Oracle® Endeca Information Discovery

Installation Guide

Version 2.4.0 • November 2012

Copyright and disclaimer

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

Part I: Before You Begin

Part II: Installing Integrator

Part III: Installing Studio

Part IV: Uninstallation Tasks

Preface

Oracle® Endeca Information Discovery Studio is an enterprise data discovery platform for advanced, yet intuitive, exploration and analysis of complex and varied data.

Information is loaded from disparate source systems and stored in a faceted data model that dynamically supports changing data. This integrated and enriched data is made available for search, discovery, and analysis via interactive and configurable applications. Oracle Information Discovery Studio includes a Provisioning Service that allows you to upload data directly from spreadsheet files.

Oracle Endeca Information Discovery Studio enables an iterative "model-as-you-go" approach that simultaneously frees IT from the burdens of traditional data modeling and supports the broad exploration and analysis needs of business users.

About this guide

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

Installing Oracle Endeca Information Discovery consists of installing:

- Integrator
- Studio

Oracle Endeca Information Discovery requires the Oracle Endeca Server.

Who should use this guide

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

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

Oracle Endeca Information Discovery Installation Overview

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

About this release

Read this section to understand the assumptions under which you can use this release. These aspects of the release may change in the future, and may represent limitations in the configuration process or feature availability.

For this release:

- **Support for the Oracle Endeca Server, Studio, and Integrator is limited to those versions that are included in this release of Oracle Endeca Information Discovery**. This release relies on the data files created with the version of the Oracle Endeca Server that supports Oracle Endeca Information Discovery. Similarly, this release only supports Studio and Integrator versions that are compatible with Oracle Endeca Information Discovery.
- **Options for loading the data sources**. In this release, the recommended way of loading the data is through Integrator.
- **Notes about the configuration and development process**. In this release, the configuration and development process takes place in Integrator and is complemented by options available to the power users in Studio.

Oracle Endeca Information Discovery language support

Oracle Endeca Information Discovery has been optimized to support ingest, display, full search, navigation and text enrichment for the English language. It also provides language packs for French, German, Spanish, Italian, Dutch and Portuguese enabling search capabilities. Currently, all product interfaces are provided in English only.

Overview of the Oracle Endeca Information Discovery modules

Oracle Endeca Information Discovery is made up of three major modules.

- Oracle Endeca Server 7.4

- Integrator
- Studio

The Oracle Endeca Server has a separate download and a separate documentation set, including a separate installation guide.

Note: These Oracle Endeca Information Discovery 2.4.x modules are compatible with each other. However, modules shipped as part of Oracle Endeca Information Discovery 2.4.x are not compatible with earlier versions.

About the Oracle Endeca Server

The Oracle Endeca Server software provides the query engine that serves as the foundation for all front-end applications utilizing it. The software includes the Oracle Endeca Server which is the management facility for administering the Endeca data stores.

Front-end applications built on top of the Oracle Endeca Server can answer end-user queries and provide business analytics information to the front-application's users.

Note: The term Oracle Endeca Server is used in two related contexts — on the one hand, it is the name of the entire software package for the server. On the other hand, the Oracle Endeca Server is the management facility for administering your data stores.

The Oracle Endeca Server lets you administer your Endeca data stores. The Endeca data store is where the source data records are stored after being loaded from your ETL client (typically, Integrator). With the Oracle Endeca Server, you can:

- Create an Endeca data store, including assigning it a name that will later be used to access and manage it.
- Stop and start Endeca data stores.
- View the status of any Endeca data store.
- List the jobs that are running on a specific Endeca data store.

In addition, the Oracle Endeca Server constantly monitors the status of the running Endeca data stores and will automatically restart them (for example, if one crashes).

As the query engine, the Dgraph process of the Oracle Endeca Server uses proprietary data structures and algorithms that allow it to provide real-time responses to client requests issued to the data store that is running on the server. The Dgraph process receives client requests via the application tier, queries the data files, and returns the results.

Each Endeca data store contains a set of versioned Web services for loading, configuring, and querying the data. These Web services, together with the Bulk Load Interface provide the interface to the Oracle Endeca Server.

About the cluster of nodes

A cluster is composed of a set of nodes, all of which can serve query requests. Only one node is identified as the leader node; All other nodes are follower nodes. One copy of the data files is shared and used by all nodes. The Cluster Coordinator provides communication between the nodes in the cluster. It also notifies the reader nodes about index updates and updates to the configuration.

If one of the cluster nodes fails, queries continue to be processed by other nodes in the cluster. A cluster also provides increased throughput by the Oracle Endeca Server. By adding nodes to a cluster you can spread the query load across multiple Oracle Endeca Servers without the need to increase storage requirements at the same rate. Nodes can be added or removed dynamically, without having to stop the cluster.

In the development environment, you can start with a single Oracle Endeca Server that is not part of a cluster. (Without the cluster services, having a single running Oracle Endeca Server is a valid configuration for starting in the development environment.) You can then expand your single instance implementation by first building a simple single-node cluster configuration and then adding more nodes. When you move to a production environment, you can duplicate a multi-node cluster that you built in the development environment.

In this release, the cluster implementation requires that you download and install the Cluster Coordinator package. For information on configuring and running the cluster, see the chapter on clustering in the *Oracle Endeca Server Administrator's Guide*.

About Integrator

Oracle Endeca Information Discovery's Integrator module is a high-performance data integration platform that lets you extract source records from a variety of source types (from flat files to databases).

Integrator then loads that data in an Endeca data store via the Data Ingest Web Service or the Bulk Load Interface.

From a high level, the Integrator suite consists of:

- **Integrator.** With its powerful graphical interface, you can use the Integrator client to build graphs that can load source data into the an Endeca data store, as well as the schema for your attributes and the index configuration documents for various Endeca features.
- **Information Discovery connectors.** Developed by Endeca, these connectors are Integrator components that are easily configured to perform various data ingest operations on the Endeca data stores.
- **Integrator Server.** The Server is used to run graphs in an enterprise-wide environment. In this environment, different users and user groups can access and run the graphs.

About Studio

Oracle Endeca Information Discovery's Studio enables rapid configuration of dashboard applications that offer the highly interactive Guided Navigation® user experience across a full range of structured and unstructured enterprise data.

Studio is easy to deploy and ideal for the agile development of enterprise-quality applications. Due to the component-based nature of Studio, these applications are simple to control, adapt, and extend. It provides granular layout and configuration control to enable users to manage and personalize their own experiences.

Studio consists of an enterprise-class portal framework and a library of UI components that embody best practices in Information Discovery applications.

About the Component SDK

The Component SDK is a packaged development environment for components, themes, layout templates, and other portal elements.

The installation and use of the Component SDK is covered in the *Oracle Endeca Information Discovery Studio Developer's Guide*.

Installation order

Following the recommended order of installation helps you minimize component dependencies.

Oracle recommends that you install the components in this order:

1. Oracle Endeca Server. For instructions on installing Oracle Endeca Server, including system requirements, see the *Oracle Endeca Server Installation Guide*.
2. Integrator (see *Installing Integrator on page 14*). After installing, open Integrator to verify that it was installed correctly.
3. Studio (see *About the Studio Installation Process on page 21*). After installing, to verify the installation, log in to Studio. Use a created Endeca data store to provide the data source.
4. Integrator Server (optional, see *Installing Integrator on page 14*). After installing, use a browser to access the Integrator Server home page and log in.

Chapter 2

System Requirements

Oracle Endeca Information Discovery's Integrator and Studio modules have the following system requirements. For details on the Oracle Endeca Server system requirements, see the *Oracle Endeca Server Installation Guide*.

Integrator system requirements

The Integrator components have the following installation requirements.

The 2.4.x version of Integrator is compatible only with the 7.4.x version of the Oracle Endeca Server.

Hardware requirements

The 64-bit version of Integrator client has the following hardware requirements documented in *Hardware requirements on page 6*.

Supported operating systems

The 64-bit version of Integrator client is supported on the operating systems documented in *Supported operating systems on page 7*.

Note: The *Oracle Endeca Information Discovery Integrator Designer Guide* lists other supported operating systems. However, that list is superseded by the list in the linked topic above.

Software requirements

- The only pre-requisite for an Integrator client installation on Linux is an X Window System.
- Integrator client requires Java version 6 (also called 1.6) JDK, which is included in the Integrator installer.
- Integrator Server must be installed on a stand-alone Apache Tomcat, version 6.0.x. Note that you should not install Integrator Server on a Tomcat that is hosting another application, such as Studio.

Hardware requirements

Integrator has the following hardware requirements.

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.

Windows and Linux on x64

Minimum hardware requirements:

- x64 processor, minimum 1.8 GHz
- At least 3 GB of RAM, depending on the size of the application data set
- 80 GB hard drive, depending on the size of the application data set

Recommended hardware requirements:

- x64 3.0+ GHz processors; Intel Xeon (including Nehalem) or AMD Opteron processors are recommended
- 8 GB of RAM or more, depending on the size of the application data set
- High performance network-attached storage (for example, attached via a dedicated iSCSI or fibre channel network), or high performance locally-attached RAID storage (for example, a RAID 6 or RAID 0+1 array with battery-backed write caching, operating on 72GB or 146 GB spindles at 10k or 15k RPM spindle speed)
- Gigabit Ethernet

Supported operating systems

Integrator supports the following 64-bit operating systems running on servers with x64 capabilities:

Operating System	Description
Oracle Linux 5	Only the Red Hat Compatible Kernel is supported.
Linux RHEL 5	• Red Hat Enterprise Linux Server (version 5 for x64) running on x64 processors. • Red Hat Enterprise Linux Advanced Platform (version 5 for x64) running on x64 processors. For best performance on Red Hat Linux version 5 (Server and Advanced), Endeca recommends the latest version of RHEL 5.
Windows 2008	Windows Server 2008 R2 Enterprise running on x64 processors.

Notes

Windows 7 is not supported for production deployment, but operates sufficiently to enable training and small-scale development work.

Windows XP is not supported.

Studio system requirements

Here are the system requirements for Studio.

The 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 7.4.x version of Oracle Endeca Server.

Hardware requirements

Minimum hardware requirements:

- x64 processor, minimum 1.8 GHz
- At least 2 GB of RAM, depending on the size of the application data set
- 80 GB hard drive, depending on the size of the application data set

Recommended hardware requirements:

- x64 3.0+ GHz processors; we recommend Intel Xeon (including Nehalem) or AMD Opteron processors
- 8 GB of RAM or more, depending on the size of the application data set
- High performance network-attached storage (for example, attached via a dedicated iSCSI or fibre channel network) or high performance locally-attached RAID storage (for example, a RAID 6 or RAID 0+1 array with battery-backed write caching, operating on 72GB or 146 GB spindles at 10k or 15k RPM spindle speed)
- 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 5	Oracle Linux 5 (version 5 for x64) running on x64 processors. Only the Red Hat Compatible Kernel is supported.
Linux RHEL 5	• Red Hat Enterprise Linux Server (version 5 for x64) running on x64 processors. • Red Hat Enterprise Linux Advanced Platform (version 5 for x64) running on x64 processors. For best performance on Red Hat Linux version 5 (Server and Advanced), we recommend the latest version of RHEL 5.
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	• Tomcat 6 • WebSphere Application Server (WAS) 7 • Oracle WebLogic Server 11gR1 (10.3.5)
Java	• Tomcat 6 and WebLogic are supported with Sun Java 6 • WAS 7 is supported with IBM Java 6 For Sun Java 6, update 18 or greater is required.
Database system	• MySQL 5.1 • Oracle 11g
Browser	• Firefox ESR on Windows • Internet Explorer 8 (with compatibility mode disabled) on Windows Firefox is recommended. ★ **Important:** Running Internet Explorer 8 in compatibility mode is not supported.
Browser plugin	Adobe Flash 10.0

Alternative database support

Studio uses a relational database to store configuration and state. By default, Studio uses Hypersonic, but this 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.

The Studio section of this guide provides instructions on how to switch to another database system.

Chapter 3

Downloading the Oracle Endeca Information Discovery software

The Oracle Endeca Information Discovery modules are downloaded from the Oracle Software Delivery Cloud.

This procedure includes the downloads for each platform for:

- Integrator
- Studio

To download the Oracle Endeca Information Discovery 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 (2.4) Media Pack v1 for Microsoft Windows x64 (64-bit)**.

   For the Linux platform, the media pack is **Oracle Endeca Information Discovery (2.4) 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. The Oracle Endeca Information Discovery package includes both Integrator and Studio.

   For Windows, download **Oracle Endeca Information Discovery (2.4) for Microsoft Windows x64 (64-bit)**. The downloaded file, `EID24_Win.zip`, contains:

`EID_2.4_Integrator_designer.exe`	Installer for Integrator 64-bit client.
`EID_2.4_Integrator_server_tomcat.zip`	Installer for Integrator Server for Tomcat.
`EID_2.4_Integrator_server_weblogic.zip`	Installer for Integrator Server for WebLogic.
`EID_2.4_Studio_portal.zip`	Studio Tomcat bundle.

`EID_2.4_Studio_portal_dependencies.zip`	Studio dependencies file for the WAS installation.
`EID_2.4_Studio_portal_war.zip`	Studio .war file for the WAS installation.
`EID_2.4_Studio_components.zip`	Studio components. Needed for the Tomcat bundle and WAS installations.
`EID_2.4_Studio_portal_weblogic.zip`	Files for the Studio WebLogic installation.
`EID_2.4_Studio_components_sdk.zip`	Studio Component SDK. Used for development of custom components. For information on developing custom components, see the *Oracle Endeca Information Discovery Studio Developer's Guide*.

For Linux, download **Oracle Endeca Information Discovery (2.4) for Linux x86-64**. The downloaded file, `EID24_Linux.zip`, contains:

`EID_2.4_Integrator_designer.sh`	Installer for Integrator client.
`EID_2.4_Integrator_server_tomcat.zip`	Installer for Integrator Server for Tomcat.
`EID_2.4_Integrator_server_weblogic.zip`	Installer for Integrator Server for WebLogic.
`EID_2.4_Studio_portal.tgz`	Studio Tomcat bundle.
`EID_2.4_Studio_portal_dependencies.zip`	Studio dependencies file for the WAS installation.
`EID_2.4_Studio_portal_war.zip`	Studio .war file for the WAS installation.
`EID_2.4_Studio_components.zip`	Studio components. Needed for the Tomcat and WAS installations.
`EID_2.4_Studio_portal_weblogic.zip`	Files for the Studio WebLogic installation.
`EID_2.4_Studio_components_sdk.zip`	Studio Component SDK. Used for development of custom components.

You can also download the **Oracle Endeca Information Discovery (2.4) Documentation**. This .zip file contains the Integrator and Studio release notes, installation guides, and third-party licensing information.

The full documentation set is available from the Endeca Information Discovery documentation page on the Oracle Technology Network (*http://www.oracle.com/technetwork/middleware/endeca/documentation/index.html*).

Part II

Installing Integrator

Chapter 4

Installing Integrator

This section describes how to install Integrator on Linux and Windows platforms.

Integrator installation packages

There are Integrator installation packages for Linux and Windows clients, and one for the Server.

Linux client installer

The Linux client installer is a shell script file named `EID_2.4_Integrator_designer.sh`.

Windows client installers

The 64-bit installer executable file is named `EID_2.4_Integrator_designer.exe`.

This installation contains a complete version of the Integrator client utility, including the Endeca connectors.

Integrator Server installation package

The Integrator Server installation package is named `EID_2.4_Integrator_server.zip`. This ZIP file contains two components:

- The `clover.war` file contains the server version of the Integrator, including the Endeca connectors.
- The `clover-license.war` file contains the license for the Integrator Server.

The contents of the package can be installed on either a Linux or Windows machine.

Integrator installation on Windows

This topic describes how to install Integrator on a Windows client machine.

Integrator is installed as a machine-wide installation on Windows. This type of installation requires administrator permissions and is used because it is intended for a production environment.

This procedure assumes that you have downloaded the Integrator installer for Windows.

To install Integrator on a Windows client:

1. Double-click the installer file to start the wizard.

   The wizard verifies the contents of the installation package.
2. Click **Next** to begin the installation process.
3. In the **Copyright and Legal** screen, click **Next**.
4. In the **Destination Folder** screen, either accept the default installation folder or select another installation location. Then click **Next**.

   You cannot install Integrator into a folder that contains content.
5. In the **Completing the Setup Wizard** screen, click **Next**.

   The wizard begins to install the Integrator files.
6. When the wizard confirms that you have successfully completed the installation, click **Finish**.

After installation, double-click the Integrator shortcut icon to start the program.

Integrator installation on Linux

This topic describes how to install Integrator on a Linux client machine.

This procedure assumes that you have downloaded the Integrator installer for Linux.

To install Integrator on a Linux client:

1. Determine where you will install the Integrator client. Verify that the target directory on which you plan to install has write permissions (is not read-only).

   For example, in this procedure we assume that the target directory is `/localdisk` and that you have write permissions for it. If you do not set these permissions, the install script will not run.
2. Run the Integrator installation script with a command similar to this example:

   ```
   ./EID_2.4_Integrator_designer.sh --target /localdisk
   ```
3. The copyright notice displays first. Scroll to the end of the copyright notice. When you reach the end of the notice, the installation begins.

   The installer displays a message that it is about to extract files in the specified directory. The installer also checks that the directory has enough available disk space, and that it can write to this directory. If these conditions are met, the installer proceeds with the installation and completes it. If they are not met, the installer issues an error and discontinues the installation.

The installation is finished when you see this confirmation message:

```
Installation complete!
```

As a result of the installation, Integrator is installed in:

```
Oracle/Endeca/Discovery/2.4.0/Integrator
```

To start up Integrator, run the `integrator` executable file in the `2.4.0/Integrator` directory.

Integrator Server installation

You can install Integrator Server on Tomcat and Oracle WebLogic web application servers.

Installing Integrator Server on Tomcat

Integrator Server is supported on Apache Tomcat version 6.0.x.

Integrator Server must be installed on a stand-alone Apache Tomcat. Do not install Integrator Server on a Tomcat server that is being used to host other applications. For example, do not install Integrator Server on the Tomcat that is hosting Studio.

Java SDK Version 1.6.x or higher is required.

Set the following environment variables:

- `JAVA_HOME`
- `JRE_HOME`

> **Note:** If `JRE_HOME` is not set, the value defaults to the value of the `JAVA_HOME` environment variable.

Download the Oracle Endeca Integrator Server for Tomcat package from the *Oracle Software Delivery Cloud*.

To install Integrator Server:

1. If you have not already installed Apache Tomcat 6.0.x:
   (a) Download the binary distribution from: *http://tomcat.apache.org/download-60.cgi*.
   (b) Unpack the ZIP file to a directory of your choosing.
   (c) By default, Tomcat starts up on HTTP connector 8080. If another application on the install machine is already using port 8080 (for example, if you have another instance of Tomcat on the machine), then change the default startup port by modifying the `conf/server.xml` file.
   (d) Run Tomcat with the `startup.sh` (for Linux) or `startup.bat` (for Windows) program file in the Tomcat `bin` directory.
   (e) Use a browser to check whether Tomcat is running on URL `http://localhost:8080`, where `8080` is the Tomcat port specified in `conf/server.xml`.

   If Tomcat is running properly and you specified the correct port, the browser displays the Tomcat homepage.

   For more detailed installation instructions, go to: *http://tomcat.apache.org/tomcat-6.0-doc/setup.html*

   For instructions on installing Tomcat as a Windows service, go to: *http://tomcat.apache.org/tomcat-6.0-doc/windows-service-howto.html*
2. Shut down Tomcat.
3. Copy the `clover-license.war` and `clover.war` files to the Tomcat `webapps` directory.
4. Create a `setenv` file and configure the Tomcat memory settings appropriate for your needs.

   For details on creating this file, see *Creating a setenv file on page 17*.
5. Restart Tomcat.

6. To verify that the Integrator Server license was successfully deployed, start a browser and enter the URL `http://localhost:8080/clover-license`, where `8080` is the Tomcat port specified in `conf/server.xml`.

   The browser should show this light-weight page:

   ```
   This webapp runs on context /clover-license, which is correct.
   ```

7. Start a browser, and enter the Integrator Server home page URL: `http://localhost:8080/clover`.

   The browser should return the Integrator Server home page.

8. To log in to the Integrator Server:

   (a) Click **Log in to Server Administration**.

   (b) At the Access Server GUI page, enter `clover` as both the **Username** and **Password** and click the **Login** button.

When the Integrator Server is running, you can access the *Reference Manual* from either the Integrator Server home page or via this URL:

```
http://localhost:8080/clover/docs/index.html
```

where `8080` is the Tomcat port specified in `conf/server.xml`.

The *Reference Manual* provides comprehensive documentation on the operations available in Integrator Server.

Tomcat configuration recommendations

This topic provides some configuration tips for getting started with your Tomcat application server.

Keep in mind that complete documentation on configuring and using the Tomcat server is available at this URL for the server:

```
http://localhost:8080/docs/
```

This topic provides some configuration tips that are specific to the Integrator Server application.

Creating a setenv file

It is recommended that you create a file named `setenv.bat` (Windows) or `setenv.sh` (Linux) and place it in the Tomcat `bin` directory. With this file (which is run by the `catalina.bat` and `catalina.sh` scripts), you can change the following Tomcat environment settings with the `JAVA_OPTS` variable:

- You can set the minimum and maximum memory heap size with the JVM `-Xms` and `-Xmx` parameters. The best limits depend on many conditions, such as transformations which Information Discovery Integrator should execute. For these types of transformations, a maximum of 1 GB is recommended. For example, to set the minimum heap size to 128 MB and the maximum heap size to 1024 MB, use:

  ```
  JAVA_OPTS="-Xms128m -Xmx1024m"
  ```

- You should set the maximum limit of the PermGen (Permanent Generation) memory space to a size larger than the default. The default of 64 MB is not enough for enterprise applications. A suitable memory limit depends on various criteria, but 256 MB would make a good choice in most cases. If the PermGen space maximum is too low, `OutOfMemoryError: PermGen space` errors may occur. You can set the PermGen maximum limit with the following JVM parameter:

```
-XX:MaxPermSize=256m
```

- For performance reasons, it is recommended that the application is run in Server mode. Apache Tomcat does not run in Server mode by default. You can set the Server mode by using the JVM `-server` parameter. You can set the JVM parameter in the `JAVA_OPTS` variable in the environment variable in the setenv file.

The following is an example of a `setenv.bat` file:

```
set "JAVA_OPTS=%JAVA_OPTS% -Xms128m -Xmx1024m -XX:MaxPermSize=256m -server"
```

About database connections

The default installation (without changes to the configuration) uses the embedded Apache Derby DB, and therefore does not need an extra database server. In addition, the installation (as regards to the DB) does not need any subsequent configuration.

Integrator Server configures itself during the first startup. Database tables and some necessary records are automatically created on the first startup with an empty database. In the **Sandboxes** tab of the Integrator Server Administration GUI, you can check that there is one default sandbox created with one test graph.

Only one Integrator Server instance can work with the embedded DB. If you need more instances, you should configure an external DB.

You will have to determine if the embedded Apache Derby DB is sufficient for your application when you are ready for production deployment. If the Apache Derby DB is not sufficient, you can configure an external DB connection such as Microsoft SQL or Oracle. For details on configuring external DBs, see the *Reference Manual* for the Integrator Server.

Installing Integrator Server on WebLogic

Oracle Endeca Integrator Server is supported on WebLogic 10.3.6.

- WebLogic 10.3.6 must be installed and a domain must be configured. Consult the WebLogic documentation (*http://www.oracle.com/technetwork/middleware/fusion-middleware/overview/index.html*) for details.
- JDK 1.6.x or higher is required.
- Default JVM memory settings are not high enough to run the Web Logic server when Oracle Endeca Integrator Server is installed. Required memory settings depend on a variety of factors, including the number and complexity of transformations performed during processing. Exceeding the specified memory settings results in failed processing and reports of OutOfMemory errors in server logs and console output. To modify maximum memory, add USER_MEM_ARGS environment variable to the domain start scripts: (Linux) or (Windows) files:

  In Linux, add the following code to startWebLogic.sh (modify the memory sizes to meet your needs):

  ```
  export USER_MEM_ARGS="-Xms128m -Xmx2048m -XX:MaxPermSize=512m"
  ```

  In Windows, add the following code to startWebLogic.cmd (modify the memory sizes to meet your needs):

  ```
  set USER_MEM_ARGS=-Xms128m -Xmx2048m -XX:MaxPermSize=512m
  ```

  Important: The minimum recommended permgen space for Web Logic when running Integrator Server is 512 MB.

Note: You must restart the domain after modifying memory sizes.

- Change HTTP Basic Authentication configuration to allow the Oracle Endeca Integrator Server to authenticate users. Add the following XML element to the configuration file (`[domainHome}/config/config.xml`) just before the end tag of the `<security-configuration>` element:

  ```
  <enforce-validbasic-auth-credentials>false</enforce-valid-basic-auth-credentials>
  ```

To install Oracle Endeca Integrator Server on Weblogic:

1. Download the Oracle Endeca Integrator Server package for your operating system from the *Oracle Software Delivery Cloud*.
2. Unzip the package to a convenient location on the machine that runs WebLogic.
3. Deploy the Oracle Endeca Integrator Server WAR file (clover.war) using the WebLogic Server Administrator Console. For details, see the Oracle Fusion Middleware Administrator's Guide: *http://docs.oracle.com/cd/E23943_01/core.1111/e10105/toc.htm*
4. Deploy the Oracle Endeca Integrator Server license WAR file (clover-license.war) using the WebLogic Server Administrator Console.
5. WebLogic starts web applications immediately after deployment. To validate that the Oracle Endeca Integrator Server is functioning correctly:

   (a) start a browser and enter the URL `http://localhost:7001/clover` where `localhost` is the name or URL where WebLogic is installed and `7001` is the WebLogic port.

   The browser displays Oracle Endeca Integrator Server Welcome page.

   (b) To log in to the server, click the **Log in to Server Administration** button and on the login page enter `clover` as the **Username** and `clover` as the **Password**.

   (c) If the administration page displays a message that the license has expired, the license was not deployed correctly. Redeploy the license.

Configure the Oracle Endeca Integrator Server. For details, see "Configuration of CloverETL Server on WebLogic" in the *Oracle Endeca Information Discovery Integrator Server Guide*.

Part III

Installing Studio

Chapter 5

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. You may also need to update the default data source.

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 23*.
- Studio Tomcat bundle for Linux. This is based on Tomcat 6 and Java 1.6. See *Installing Studio Using the Tomcat Bundle for Linux on page 30*.
- Studio as a standalone application on a Tomcat application server. See *Installing Studio on Tomcat 6.0 on page 33*.
- Studio as a standalone application on WebSphere Application Server 7. WebSphere can be used on both Linux and Windows. *Installing Studio on WebSphere Application Server Version 7 on page 38*.
- Studio as a standalone application on Oracle WebLogic Server 11g. WebLogic can be used on both Linux and Windows. *Installing Studio on Oracle WebLogic Server 11gR1 (10.3.5) on page 47*.

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 59*.

Briefly, to deploy an alternate database, you modify the `portal-ext.properties` file to specify the appropriate JDBC connection information for the desired database.

Note on maintaining the Endeca Theme

To start up, Studio requires the Endeca Theme. Even if you do not intend to use the Endeca Theme in production, you should not uninstall the Endeca Theme (`endeca-theme-<version>.war`) from the `endeca-portal\deploy` directory.

Note on configuring the default data source

The data sources in your Studio application must always include a default data source. This data source is automatically assigned to all data-source-backed components when they are initially added to a page.

When you install a new instance of Studio, it creates a placeholder `default` data source. For details on configuring this data source to point to your server, see *Updating the default data source to point to your Endeca Server on page 55*.

Chapter 6

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

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

Installing the Studio Tomcat bundle requires the following files from the Oracle Endeca Information Discovery download for Windows (`EID24_Win.zip`):

- `EID_2.4.`*`x`*`_Studio_portal.zip`
- `EID_2.4.`*`x`*`_Studio_components.zip`

To install the Studio Tomcat bundle:

1. Unzip `EID_2.4.`*`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\2.4.0\` to extract the file into.

2. Extract the `.war` files from `EID_2.4.`*`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 themes, hooks, and layouts required by the portal. It is safe to overwrite these files with the versions in `EID_2.4.`*`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-6.0.35`, and set `JAVA_HOME` to `C:\<`*`path_to_endeca-portal`*`>\tomcat-6.0.35\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-6.0.35\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.

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

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

For better security, after logging in for the first time, you should probably either:

- Change the password for this user
- Create new administrative users for your instance, and then remove this user

6. 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 *Oracle Endeca Information Discovery Studio User's Guide*.

Changing the context root for the Windows Tomcat bundle

Optionally, after installing the Windows Tomcat bundle, you can change the context root.

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

Note that any change to the context root only affects the Studio application. Components, themes, and other hooks still use their default context paths.

To change the context root:

1. Stop the server.
2. Rename `endeca-portal\tomcat-6.0.35\conf\Catalina\localhost\ROOT.xml` file to `<`*`context root`*`>.xml`.

   For example, if your 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. Modify the XML file created in the previous step as needed:
   - For a root context: `<Context path="" />`
   - For a context of `/sales`: `<Context path="/sales"/>`
   - For a context of `/sales/east`: `<Context path="/sales/east"/>`
4. Rename the `endeca-portal\tomcat-6.0.35\webapps\ROOT` directory to `endeca-portal\tomcat-6.0.35\webapps\<`*`context root`*`>`.

For multi-level context paths, use a multi-level path such as: `endeca-portal\tomcat-6.0.35\webapps\sales#east`.

5. Edit the `endeca-portal\portal-ext.properties` file.

   Find the `portal.ctx` property at the beginning of the file, and change the value to be the same context root value you used above.

   Do not include a trailing slash in the `portal.ctx` value. For example, use this value:

   ```
   portal.ctx=/mycompany/portal
   ```

   Do not use this value:

   ```
   portal.ctx=/mycompany/portal/
   ```

6. Restart the server.

Running Studio as a Windows service

If you have installed the Windows Tomcat bundle, 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.

The Studio bundle does not include the Tomcat service installer files. You will need to obtain those 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.

You also should install the Tomcat service monitor. The monitor is used to configure and monitor the Windows service, and is useful for troubleshooting. The service monitor executable also is available from the Tomcat download.

Obtaining the service installer files

The service installer and monitor files are part of the Tomcat download.

You must use the files for Tomcat version 6.0.35.

To obtain the files and add them to Studio:

1. From the Apache Tomcat website (*http://tomcat.apache.org*), download the Tomcat file `apache-tomcat-6.0.35-windows-x86.zip` .

   A sample URL for the archive directory is: *http://archive.apache.org/dist/tomcat/tomcat-6/v6.0.35/bin/*

If you are not using the bundled JVM, and your JVM is 64-bit, then you must download the 64-bit version of the zip file (`apache-tomcat-6.0.35-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-6.0.35\bin`

Configuring the service

In the `service.bat` file, you need to configure the service name, description, and memory allocation. The Tomcat monitor file name also must be updated to reflect the change to the service name.

After you copy the Tomcat service installation files, before you can start the service, you need to update `service.bat` 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=Studio24
   set PR_DISPLAYNAME=Studio 2.4
   ```

4. Next, find the following line:

   ```
   set PR_DESCRIPTION=Apache Tomcat 6.0.35 Server -
   http://tomcat.apache.org/
   ```

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

   ```
   set PR_DESCRIPTION=Studio server, version 2.4
   ```

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=Studio24` in `service.bat`, then you must rename the copy of the `tomcat6w.exe` to `Studio24w.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.

Before you install the service, make sure that you have updated the configuration.

Also, 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-6.0.35\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-6.0.35\jre1.6.0_21\win\bin\client\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-6.0.35\logs\jakarta_service*.log`) for the relevant messages.

If the service will not start:

1. Make sure that you have used the correct version of the Tomcat download:
   - Tomcat version 6.0.35
   - For a 32-bit JVM (including the bundled JVM), `apache-tomcat-6.0.35-windows-x86.zip`
   - For a 64-bit JVM, `apache-tomcat-6.0.35-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-6.0.35\jre1.6.0_21\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 7

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.

Installing the Studio Linux Tomcat bundle requires the following files from the Oracle Endeca Information Discovery for Linux download (EID24_Linux.zip):

- EID_2.4.*x*_Studio_portal.tgz
- EID_2.4.*x*_Studio_components.zip

To install the Studio Linux Tomcat bundle:

1. Extract EID_2.4.*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/2.4.*x*/ to extract the file into.

2. Extract the .war files from EID_2.4.*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 themes, hooks, and layouts required by the portal. It is safe to overwrite these files with the versions in EID_2.4.*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-6.0.35/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/) in your browser. Log in using the following default credentials:

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

For better security, after logging in for the first time, you should probably either:

- Change the password for this user
- Create new administrative users for your instance, and then remove this user

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 *Oracle Endeca Information Discovery Studio User's Guide*.

Changing the context root in the Linux Tomcat bundle

Optionally, you can change the context root used by your Studio application.

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

Note that any change to the context root only affects the Studio application. Components, themes, and other hooks still use their default context paths.

To change the context root:

1. Stop the server.
2. Rename `endeca-portal/tomcat-6.0.35/conf/Catalina/localhost/ROOT.xml` file to `<`*`context root`*`>.xml`.

   For example, if your 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. Modify the XML file created in the previous step as needed:
   - For a root context: `<Context path="" />`
   - For a context of `/sales`: `<Context path="/sales"/>`
   - For a context of `/sales/east`: `<Context path="/sales/east"/>`
4. Rename the `endeca-portal/tomcat-6.0.35/webapps/ROOT` directory to `endeca-portal/tomcat-6.0.35/webapps/<`*`context root`*`>`.

   For multi-level context paths, use a multi-level path such as: `endeca-portal/tomcat-6.0.35/webapps/sales#east`.

5. Edit the `endeca-portal/portal-ext.properties` file.

   Find the `portal.ctx` property at the beginning of `portal-ext.properties`.

   Change the value of this setting to be the same context root value you used above.

Do not include a trailing slash in the `portal.ctx` value. For example, use this value:

```
portal.ctx=/sales/east
```

Do not use this value:

```
portal.ctx=/sales/east/
```

6. Restart the server.

Chapter 8

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

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.

Installing Studio on Tomcat 6.0 requires the following files from the Oracle Endeca Information Discovery download for Windows or Linux (EID24_Win.zip or EID24_Linux.zip).

- `EID_2.4.x_Studio_portal_dependencies.zip`
- `EID_2.4.x_Studio_portal_war.zip`
- `EID_2.4.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_2.4.`*`x`*`_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`
   - `cxf-2.2.8.jar`
   - `cxf-orawsdl-adapter.jar`
   - `endeca-images.jar`
   - `endeca-portal.jar`
   - `hsql.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`
- `mysql-connector-java-commercial-5.1.17-bin.jar`
- `ojdbc6.jar`
- `oracle.http_client_11.1.1.jar`
- `orawsdl.jar`
- `portal-kernel.jar`
- `portal-service.jar`
- `portlet-container.jar`
- `portlet.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 adding the `JAVA_OPTS` (for Linux) or `set JAVA_OPTS` (for Windows) line.

   This line should be added under the line `Execute The Requested Command` as follows (for Windows, the line needs to start with `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
   ```

   Adding this line increases the memory size for the server and establishes security configuration for Studio.

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 root context, create a new file called `ROOT.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`.

4. Modify the XML file created in the previous step as needed:
   - For a root context: `<Context path="" crossContext="true" />`
   - For a context of `/sales`: `<Context path="/sales" crossContext="true" />`
   - For a context of `/sales/east`: `<Context path="/sales/east" 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. Delete the contents of the `endeca-portal/apache-tomcat-<version>/webapps/ROOT` directory.

   This directory contains the standard Web application that is installed with Tomcat by default. We will replace this standard web application with Studio in the next step.
2. Unzip `EID_2.4.x_Studio_portal_war.zip` into a temporary directory.

   This zip file contains the Studio .war file and the `copyright.txt` file.
3. Read the `copyright.txt` file and then save it to the location of your choice.
4. Unzip the contents of the .war file into `endeca-portal/apache-tomcat-<version>/webapps/ROOT`.
5. If using a non-root context for your deployment, rename the `endeca-portal\apache-tomcat-<version>\webapps\ROOT` directory to `endeca-portal\apache-tomcat-<version>\webapps\<context root>`.

   For multi-level context paths, use the multi-level path.

   For example: `endeca-portal\apache-tomcat-<version>\webapps\ROOT` to `endeca-portal\apache-tomcat-<version>\webapps\mycompany\sales`.
6. Copy the `portal-ext.properties` file from the temporary directory you created for the `EID_2.4.x_Studio_portal_dependencies.zip` to the `endeca-portal` directory.
7. Edit `endeca-portal\portal-ext.properties`.

   Find the `portal.ctx` property at the beginning of `portal-ext.properties`.

Change the value of this setting to be the same context root value you set earlier. However, do not include a trailing slash in the `portal.ctx` value.

For example, use this value: `portal.ctx=/mycompany/portal`

Do not use this value: `portal.ctx=/mycompany/portal/`

8. Under the `endeca-portal` directory, create a `data` directory, and then create an `endeca-data-sources` directory below that.
9. In the `endeca-portal` directory, create a `deploy` directory.
10. Extract the .war files from `EID_2.4.`*`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.
11. 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.
12. To test that the application is running, go to the portal (`http://localhost:8080/`) in your browser. Log in using the following default credentials:

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

For better security, after logging in for the first time, you should probably either:

- Change the password for this user
- Create new administrative users for your instance, and then remove this user

13. 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 *Oracle Endeca Information Discovery Studio User's Guide*.

Chapter 9

Installing Studio on WebSphere Application Server Version 7

You can deploy Studio as a standalone application on WebSphere Application Server (WAS) version 7.0.

Note: The examples in this section are based on a Linux server WAS deployment. If you are installing on Windows, the steps will be similar, though you will need to substitute Windows executables and paths. In certain examples, backslashes are used to escape the dollar sign ($) character on Linux, because the shell would otherwise attempt a variable substitution for this character. These backslashes should not be required on a Windows system.

High-level overview of WebSphere Application Server 7 deployment

Here is an overview of the steps for deploying Studio on WAS 7. This overview is followed by the details for each step.

Installing Studio on WAS 7 requires the following files from the Oracle Endeca Information Discovery download for Windows or Linux (`EID24_Win.zip` or `EID24_Linux.zip`).

- `EID_2.4.`*`x`*`_Studio_portal_dependencies.zip`
- `EID_2.4.`*`x`*`_Studio_portal_war.zip`
- `EID_2.4.`*`x`*`_Studio_components.zip`

Here is an overview of each step in the process of deploying Studio on WAS 7. This overview is followed by specific instructions and details for each step:

1. Add additional JVM properties to your WAS 7 installation.

2. Deploy the dependency `.jar` files.

   For the exact list of required files and destination directories, see the specific instructions for this step.
3. Start (or restart) the WAS server.
4. Install the Studio `.war` file as an enterprise application.
5. Edit and deploy `portal-ext.properties`.
6. Install the Endeca theme, components, and other framework `.war` files.
7. Start the Studio enterprise application.
8. Optionally, repeat step 6 for any additional plugins you want to add.

Adding JVM properties to your WAS 7 installation

Studio requires you to add additional JVM properties to the WAS 7 installation.

To use the WAS 7 **Integrated Solutions Console** to add these properties.

1. Log in to the Integrated Solutions Console.
2. In the left menu, click **Servers**, then **Server Types**.
3. Under **Server Types**, click **WebSphere application servers**.
4. In the list of servers, click the server instance (for example, server1).

   The details page for the server is displayed.
5. On the **Configuration** tab, in the menu at the right, under **Server Infrastructure**, expand the **Java and Process Management** option, then click **Process definition**.
6. On the **Configuration** tab of the **Process definition** page, under **Additional Properties**, click **Java Virtual Machine**.
7. On the **Configuration** tab of the **Java Virtual Machine** page:
   (a) In the **Initial heap size** field, type `256`.

   The value is in MB.

   (b) In the **Maximum heap size** field, type `1024`.

   (c) In the **Generic JVM arguments** field, enter the following arguments, separated by spaces:

   ```
   -XX:MaxPermSize=256m -Dfile.encoding=UTF8 -Duser.timezone=GMT
   ```
8. Click **OK**.
9. Save the changes to the master configuration.
10. Restart WAS.

Deploying Studio dependency libraries on WAS 7

For WAS 7, Studio requires the deployment of several Java libraries.

These libraries are deployed to a global class loader, making them available to multiple applications.

To deploy the Studio dependency libraries:

1. Unzip the `.jar` files found in `EID_2.4.x_Studio_portal-dependencies.zip`.
2. Upload the following `.jar` files from the `.zip` file to the WAS server's external library directory.

   (For example, if WAS is installed in `/usr/local/WAS/AppServer`, you would deploy the selected `.jar` files into `/usr/local/WAS/AppServer/lib/ext/`.)

   ```
   annotations.jar
   commons-lang.jar
   container.jar
   cxf-2.2.8.jar
   cxf-orawsdl-adapter.jar
   dom4j.jar
   endeca-images.jar
   endeca-portal.jar
   hibernate3.jar
   hsql.jar
   jabsorb.jar
   jackson-core-lgpl-1.7.2.jar
   jackson-mapper-lgpl-1.7.2.jar
   log4j.jar
   mdex_bindings.jar
   oracle.http_client_11.1.1.jar
   orawsdl.jar
   portal-kernel.jar
   portal-service.jar
   portlet.jar
   portlet-container.jar
   saw-api.jar
   slf4j-api.jar
   slf4j-log4j12.jar
   XmlSchema-1.4.5.jar
   ```
3. Restart the WAS server so that it can pick up the newly available `.jar` files.

Extracting the standalone portal WAR on WAS 7

Before you can install the standalone portal WAR, you must extract it from its download package.

To extract the standalone portal WAR on WAS 7:

1. Unzip `EID_2.4.x_Studio_portal_war.zip` into a temporary directory.

   This zip file contains the `endeca-portal-<version>.war` file and the `copyright.txt` file.
2. Read the `copyright.txt` file, then save it to the location of your choice.
3. Note the location of the `endeca-portal-<version>.war` file, as you will need it for the next step.

Deploying the standalone portal WAR on WAS 7

After downloading and extracting the necessary files, you can deploy Studio as an enterprise application in WebSphere Application Server, and then install portlets, themes, and other plugins as modules in that enterprise application.

The following steps document the installation procedure by using the IBM Integrated Solutions Console for a WebSphere Application Server installed and maintained without the use of the Deployment Manager, and consisting of one cell with one node and one server.

The instructions may need to be adjusted for clustered environments, environments maintained with the Deployment Manager, or for environments where administration is performed using tools such as `wsadmin`, rather than the Integrated Solutions Console.

The following steps assume that no other applications are deployed in the same application server. If there are other applications, ensure that no applications are bound to context root `/` (or that any such applications are stopped during the Studio deployment).

After following these steps, you can adjust the context root for the Studio application, to ensure it does not conflict with other applications.

To deploy the Studio standalone portal WAR on WAS 7:

1. Start the WAS server.
2. Log in to the WAS Integrated Solutions Console, using the appropriate administrator credentials.
3. In the WAS Integrated Solutions Console, select **Applications>New Application>New Enterprise Application**.
4. Click to browse to and select the Studio WAR you downloaded earlier (`endeca-portal-<`*`version`*`>.war`), and then click **Next**.
5. Select **Choose to generate default bindings and mappings**. Check the following options:
    - **Generate default bindings**
    - **Override existing bindings**
6. Still in the **Choose to generate default bindings and mappings** section, check **Use default virtual host name for Web and SIP modules**, and then enter `default_host` in the text field.

    Click **Next**.
7. By default, the application name is **endeca-portal-<*version*>_war**. Set the application name to a more relevant name (for example, `StudioDevInstance`). All other installation options can remain unchanged. Click **Next**.

    Note: Do not use spaces in the application name. For example, use **StudioDevInstance** instead of **Studio Dev Instance**.
8. In **Map modules to servers**, accept the default settings, and then click **Next**.
9. In **Map context roots for Web modules**, set the context root to the desired path for your Studio installation, and then click **Next**.

    Note: Make a note of your context root, as you will need to reference it several times during the deployment process.
10. In **Install New Application**, confirm that your settings are correct, and then click **Finish**.
11. Wait for installation. If the installation is successful, click **Save directly to master configuration**.

Creating the Liferay Home directory on WAS 7

The remaining instructions for the WAS 7 installation refer to a directory called Liferay Home. The Liferay Home directory is created relative to the user's home directory.

Manually create a Liferay Home directory (`/home/endeca/liferay/`), along with the following subdirectories:

- `/home/endeca/liferay/data`
- `/home/endeca/liferay/data/endeca-data-sources`
- `/home/endeca/liferay/websphere-deploy`

Configuring and deploying portal-ext.properties for WAS 7

After you edit the default version of the `portal-ext.properties` file, you then deploy it in WAS.

To deploy the file, you can either:

- Update the application to include the `portal-ext.properties` file.
- Upload the `portal-ext.properties` file to the Liferay Home directory on the server.

Editing portal-ext.properties for WAS 7 deployment

Updating the application to include the portal-ext.properties file on WAS 7

Uploading portal-ext.properties to Liferay Home on the server on WAS 7

Editing portal-ext.properties for WAS 7 deployment

Studio provides a default version of `portal-ext.properties` to use as a starting point. Before you can deploy the file, you must add and edit settings to reflect your WAS deployment.

The default version of `portal-ext.properties` is included in the package `EID_2.4.`*`x`*`_Studio_portal_dependencies.zip`.

To update the file:

1. Open the `portal-ext.properties` file.
2. Add the following lines to the end of the file:

   ```
   # Specify a directory where Liferay will "deploy" processed plugins.
   # From this directory, WAS users will deploy WARs as modules in the
   # Information Discovery Studio enterprise application.
   #
   auto.deploy.dest.dir=${liferay.home}/websphere-deploy
   #
   # Set this to true to enable JMX integration in
   # com.liferay.portal.cache.EhcachePortalCacheManager.
   #
   ehcache.portal.cache.manager.jmx.enabled=false
   ```

   The destination directory (specified by the `auto.deploy.dest.dir` setting) must exist before the plugin is hot-deployed.

In the above example, you must manually create the `websphere-deploy` directory if it does not exist.

3. Find the `portal.ctx` property at the beginning of `portal-ext.properties`.

   Change the value of this setting to be the same context root value you used when deploying the standalone portal WAR. However, do not include a trailing slash in the `portal.ctx` value.

   For example, use this value:

   ```
   portal.ctx=/mycompany/portal
   ```

   Do not use this value:

   ```
   portal.ctx=/mycompany/portal/
   ```

4. Save the file.

Updating the application to include the portal-ext.properties file on WAS 7

After you create the `portal-ext.properties` file, you can use the IBM Integrated Solutions Console to include it in the `portal.war` module.

These steps may be performed using the `wsadmin` tool instead of the Integrated Solutions Console, and may need to be adjusted for alternate WAS configurations.

Note: Whenever users make changes to `portal-ext.properties`, they will need to repeat these steps to update the file in the application.

In some environments, it may therefore be more appropriate to deploy `portal-ext.properties` to the Liferay Home directory, where it can be updated without updating the deployed application. See *Uploading portal-ext.properties to Liferay Home on the server on WAS 7 on page 43*.

To deploy `portal-ext.properties` in the Integrated Solutions Console:

1. Go to **Applications>Application Types>WebSphere Enterprise Applications**.
2. Select the enterprise application created when you deployed the portal WAR, then click **Update**.
3. Select **Replace or add a single file**.
4. Specify the path to deploy the file into the `WEB-INF/classes` directory of the portal Web application.

   For example: `portal-<`*`version`*`>.war/WEB-INF/classes/portal-ext.properties`
5. Browse to where you created the file on your computer.
6. When the file has successfully updated, click **Save directly to master configuration**.

Uploading portal-ext.properties to Liferay Home on the server on WAS 7

Another option for deploying the updated `portal-ext.properties` file is to manually upload it to WAS.

To manually upload the `portal-ext.properties` file:

1. Upload the `portal-ext.properties` file to the Liferay Home directory.

For example: `/home/endeca/liferay/portal-ext.properties`.

2. When the Studio application is started, these properties are read.

Starting the Studio application on WAS 7

Once the Studio application has been deployed, and the `portal-ext.properties` file has been configured and deployed, the application needs to be started.

The following steps describe this process in the IBM Integrated Solutions Console.

To start the application:

1. Go to **Applications>Application Types >WebSphere Enterprise Applications**.
2. Select the enterprise application created when you deployed the portal WAR.
3. If the application is not already running, click **Start** to start it.

Deploying components and other plugins in WAS 7

Next, you need to deploy components, themes, hooks, and other plugins in WAS 7.

These plugins are located in `EID_2.4.`*`x`*`_Studio_components.zip`.

About component pre-processing in WAS 7

Deploying components in WAS 7

Deploying generated .war files on WAS 7 with the Integrated Solutions Console

Using wsadmin to deploy the generated .war file on WAS 7

About component pre-processing in WAS 7

WAS does not support the hot deployment of components. However, the Studio deployment code must update plugins by adding necessary libraries and configuration files.

For example, the portlet deployment code adds the following important piece of configuration to a portlet component's `web.xml` file:

```
<context-param>
  <param-name>com.ibm.websphere.portletcontainer.PortletDeploymentEnabled</param-name>
  <param-value>false</param-value>
</context-param>
```

This context parameter is important for WAS deployment, as it ensures that WAS's portal server does not attempt to load the new portlet, and instead allows Studio to load the newly deployed portlet.

For this reason, components must be pre-processed before they are deployed to WAS. You upload your `.war` files to the Studio `deploy` directory so that the deployer can find and process them.

Deploying components in WAS 7

Before you can deploy the Studio components in WAS 7, they must first be pre-processed.

Important: To start up, Studio requires the Endeca Theme. Even if you do not intend to use the Endeca Theme in production, you should deploy the Endeca Theme (`endeca-theme-<version>.war`).

To deploy Studio components in WAS 7:

1. Copy all component `.war` files from `EID_2.4.x_Studio_components.zip` to `${liferay.home}/deploy`.
2. Wait while Liferay pre-processes the .war files and places them in the `${liferay.home}/websphere-deploy` directory.
3. Deploy the `.war` files generated in the previous step as modules in the Studio enterprise application. To do this, you can either use:
   - The WebSphere Integrated Solutions Console.
   - The command line, using `wsadmin`.

Deploying generated .war files on WAS 7 with the Integrated Solutions Console

You can use the IBM Integrated Solutions Console to deploy the `.war` files it finds in the `websphere-deploy` directory.

Note: These steps may need to be adjusted for alternate WAS configurations.

To deploy a generated `.war` file with the Integrated Solutions Console:

1. Go to **Applications>Application Types>WebSphere Enterprise Applications**.
2. Select the enterprise application created when you deployed the portal `.war` file, then click **Update**.
3. Select **Replace or add a single module**.
4. Specify the path to deploy the file as the display name of the new module.

   For example, if you are adding `endeca-navigation-portlet.war`, specify the path as `endeca-navigation-portlet`.
5. Browse the remote file system to the newly created `.war` file in the deploy output directory.

   Continuing the example above, this might be `/home/endeca/liferay/websphere-deploy/endeca-navigation-portlet.war`.
6. After finding the file, click **Ok**.
7. Select the detailed install path. Keep the defaults on all screens except the context root.

   Set the context root to match the display name of the new plugin (in this example, `/endeca-navigation-portlet/`).
8. Once it has successfully updated, click **Save directly to master configuration**.

Using wsadmin to deploy the generated .war file on WAS 7

You can also use the `wsadmin` tool to deploy the generated `.war` file from the command line.

Note: These steps may need to be adjusted for alternate WAS configurations.

In the `wsadmin` tool, enter a command similar to the example below, where the command is executed from the deploy output directory (that is, the directory containing the `endeca-navigation-portlet.war` file):

```
[WAS]/AppServer/bin/wsadmin.sh -c "\$AdminApp update Studio
modulefile {-operation addupdate -contents endeca-navigation-port¬
let.war -contextroot /endeca-navigation-portlet/ -contenturi endeca-navi¬
gation-portlet -usedefaultbindings}" -c "\$AdminConfig save"
```

In this example:

- The enterprise application is named `Studio`.
- The file name for the module being added is `endeca-navigation-portlet.war`.
- The module display name is `endeca-navigation-portlet`
- The command is executed in `/home/endeca/liferay/websphere-deploy/` .

Troubleshooting WAS 7 deployment

When deploying Studio on WAS 7, keep the following in mind.

Updating the Studio .war file

If you need to update the Studio `.war` file (not any individual plugin, but the portal `.war` itself), you must restart the WAS server. If you only restart the module, the restart might not be successful.

Chapter 10

Installing Studio on Oracle WebLogic Server 11gR1 (10.3.5)

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

Installing Studio on WebLogic requires the file `EID_2.4.x_Studio_portal_weblogic.zip` file from the Oracle Endeca Information Discovery download for Windows or Linux (`EID24_Win.zip` or `EID24_Linux.zip`).

Prerequisites for installing Studio on WebLogic

Before you can install Studio for WebLogic, the following items must already be in place.

You must have:

- Installed Oracle WebLogic Server 11gR1 (10.3.5)
- Installed Sun Java 6
- Created a WebLogic domain

Also make sure you are logged in as an administrative user.

For information on how to install WebLogic and Java, see the Oracle documentation.

Updating WebLogic domain settings in the setDomainEnv script file

Before you can deploy Studio, you need to update some WebLogic settings.

The relevant settings are in 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 (`<WebLogicInstallDirectory>/user_projects/domains/<DomainName>/bin/`) where:

- `<WebLogicInstallDirectory>` is the directory where you installed WebLogic.
- `<DomainName>` is the name of the domain where you are installing Studio.

In the file:

1. Add the `JAVA_OPTIONS` argument close to the top of the file.

   For `setDomainEnv.cmd` (Windows):

   ```
   set JAVA_OPTIONS=-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 `setDomainEnv.sh` (Linux):

   ```
   JAVA_OPTIONS="-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"
   export JAVA_OPTIONS
   ```

2. If you are using SSL-enabled data sources, then add the following to the `JAVA_OPTIONS` argument:

   ```
   -DUseSunHttpHandler=true
   ```

3. Set the max perm size setting to `"-XX:MaxPermSize=512m"`.
4. Update the memory arguments setting to `"-Xms256m -Xmx1024m"`.

Creating and configuring the Liferay Home directory for WebLogic

For the WebLogic installation, the default location of the Liferay Home directory is `<`*`WebLogicInstallDirectory`*`>/user_projects/domains`, where `<`*`WebLogicInstallDirectory`*`>` is the directory where you installed WebLogic. You can, if needed, use a different directory as the Liferay Home directory.

To create and configure a different location for the Liferay Home directory:

1. Create the directory you want to be Liferay Home directory.
2. Next, you need to update the version of `portal-ext.properties` located in the `web-inf/classes` directory of the `endeca-portal-2.4.`*`x`*`.war` file, which is located in the .ear file, which is contained in the .zip file.

   This version of `portal-ext.properties` is only used to provide a pointer to the Liferay Home directory.
3. Add the `liferay.home` parameter to the top of the file.

   For example:

   ```
   liferay.home=/localdisk/liferay
   ```

4. Repackage the file into the .war file.

Configuring a non-root context for a WebLogic instance

By default, Studio uses the root context. If you are installing Studio on a non-root context, then before you deploy the Studio .ear file, you need to update the `application.xml` file to specify the context root.

Note that any change to the context root only affects the Studio application. Components, themes, and other hooks still use their default context paths.

To change the context root:

1. Open the `META-INF/application.xml` file in `endeca-portal-weblogic.ear`.
2. Find the following xml snippet:

   ```
   <module>
     <web>
       <web-uri>endeca-portal.war</web-uri>
       <context-root>/</context-root>
     </web>
   </module>
   ```

3. Update the value of the `context-root` element to your chosen context root path.

   The value should have a trailing slash. For example:

   ```
   <module>
     <web>
       <web-uri>endeca-portal.war</web-uri>
       <context-root>/mycompany/portal/</context-root>
     </web>
   </module>
   ```

For a non-root context, you also must specify the context root in the `portal-ext.properties` that you install in the Liferay Home directory. *Deploying the Studio .ear file on page 49* includes instructions for installing and updating this version of `portal-ext.properties`.

Deploying the Studio .ear file

To install Studio, you deploy the .ear file into the WebLogic domain. For a development environment, you can use the autodeploy function. In a production environment, you must use the WebLogic Administration Console.

To deploy the Studio .ear file:

1. Extract the `portal-ext.properties` file from `EID_2.4.x_Studio_portal_weblogic.zip`, and place it in the Liferay Home directory.

   This version of the file is at the top level of the .zip file, and contains the complete set of Studio default settings.

2. If you have changed the Liferay Home directory from the default, then you must add the `liferay.home` parameter to the top of the file.

   For example:

   ```
   liferay.home=/localdisk/liferay
   ```

3. If you are using a non-root context, then you also need to update the `portal.ctx` property to use the same context root value you set in the `META-INF/application.xml` (See *Configuring a non-root context for a WebLogic instance on page 49*).

The property is at the beginning of the file.

When setting the `portal.ctx` value, do not include a trailing slash. For example, use this value:

```
portal.ctx=/mycompany/portal
```

Do not use this value:

```
portal.ctx=/mycompany/portal/
```

4. Create the `/data/endeca-data-sources` directory in the Liferay Home directory.

   If your data sources are connected to a secured Endeca Server, then this directory will contain the associated Java Keystore (JKS) files.

   For details on connecting data sources to a secured Endeca Server, see the *Oracle Endeca Information Discovery Studio User's Guide*.

5. Create the `deploy` directory in the Liferay Home directory.

6. Create the `weblogic-deploy` directory in the Liferay Home directory.

7. If WebLogic isn't already started, then to start it, run the WebLogic startup script. For Linux, the script name is `startWebLogic.sh`.

   The startup script is in the `bin` subdirectory of the domain directory.

8. Extract the .ear file from `EID_2.4.`*`x`*`_Studio_portal_weblogic.zip`, and then deploy the .ear file into the domain.

   For a development environment, you can use the autodeploy option to deploy the file. For example, for Linux:

   ```
   cp endeca-portal-weblogic-2.4.ear <DomainDirectory>/autodeploy
   ```

   Where <*`DomainDirectory`*> is the path to the domain directory.

   For a production environment, you need to use the WebLogic Administration Console to deploy the file:

   (a) Go to the **Administration Console** at <*`hostname`*>`:7001/console`.

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

   (c) Click **Install**.

   (d) Use the file browser to find the `endeca-portal-weblogic-2.4.ear` file, then click the radio button to the left of it.

   (e) Click **Next**.

   (f) Accept the default selection, then click **Next**.

   (g) Accept the default selections, then click **Finish**.

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

   Note that when you deploy the .ear file, you are deploying the complete set of standard Studio components. To remove components, or add custom components, you need to update the .ear file. For information on adding components to and removing components from the .ear file, see the *Oracle Endeca Information Discovery Studio Developer's Guide*.

9. To verify that Studio has deployed successfully:

   (a) Open a Web browser.

(b) In the browser, go to `http://<hostName>:<portNumber>`, where *<hostName>* is the name of the server, and *<portNumber>* is the number of the port used by WebLogic.

By default, WebLogic uses port 7001.

You should see the Studio welcome page.

Chapter 11

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

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 continuing 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.

Chapter 12

Getting Started with Studio

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

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:	`admin`

For better security, after logging in for the first time, you should probably either:

- Change the password for this user
- Create new administrative users for your instance, and then remove this user

Using the Control Panel to configure Studio

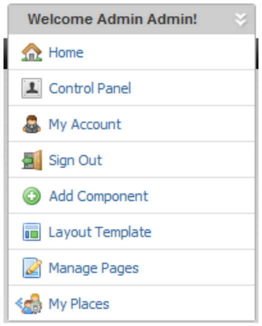
The **Control Panel** provides access to a wide range of functions, including configuring Studio settings, managing data used in Studio applications, managing Studio users, and monitoring performance.

For information on using the **Control Panel** to configure and monitor Studio, see the *Oracle Endeca Information Discovery Studio User's Guide*.

To display the **Control Panel**:

1. Click the Studio menu in the upper-right corner of the page.

   The Studio menu is labeled "Welcome *<user name>*!"

2. From the Studio menu, select **Control Panel**.

Updating the default data source to point to your Endeca Server

When you first install Studio, all components that require a backing data source are bound to a data source called `default`.

If no other data sources are present, then Studio creates a placeholder `default` data source.

Before you can start working with components in Studio, you must either:

- Use the **Data Sources** page to update this data source to point to the correct server, port, and data store name.
- Add a new data source to Studio, then use the **Framework Settings** page to configure that data source to be the default.

The **Data Sources** and **Framework Settings** pages are both available from the **Control Panel**.

To update the `default` data source:

1. In the Studio **Control Panel** menu, in the **Information Discovery** section, click **Data Sources**.
2. On the **Data Sources** page, click the **Edit** button for the `default` data source.
3. In the data source definition, edit the server, port, and data store name to reflect your Endeca Server.

   For example:

   ```
   {
   "server":"localhost",
   "port":"5555",
   "datastoreName":"acmeDB"
   }
   ```

4. Click **Save**.

For details on creating and managing data sources, see the *Oracle Endeca Information Discovery Studio User's Guide*.

Importing the Data Explorer and Schema Explorer sample pages

Studio provides a set of sample pages you can use as a starting point for working with components and viewing data and configuration settings.

One page, called **Data Explorer**, contains components bound to the `default` data source. This page allows you to explore the data in the `default` data source.

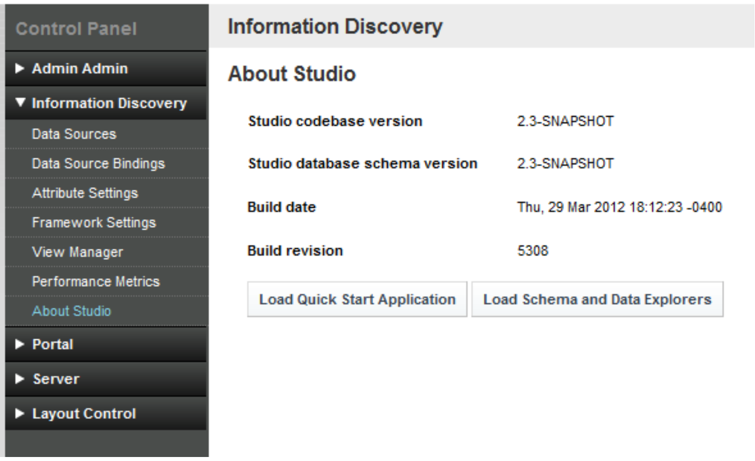
The other page, **Schema Explorer**, contains similar components. The components on the **Schema Explorer** page are bound to the `default-schema` data source, which only displays configuration records. This page allows you to explore the configuration settings for the `default` data source.

The **About Studio** page includes a button to import these pages into Studio. The pages are loaded into a new community called **explorers**. If needed, Studio also creates the data sources.

To import these sample pages into Studio:

1. From the Studio menu, select **Control Panel**.
2. On the **Control Panel** menu, under **Information Discovery**, click **About Studio**.

   The **About Studio** page is displayed.

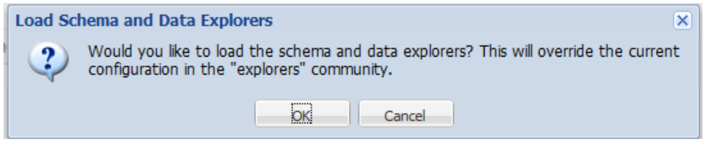


About Studio

Studio codebase version	2.3-SNAPSHOT
Studio database schema version	2.3-SNAPSHOT
Build date	Thu, 29 Mar 2012 18:12:23 -0400
Build revision	5308

3. On the **About Studio** page, click **Load Schema and Data Explorers**.

   Studio prompts you to confirm that you want to load the pages.

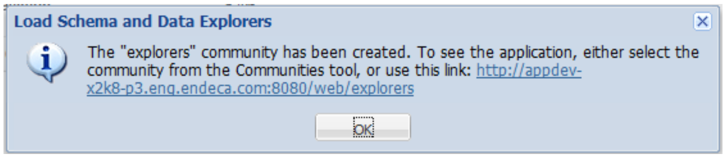


Load Schema and Data Explorers

Would you like to load the schema and data explorers? This will override the current configuration in the "explorers" community.

4. Click **OK**.

Studio displays a message confirming that it has created the new **explorers** community with the **Data Explorer** and **Schema Explorer** pages. The message dialog includes a link to the new community, or you can just click **OK** to close the dialog and continue.



Load Schema and Data Explorers

The "explorers" community has been created. To see the application, either select the community from the Communities tool, or use this link: http://appdev-x2k8-p3.eng.endeca.com:8080/web/explorers

If neither the default or default-schema data sources exist, then Studio creates them. In this case, both data sources contain placeholder values for the server, port, and data store name.

If the default data source exists, but default-schema does not exist, then Studio uses the server, port, and data store name values from default to create default-schema, and adds the configuration to only display configuration records.

5. If needed, update the default and default-schema data sources to point to your Endeca Server.

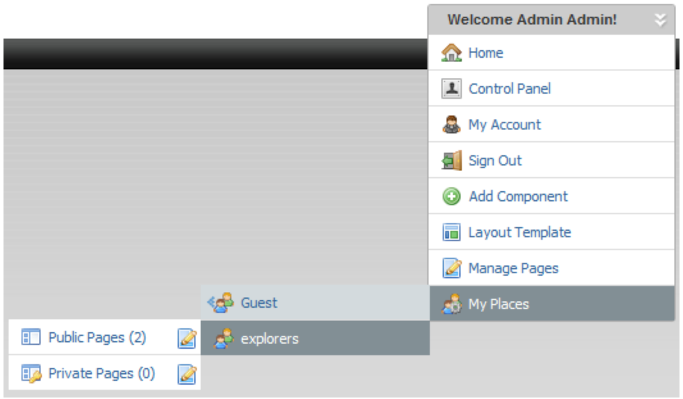
   See *Updating the default data source to point to your Endeca Server on page 55*. The same process would apply for updating default-schema.

   For default-schema, make sure you do not edit or remove the mdex-records parameter.

6. To navigate to the **explorers** community:

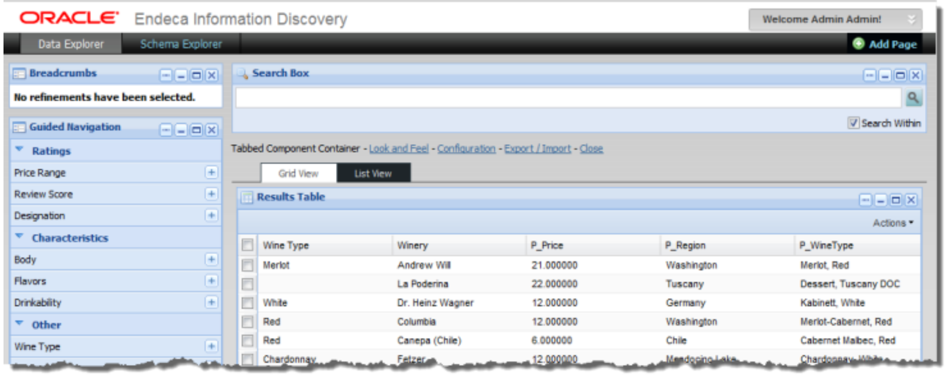
   (a) From the Studio menu, select **My Places**.

   (b) Click **explorers**.



   (c) Click **Public Pages**.

The **Data Explorer** and **Schema Explorer** pages are displayed.



No refinements have been selected.

☑ Search Within

	Wine Type	Winery	P_Price	P_Region	P_WineType
□	Merlot	Andrew Will	21.000000	Washington	Merlot, Red
□		La Poderina	22.000000	Tuscany	Dessert, Tuscany DOC
□	White	Dr. Heinz Wagner	12.000000	Germany	Kabinett, White
□	Red	Columbia	12.000000	Washington	Merlot-Cabernet, Red
□	Red	Canepa (Chile)	6.000000	Chile	Cabernet Malbec, Red
□	Chardonnay	Fetzer	12.000000	Mendocino Lake	Chardonnay, White

Exploring the Quick Start Sample Application in Studio

Endeca Information Discovery's Quick Start allows you to quickly and easily create a new Endeca Server, then explore a Sample Application created for that data.

For instructions on how to download, run, and explore the Quick Start project and the Sample Application, see the *Oracle Endeca Information Discovery Quick Start Guide*.

Chapter 13

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 preferences, user permissions, system settings, and more.

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.

When changing to a different database, keep the following details in mind:

- Studio ships with a `portal-ext.properties` file (in the portal distribution's root directory). You can modify this file instead of creating a new one.
- Studio has been tested on MySQL and Oracle 11g. Other databases are expected to work but have not been explicitly tested.
- 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.

To switch to a different database:

1. Install and verify that your database is working.
2. Create a new empty database or schema for Studio.
3. Create a database user for Studio.
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. Edit the `portal-ext.properties` file.

   In the JDBC section, comment out the settings for Hypersonic, and uncomment the settings for your database.
7. Edit the settings for your database of choice, adding the appropriate username and password and editing the JDBC connection string as necessary.
8. Start Studio. Monitor its logs to ensure for any error messages while connecting to the database and creating tables.
9. After tables have been created and you have validated that Studio is running, you may remove the user's alter table privileges.

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

Chapter 14

Creating a Studio Cluster

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

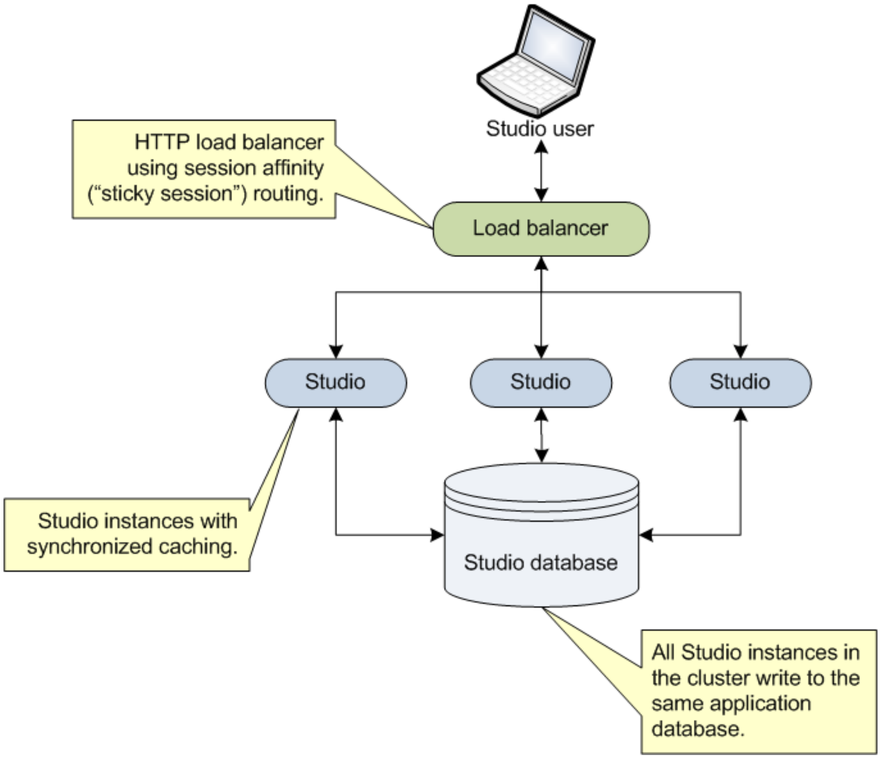
About Studio clustering

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.


Studio user
HTTP load balancer using session affinity (“sticky session”) routing.
Load balancer
Studio
Studio
Studio
Studio instances with synchronized caching.
Studio database
All Studio instances in the cluster write to the same application database.

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 59*. 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
net.sf.ehcache.configurationResourceName	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. In the default portal.properties file, the configuration file is set to hibernate.xml, to implement caching in a non-clustered implementation. When you uncomment this property in portal-ext.properties, which changes the configuration file to hibernate-clustered.xml, then Hibernate synchronizes the cache with the other members of the cluster.
ehcache.multi.vm.config.location	The name and location of the XML configuration file for Ehcache. In the default portal.properties file, the file is set to liferay-multi-vm.xml, to implement caching in a non-clustered implementation. When you uncomment this property in portal-ext.properties, which changes the configuration file to liferay-multi-vm-clustered.xml, then the cache is synchronized with the other members of the cluster.
org.quartz.jobStore.isClustered	Enables clustering on the built-in Quartz job scheduling engine.

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 WebSphere, in `endeca-portal-`*`versionNumber`*`.war`
   - 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. To deploy the customized files in Websphere:

   (a) Log in to the Websphere Integrated Solutions Console.

   (b) In the menu, expand the **Applications** section.

   (c) Under **Application Types**, select **WebSphere enterprise applications**.

   (d) Check the checkbox next to `endeca-portal-`*`versionNumber`*`.war`, then click Update.

   (e) Select **Replace or add a single file**.

   (f) In the Specify the path of the file to be replaced or added field, enter:

   `endeca-portal-`*`versionNumber`*`.war/WEB-INF/classes/ehcache/hibernate-clustered-custom.xml`

   Where *versionNumber* is the actual version string of the endeca-portal WAR file.

   (g) Specify the path to `hibernate-clustered-custom.xml` either locally or remotely.

   (h) Click **Next**, then click **OK**.

   (i) After the file is deployed, save changes.

   (j) Repeat steps d through I for the `liferay-multi-vm-clustered-custom.xml` file.

(k) Restart WAS.

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

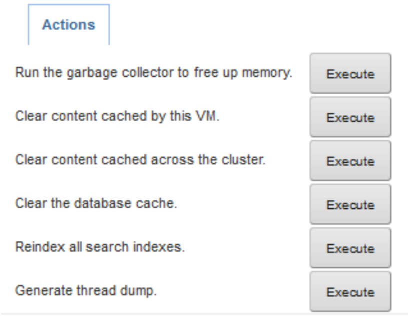
```
net.sf.ehcache.configurationResourceName=/ehcache/hibernate-clustered-custom.xml
ehcache.multi.vm.config.location=/ehcache/liferay-multi-vm-clustered-custom.xml
```

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 Studio 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:



Run the garbage collector to free up memory.	Execute
Clear content cached by this VM.	Execute
Clear content cached across the cluster.	Execute
Clear the database cache.	Execute
Reindex all search indexes.	Execute
Generate thread dump.	Execute

- 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 IV

Uninstallation Tasks

Chapter 15

Uninstalling Oracle Endeca Information Discovery

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

Uninstalling Integrator

This section contains the procedures for uninstalling the Integrator package.

Uninstalling Integrator on Windows

Follow these steps to uninstall Integrator from your Windows client machine.

Although the uninstall procedure does not delete the workspace folder used by the Integrator, it is a good practice to back up your Integrator projects before uninstalling.

To uninstall Integrator from your Windows machine:

1. Close the Integrator if you have it open.
2. Open **All Programs** in the Start menu.
3. Open the **Oracle Endeca Information Discovery 2.4.0** entry.
4. Click on **Uninstall Integrator**.

   The Uninstall wizard is launched with the **Uninstall** option selected.
5. In the Uninstall wizard, click **Next**.
6. In the Completing the Setup Wizard dialog, click **Next**.
7. When the uninstallation procedure is completed, click **Finish**.

Note that alternatively, you can uninstall Integrator from the Windows Control Panel by selecting **Programs> Uninstall a program** and then choosing **Oracle Endeca Information Discovery Integrator 2.4.0** from the list of installed software.

Uninstalling Integrator on Linux

Follow these steps to uninstall Integrator from your Linux client machine.

Although the uninstall procedure does not delete the workspace directory for Integrator, it is a good practice to back up your Integrator projects before uninstalling.

To uninstall Integrator from your Linux machine:

1. Change to the parent directory of the Integrator install directory.
   The default parent directory is `Oracle/Endeca/Discovery`.
2. Issue an `rm` command as in this example:
   ```
   rm -rf Integrator
   ```

As mentioned above, the Integrator workspace folder is not deleted after Step 2. If you do not delete the workspace directory, you can re-use it if you re-install Integrator.

Uninstalling Integrator Server

Uninstallation procedures are different on Tomcat and WebLogic.

Uninstalling Integrator Server from Tomcat

Follow these steps to uninstall Integrator Server from Tomcat.

To uninstall Integrator Server from your Windows or Linux machine:

1. Shut down Tomcat.
2. Delete the Tomcat install directory.

Uninstalling Integrator Server from WebLogic

Use the standard WebLogic procedure for uninstalling web applications to uninstall Integrator Server from WebLogic

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

Uninstalling Studio

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

Index