Endeca® Discovery Framework

Installation Guide

Contents

Preface.....7
About this guide.....7
Who should use this guide.....7
Conventions used in this guide.....7
Contacting Endeca Customer Support.....8

Chapter 1: Before you install.....9
Overview of the Endeca Discovery Framework.....9
    Features available in this version.....9
System requirements.....10
Compatibility with Endeca components.....11
Upgrading from a previous version of the Discovery Framework.....11
    Upgrading from Discovery Framework 1.1 to 1.2.....11
    Upgrading from Discovery Framework version 1.0.....12
Obtaining more information.....13

Chapter 2: Installing the Discovery Framework.....15
Downloading the Endeca Discovery Framework software.....15
Installing your package.....16
    Installing the Windows Tomcat bundle.....16
    Installing the Linux Tomcat bundle.....17
    Installing the Discovery Framework on Tomcat 5.5.....18
    Installing the Discovery Framework on the WebSphere Application Server.....21

Chapter 3: About Discovery Framework logging.....29
Modifying logging in your Discovery Framework components.....29
    Adjusting the logging verbosity level in the Control Panel.....29
    Modifying portal-log4j-ext.xml.....31
Setting up logging for your Discovery Framework application.....31
About log4j.properties files.....32
Learning about log4j.....32

Chapter 4: Getting started with the Discovery Framework.....33
Starting the Discovery Framework.....33
Accessing the Control Panel.....33
    About Discovery Framework settings.....34
Adding Endeca components 36
About data sources.....36
    About the sample MDEX Engine data sources.....36
    Adding data sources to the Discovery Framework.....36
    Changing an Endeca component's data source.....37

Chapter 5: Other installation tasks.....39
Using a different database.....39
    Overview of switching to a different database.....39
Installing Corda.....40
    About the Corda Server servlet.....41
    Obtaining the Corda software.....41
    Deploying the Corda Server servlet in an application server.....41
    Confirming the Corda Server servlet deployment.....42
    Deploying PCXML templates.....42
    Updating the Chart component with changes to the Corda Server 43
    Adding or removing PCXML templates.....43
    Troubleshooting Corda.....44
Uninstalling the Discovery Framework.....44

Copyright and disclaimer

Product specifications are subject to change without notice and do not represent a commitment on the part of Endeca Technologies, Inc. The software described in this document is furnished under a license agreement. The software may not be reverse engineered, decompiled, or otherwise manipulated for purposes of obtaining the source code. The software may be used or copied only in accordance with the terms of the license agreement. It is against the law to copy the software on any medium except as specifically allowed in the license agreement.

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose without the express written permission of Endeca Technologies, Inc.

Portions of this document and the software are subject to third-party rights, including:

Trademarks

Endeca, the Endeca logo, Guided Navigation, MDEX Engine, Find/Analyze/Understand, Guided Summarization, Every Day Discovery, Find Analyze and Understand Information in Ways Never Before Possible, Endeca Latitude, Endeca Profind, Endeca Navigation Engine, and other Endeca product names referenced herein are registered trademarks or trademarks of Endeca Technologies, Inc. in the United States and other jurisdictions. All other product names, company names, marks, logos, and symbols are trademarks of their respective owners.

The software may be covered by one or more of the following patents: US Patent 7035864, US Patent 7062483, US Patent 7325201, US Patent 7424528, US Patent 7567957, US Patent 7617184, Australian Standard Patent 2001268095, Republic of Korea Patent 0797232, Chinese Patent for Invention CN10461159C, Hong Kong Patent HK1072114, European Patent EP1459206B1, and other patents pending.

Endeca Discovery Framework Installation Guide • July 2010

Preface

The Endeca® Information Access Platform is the foundation for building applications that help people understand complex information, fostering discovery and improving daily decision-making. These applications instantly summarize data and content for users—even for unanticipated requests. The Guided Summarization™ experience leads to unexpected insights in millions of everyday decisions, increasing revenue, decreasing costs, and accelerating operations.

The Endeca Information Access Platform is powered by MDEX Engine™ technology, a new class of database designed for exploring information, not managing transactions. The MDEX Engine is supported by:

- An adaptive application component library that enables the rapid development of information access applications that automatically adapt to changes in the data and content.
- A Web-based management suite that empowers managers to highlight the right information at the right time to end users through adaptive presentation rules and dynamic pages.

These essential capabilities are delivered as an enterprise-class platform, with the scalability, reliability, and security that leading organizations demand.

About this guide

This guide contains installation instructions for setting up the Endeca Discovery Framework on Windows and Linux.

Who should use this guide

This guide is intended for developers who are building applications using the Endeca Discovery Framework on Windows or Linux.

Conventions used in this guide

This guide uses the following typographical conventions:

Code examples, inline references to code elements, file names, and user input are set in `monospace` font. In the case of long lines of code, or when inline monospace text occurs at the end of a line, the following symbol is used to show that the content continues on to the next line: ¬

When copying and pasting such examples, ensure that any occurrences of the symbol and the corresponding line break are deleted and any remaining space is closed up.

Contacting Endeca Customer Support

The Endeca Support Center provides registered users with important information regarding Endeca software, implementation questions, product and solution help, training and professional services consultation as well as overall news and updates from Endeca.

You can contact Endeca Standard Customer Support through the Support section of the Endeca Developer Network (EDeN) at *http://eden.endeca.com*.

Chapter 1

Before you install

This section provides an overview of the Endeca Discovery Framework, system requirements, and other information you need to know before installing.

Overview of the Endeca Discovery Framework

The Endeca Discovery Framework enables rapid configuration of search applications—often in just minutes—that offer the highly interactive Guided Navigation® user experience across a full range of structured and unstructured enterprise data.

Lightweight and easy to deploy, the Discovery Framework is ideal for the development of enterprise-quality search applications. Due to the Discovery Framework's component-based nature, applications built with it are simple to control, adapt, and extend. Granular layout and configuration control let users manage and personalize their own experiences.

About the Component SDK

The Component SDK is a packaged development environment for portlets, themes, layout templates, and other portal element. Endeca has modified Liferay's version of its Plugins SDK to include the Endeca enhancements, such as the `EndecaPortlet` core class. The installation and use of the Component SDK is covered in the *Discovery Framework Extension Guide*.

Features available in this version

The Discovery Framework 1.2 provides a core set of application capabilities designed to address the most common requirements for search applications.

The Discovery Framework contains the following Endeca components:

- The **Bookmarks** component allows you to save the navigation state and component state of a given component so that you can return to them.
- The **Breadcrumbs** component provides breadcrumb navigation aid functionality.
- The **Chart** component lets you access Corda-based Analytics charting.
- The **Data Sources** component allows you to view configured data sources and test the connection to them. In addition, you can reload updated configuration based on edits you have made on disk.
- The **Endeca Attribute Settings** component allows you to modify the display names of attributes in your data sources. Note that this component appears in the Liferay Control Panel.

- The **Framework Settings** component provides access to state, security, and other settings. Note that this component appears in the Liferay Control Panel.
- The **Guided Navigation** component provides Endeca Guided Navigation functionality.
- The **Performance Metrics** component displays information about component and MDEX Engine query performance.
- The **Range Filter** component allows you to add and modify range filters.
- The **Record Details** component displays all of the properties for the record in question.
- The **Results Table** component provides a simple interface for displaying results, along with an example of view transitions.
- The **Sample** component provides developers with a template from which they can build their own custom components.
- The **Searchbox** component provides searchbox functionality.
- The **Tabbed Component Container** allows you to create a tabbed interface within a region of a page and then store different components on various tabs.
- The **Advanced Visualization** component provides Xcelsius dashboarding within your application.

Note: For more information about components, see the *Discovery Framework Component Catalog*.

Liferay components in the Discovery Framework

The Discovery Framework includes a set of Liferay Portal content management components that allow you to do things like publish HTML content or embed an external Web site or application in a component. For an overview of Liferay Web content management, see *here*. For more details, see this Liferay *blog post*, or search the *Liferay Portal* site.

The Discovery Framework includes Liferay's **Languages** component, which lets you change the server locale.

System requirements

The Endeca Discovery Framework version 1.2 has the following requirements:

Hardware requirements

The hardware requirements for the Discovery Framework 1.2 are the same as those for Endeca MDEX Engine version 6.1.x. For details, see the *Endeca MDEX Engine Installation Guide*.

Supported operating systems

The Discovery Framework 1.2 is supported on the same Windows and Linux operating systems as the Endeca MDEX Engine version 6.1.x, with the exception noted below. For details, see the *Endeca MDEX Engine Installation Guide*.

Important: The Discovery Framework is not supported on Sparc Solaris.

Software requirements

The Discovery Framework is a Web-based application that runs in an application server.

- Supported browsers: Firefox 3.6, Internet Explorer 8

Tip: Firefox is recommended.

- Supported application servers: Tomcat 6, Tomcat 5.5, WebSphere Application Server (WAS) 7
- Supported Java versions: Tomcat 6 is supported with Sun Java 6; Tomcat 5.5 is supported with Sun Java 5; WAS 7 with IBM Java 6
- Supported database systems: MySQL 5.1, DB2 9.5

Compatibility with Endeca components

This document assumes that you already have a running MDEX Engine at which you can point the Discovery Framework.

The Endeca Discovery Framework version 1.2 is compatible with the following Endeca components:

- MDEX Engine 6.1.x
- Platform Services 6.0.1 or 6.1
- Developer Studio 6.0.1 or 6.1
- Deployment Template 3.1 or 3.2

Upgrading from a previous version of the Discovery Framework

This section describes how to upgrade your Discovery Framework application from a previous version.

Note: You can only have one version of the Discovery Framework installed on your machine at a time.

Upgrading from Discovery Framework 1.1 to 1.2

Data migration is supported between Discovery Framework 1.1 and 1.2.

Backing up your Discovery Framework 1.1 files

The first step in migrating to Discovery Framework 1.2 is backing up your 1.1. files. After backing up your files, you can uninstall Discovery Framework 1.1.

To back up your Discovery Framework 1.1 files and uninstall:

1. Stop your Discovery Framework 1.1 server.
2. Back up your database as follows:
    - If you are using HSQL, skip this step. Your database will be backed up as part of step 4 below.
    - If you are using MYSQL, DB2, or some other RDBMS, follow the backup procedures from your vendor.
3. Back up any Discovery Framework 1.1 customizations you have made, such as database connectivity options in `portal-ext.properties`, Corda PCXML files, or Xcelsius (SAP Crystal Dashboard Design) visualizations.

4. Back up the entire `endeca-portal/data` directory to a safe backup location.
   If you have changed the location of anything normally kept in `endeca-portal/data`, such as your Endeca data source definitions, JCR repository, or Lucene search indexes, back up your custom location(s) as well.
5. Back up your license, at `endeca-portal/ee/license`.
6. Uninstall Discovery Framework 1.1.
   Uninstalling consists of removing the packages and directories created in your installation.

Updating your database schema

Next, you need to update your database schema with 1.2 changes.

To update your database schema:

Using the appropriate tools for your database vendor, make the following changes:

- Change `endeca_metadata.description` to varchar(250).
- Change `endeca_metadata_property.description` to varchar(250).
- Add `endeca_bookmarks_datasources.serializedRequestState` as CLOB.

Installing the Discovery Framework 1.2 and restoring backups

To install the Discovery Framework 1.2 and restore files backed up from version 1.1:

1. Install Discovery Framework 1.2, following the steps in the next section of this guide. Do not start the server.
2. Restore the entire `endeca-portal/data` directory from your safe backup location. Overwrite the `endeca-portal/data` files that were installed with the Discovery Framework 1.2.

   If you have changed the location of anything normally kept in `endeca-portal/data`, such as your Endeca data source definitions, JCR repository, or Lucene search indexes, restore and re-verify your custom locations as well.
3. Restore your database, as follows:
   - If you are using HSQL, skip this step, since your database was restored as part of step 2.
   - If you are using MYSQL, DB2, or some other RDBMS, your database should still be intact. If it is not, follow the restore procedures from your vendor.
4. Configure the Discovery Framework to connect to your database. If you are not using HSQL, you should have created a backup of your database connection strings in the previous procedure.
5. Restore any Discovery Framework 1.1 customizations you have made.

   Note: These customizations may not be supported in Discovery Framework 1.2.
6. Restore your license at `endeca-portal/ee/license`.
7. Start the Discovery Framework.

Upgrading from Discovery Framework version 1.0

Data migration is not supported between Discovery Framework 1.0 and 1.2.

To upgrade from Discovery Framework 1.0 to Discovery Framework 1.2:

1. Uninstall Discovery Framework 1.0.

   Uninstalling consists of removing the packages and directories created in your installation.

   Note: If your 1.0 deployment was installed in an existing application server, follow the appropriate steps to stop and uninstall the DF 1.0 application from the application server before installing DF 1.2.

2. Install Discovery Framework 1.2, following the instructions in the next chapter.
3. Rebuild your application.

Obtaining more information

Because the Discovery Framework is built upon the Liferay Portal, you can access Liferay's documentation for more information about how to perform administrative tasks.

Specifically, the *Liferay Portal Administrator's Guide* provides extensive information about installing, configuring, and maintaining a portal. To access a free PDF download of this guide, go to *http://www.liferay.com* and navigate to Documentation.

Liferay developer resources

In addition to its formal administrator documentation, Liferay offers developer assistance in the form of blogs, wikis, and forums. To access this, go to *http://www.liferay.com* and navigate to Community.

The Endeca Developer Network (EDeN)

You can obtain more information about the Discovery Framework and other Endeca products at the Endeca Developer Network (EDeN) at *http://eden.endeca.com*. In particular, EDeN's Discovery Framework forum provides discussions for technical and business users of Endeca's Discovery Framework and its components, including topics such as development, extension, deployment, and configuration.

Additional Endeca documentation

The Discovery Framework doc directory contains the following documents:

- The *Discovery Framework Component Catalog*, which provides an overview of each of the standard components.
- The *Discovery Framework Extension Guide* and Discovery Framework javadoc, both aimed at developers using the Discovery Framework.

Chapter 2

Installing the Discovery Framework

This section contains the Discovery Framework installation procedures for the supported application servers.

Downloading the Endeca Discovery Framework software

You can download the Endeca Discovery Framework from the Downloads section of the Endeca Developer Network (EDeN).

To download the Discovery Framework software:

1. If you have not previously done so, establish a Support account with download access through the Support section of the Endeca Developer Network (EDeN) at *http://eden.endeca.com*. This enables the Endeca Support and Customer Care groups to track which versions of the software you are using.
2. Navigate through the EDeN site as follows:
   a) On the EDeN homepage, click **Downloads**.
   b) On the **Tools and Utilities** page, find the **Product Downloads** section and click **View and download purchased products**.
   c) On the **Product Downloads** page, find and click **Discovery Framework**.
   d) In the **Current Releases** table, click **Discovery Framework 1.2**.

   The **Product Download** page contains links to all available Discovery Framework packages.
3. Download the appropriate Discovery Framework zip files, depending on your installation environment:

   - To install the Tomcat 6 bundle for Windows, download `endeca-portal-1.2.zip` and `components-1.2.zip` to your development server.
   - To install the Tomcat 6 bundle for Linux, download `endeca-portal-1.2.tgz` and `components-1.2.zip` to your development server.
   - To install Discovery Framework for Tomcat 5.5 or the Websphere Application Server 7, download `endeca-portal-1.2.war`, `endeca-portal-dependencies-1.2.zip`, and `components-1.2.zip`.

   Note: For instructions on downloading and installing the Corda package (which is only required if you plan to use the Chart component), see the section "Installing Corda."

Installing your package

After downloading the Discovery Framework software, you can install it on your development server.

There are four options for installing this release of the Discovery Framework:

- Discovery Framework with the Windows Tomcat bundle. This is based on Tomcat 6 and Java 1.6.
- Discovery Framework with the Linux Tomcat bundle. This is based on Tomcat 6 and Java 1.6.
- Discovery Framework as a standalone application on Tomcat 5.5 application server.
- Discovery Framework as a standalone application on Websphere Application Server 7 application server.

Note: The following steps will deploy the portal using the default embedded Hypersonic database, which is not intended for production use. In production, you must deploy using an alternate database. More information about this process can be found in chapter 5 of this guide. Briefly, deploying an alternate database can be accomplished by modifying the `portal-ext.properties` file to specify the appropriate JDBC connection information for the desired database. Alternatively, you can follow the instructions in the *Liferay Portal Administrator's Guide* to set up a JDBC provider and data source in your application server, and configure the `portal-ext.properties` to look up the data source by JNDI name.

Installing the Windows Tomcat bundle

This topic provides the steps for installing the Discovery Framework Windows Tomcat bundle on your development server. In this version Tomcat 6 and the JVM 1.6 are embedded.

To install the Discovery Framework Tomcat bundle:

1. Unzip `endeca-portal-1.2.zip` to the directory of your choice.
2. Extract the `.war` files from `components-1.2.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.
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=C:\path\to\endeca-portal\tomcat-6.0.18`. (If you do not have these environment variables set, you can leave them un-set.)
4. Start the portal's Tomcat instance by running `endeca-portal\tomcat-6.0.18\bin\startup.bat`.

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

5. Go to the portal (`http://localhost:8080/`) in your browser, and log in using the following default credentials:

Option	Description
Email address	test@endeca.com
Password	test

6. Upon first use, enter the license key, as described in the topic that appears later in this section.

7. Optionally, you can set up *log4j* logging. `log4j` provides configurable, Java-based logging in an open-source utility.

   Note: For more information about Discovery Framework logging, see Chapter 3.

Installing the Linux Tomcat bundle

This topic provides the steps for installing the Discovery Framework Linux Tomcat bundle on your development server. In this version Tomcat 6 and the JVM 1.6 are embedded.

To install the Discovery Framework Tomcat bundle:

1. Extract `endeca-portal-1.2.tgz` to the directory of your choice.
2. Extract the `.war` files from `components-1.2.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.
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. (If the `JAVA_HOME` environment variable is not set, you must set it.)
4. Start the portal's Tomcat instance by running `endeca-portal/tomcat-6.0.18/bin/start¬ up.sh`.

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

5. Upon first use, enter the license key, as described in the next topic.
6. Go to the portal (`http://localhost:8080/`) in your browser, and log in using the following default credentials:

Option	Description
Email address	test@endeca.com
Password	test

7. Optionally, you can set up *log4j* logging. `log4j` provides configurable, Java-based logging in an open-source utility.

   Note: For more information about Discovery Framework logging, see Chapter 3.

Entering the license key

The Discovery Framework is build upon Liferay Portal Enterprise Edition. When the Discovery Framework is initially deployed and started, the first user to access the application is prompted for a license key.

To license your version of the Discovery Framework:

1. Download the license key from the Discovery Framework section of EDeN.
2. In the **Liferay Portal license** dialog box, enter the key in the **License Key** field.

3. Click **Save**. The portlet updates the dialog box with more detailed information about the license key.
4. Click **Save** again. The dialog box closes and you can proceed with your work.

Installing the Discovery Framework on Tomcat 5.5

You can deploy the Discovery Framework as a standalone application on Tomcat 5.5.

Note: Before following the steps here, consult the *Liferay Portal Administrator's Guide*, which contains portal deployment instructions and examples for Tomcat 5.5.

Note: The examples in this section are based on a Windows server Tomcat deployment. If you are installing on Linux, the steps will be similar, though you will need to substitute Linux binaries and paths. Where there is a significant difference, this is called out.

Note: These instructions assume that you have obtained the `apache-tomcat-5.5.x.zip` or `tar.gz` file from the *Apache Foundation* but that you have not yet installed it. The rest of these instructions will refer to the installation directory as `apache-tomcat-5.5.x`, leaving off the minor version number.

High-level overview of Tomcat 5.5 deployment

This topic provides an overview of the steps you need to take to deploy the Discovery Framework as a standalone application on Tomcat 5.5.

Details on each of these steps appear in the topics that follow.

To deploy the Discovery Framework on Tomcat 5.5:

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

Installing Tomcat 5.5 and deploying the dependency libraries

The Discovery Framework requires the deployment of several Java libraries.

To install the Tomcat software and deploy the Discovery Framework dependency libraries:

1. Create an `endeca-portal` directory. This will be the home directory for your Discovery Framework installation.
2. Create an `apache-tomcat-<version>` directory under the `endeca-portal` directory.
3. Unzip `apache-tomcat-5.5.x.zip` into `endeca-portal/apache-tomcat-5.5.x`, where `x` indicates the minor version number.

   Unzipping this file creates much of the directory structure mentioned below.
4. Unzip `endeca-portal-dependencies-1.2.zip` into a temporary directory.

   This zip file contains a collection of `.jar` files and other dependency files.

5. From the temporary directory, copy the following `.jar` files into the `endeca-portal/apache-tomcat-5.5.x/common/endorsed` directory:

   ```
   log4j.jar
   log4j.properties.jar
   ccpp.jar
   jutf7.jar
   ```

6. Under the `endeca-portal/apache-tomcat-5.5.x/common/lib` directory, create an `ext` directory.
7. From the temporary directory you created in step 4, copy the following `.jar` files into the `endeca-portal/apache-tomcat-5.5.x/common/lib/ext` directory that you just created:

   ```
   activation.jar
   annotations.jar
   commons-lang.jar
   cs_bindings.jar
   cxf-2.2.8.jar
   cxf-rt-databinding-jaxb-2.2.8.jar
   endeca-images.jar
   endeca-portal.jar
   endeca_navigation.jar
   ext-service.jar
   geronimo-activation_1.1_spec-1.0.2.jar
   geronimo-annotation_1.0_spec-1.1.1.jar
   geronimo-jaxws_2.1_spec-1.0.jar
   geronimo-saaj_1.3_spec-1.0.1.jar
   geronimo-stax-api_1.0_spec-1.0.1.jar
   geronimo-ws-metadata_2.0_spec-1.1.2.jar
   hsql.jar
   jabsorb.jar
   jaxb-api-2.1.jar
   jaxb-impl-2.1.7.jar
   jms.jar
   jsr173_1.0_api.jar
   jta.jar
   jtds.jar
   mail.jar
   mysql.jar
   portal-kernel.jar
   portal-service.jar
   portlet.jar
   postgresql.jar
   stax-1.2.0.jar
   wsdl4j-1.6.2.jar
   wstx.jar
   XmlSchema-1.4.3.jar
   ```

Modifying Tomcat configuration to work with the Discovery Framework

Before proceeding further, you must modify some Tomcat configuration files.

1. In the `endeca-portal/apache-tomcat-5.5.x/bin/` directory, modify `catalina.bat` (on Windows) or `catalina.sh` (on Linux) by adding the JAVA_OPTS line. This line should appear under the line `Execute The Requested Command` as follows:

   ```
   set JAVA_OPTS=%JAVA_OPTS% -Xmx1024m -XX:MaxPermSize=256m -Dfile.encod¬
   ing=UTF8 -Duser.timezone=GMT -Dorg.apache.catalina.loader.WebappClassLoad¬
   er.ENABLE_CLEAR_REFERENCES=false
   ```

   This increases the memory size for the server and establishes security configuration for the Discovery Framework.
2. Modify the `endeca-portal/apache-tomcat-5.5.x/conf/catalina.properties` file as follows to add the `ext` directory to the common class loader:

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

3. Create a new file called `ROOT.xml` and place it in `endeca-portal/apache-tomcat-5.5.x/conf/Catalina/localhost/`. Add the following line to the new file:

   ```
   <Context path="" />
   ```

4. To support UTF-8 URI encoding, edit the `server.xml` file located in the `endeca-portal/apache-tomcat-5.5.x/conf` directory as follows:

   ```
   <!-- Define a non-SSL HTTP/1.1 Connector on port 8080 -->
      <Connector
         port="8080"
         maxHttpHeaderSize="8192"
         maxThreads="150"
         minSpareThreads="25"
         maxSpareThreads="75"
         enableLookups="false"
         redirectPort="8443"
         acceptCount="100"
         connectionTimeout="20000"
         disableUploadTimeout="true"
         URIEncoding="UTF-8"
      />
   ```

Deploying and starting the Discovery Framework application

Once Tomcat configuration is complete, the Discovery Framework application can be deployed and started.

To deploy and start the Discovery Framework application:

1. Delete the contents of the `endeca-portal/apache-tomcat-5.5.x/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 the Discovery Framework application in the next step.
2. Unzip the contents of `endeca-portal-1.2.war` into the `endeca-portal/apache-tomcat-5.5.x/webapps/ROOT` directory.
3. Copy the `portal-ext.properties` file from the temporary directory you created for the `endeca-portal-dependencies-1.2.zip` file in step 4 to the `endeca-portal` directory. Do not edit the file.

4. Under the `endeca-portal` directory, create a `data` directory, and then create an `endeca-data-sources` directory below that.
5. Create a data source to place in the `endeca-portal/data/endeca-data-sources` directory. For information about data sources, see the *Discovery Framework Extension Guide*. In addition, you can reference the sample data source files, which are located in the `endeca-data-sources` directory in the temporary directory you created for the `endeca-portal-dependencies-1.2.zip` file in a previous step.
6. Start the portal's Tomcat instance by running `endeca-portal\tomcat-5.5.x\bin\start¬up.bat`.

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

7. Go to the portal (`http://localhost:8080/`) in your browser, and log in using the following default credentials:

Option	Description
Email address	test@endeca.com
Password	test

8. Upon first use, enter the license key, as described in the topic that appears later in this section.

Entering the license key

The Discovery Framework is build upon Liferay Portal Enterprise Edition. When the Discovery Framework is initially deployed and started, the first user to access the application is prompted for a license key.

To license your version of the Discovery Framework:

1. Download the license key from the Discovery Framework section of EDeN.
2. In the **Liferay Portal license** dialog box, enter the key in the **License Key** field.
3. Click **Save**. The portlet updates the dialog box with more detailed information about the license key.
4. Click **Save** again. The dialog box closes and you can proceed with your work.

Installing the Discovery Framework on the WebSphere Application Server

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

Note: Before following the steps here, consult the *Liferay Portal Administrator's Guide*, which contains portal deployment instructions and examples for WebSphere Application Server 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.

High-level overview of Websphere Application Server deployment

This topic provides an overview of the steps you need to take to deploy the Discovery Framework on WAS.

Details on each of these steps appear in the topics that follow.

To deploy the Discovery Framework on WAS:

1. Deploy dependency `.jar` files. The exact list of required files appears below.
2. Start (or restart) the WAS server.
3. Install the Discovery Framework `.war` file as an enterprise application.
4. Edit and deploy `portal-ext.properties`.
5. Create the `endeca-data-sources/*.json` data source configuration files.

   For more information, see the section "About data sources."
6. Install the Endeca theme, portlet components, and post-render hook.
7. Start the Discovery Framework enterprise application, entering the license key upon first use.
8. Optionally, repeat step 7 for any additional plugins you want to add.

About the Liferay Home directory

The following instructions refer to a directory called Liferay Home. This topic explains how this directory is created.

When the Discovery Framework application is started, it creates a Liferay Home directory. By default, when deployed on WAS, the Liferay Home directory is created relative to the user's home directory. For example, if WAS is run as user `endeca` with home directory `/home/endeca`, the Liferay Home directory is created at `/home/endeca/liferay`.

Deploying Discovery Framework dependency libraries

The Discovery Framework 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 Discovery Framework dependency libraries:

1. Unzip the `.jar` files found in `endeca-portal-dependencies-1.2.zip`.
2. Upload the following list of `.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
cs_bindings.jar
cxf-2.2.8.jar
cxf-rt-databinding-jaxb-2.2.8.jar
endeca-images.jar
endeca-portal.jar
endeca_navigation.jar
ext-service.jar
geronimo-activation_1.1_spec-1.0.2.jar
geronimo-annotation_1.0_spec-1.1.1.jar
geronimo-jaxws_2.1_spec-1.0.jar
geronimo-saaj_1.3_spec-1.0.1.jar
geronimo-stax-api_1.0_spec-1.0.1.jar
geronimo-ws-metadata_2.0_spec-1.1.2.jar
hsql.jar
```

```
jabsorb.jar
jaxb-api-2.1.jar
jaxb-impl-2.1.7.jar
jsr173_1.0_api.jar
log4j.jar
portal-kernel.jar
portal-service.jar
portlet.jar
slf4j-api.jar
slf4j-log4j12.jar
stax-1.2.0.jar
wsdl4j-1.6.2.jar
wstx.jar
XmlSchema-1.4.3.jar
```

3. Restart the WAS server so that it can pick up the newly available `.jar` files.

Deploying the standalone portal WAR

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

Note: 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 by using tools like `wsadmin`, rather than the Integrated Solutions Console.

Note: 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 Discovery Framework deployment). After following these steps, you will be able to adjust the context root for the Discovery Framework application, to ensure it does not conflict with other applications.

To deploy the Discovery Framework standalone portal WAR on WebSphere Application Server:

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 and select the Endeca Discovery Framework WAR you downloaded earlier (`endeca-portal-1.2.war`).
5. Select **Choose to generate default bindings and mappings** and 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 enter `default_host` in the text field.
   Click Next.

7. By default, the application name is **endeca-portal-1_2_war**. Set the application name to a more relevant name (for example, `DiscoveryFramework`). All other installation options can remain unchanged. Click Next.

   Note: Do not use spaces in the application name. For example, use **DiscoveryFramework** instead of **Discovery Framework**.

8. Keep default server bindings, and click **Next**.
9. Set the context root to `/`, and click **Next**.

   Note: This can be modified later to allow the Discovery Framework to share an application server with other applications, rather than taking over the root context.

10. Click **Finish**.
11. Wait for installation and, if it is successful, click **Save directly to master configuration**.

Editing the portal-ext.properties file for WebSphere deployment

Before deploying your `portal-ext.properties` file, you must edit it.

1. Open the `portal-ext.properties` file and 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
# Discovery Framework enterprise application.
auto.deploy.dest.dir=${liferay.home}/websphere-deploy
```

2. Save the file.

Configuring portal-ext.properties for WebSphere deployment

After you edit your `portal-ext.properties` file, there are two ways to deploy it in WAS.

- By updating the application to include the `portal-ext.properties` file.
- By uploading the `portal-ext.properties` file to the Liferay Home directory on the server.

Both methods are described in the following topics.

Updating the application to include the portal-ext.properties file

After you create the `portal-ext.properties` file, you can use the IBM Integrated Solutions Console to update the portal WAR module with the additional file.

This topic documents the use of the Integrated Solutions Console to update the Discovery Framework application, to include `portal-ext.properties` in the `endeca-portal.war` module. These steps may be performed with the `wsadmin` tool instead of the Integrated Solutions Console and may need to be adjusted for alternate WAS configurations.

Note: In order to make changes to the `portal-ext.properties` file, users will need to repeat these steps to update the application with updated versions of the `portal-ext.properties` file. In some environments, it may be more appropriate to deploy the `portal-ext.properties` file to the Liferay Home directory, where it can be updated without updating the deployed application. That option is described in the next topic.

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

1. Go to **Applications** > **Application Types** > **WebSphere Enterprise Applications** and select the enterprise application created when you deployed the portal WAR. Click Update.
2. Select **Replace or add a single file**.
3. Specify the path to deploy the file into the `WEB-INF/classes` directory of the portal Web application. For example: `endeca-portal-1.2.war/WEB-INF/classes/portal-ext.properties`
4. Browse to where you created the file on your computer.
5. Once the file has successfully updated, click **Save directly to master configuration**.

Uploading the portal-ext.properties file to the Liferay Home directory on the server
After you create the `portal-ext.properties` file, you can manually upload it to WAS.

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

Upload the `portal-ext.properties` file to the Liferay Home directory. For example: `/home/endeca/liferay/portal-ext.properties`.
Liferay reads these properties when the Discovery Framework application is started. When the file is modified, Liferay needs to be restarted to read the updated properties.

Example settings for portal-ext.properties
Endeca's default version of `portal-ext.properties` is included in the package `endeca-portal-dependencies-1.2.zip`.

This file serves as a useful starting point for configuration of the portal properties, and should be deployed to the application server according to the steps described in a previous topic.

Note: Most of the settings in the default `portal-ext.properties` file are not specific to deployment on WAS. However, the following additional setting included in the file is important for portlet deployment on WAS:

```
# Specify a directory where Liferay will "deploy" processed plugins.
# This is where the WAS user will deploy WARs from (as modules within
# the enterprise app)
auto.deploy.dest.dir=${liferay.home}/websphere-deploy
```

Keep in mind that 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.

Deploying Endeca data source configuration

To configure one or more MDEX Engines as data sources for the Discovery Framework, a JSON configuration file needs to be deployed for each MDEX Engine.

These files should be deployed relative to the Liferay Home directory. Sample data source configuration files are provided as `.json.sample` files in the `endeca-portal-dependencies-1.2.zip` file you downloaded.

To deploy Endeca data source configuration:

Upload the files to the `data/endeca-data-sources/` subdirectory.

For example: `/home/endeca/data/endeca-data-sources/default.json`

Deploying the Endeca theme

Themes are plugin components that control the look and feel of your Discovery Framework application.

To deploy the Endeca theme (endeca-theme), follow the steps given in the topic "Deploying components in WebSphere Application Server."

Note: The default `portal-ext.properties` file specifies the Endeca theme as the default, so things may not work as expected until the Endeca theme is deployed.

Starting the application

Once the Discovery Framework 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** and select the enterprise application created when you deployed the portal WAR.
2. If it is not already running, click **Start** to start it.
3. View your deployed application at the root context of the server.

Entering the license key

The Discovery Framework is build upon Liferay Portal Enterprise Edition. When the Discovery Framework is initially deployed and started, the first user to access the application is prompted for a license key.

To license your version of the Discovery Framework:

1. Download the license key from the Discovery Framework section of EDeN.
2. In the **Liferay Portal license** dialog box, enter the key in the **License Key** field.
3. Click **Save**. The portlet updates the dialog box with more detailed information about the license key.
4. Click **Save** again. The dialog box closes and you can proceed with your work.

Deploying components in WebSphere Application Server

This topic describes how to deploy components, themes, and other plugins in WAS.

WAS does not support the hot deployment of components. However, Liferay's deployment code must update plugins by adding necessary libraries and configuration files. For example, Liferay's 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.PortletDeploymentEn¬
abled</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 the Discovery Framework to load the newly deployed portlet.

To deploy plugins in WAS:

1. Make the plugin `.war` file available to Liferay for "hot-deploy."

2. Deploy the `.war` file generated in step 1 as a module in the Discovery Framework enterprise application. There are two ways to do this:
   - Through the Websphere Integrated Solutions Console.
   - At the command line, using `wsadmin`.

More details on each of these steps appear in the following topics.

About Liferay deploy directories

This topic describes the relationship between Liferay deploy and websphere-deploy directories.

By default, Liferay reads `${liferay.home}/deploy` as its `deploy` directory. This is typically `liferay/deploy` within the home directory of the user running the WAS server (for example, `/root/liferay/deploy` or `/home/endeca/liferay/deploy`).

Liferay also utilizes a `deploy` output directory, set to `${liferay.home}/websphere-deploy` in the `portal-ext.properties` example above. The `deploy` output directory must already exist. If it does not, you must manually create it before following this procedure.

Allowing Liferay to hot-deploy a plugin

After you upload your `.war` files to Liferay's `deploy` directory, Liferay's deployer can find and process them.

Upload portlet, theme, hook, and other plugin `.war` files to the this `deploy` directory.
The Liferay deployer picks them up, modifies them, and delivers them to the `deploy` output directory.

Manually deploying generated .war files

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 manually deploy a generated `.war` file:

1. Go to **Applications** > **Application Types** > **WebSphere Enterprise Applications** and select the enterprise application created when you deployed the portal `.war` file. Click Update.
2. Select "Replace or add a single module."
3. 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`.
4. Browse the remote file system to the newly created `.war` file in the Liferay deploy output directory. Continuing the example above, this might be `/root/liferay/websphere-deploy/endeca-navigation-portlet.war`.
5. Select the detailed install path and 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/`).
6. Once it has successfully updated, click "Save directly to master configuration."

Using wsadmin to deploy the generated .war file

You can also deploy the generated `.war` file at the command line using the `wsadmin` tool.

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

In the `wsadmin` tool, enter a command similar to the example below.

In this example, the enterprise application is named `DiscoveryFramework`. The module being added has the file name `endeca-navigation-portlet.war` and the display name `endeca-navigation-portlet`. This command is executed from the Liferay deploy output directory (that is, the directory containing the `endeca-navigation-portlet.war` file). In our example, this command is executed in `/root/liferay/websphere-deploy/.` .

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

Troubleshooting WAS deployment

This topic discusses an issue to keep in mind when deploying the Discovery Framework on WAS.

Updating the Discovery Framework .war file

If you need to update the Discovery Framework `.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 3

About Discovery Framework logging

The Discovery Framework uses the Apache `log4j` logging utility.

Modifying logging in your Discovery Framework components

There are two ways to modify logging for your Discovery Framework components.

From a developer's perspective, you will most frequently need to adjust a logging verbosity level for a given class or class hierarchy. The easiest way to do this is through the Liferay Control Panel. If you need to modify logging in a more complex manner, or want to change default settings, you can modify the `portal-log4j-ext.xml`.

Both of these methods are described in this section.

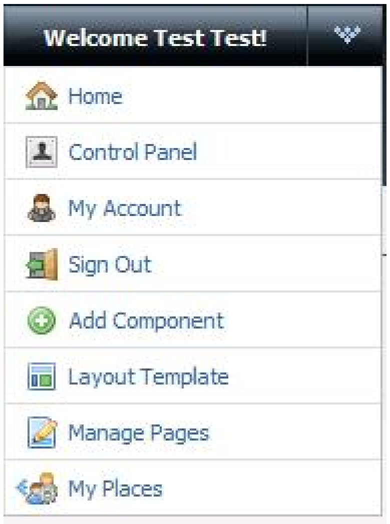
Note: Using either of these methods causes Liferay to adjust the log verbosity for both `log4j` and the Java Utility Logging Implementation (JULI). Code using either of these loggers should respect this configuration.

Adjusting the logging verbosity level in the Control Panel

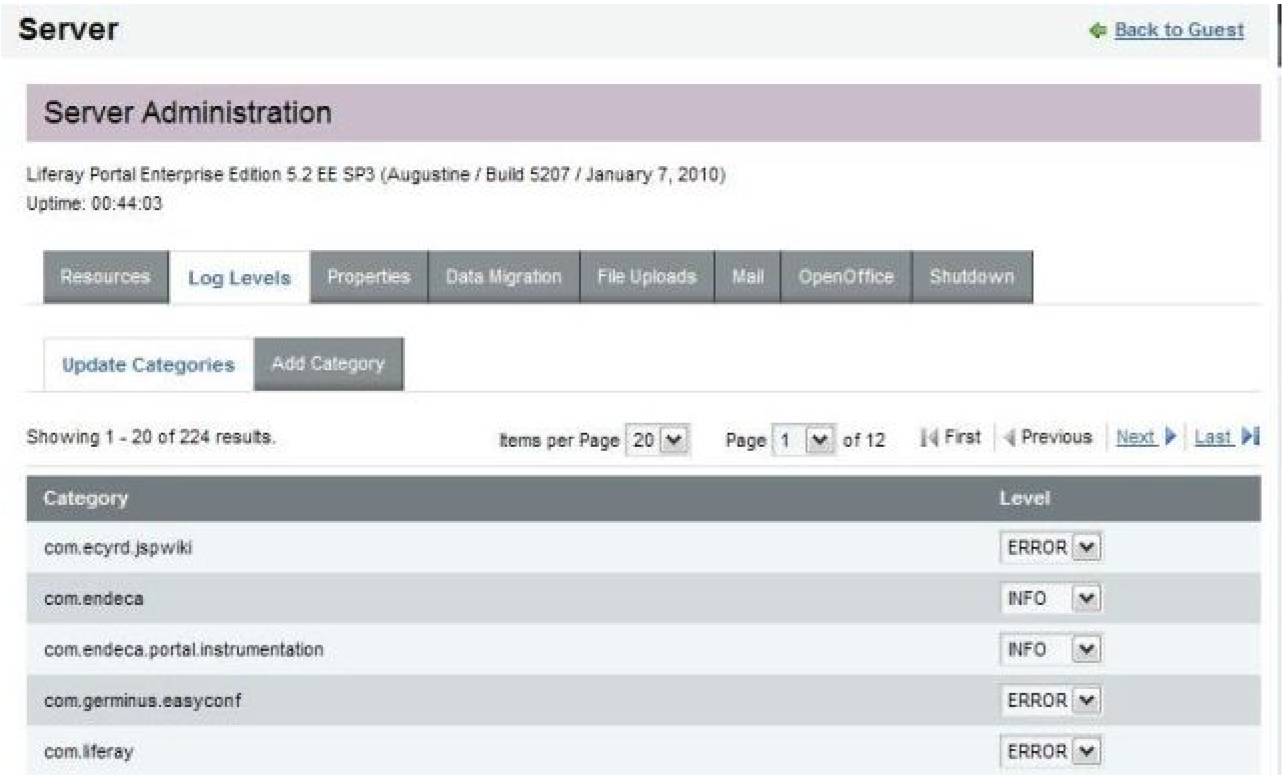
The easiest way to dynamically adjust logging verbosity levels for any class hierarchy is from the Liferay Control Panel.

To adjust logging verbosity in the Liferay Control Panel:

1. In the Discovery Framework, point the cursor at the Dock in the upper-right corner of the page. The Dock is labeled "Welcome <*user name*>!"



2. From the drop-down menu, choose **Control Panel**.
3. From the **Control Panel** tool menu, choose **Server Administration**.
4. In the **Server Administration** pane, choose the **Log Levels** tab.



Liferay Portal Enterprise Edition 5.2 EE SP3 (Augustine / Build 5207 / January 7, 2010)
Uptime: 00:44:03

Showing 1 - 20 of 224 results.

Category	Level
com.ecyrd.jspwiki	ERROR
com.endeca	INFO
com.endeca.portal.instrumentation	INFO
com.germinus.easyconf	ERROR
com.liferay	ERROR

5. Scroll to find the class hierarchy you want to modify, and then adjust the logging level in the drop-down list. The available options are:
   - OFF
   - FATAL
   - ERROR
   - WARN
   - INFO

- DEBUG
- ALL

> **Note:** When you modify a class hierarchy, all classes that fall under that class hierarchy are also changed.

6. When you have finished adjusting log levels, click **Save**.

> **Note:** By default, Endeca sets log levels for `com.endeca` and `com.endeca.portal.instrumentation`. You can adjust these levels. In addition, you can set the verbosity for a specific class or package by using the **Add Category** tab.

Modifying portal-log4j-ext.xml

Liferay's primary log configuration is managed in the `portal-log4j.xml` file (which is packed inside the portal application's `WEB-INF/lib/portal-impl.jar`).

As with many other configuration files, Liferay provides administrators with a second configuration file, `portal-log4j-ext.xml` (located in the portal application's `/WEB-INF/classes/META-INF/` directory), which can be used to override settings in the main `portal-log4j.xml` file.

Both of these files are in standard `log4j` XML configuration format. Both files allow creating and modifying appenders, binding appenders to loggers, and adjusting the default log verbosity of different classes and packages. By default, the Endeca override file `portal-log4j-ext.xml` specifics a log verbosity of INFO for the `com.endeca` and `com.endeca.portal.instrumentation` packages. Endeca does not override any of the default log verbosity settings specified for non-Endeca components packaged in `portal-log4j.xml`.

Setting up logging for your Discovery Framework application

The Discovery Framework uses the Apache log4j logging utility.

Liferay's primary log configuration is managed in `portal-log4j.xml` file (which is packed inside portal application's `WEB-INF/lib/portal-impl.jar`). As with many other configuration files, Liferay provides administrators with a second configuration file, `portal-log4j-ext.xml`, which can be used to override settings in the main `portal-log4j.xml` file.

Both of these files are in standard log4j XML configuration format. Both files allow creating and modifying appenders, binding appenders to loggers, and adjusting the default log verbosity of different classes/packages. By default, the Endeca override file specifics a log verbosity of INFO for the `com.endeca` and `com.endeca.portal.instrumentation` packages. Endeca does not override any of the default log verbosity settings specified for non-Endeca components packaged in `portal-log4j.xml`.

The Endeca log configuration specifies three appenders. The main root logger prints all messages to two locations: the console, which is typically redirected to the application server's output log (`catalina.out` in Tomcat and `SystemOut.log` in WAS), and a file called `df.log`. That file is specified relative to the working directory. If not adjusted, the `df.log` file can typically be found in one of the following locations:

1. If Tomcat was started by running the `startup.bat` or `startup.sh` script, the log is found wherever the script was run. For example, if you navigate to `tomcat-5.5.27/bin` and execute the startup script, your logs appear in `tomcat-5.5.27/bin/df.log`.
2. If Tomcat was registered and started as a Windows service, the log files may be located in `C:\Windows\System32\df.log` or `C:\Windows\SysWOW64\df.log`.
3. If Tomcat is a server inside of Eclipse, the log files may be located in the root of the Eclipse directory (such as `C:\eclipse\df.log`).
4. If running WAS 7, the log files may be located relative to the profile's working directory (such as `/localdisk/WAS/AppServer/profiles/AppSrv01/df.log`).

In addition to the console and `df.log` appenders, Endeca also provide a second file appender for capturing metrics logging. This appender creates a file called `df-metrics.log`, which is generated in the same location as `df.log`. All log entries produced by classes in `com.endeca.portal.instrumentation` are routed to `df-metrics.log`—they are not printed to the console or to `df.log`. By keeping performance information separate, administrators can easily distinguish server logs from performance logs, and can easily run analysis scripts on the performance logs.

Note: For further details on `log4j` logging in Liferay, see the *Liferay Portal Administrator's Guide*.

About log4j.properties files

This topic describes the different versions of the `log4j.properties` file.

The version of the `log4j.properties` file that is located in `common/endorsed/log4j.properties.jar` is used to configure logging for the Tomcat bundle. The file ensures that there is some preliminary `log4j` configuration, because`log4j` is initialized before the Discovery Framework in the Tomcat bundle. This `log4j.properties` file provides minimal configuration, which ensures that initial messages are logged to the console in the same format as the default configuration in `portal-log4j-ext.xml`. The settings in the `log4j.properties` file only affect a small number of messages printed as the server is starting. Once the Discovery Framework starts and loads its XML configuration file, it overrides the settings in the `log4j.properties` file. Therefore, it should not be necessary for administrators to modify this properties file.

In addition, all deployed portlets, as well as the Discovery Framework application itself, have their own `log4j.properties` files, located in `WEB-INF/classes`. Because the Discovery Framework uses XML configuration files, these properties files have no effect.

Learning about log4j

This topic provides links to additional information about `log4j`.

The *Apache log4j site* provides general information about `log4j`, along with documentation.

For more information about `log4j` logging in Liferay, see the *Liferay documentation*, including the *Liferay Portal Administrator's Guide*.

Chapter 4

Getting started with the Discovery Framework

This section describes how to launch and configure the Discovery Framework and begin to work with it.

Starting the Discovery Framework

You start the Discovery Framework by starting your application server, going to the portal in your Web browser, and logging in.

The default login is `test@endeca.com`, and the default password is `test`.

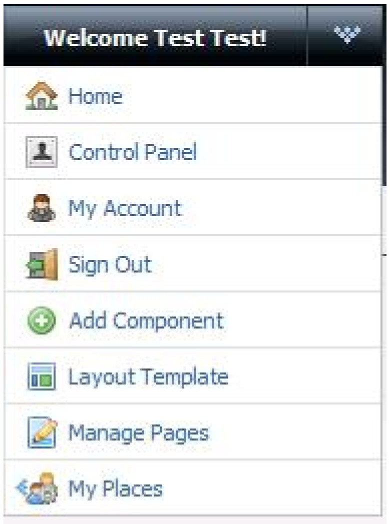
Accessing the Control Panel

After logging in to the Discovery Framework, you may also want to access edit controls. This is done through the **Control Panel**.

The **Control Panel** provides access to a wide range of edit controls, including managing accounts, adding new users, and monitoring performance. For full documentation on **Control Panel** capabilities, see the *Liferay Portal Administrator's Guide*. To access a free PDF download of this guide, go to *http://www.liferay.com* and navigate to Documentation.

To access the **Control Panel**:

1. Point the cursor at the **Dock** in the upper-right corner of the page. The **Dock** is labeled "Welcome *<user name>*!"



2. From the drop-down menu, choose **Control Panel**.

About Discovery Framework settings

Many settings related to Discovery Framework can be adjusted from the **Discovery Framework Settings** section of the **Control Panel**.

Configurable settings include the following:

- **df.container:** Controls AJAX behavior of the Discovery Framework. Should not be edited.
- **df.defaultDataSource:** The name of the data source to use as the default.
- **df.maxExportBaseErrors:** The maximum allowable number of non-Analytics records that can be exported.
- **df.maxExportAnalyticsRecords:**The maximum allowable number of Analytics records that can be exported.
- **df.dataSourceDirectory:** The directory on disk from which to load data source definition files.
- **df.mdexStateManager:** The fully-qualified class name to use for the MDEX State Manager.
- **df.mdexSecurityManager:** The fully-qualified class name to use for the MDEX Security Manager.
- **df.cordaServerExternalUrl** and **df.cordaServerInternalUrl:** The externally and internally accessible URLs of the Corda Server, which is used by the **Chart** component.

The default values of these settings are created automatically upon first use. You cannot add or delete settings from the **Control Panel**—you can only edit them. Settings only appear after the feature(s) that use them have been executed at least once. For example, if you have never used the **Chart** component, the Corda Server URL settings will not appear.

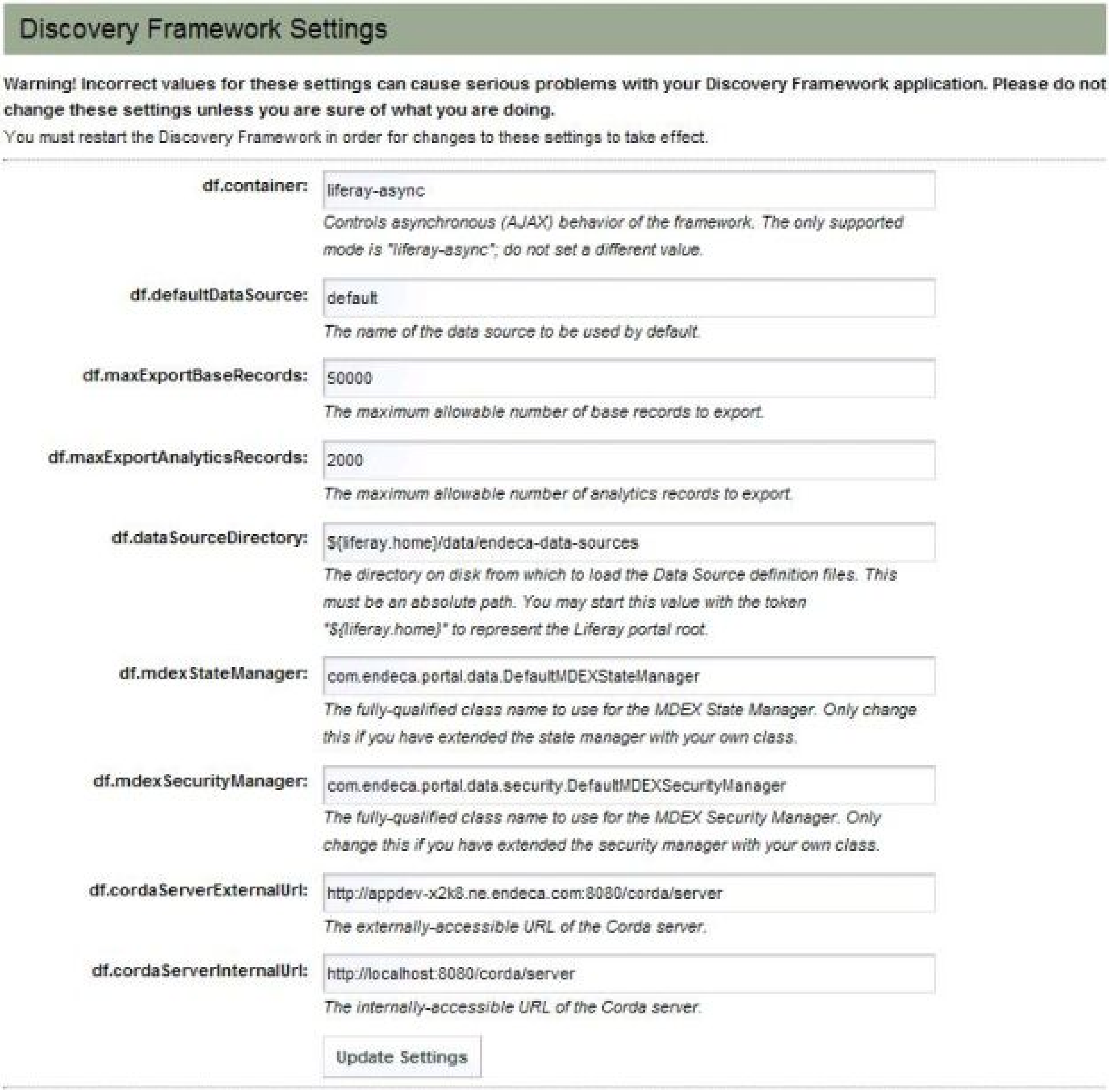
Modifying Discovery Framework settings

You modify Discovery Framework settings in the Control Panel.

Important: Take care when modifying these settings, as incorrect values can cause problems with your Discovery Framework application.

To modify Discovery Framework settings:

1. In the Discovery Framework, point the cursor at the **Dock** in the upper-right corner of the page. The **Dock** is labeled "Welcome <user name>!"
2. From the drop-down menu, choose **Control Panel**.
3. From the **Control Panel Portal** menu, choose **Discovery Framework Settings**.



Discovery Framework Settings

Warning! Incorrect values for these settings can cause serious problems with your Discovery Framework application. Please do not change these settings unless you are sure of what you are doing.

You must restart the Discovery Framework in order for changes to these settings to take effect.

df.container: liferay-async

Controls asynchronous (AJAX) behavior of the framework. The only supported mode is "liferay-async"; do not set a different value.

df.defaultDataSource: default

The name of the data source to be used by default.

df.maxExportBaseRecords: 50000

The maximum allowable number of base records to export.

df.maxExportAnalyticsRecords: 2000

The maximum allowable number of analytics records to export.

df.dataSourceDirectory: ${liferay.home}/data/endeca-data-sources

The directory on disk from which to load the Data Source definition files. This must be an absolute path. You may start this value with the token "${liferay.home}" to represent the Liferay portal root.

df.mdexStateManager: com.endeca.portal.data.DefaultMDEXStateManager

The fully-qualified class name to use for the MDEX State Manager. Only change this if you have extended the state manager with your own class.

df.mdexSecurityManager: com.endeca.portal.data.security.DefaultMDEXSecurityManager

The fully-qualified class name to use for the MDEX Security Manager. Only change this if you have extended the security manager with your own class.

df.cordaServerExternalUrl: http://appdev-x2k8.ne.endeca.com:8080/corda/server

The externally-accessible URL of the Corda server.

df.cordaServerInternalUrl: http://localhost:8080/corda/server

The internally-accessible URL of the Corda server.

4. Make your modifications and then click **Update Settings**.
5. Restart the Discovery Framework so your changes can take effect.

Note: If you do not see the **Discovery Framework Settings** in the **Control Panel**, it probably means you did not install the `endeca-framework-settings-portlet-1.2.war` file. Please review your installation settings.

Adding Endeca components

The Discovery Framework version 1.2 contains several Endeca components. These components make it possible for you to add only the desired Endeca functionality to your application.

To add an Endeca component to your Discovery Framework application:

1. Point the cursor at the **Dock** in the upper-right corner of the page.
2. In the drop-down menu, select **Add Component**.
   The**Add Component** dialog box opens.
3. In the **Add Component** dialog box, expand the Endeca category.
   A list of the available Endeca components appears.
4. Drag the components you want to include, one by one, into the main page layout.

About data sources

Every instance of a component that needs to query the MDEX Engine is backed by a particular data source. That data source, which represents a pointer to a specific MDEX Engine, is used to maintain application state for each user's session.

The different components in your Discovery Framework application can connect to different data sources.

Note: The Data Sources, Performance Metrics, and Bookmarks components do not require a backing data source.

Note: Upon installation, all components that require a backing data source are bound to a data source called `default.json`. This file does not exist by default; you must either manually create it under `endeca-portal\data\endeca-data-sources`, or switch to another data source.

About the sample MDEX Engine data sources

The Discovery Framework ships with sample data sources.

These sample data sources include configuration for the following:

- Host and port only.
- Host, port, and initial query state.
- Host, port, initial query state, and security filter configuration.

Adding data sources to the Discovery Framework

It is possible to add new data sources to the Discovery Framework. Your Endeca components can then access the data sources you have added.

To add a data source to the Discovery Framework:

1. Create a new JSON file in `endeca-portal\data\endeca-data-sources`.

For definition examples, see the sample data sources located in the same directory.

2. After creating the new file on disk, do one of the following:
   - Restart the Discovery Framework.
   - In the **Data Sources** component, click **Update data sources**.

Note: If your data source does not appear after completing step 2, it probably means that your data source contains invalid JSON syntax. You can confirm this by looking for a message about invalid syntax in the Discovery Framework log `df.log`. Check the log, edit your syntax, and try the steps above again.

Changing an Endeca component's data source

If more than one data source has been configured for the application, the data source for an individual portlet instance can be changed.

To change the data source for an Endeca component that can be bound to a data source:

1. In the header of the component whose data source you want to change, select the `...` icon, and then select **Preferences**.
2. Select the new data source in the drop down menu, and click **Update data source**.
   You should see that the component has been successfully bound to a new data source.
3. Click **Return to Full Page**.

   Note: This procedure only changes the data source for that single instance of the component.

Chapter 5

Other installation tasks

This section discusses some other installation tasks related to your Discovery Framework installation.

Using a different database

The Liferay portal server uses a relational database to store configuration and state, such as portlet preferences, user permissions, system settings, and more.

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

For instructions on switching to another supported database system, see the *Liferay Portal Administrator's Guide*. Keep the following details in mind:

- The Discovery Framework 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.
- Endeca has tested the Discovery Framework on MySQL and DB2. Other databases are expected to work but have not been explicitly tested.

Overview of switching to a different database

This topic provides a high-level overview of the steps involved in switching from the default Hypersonic database to the production RDBMS of your choice.

Note: Because the details vary from database to database, this topic only provides a high-level overview of this process. For detailed information, see the *Liferay Portal Administrator's Guide*.

To switch to a different database:

1. Install and verify that your database is working.
2. Create a new empty database or schema for the Liferay portal.
3. Create a database user for the Liferay portal.
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 Liferay application servers.
5. Stop Liferay if it is running.

6. Edit the `portal-ext.properties` file. In the JDBC section, comment out the settings for Hypsersonic, 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 the Discovery Framework. 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 Liferay is running, you may remove the liferay user's alter table privileges. Note you may have to add these back later if you upgrade Liferay or install components that require schema changes.

Installing Corda

The Chart component requires the installation of Corda charting software. Endeca recommends deploying Corda Server as a servlet when using it with the Discovery Framework. If you plan to use the Chart component, make sure to download `corda-1.2.zip` and deploy the Corda Server servlet in that package.

Related Links

Other installation tasks on page 39
This section discusses some other installation tasks related to your Discovery Framework installation.

About the Corda Server servlet on page 41
This topic describes the Corda servlet shipped with the Discovery Framework.

Obtaining the Corda software on page 41
You download the Corda software package, along with the rest of the Discovery Framework, from the EDeN downloads page.

Deploying the Corda Server servlet in an application server on page 41
The Corda Server servlet can be deployed by following the standard servlet deployment procedure for the application server in question.

Confirming the Corda Server servlet deployment on page 42
After deploying the Corda Server servlet, you should ensure that it is running.

Deploying PCXML templates on page 42
PCXML templates are XML-based templates that describe and define the charts and maps used by Corda. This topic describes how you can deploy the PCXML templates distributed with the Corda Server servlet.

Updating the Chart component with changes to the Corda Server on page 43
By default, the Chart component is configured to look for the local instance of the Corda Server, and fails if the servlet is not deployed.

Adding or removing PCXML templates on page 43
After initial deployment, you can add or remove PCXML templates.

Troubleshooting Corda on page 44
When attempting to render a chart, the Chart component may fail to reach the Corda Server at the specified host and port.

About the Corda Server servlet

This topic describes the Corda servlet shipped with the Discovery Framework.

The Corda Server servlet is a Java servlet version of the Corda Server. It is designed to run on, and be accessed through, Java-enabled application servers such as Tomcat and WAS. Because it is packaged as a servlet, you do not need to run this version of the Corda Server as a separate process over a separate server port.

For details on how the Corda Server servlet is packaged, see the *Corda documentation*.

Note: If you choose to deploy Corda as a server, rather than the recommended approach of deploying Corda as a servlet in an existing application server, see the Corda installation instructions in the *Endeca Platform Services Installation Guide*.

Important: Deploying the Corda servlet on the same Tomcat server as the Discovery Framework is intended for development purposes only. You should install Corda on a separate application server (or as a standalone server) for production use. If you purchased the Advanced Visualization for Java and .NET module, your license entitles you to run a single production instance of the Corda server (whether deployed as a servlet or as a standalone server).

Obtaining the Corda software

You download the Corda software package, along with the rest of the Discovery Framework, from the EDeN downloads page.

The Corda software is packaged in the `corda-1.2.zip` file.

Deploying the Corda Server servlet in an application server

The Corda Server servlet can be deployed by following the standard servlet deployment procedure for the application server in question.

Note: Hot-deploying the Corda Server servlet into Liferay's deploy directory is not supported.

Deploying on a Tomcat server:

1. Unzip the `corda-1.1.zip` file into Tomcat's `webapps` directory (such as `/path/to/tomcat-5.5.27/webapps`). This should make `corda.war` available at the top level of that directory.
2. Restart Tomcat.

Depending on your Tomcat configuration, the servlet container may unpack the `.war` archive, or it may operate directly from the archive. If you plan to modify Corda configuration files, or to deploy or modify PCXML chart templates, you may prefer to unpack the `corda.war` archive, to provide easier access to files inside the archive.

Deploying on WebSphere Application Server:

1. Unzip the `corda-1.1.zip` file to your hard drive to make the `corda.war` file available.
2. Use the IBM Integrated Solutions Console, Deployment Manager, or `wsadmin` utility to deploy `corda.war`. The servlet may be deployed as an enterprise application with context root `/corda`.

In all cases, the server log should display messages similar to the following example when the Corda Server servlet starts successfully:

```
Corda Server (PopChart) Version 6.0.727
  PopChart: Valid Key, OEM build for: ENDECA.
  OptiMap: No key entered, or key invalid.
  Highwire: No key entered, or key invalid.
  Cluster: No key entered, or key invalid.

Copyright 1997 - 2006, Corda Technologies, Inc. (www.corda.com) Protected
by U.S
. Patent 5,933,830. Other patents pending.

server_root: /Corda60
chart_root: chart_root
Cache Segment Size: 0
Password is Enabled, Required for Save
Maximum Threads: 64
Default Image Type is: Flash
Auto Detect PNG Support. Compression Mode: DEFAULT
```

Confirming the Corda Server servlet deployment

After deploying the Corda Server servlet, you should ensure that it is running.

To test that the Corda Server servlet is running and that your PCXML is deployed:

1. Access a URL similar to the following:

   ```
   http://localhost:8080/corda/server/?@_FILEapfiles/Bar.pcxml
   ```

2. Substitute the path (from chart root) to your PCXML file. In the example above, Corda is set to load `apfiles/Bar.pcxml`.
   If the chart loads, the server (and PCXML) are ready to use.

Deploying PCXML templates

PCXML templates are XML-based templates that describe and define the charts and maps used by Corda. This topic describes how you can deploy the PCXML templates distributed with the Corda Server servlet.

To deploy the PCXML templates, update the servlet with the new PCXML templates. The steps to update the servlet may differ, depending on the application server and configuration used when deploying the servlet. In all cases, this can be accomplished by updating `corda.war` with the required changes and repeating the steps in topic "Deploying the Corda Server servlet in an application server" to deploy the modified `.war` file.

The PCXML chart templates are located in the following location in `corda.war`: `WEB-INF/classes/Corda60/chart_root/apfiles`.

Note: Updating `corda.war` requires the use of an archiving tool to expand the archive file. This Java archive can be expanded and re-packaged with Java's jar tool or with a zip utility.

Note: Adding new PCXML chart templates requires updates to the Chart component to use the newly deployed PCXML files. For more information, see the topic "Adding or removing PCXML templates."

Updating the Chart component with changes to the Corda Server

By default, the Chart component is configured to look for the local instance of the Corda Server, and fails if the servlet is not deployed.

You can change where the Chart component looks for Corda, if you are installing the servlet on a non-localhost machine or you have a Corda server already running elsewhere.

By default, the Chart component is configured to use a Corda Server deployed as a servlet on the same application server as the Discovery Framework. This is a convenient configuration for single-server deployments and development and demonstration environments. However, production environments (especially those with clustered application servers) may require alternate configuration to specify a separate location for the Corda Server.

The internal and external hosts used by Corda Server differ in cases where Corda is deployed as a standalone server and cases where Corda is deployed as a servlet.

- When deployed as a servlet, the internal and external URL typically take a form similar to `http://server.example.com:[app server port]/corda/server`.
- When deployed as a standalone server, the internal and external hosts typically take a form similar to `http://cordaserver.example.com:[corda server port]/` (where the Corda server port differs for the internal and external URLs). For details, refer to the *Corda documentation*.

Configuring the location of the Corda Server

You can configure Corda Server location in the Discovery Framework's **Control Panel**.

Important: Because the **Control Panel** only shows settings that exist, in order to be able to edit the Corda Server URL properties, you must put the Chart component on a page first.

To configure the Corda Server URL:

1. Point the cursor at the **Dock** in the upper-right corner of the page.
2. In the drop-down menu, choose **Control Panel**.
3. In the **Portal** section of the **Control Panel** navigation panel, select **Discovery Framework Settings**.
4. Change the appropriate property—`df.cordaServerExternalUrl` or `df.cordaServerInternalUrl`—following the URL formatting guidelines above. For example:

   ```
   df.cordaServerInternalUrl = http://localhost:8080/corda/server
   ```

5. Click **Update Settings**.
6. Restart the Discovery Framework.

Adding or removing PCXML templates

After initial deployment, you can add or remove PCXML templates.

The `endeca-corda-chart-portlet-1.2.war` archive file is included in `components-1.1.zip`. This Java archive file can be expanded and re-packaged with Java's jar tool or with a zip utility.

To add or remove PCXML templates:

Update the `WEB-INF/analytics-portlet-config.xml` file in `endeca-corda-chart-portlet-1.1.war` and add or remove the CordaChartConfiguration elements.

For example, to add a new chart called 3D Pie with PCXML template `3DPie.pcxml`, update the file to include the following XML:

```
<bean id="3dPieChart" class="com.endeca.portlet.corda.CordaChartConfigu¬
ration">
  <property name="chartDisplayName" value="3D Pie" />
  <property name="pcxml" value="3DPie.pcxml" />
</bean>
```

This configuration causes the Chart component to display an additional option on its Preferences panel, allowing the use of 3D Pie as a chart style.

Troubleshooting Corda

When attempting to render a chart, the Chart component may fail to reach the Corda Server at the specified host and port.

If the host and port configuration is correct, you may need to configure Corda to allow connections from the application server hosts that are hosting the Discovery Framework.

To change the Corda configuration:

1. Update the `corda-web.war` file and redeploy the servlet to your application server.
2. Update the `WEB-INF/classes/Corda60/config/path.xml` configuration file in `corda-web.war` to include entries for the hosts that need to embed charts powered by the Corda Server.

   For example, entries like the following may be added to enable access from a specific host, from a range of domain names, or from a range of IP addresses, respectively:

   ```
   <PathMaps Version="1.0">
      …
      <Map Name="ValidDomain" Path="appserver.prod.example.com" Action="al¬
   lowDomain"/>
     <Map Name="ValidDomain" Path="*.prod.example.com" Action="allowDomain"/>

      <Map Name="ValidDomain" Path="192.168.*" Action="allowDomain"/>
   </PathMaps>
   ```

   Refer to *Corda's documentation* for details about this configuration file.

Uninstalling the Discovery Framework

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

If your 1.0 deployment was installed in an existing application server, follow the appropriate steps to stop and uninstall the DF 1.0 application from the application server before installing DF 1.1.

Note: You cannot have multiple versions of the Discovery Framework installed on the same machine simultaneously.

Index

A

accessing
- the Control Panel 33

adding
- data sources 36
- Endeca components 36
- PCXML templates 42

B

backing up your 1.1 files 11

C

Chart component
- changing Corda Server source 43

components
- changing data source for 37

components, in this version 9

configuring
- the Corda server location 43
- Tomcat 5.5 20

confirming
- Corda Server servlet deployment 42

Control Panel
- accessing 33
- adjusting logging level in 29

Corda
- adding PCXML templates 42
- confirming the servlet deployment 42
- deploying the servlet 41
- installing 40
- obtaining 41

Corda server
- configuring location 43

Corda, fixing connection problems
- Corda
  - troubleshooting 44
- troubleshooting
  - Corda 44

D

data source configuration
- deploying 25

data sources
- about 36
- adding 36
- changing for a component 37
- samples 36
- troubleshooting 36

databases
- switching 39

deploying
- components in WAS 26
- data source configuration 25

deploying Discovery Framework dependency libraries on WAS 22

deploying the Corda Server servlet 41

deploying the standalone portal WAR 23

directory
- Liferay Home 22

Discovery Framework
- about settings for 34
- deploying and starting on Tomcat 5.5 20
- downloading 15
- license key 17, 21, 26
- Linux installation steps 17
- migrating 11
- modifying logging 29
- modifying settings 34
- obtaining more information 13
- starting 33
- uninstalling 44
- updating your database schema 12
- Windows installation steps 16

downloading Corda software 41

E

edit controls
- turning on 33

Endeca components
- adding 36

Endeca theme
- deploying 26

entering the license key 17, 21, 26

F

features in this version 9

H

hot-deploying a plugin in Liferay 27

I

installing
- Discovery Framework 16
- Discovery Framework on Linux 17
- Discovery Framework on Tomcat 5.5 18

installing *(continued)*
  Discovery Framework on WebSphere Application Server 21
  Discovery Framework on Windows 16

L

Liferay
  deploy directories for WAS 27
  hot-deploying a plugin 27
Liferay Home
  about 22
Liferay portal
  accessing documentation for 13
log4j
  obtaining more information 32
log4j.properties files 32
logging
  setting up 31
  ways to modify 29
logging verbosity
  adjusting 29

M

manually deploying generated .war files in WAS 27
MDEX Engine
  sample data sources 36
migrating
  from version 1.1 12
modifying
  PCXML templates 42
  portal-log4j-ext.xml 31

O

obtaining additional information 13
obtaining the Discovery Framework software 15
overview
  of deploying on WAS 22
  of switching databases 39
  Tomcat 5.5 deployment 18
overview of package 9

P

package overview 9
PCXML
  adding templates 43
  removing templates 43
PCXML templates
  adding 42
portal-ext.properties
  configuring for WAS 24
  editing for WebSphere 24
  updating the application with 24
  uploading to WAS 25
portal-ext.properties, example for WAS 25
portal-log4j-ext.xml
  about 31
prerequisites 10, 11
previous version of Discovery Framework 12

R

rebuilding your application 12
relational database, changing
  database
    changing 39
  Hypersonic
    replacing with another database 39

S

sample data sources 36
setting up log4j 31
settings
  about 34
  modifying 34
standalone portal WAR
  deploying 23
starting the Discovery Framework 33
system requirements 10, 11

T

themes
  controlling 26
Tomcat 5.5
  deploying and starting on 20
  deploying the dependency libraries 18
  high-level deployment overview 18
  installing 18
  installing Discovery Framework on 18
  modifying configuration 20
troubleshooting
  WAS deployment 28
troubleshooting data sources 36
turning on edit controls 33

U

uninstalling
  the Discovery Framework 44
updating your database schema for 1.2 12
upgrading from a previous version 12
upgrading your installation 12

V

verbosity
  logging 29

W

WAS
- configuring portal-ext.properties for 24
- deploying components in 26
- deploying dependency libraries 22
- high-level deployment overview 22
- Liferay deploy directories 27
- manually deploying generated .war files 27
- manually uploading portal-ext.properties to 25

WAS *(continued)*
- starting the application 26
- troubleshooting 28

webSphere
- editing portal-ext.properties 24

WebSphere Application Server
- installing 21

wsadmin
- using to deploy 27