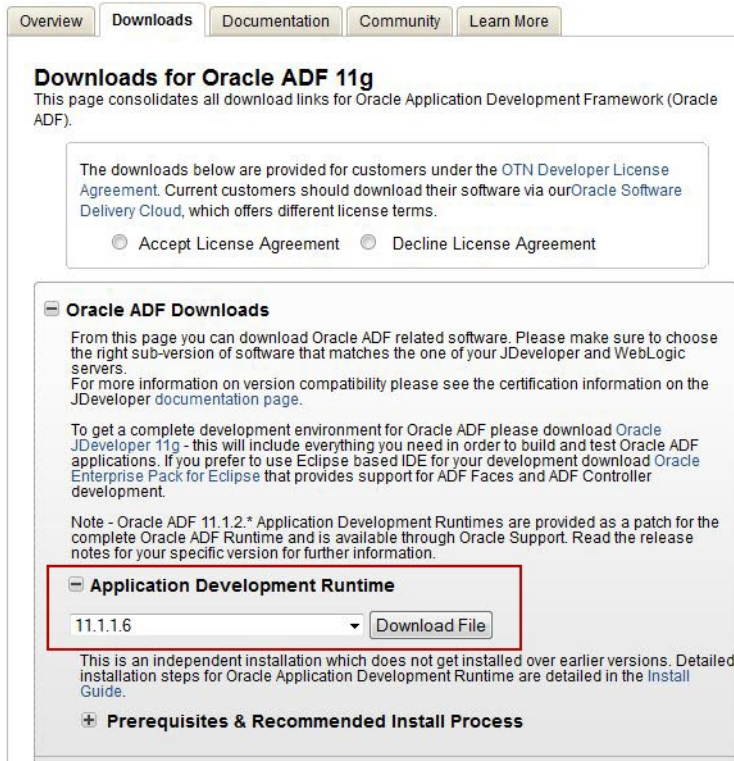
Important: Before installing Oracle ADF Runtime on Windows, ensure that the path to the location where you have installed the Java JDK is C:\Java, or another path that does not include spaces. If you install the Java JDK to the default location (C:\Program Files), installation of Oracle ADF Runtime will fail.

To install Oracle ADF Runtime:

1. Download the ADF package:

(a) Go to <http://www.oracle.com/technetwork/developer-tools/adf/downloads/index.html>.

The following graphic illustrates the location of the **Application Development Runtime** download link:



(b) Click the **Accept License Agreement** radio button at the top of the page.

(c) In the **Application Development Runtime** list, choose 11.1.1.6 and then click **Download File**.

(d) If you are not already logged in, sign in (on the Sign In page) with your Oracle username and password.

(e) Click **Save File** on the download dialog.

2. Unpack the ADF zip package.

The package contains:

- readme.htm file
- Disk1 directory
- Disk2 directory

3. From a command prompt, change to the Disk1 directory.

4. Run the installer:

```
setup.exe -jreLoc <jre_location>
```

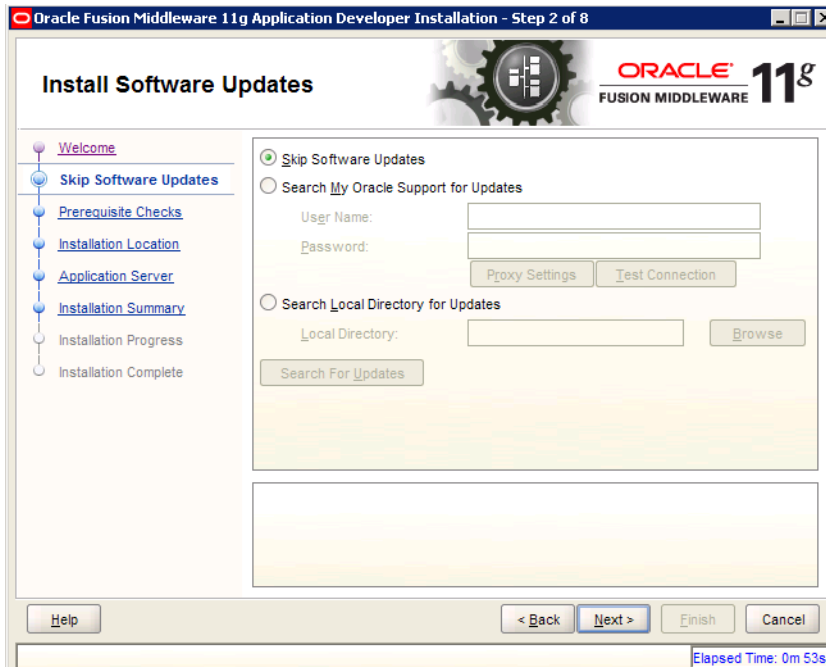
<jre_location> is the full path to the location of the Sun Java 6 JRE (Java Runtime Environment) that you installed. The JRE is located in the JDK installation directory.

For example:

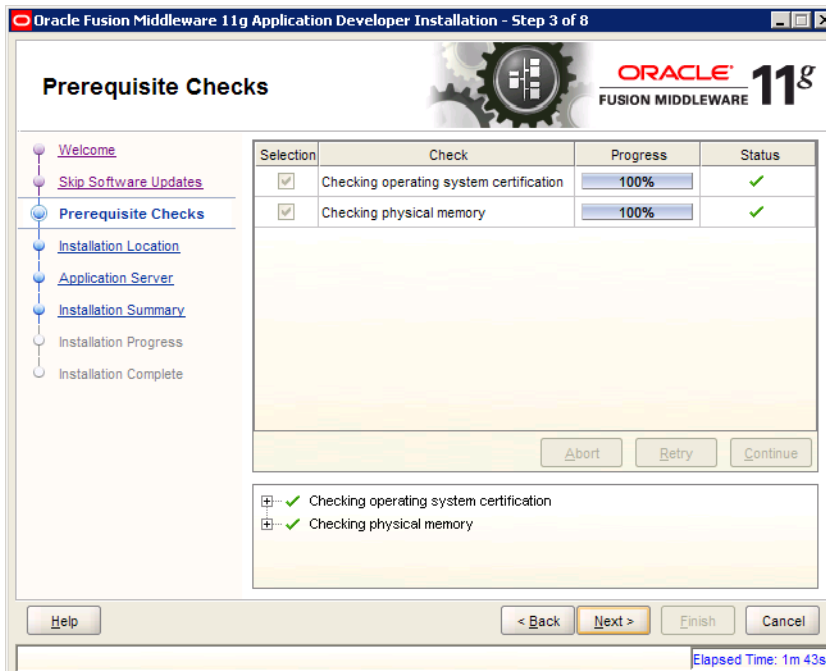
```
setup.exe -jreLoc c:\Java\jdk1.6.0_43
```

The **Welcome** page of the ADF installation wizard is displayed.

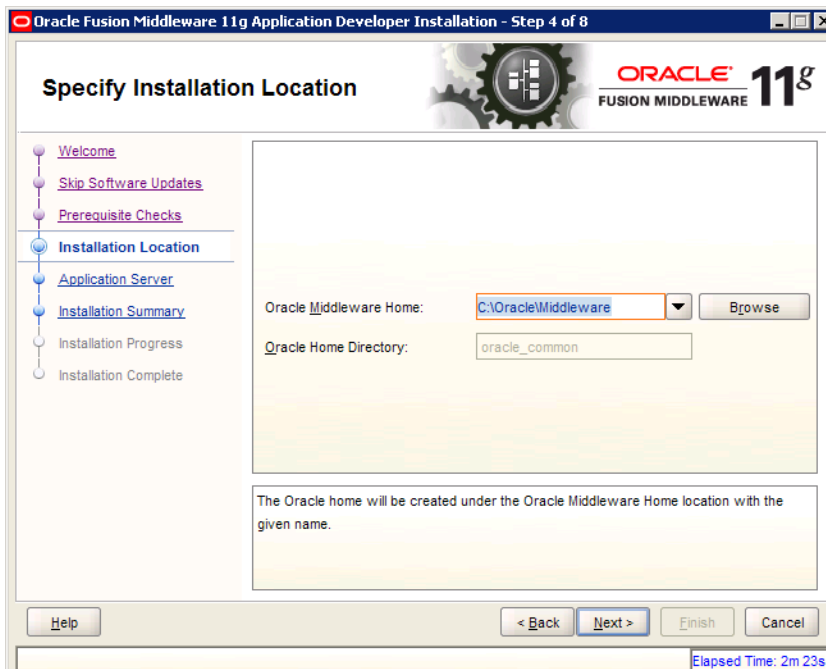
5. On the **Welcome** page, read the information, then click **Next**.
6. On the **Install Software Updates** page, click the **Skip Software Updates** radio button, then click **Next**.



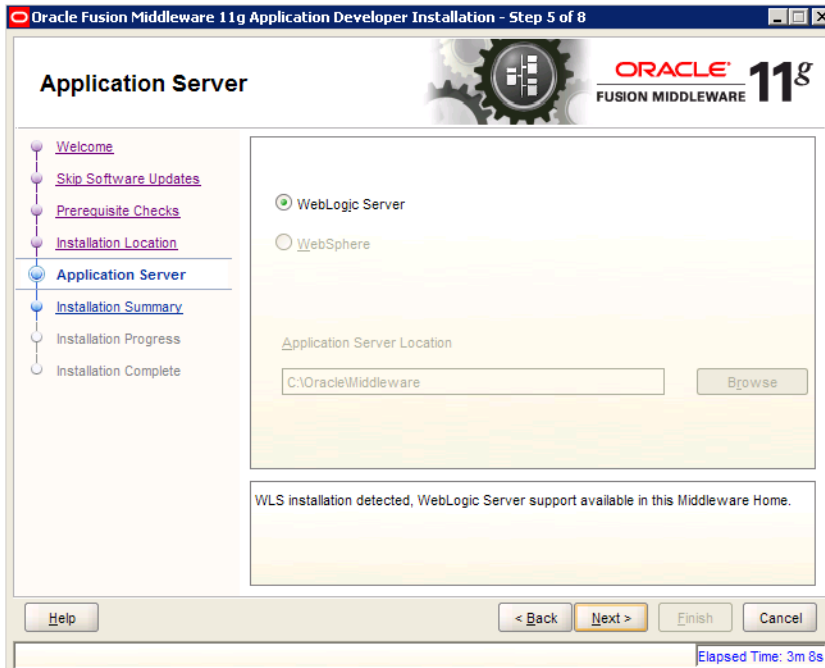
7. On the **Prerequisite Checks** page, wait until the installation process passes all the necessary checks. If everything passes the checks, click **Next**.



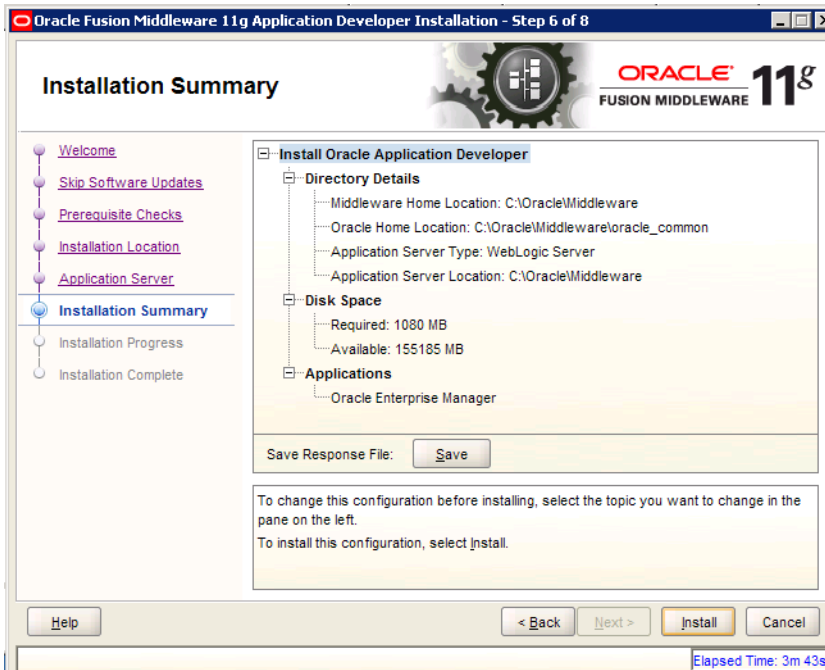
8. On the **Specify Installation Location** page, verify that the Oracle Middleware Home directory is the location where you installed WebLogic, then click **Next**.



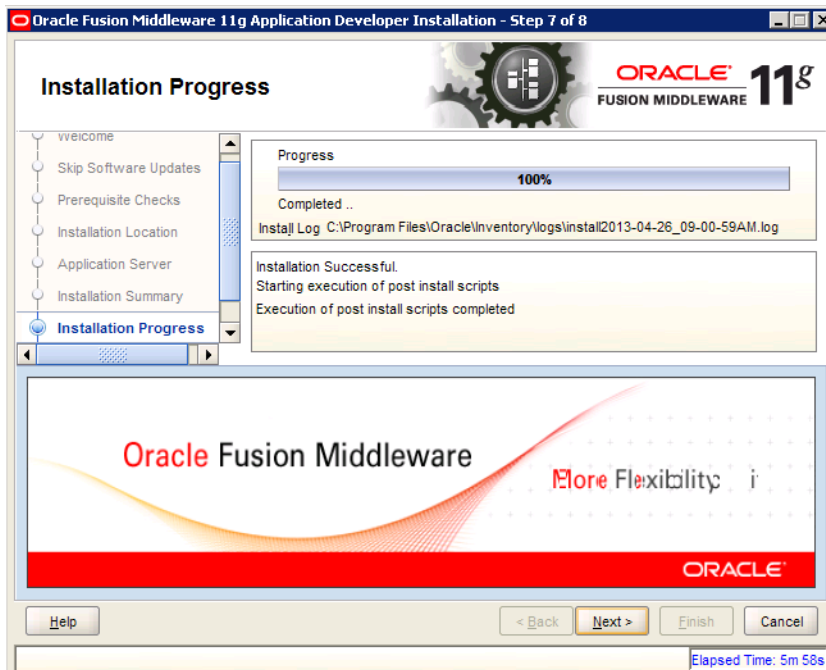
- On the **Application Server** page, click the **WebLogic Server** radio button, then click **Next**.



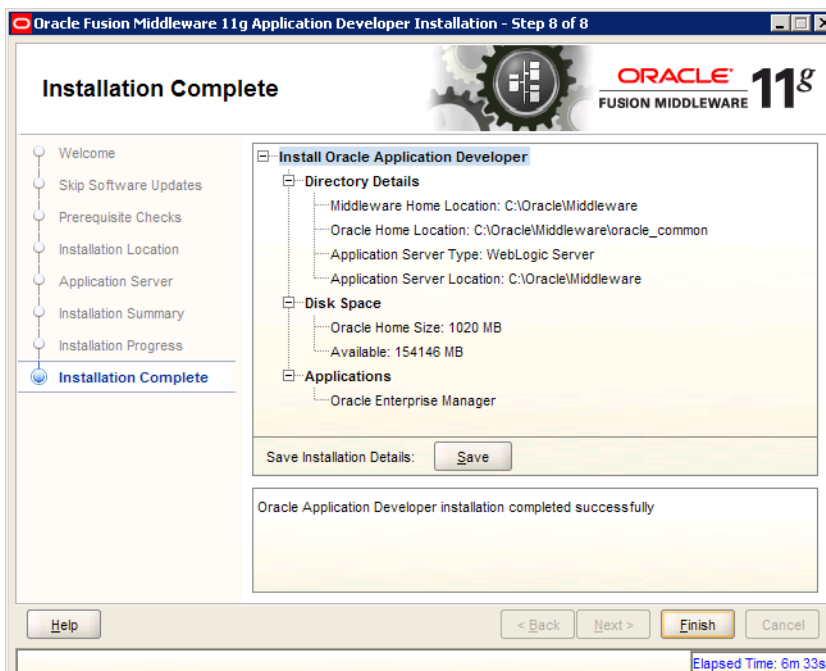
- On the **Installation Summary** page, verify the installation details, then click **Install**.



11. On the **Installation Progress** page, when the installation process has completed, click **Next**.



12. On the **Installation Complete** page, to exit the installer, click **Finish**.



Two shortcuts are created in the **Start Menu**:

- Oracle Application Developer 11g
- Oracle Common Home 11g

Running the Provisioning Service installation script

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

Before running the installation script, be sure you have:

- installed the Oracle Sun JDK. For details, see [Downloading and installing the Sun JDK on page 39](#).
- installed WebLogic Server. For details, see [Downloading and installing WebLogic Server on page 42](#).
- installed the Application Development Framework (ADF). For details, see [Downloading and installing the ADF Runtime package on page 83](#).

To run the Provisioning Service installation script:

1. Unzip the Provisioning Service installation package into a convenient directory.

For example, you could create a directory called `eid_ProvisioningService` at the root of your WebLogic Server installation and unzip the Provisioning Service installation package to that location.

2. Optionally, if you are using SSL, copy the identity keystore and trust keystore from the Endeca Server to the directory you created in Step 1.

The Endeca Server keystore files are stored in the directory `$DOMAIN_HOME/config/ssl` where `$DOMAIN_HOME` is the domain directory of the Endeca Server domain. Copy the following files:

- `endecaServerClientCert.ks`
- `endecaServerTrustStore.ks`

3. Open a terminal window and change to the directory where you unzipped the installation package.

4. Enter the command `$MW_HOME\wlserver_10.3\common\bin\wlst.cmd install.py`.

WebLogic Scripting Tool executes the installation script.

5. The name of the Provisioning Service domain defaults to `endeca_provisioning_service_domain`. Press the enter key to accept the default name, or type a new name and press the enter key.
6. Enter the name of the WebLogic user for the Provisioning Service domain.
7. Enter the password for the WebLogic user for the Provisioning Service domain. The password must include at least eight characters, and one or more of the characters must be a number or special character.
8. The non-SSL port of the Provisioning Service defaults to 8201. Press the enter key to accept the default port, or type a new port and press the enter key.
9. Define the SSL configuration for the Provisioning Service.



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.

- If you do not use SSL for your Oracle Endeca Information Discovery implementation, at the **Would you like to enable SSL to connect to Provisioning Service?** prompt, enter `n` and press the enter key.

- If you do use SSL for your Oracle Endeca Information implementation:
 1. At the **Would you like to enable SSL to connect to Provisioning Service?** prompt, enter `y` and press the enter key.
 2. The default SSL port of the Provisioning Service is 8202. Press the enter key to accept the default port, or type a new port and press the enter key.
 3. Enter the identity keystore password you specified when creating the identity keystore on Endeca Server.
 4. Enter the trust keystore password you specified when creating the trust keystore on Endeca Server.
- 10. Enter the host of the Endeca Server. You can enter either the host name or the IP address.
- 11. 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.
- 12. When prompted, specify the database configuration for the backend database.

The Provisioning Service supports three options for the backend database:

- Derby

The Provisioning Service can use 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.
- 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.

Running the Provisioning Service installation script in silent mode

Running the Provisioning Service installation script in silent mode

You can run the Provisioning Service installation script in silent mode.

Before running the installation script, be sure you have:

- installed the Oracle Sun JDK. For details, see [Downloading and installing the Sun JDK on page 39](#).
- installed WebLogic Server. For details, see [Downloading and installing WebLogic Server on page 42](#).
- installed the Application Development Framework (ADF). For details, see [Downloading and installing the ADF Runtime package on page 83](#).

To run the Provisioning Service installation script in silent mode:

1. Unzip the Provisioning Service installation package into a convenient directory.

For example, you could create a directory called `eid_ProvisioningService` at the root of your WebLogic Server installation and unzip the Provisioning Service installation package to that location.

2. Copy the identity keystore and trust keystore from the Endeca Server to the directory you created in Step 1.

The Endeca Server keystore files are stored in the directory `$DOMAIN_HOME/config/ssl` where `$DOMAIN_HOME` is the domain directory of the Endeca Server domain. Copy the following files:

- `endecaServerClientCert.ks`
- `endecaServerTrustStore.ks`



Note: You can skip this step if you are implementing non-secure communication (in other words, if you are not using SSL for the Endeca Server and Studio).

3. Configure the installation script to run in silent mode.
 - (a) Using a simple text editor, open the installation script, `install.py`.
 - (b) Change the value of the `SILENT_INSTALL` property to `true`.
 - (c) Edit the following data:

- `PROVISIONING_SERVICE_USERNAME`
The name of the admin user for the Provisioning Service domain.
- `PROVISIONING_SERVICE_PASSWORD`
The password of the admin user for the Provisioning Service domain.

- **JDBC_DRIVER**

The driver used to connect to the back-end database. The following drivers are included with the Provisioning Service:

- Derby: `org.apache.derby.jdbc.EmbeddedDriver`

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.

- Oracle: `oracle.jdbc.xa.client.OracleXaDataSource`
- MySQL: `com.mysql.jdbc.Driver`

Defaults to `org.apache.derby.jdbc.EmbeddedDriver`

- **JDBC_URL**

The connect string used to connect to the back-end database for provisioning service.

Defaults to `'jdbc:derby:memory:testDB;create=true'` to support a Derby data base.

- If you use an Oracle database as the backend database for the Provisioning Service, the connect string format is `jdbc:oracle:thin:@[HOST]:[PORT]:[SID]`.
- If you use a MySQL database as the backend database for the Provisioning Service, the connect string format is `jdbc:mysql://[HOST]:[PORT]/[DATABASENAME]`.

- **JDBC_USER**

The user name that the Provisioning Service uses to connect to the backend database.

- **JDBC_PASS**

The password of the user that the Provisioning Service uses to connect to the backend database.

- **PROVISIONING_SERVICE_PORT**

The port on which the Provisioning Service will listen for non-SSL requests. Defaults to 8201.

- **PROVISIONING_SERVICE_PORT_SSL**

The port on which the Provisioning Service will listen for SSL requests. Defaults to 8202.

- **ENDECA_SERVER_ADDRESS**

Specifies the machine where the Endeca Server is installed. The value can be either a host name or an IP address. Defaults to `localhost`, or an Endeca Server installed on the same machine as the Provisioning Service.

- **ENDECA_SERVER_PORT**

Specifies the port on which the Endeca Server listens. Defaults to 7002, the default Endeca Server SSL port.

- **ENDECA_SERVER_SSL**

Flag specifying whether SSL is enabled on the Endeca Server. Defaults to `true`.

(d) Save your changes.

4. To configure SSL:



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.

- To configure the Provisioning Service not to use SSL, change the value of the `USE_SSL` property to `false`.
- To configure the Provisioning Service to use SSL:
 1. Leave the value of the of the `USE_SSL` property as `true`.
 2. In the `KEYSTORE_IDENTITY_PASS` property, enter the password you specified when creating for the `EndecaServerClientCert.ks` keystore file on the Endeca Server.
 3. In the `KEYSTORE_TRUST_PASS` property, enter the password you specified when creating for the `EndecaServerTrustStore.ks` keystore file on the Endeca Server.
- 5. Save your changes.
- 6. Open a terminal window and change to the directory where you unzipped the installation package.
- 7. Run the installation script. In the commands, `$MW_HOME` is the directory where you installed WebLogic Server and `path` is the path to the directory you created in Step 1.
 - In Windows, run the command `$MW_HOME/oracle_common/common/bin/wlst.cmd path/install.py`.
 - In Linux, run the command `$MW_HOME/oracle_common/common/bin/wlst.sh path/install.py`.

WebLogic Scripting Tool executes the installation script.

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.

Checking the Provisioning Service 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 3.1.0.0 Revision 2426

Endeca Server Communication SUCCESS

Backend Database Access SUCCESS

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

Provisioning Service data domain profile

During installation, an Endeca Server data domain profile is created.

The Provisioning Service data domain profile is named `eid-provisioning-service`. It is a clone of 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 sets up the backend database for the Provisioning Service. If you want to use a different RDBMS, change the configuration manually.

The database schema you want to use to support the Provisioning Service must be set up before you can begin to use the Provisioning Service. Scripts are provided to create the schema.

- `oracleCreateDDL.sql`

Use this script to create the schema in an Oracle database.

- `mysqlCreateDDL.sql`

Use this script to create the schema in a MySQL database.

These scripts are stored 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 with the user used to install 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. Click **Delete**. WebLogic Server deletes the selected datasource.
4. Create a new datasource with the JNDI name `jdbc/oracle.eid-ps`. For an Oracle database:
 - For an Oracle database:
 - (a) Click **New** and from the drop menu, choose `Generic Data Source`. 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, choose `Oracle` (if you want to use an Oracle database) or `MySQL` (if you want to use a MySQL database).
 - (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. Expand the **Advanced** section.
 - (m) Check the **Check Connections On Reserve** box and 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 communication security configurations for the Provisioning Service

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.

[Implementing SSL communication with Endeca Server and Studio](#)

[Configuring the Provisioning Service for non-secure communication](#)

Implementing SSL communication with Endeca Server and Studio

By default, the Provisioning Service is configured to support SSL. Additional implementation is required to enable SSL communication with Endeca Server and Studio.

To implement secure communication with Endeca Server and Studio over SSL:

1. Copy the keystore cert files from the home directory of the Endeca Server installation to the domain home directory of the Provisioning Service.
Copy the following cert files:
 - `endecaServerClientCert.ks`

- `endecaServerTrustStore.ks`
2. Configure the identity keystore and trust keystore on the WebLogic Server.
 - (a) Start the Provisioning Service WebLogic domain.
 - (b) Start a browser, go to the Administration Console for your server and log in.
 - (c) On **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**.

Configuring the Provisioning Service for non-secure communication

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, in `plan.xml`:

- Change the value of the `endeca-server-security-enabled` variable to `false`.
- Change the value of the `transport-guarantee` variable to `NONE`.
- Change the value of the `protected-url-pattern` variable to `/DISABLED`.

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



Chapter 16

Configuring the Studio Connection to the Provisioning Service

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

If the connection configuration is not valid, 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

Uninstalling Oracle Endeca Information Discovery



Chapter 17

Uninstalling Oracle Endeca Information Discovery Studio

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

A

- ADF Runtime
 - installing 83

C

- clustering, Studio
 - about 75
 - clearing the cache 79
 - customizing the cache configuration 78
 - enabling synchronized caching 77
 - installing instances 76
- context root
 - changing for the Linux Tomcat bundle 25
 - changing for the Windows Tomcat bundle 22
- Control Panel, displaying in Studio 66

D

- database
 - changing from Hypersonic in Studio 68
 - configuring Provisioning Service for production 94
- data domain profile 94

I

- installation
 - Provisioning Service 82
 - Studio Linux Tomcat bundle 24
 - Studio on Oracle WebLogic Server 38
 - Studio on Tomcat 6.0 27
 - Studio Windows Tomcat bundle 21
- installing
 - ADF Runtime 83

L

- languages, supported 9

M

- MySQL database 94

O

- Oracle database 94

P

- Provisioning Service 89
 - connecting Studio to 97
 - installing 82
 - silent installation 91
 - SSL 95

- system requirements 15
- uninstalling 101

R

- reverse proxy, using with Studio 70

S

- SSL
 - Provisioning Service 95
- Studio
 - changing from the Hypersonic database 68
 - clustering, about 75
 - clustering, about synchronized caching 77
 - clustering, clearing the cache 79
 - clustering, customizing the cache configuration 78
 - clustering, enabling synchronized caching 77
 - clustering, installing instances 76
 - connecting to the Provisioning Service 97
 - Control Panel, displaying 66
 - downloading 17
 - installing the Sun JDK for WebLogic Server 39
 - installing WebLogic Server 42
 - Linux Tomcat bundle installation steps 24
 - Oracle WebLogic Server installation steps 38
 - running as Windows service (Tomcat) 33
 - running as Windows service (WebLogic) 58
 - starting 66
 - supported languages 9
 - system requirements 13
 - Tomcat 6.0 installation 27
 - uninstalling 101
 - using a reverse proxy 70
 - Windows Tomcat bundle installation steps 21
- Studio troubleshooting
 - JavaScript timeout 64
 - starting the Tomcat Windows service 36
 - Tomcat connector keepAliveTimeout 65
- Sun JDK
 - downloading for WebLogic Server 39
 - installing for WebLogic Server 39
- system requirements
 - Provisioning Service 15
 - Studio 13

T

- Tomcat 6.0
 - about the Studio install 27
 - configuring Tomcat 29
 - deploying Studio 30
 - deploying Studio dependency libraries 28
 - installing Tomcat 28

- starting Studio 30
- Tomcat bundle (Linux)
 - changing the context root 25
 - installing 24
- Tomcat bundle (Windows)
 - changing the context root 22
 - installing 21

U

- uninstalling
 - Provisioning Service 101
 - Studio 101

W

- WebLogic Server

- context path, changing for Studio 62
- creating the Studio domain 48
- deploying Studio 53
- downloading 42
- installing 42
- installing the Sun JDK 39
- running Studio as a Windows Service 58
- Studio home directory, changing 60
- Windows service for Studio (Tomcat)
 - about 33
 - configuring 34
 - installing 35
 - installing Tomcat monitor 33
 - obtaining installer files 33
 - starting 35
 - troubleshooting 36
- Windows Service for Studio (WebLogic) 58