

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
Interaction with Liferay Portal.....	9
System requirements.....	10
Changing the JavaScript time-out value on Internet Explorer 8.....	11
Compatibility with other Endeca products.....	11
Upgrading from a previous version of the Discovery Framework.....	12
Obtaining more information.....	12
 Chapter 2: Installing the Discovery Framework.....	 13
Installing the Discovery Framework package.....	13
Downloading the Endeca Discovery Framework software for the full installation.....	13
Installing the Windows Tomcat bundle.....	14
Installing the Linux Tomcat bundle.....	19
Installing the Discovery Framework on Tomcat 5.5.....	21
Installing the Discovery Framework on the WebSphere Application Server version 6.1.....	25
Installing the Discovery Framework on the WebSphere Application Server version 7.....	33
Discovery Framework interaction with IAP 6.1.4.....	41
Downloading the Endeca Discovery Framework documentation.....	41
 Chapter 3: Getting started with the Discovery Framework.....	 43
Starting the Discovery Framework.....	43
Creating a default.json file.....	43
Accessing the Control Panel.....	44
About Framework Settings.....	44
Adding Endeca standard components	46
 Chapter 4: Other installation tasks.....	 49
Using a different database.....	49
Overview of switching to a different database.....	49
Installing Corda.....	50
Obtaining the Corda software.....	50
About the Corda Server servlet.....	50
About the Discovery Framework implementation of Corda.....	51
Deploying the Corda Server servlet in an application server.....	51
Updating the Chart component with changes to the Corda Server servlet.....	53
Troubleshooting Corda.....	55
Uninstalling the Discovery Framework.....	56



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.

Copyright © 2003-2011 Endeca Technologies, Inc. All rights reserved. Printed in USA.

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

Corda PopChart® and Corda Builder™ Copyright © 1996-2005 Corda Technologies, Inc.

Outside In® Search Export Copyright © 2008 Oracle. All rights reserved.

Rosette® Globalization Platform Copyright © 2003-2005 Basis Technology Corp. All rights reserved.

Teragram Language Identification Software Copyright © 1997-2005 Teragram Corporation. All rights reserved.

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 InFront, Profind, Endeca Navigation Engine, Don't Stop at Search, 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 7.035.864, US Patent 7.062.483, US Patent 7.325.201, US Patent 7.428.528, US Patent 7.567.957, US Patent 7.617.184, US Patent 7.856.454, Australian Standard Patent 2001268095, Republic of Korea Patent 0797232, Chinese Patent for Invention CN10461159C, Hong Kong Patent HK1072114, European Patent EP1459206, European Patent EP1502205B1, and other patents pending.

Endeca Discovery Framework Installation Guide • March 2011

Version 1.5

Preface

Endeca® Latitude applications guide people to better decisions by combining the ease of search with the analytic power of business intelligence. Users get self-service access to the data they need without needing to specify in advance the queries or views they need. At the same time, the user experience is data driven, continuously revealing the salient relationships in the underlying data for them to explore.

The heart of Endeca's technology is the MDEX Engine.™ The MDEX Engine is a hybrid between an analytical database and a search engine that makes possible a new kind of Agile BI. It provides guided exploration, search, and analysis on any kind of information: structured or unstructured, inside the firm or from external sources.

Endeca Latitude includes data integration and content enrichment tools to load both structured and unstructured data. It also includes the Discovery Framework, a set of tools to configure user experience features including search, analytics, and visualizations. This enables IT to partner with the business to gather requirements and rapidly iterate a solution.

About this guide

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

The Discovery Framework 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.

The Discovery Framework is easy to deploy and ideal for the agile development of enterprise-quality applications. Due to component-based nature of the Discovery Framework, 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.

The Discovery Framework consists of an enterprise-class portal framework and a library of UI components that embody best practices in Endeca applications. In addition, it includes a Component SDK, which 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.

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 Discovery Framework 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.

The Discovery Framework is easy to deploy and ideal for the agile development of enterprise-quality applications. Due to component-based nature of the Discovery Framework, 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. The Discovery Framework consists of an enterprise-class portal framework and a library of UI components that embody best practices in Endeca applications.

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*.

Interaction with Liferay Portal

The Discovery Framework is built upon the Liferay Portal Enterprise Edition.

Liferay Portal is an open-source JSR-286 portal technology. The Discovery Framework extends basic Liferay functionality to provide enhanced user management, security, and cross-component interaction, as well as performance-optimized communication with Endeca MDEX Engines.

This version of the Discovery Framework is built upon Liferay Portal 5.2 Enterprise Edition Service Pack 5.

System requirements

This version of the Endeca Discovery Framework has the following requirements:

Hardware requirements

The hardware requirements for the Discovery Framework 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 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 on Windows, Internet Explorer 8 (with compatibility mode disabled) on Windows



Tip: Firefox is recommended.



Important: Running Internet Explorer 8 in compatibility mode is not supported.



Note: The Discovery Framework supports Adobe Flash 10.0.

- Supported application servers: Tomcat 6, Tomcat 5.5, WebSphere Application Server (WAS) 6.1.0.1 or higher, 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 6.1 with IBM Java 5, WAS 7 with IBM Java 6



Important: For Sun Java 6, update 18 or greater is required.

- Supported database systems: MySQL 5.1, DB2 9.5

Alternative database support

The Liferay Portal server uses a relational database to store configuration and state. By default, Liferay uses Hypersonic, but this is not recommended for production use due to performance issues. Endeca tests the Discovery Framework on MySQL and DB2. 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 Discovery Framework code and can be reproduced on a supported database.

The *Discovery Framework Installation Guide* combined with the *Liferay Portal Administrator's Guide* provides detailed instructions on how to switch to another database system.

Changing the JavaScript time-out value on Internet Explorer 8

Internet Explorer 8 keeps track of the number of JavaScript lines executed. After a fixed value, the browser issues an error message, prompting the user to decide whether he or she would like to continue running the script. Because the Discovery Framework 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 the Discovery Framework on Internet Explorer 8, Endeca strongly recommends that you apply the fix outlined in this topic.

To change the script time-out 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`.

Compatibility with other Endeca products

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

This version of the Discovery Framework is compatible with the following Endeca components:

- MDEX Engine 6.1.x (up to 6.1.4) or 7.0.1 Early Access (see below)



Note: For best stability and performance, if you are using Discovery Framework with the MDEX Engine 6.1.3, ensure that you obtain the latest patches for that release.

- Platform Services 6.0.1 or 6.1
- Developer Studio 6.0.1 or 6.1
- Deployment Template 3.1 or 3.2



Important: The Discovery Framework is supported on Dgraph deployments only. It is not supported for use with the Agraph.

Upgrading from a previous version of the Discovery Framework

For information about upgrading from a previous version of the Discovery Framework, see the *Discovery Framework Migration Guide*.



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

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 Migration Guide*, which provides information on upgrading from earlier versions of the Discovery Framework.
- 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. It covers both full portal installations and component-only installations.

Installing the Discovery Framework package

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

There are five 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 6.1.
- Discovery Framework as a standalone application on WebSphere Application Server 7.



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 6 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.



Important: The Discovery Framework requires the Endeca Theme in order to start up. 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.

Downloading the Endeca Discovery Framework software for the full installation

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 <version>**.

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-<version>.zip` and `components-<version>.zip` to your development server.
 - To install the Tomcat 6 bundle for Linux, download `endeca-portal-<version>.tgz` and `components-<version>.zip` to your development server.
 - To install Discovery Framework for Tomcat 5.5 or the WebSphere Application Server version 6.1 or 7, download `endeca-portal-<version>.war.zip`, `endeca-portal-dependencies-<version>.zip`, and `components-<version>.zip`.



Important: If you are doing a component-only installation, see the separate download list in the section "Installing the components-only package."



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 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.



Note: Among the data sources in your Discovery Framework application, you 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. For details, see the topic "Specifying a default data source," located in the chapter "About data source configuration."

To install the Discovery Framework Tomcat bundle:

1. Unzip `endeca-portal-<version>.zip` to the directory of your choice.
The Discovery Framework creates a directory called `endeca-portal`. For example, if you unzip into `C:`, the Discovery Framework installs into `C:\endeca-portal`.
2. Extract the `.war` files from `components-<version>.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 `components-<version>.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=C:\path\to\endeca-portal\tomcat-6.0.29`. (If you do not have these environment variables set, you can leave them un-set.)
4. Install the Liferay license.
The instructions for obtaining and installing the license are provided later in this section.
5. Start the portal's Tomcat instance by running `endeca-portal\tomcat-6.0.29\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 the Discovery Framework is running.

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 the *Power User's Guide*.

Changing the context root for the Windows Tomcat bundle

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

To change the context root:

1. Rename `endeca-portal\tomcat-6.0.29\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`.
2. 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"/>`
3. Rename the `endeca-portal\tomcat-6.0.29\webapps\ROOT` directory to `endeca-portal\tomcat-6.0.29\webapps\<context root>`.
For multi-level context paths, use a multi-level path like the following:
`endeca-portal\tomcat-6.0.29\webapps\sales#east`.
4. 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. 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/
```

Installing the Liferay license

The Discovery Framework is built upon the Liferay Portal Enterprise Edition. Before you can start the Discovery Framework, you must install the Liferay license, which is available from the Endeca Developer Network (EDeN).

To install the license:

1. Download the license (Discovery Framework License) from the Discovery Framework section of EDeN (<http://eden.endeca.com>).
For details on navigating to this section, see the topic "Downloading the Endeca Discovery Framework software for the full installation."
2. Save the file to the `endeca-portal/deploy` directory of your Discovery Framework installation.

When you start the Discovery Framework, the license is installed.

Running the Discovery Framework as a Windows service

If you have installed the Windows Tomcat bundle, then you can run the Discovery Framework as a Windows service.

About running Discovery Framework as a Windows service

Running Discovery Framework as a Windows service requires the Tomcat service installer files.

The Discovery Framework 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 6.0.29.

To obtain the files and add them to the Discovery Framework:

1. From the Apache Tomcat website (<http://tomcat.apache.org>), download the Tomcat file `apache-tomcat-6.0.29-windows-x86.zip`.
If you are not using the bundled JVM, and your JVM is 64-bit, then you must download `apache-tomcat-6.0.29-windows-x64.zip`. This is the 64-bit version of the Tomcat download.
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 the Discovery Framework:

```
endeca-portal\tomcat-6.0.29\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 Discovery Framework installation
- Increase the memory allocation. The Discovery Framework 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 Discovery Framework installation. For example:

```
set SERVICE_NAME=DF15
set PR_DISPLAYNAME=Discovery Framework 1.5
```

4. Next, find the following line:

```
set PR_DESCRIPTION=Apache Tomcat 6.0.29 Server -
http://tomcat.apache.org/
```

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

```
set PR_DESCRIPTION=Endeca Discovery Framework server, version 1.5
```

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" --Jvms 128 --Jvmmx 256
```

7. Replace the last part of the line:

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

with:

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

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

8. Save and close the file.

9. Rename the Tomcat monitor file (`tomcat6w.exe`) to be `<value of SERVICE_NAME> w.exe`.

For example, if you set `SERVICE_NAME=DF15` in `service.bat`, then you must rename `tomcat6w.exe` to `DF15w.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 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 Discovery Framework service:

1. From the command line, navigate to the Discovery Framework Tomcat bin directory.
`endeca_portal\tomcat-6.0.29\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.
 - 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-6.0.29\jre1.6.0_21\win\bin\server\jvm.dll`.
 If you are not using the bundled JVM, then set the path to your JVM.
 - e) Click **OK**.
4. When you install the service, it is set up to be started manually. To configure the service to start automatically:
 - a) Display the **Services** list (**Control Panel > Administrative Tools > Services**).
 - b) In the list, double-click the Discovery Framework service.
 The properties dialog for the service is displayed.
 - c) From the **Startup type** drop-down, 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-6.0.29\logs\jakarta_service*.log`) for the relevant messages.

If the service will not start, try the following:

1. Make sure that you have used the correct version of the Tomcat download:
 - Tomcat version 6.0.29
 - For a 32-bit JVM (including the bundled JVM), `apache-tomcat-6.0.29-windows-x86.zip`
 - For a 64-bit JVM, `apache-tomcat-6.0.29-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-6.0.29\jre1.6.0_21\win\bin\server\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, select **Java**.
 - d) Click the **Shutdown** tab.
 - e) From the **Mode** drop-down, select **Java**.
 - f) Click **OK**.

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 is embedded.



Note: Among the data sources in your Discovery Framework application, you 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. For details, see the topic "Specifying a default data source," located in the chapter "About data source configuration."

To install the Discovery Framework Tomcat bundle:

1. Extract `endeca-portal-<version>.tgz` to the directory of your choice.
2. Extract the `.war` files from `components-<version>.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 `components-<version>.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. Install the Liferay license.

The instructions for obtaining and installing the license are provided later in this section.

6. Start the portal's Tomcat instance by running `endeca-portal/tomcat-6.0.29/bin/start-up.sh`.



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. 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 the *Power User's Guide*.

Changing the context root in the Linux Tomcat bundle

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

To change the context root:

1. Rename `endeca-portal/tomcat-6.0.29/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`.

2. 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"/>`

3. Rename the `endeca-portal/tomcat-6.0.29/webapps/ROOT` directory to `endeca-portal/tomcat-6.0.29/webapps/<context root>`.

For multi-level context paths, use the multi-level path here:

`endeca-portal/tomcat-6.0.29/webapps/ROOT` directory to
`endeca-portal/tomcat-6.0.29/webapps/sales#east`.

4. 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. 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/
```

Installing the Liferay license

The Discovery Framework is built upon the Liferay Portal Enterprise Edition. Before you can start the Discovery Framework, you must install the Liferay license, which is available from the Endeca Developer Network (EDeN).

To install the license:

1. Download the license (Discovery Framework License) from the Discovery Framework section of EDeN (<http://eden.endeca.com>).
For details on navigating to this section, see the topic "Downloading the Endeca Discovery Framework software for the full installation."
2. Save the file to the `endeca-portal/deploy` directory of your Discovery Framework installation.

When you start the Discovery Framework, the license is installed.

Installing the Discovery Framework on Tomcat 5.5

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

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. 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: Among the data sources in your Discovery Framework application, you 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. For details, see the topic "Specifying a default data source," located in the chapter "Discovery Framework data sources" in the *Power User's Guide*.

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-<version>.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-saaaj_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
hsqldb.jar
jabstorb.jar
jackson-core-lgpl-1.7.2.jar
jackson-mapper-lgpl-1.7.2.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.encoding=UTF8 -Duser.timezone=GMT -Dorg.apache.catalina.loader.WebappClassLoader.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. To deploy the Discovery Framework in the root context, create a new file called `ROOT.xml` and place it in `endeca-portal/apache-tomcat-5.5.x/conf/Catalina/localhost/`. To deploy the Discovery Framework into any other context, create a new file called `<context root>.xml` and place it in `endeca-portal/apache-tomcat-5.5.x/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="" />`
- For a context of `/sales`: `<Context path="/sales"/>`
- For a context of `/sales/east`: `<Context path="/sales/east"/>`

5. Rename the `endeca-portal\apache-tomcat-5.5.x\webapps\ROOT` directory to `endeca-portal\apache-tomcat-5.5.x\webapps\<context root>`. For multi-level context paths, use the multi-level path here: `endeca-portal\apache-tomcat-5.5.x\webapps\ROOT` directory to `endeca-portal\apache-tomcat-5.5.x\webapps\mycompany\sales`.

6. 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 `endeca-portal-<version>.war.zip` into a temporary directory.
This zip file contains the `endeca-portal-<version>.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 `endeca-portal-<version>.war` into the `endeca-portal/apache-tomcat-5.5.x/webapps/ROOT` directory.
5. Copy the `portal-ext.properties` file from the temporary directory you created for the `endeca-portal-dependencies-<version>.zip` file in step 4 to the `endeca-portal` directory.
6. 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 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/
```
7. Under the `endeca-portal` directory, create a `data` directory, and then create an `endeca-data-sources` directory below that.
8. 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-<version>.zip` file in a previous step.
9. Install the Liferay license.
The instructions for obtaining and installing the license are provided later in this section.
10. 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. Do not shut down the Tomcat window while the Discovery Framework is running.

11. 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

Installing the Liferay license

The Discovery Framework is built upon the Liferay Portal Enterprise Edition. Before you can start the Discovery Framework, you must install the Liferay license, which is available from the Endeca Developer Network (EDeN).

To install the license:

1. Download the license (Discovery Framework License) from the Discovery Framework section of EDeN (<http://eden.endeca.com>).
For details on navigating to this section, see the topic "Downloading the Endeca Discovery Framework software for the full installation."
2. Save the file to the `endeca-portal/deploy` directory of your Discovery Framework installation.

When you start the Discovery Framework, the license is installed.

Installing the Discovery Framework on the WebSphere Application Server version 6.1

You can deploy the Discovery Framework as a standalone application on WebSphere Application Server (WAS) version 6.1. (6.1.0.1 or higher).

Before following the steps here, consult the *Liferay Portal Administrator's Guide*, which contains portal deployment instructions and examples for WebSphere Application Server 6.1.



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.



Note: Among the data sources in your Discovery Framework application, you 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. For details, see the topic "Specifying a default data source," located in the chapter "About data source configuration."



Note: When the Discovery Framework is running on WAS 6.1 on Linux, some of its administrative components require the `compat-libstdc++-33` library to function. If this library is not already installed, you can install it with a command such as `yum install compat-libstdc++-33`. For more information, see the [IBM WebSphere documentation](#).

High-level overview of WebSphere Application Server 6.1 deployment

This topic provides an overview of the steps you need to take to deploy the Discovery Framework on WAS 6.1.

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

To deploy the Discovery Framework on WAS 6.1:

1. Deploy dependency .jar files. The exact list of required files and destination directories 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 "Discovery Framework data sources" in the *Power User's Guide*.
6. Install the Endeca theme, portlet components, and other framework .war files.
7. Install the Liferay license.
The instructions for obtaining and installing the license are provided later in this section.
8. Start the Discovery Framework enterprise application.
9. Optionally, repeat step 6 for any additional plugins you want to add.

Deploying Discovery Framework dependency libraries on WAS 6.1

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 on WAS 6.1:

1. Unzip the .jar files found in `endeca-portal-dependencies-<version>.zip`.
2. Upload the following list of 33 .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
ccpp.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
hsqldb.jar
jabstorb.jar
jackson-core-lgpl-1.7.2.jar
jackson-mapper-lgpl-1.7.2.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. Do one of the following, depending on the version of the MDEX Engine you are using with your Discovery Framework application:
 - If you are using MDEX 6 data sources, move the file `portlet.jar` from the WAS server's external library directory (`/usr/local/WAS/AppServer/lib/ext/` in the example above) to `/usr/local/WAS/AppServer/java/jre/lib/ext/`.
 - If you are using MDEX 7 data sources, move the following list of `.jar` files from the WAS server's external library directory (`/usr/local/WAS/AppServer/lib/ext/` in the example above) to `/usr/local/WAS/AppServer/java/jre/lib/ext/`:

```
cxfr-2.2.8.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
jaxb-api-2.1.jar
jaxb-impl-2.1.7.jar
portlet.jar
wsdl4j-1.6.2.jar
XmlSchema-1.4.3.jar
```

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

Extracting the standalone portal WAR on WAS 6.1

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

To extract the standalone portal WAR on WAS 6.1:

1. Unzip `endeca-portal-<version>.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 and then save it to the location of your choice.
3. Note the location of the `endeca-portal-<version>.war` file, as you will need to navigate to it in the next step.

Deploying the standalone portal WAR on WAS 6.1

After downloading and extracting the necessary files, you can deploy the Discovery Framework as an enterprise application in WebSphere Application Server, and then install components, 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 by using tools like `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 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 version 6.1:

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 > Install New Application**.
4. Click to browse to and select the Endeca Discovery Framework WAR you downloaded earlier (`endeca-portal-<version>.war`).
5. Set the context root to the desired path for your Discovery Framework installation.



Note: Make a note of your context root, as you will need to reference it several times during the deployment process.

6. Select **Show me all installation options and parameters**, and then click **Next**.
7. In the **Choose to generate default bindings and mappings** section, check **Generate Default Bindings**, and then select the following options:
 - **Override existing bindings**
 - **Use default virtual host name for Web and WIP modules** (enter `default_host` in the text field)

Click **Next**.

8. In the **Select installation options** step, the default application name is **endeca-portal-<version>_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**.

9. Keep the defaults for all other options and then click **Finish**.
10. Wait for installation and, if it is successful, click **Save directly to master configuration**.

Creating the Liferay Home directory for WAS 6.1

The remaining instructions in this section 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`

Editing the portal-ext.properties file for WAS 6.1 deployment

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

Endeca's default version of `portal-ext.properties` is included in the package `endeca-portal-dependencies-<version>.zip`.

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. Replace ${liferay.home}
# with the appropriate directory path, such as /home/endeca.
#
auto.deploy.dest.dir=${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
```



Note: 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.

2. 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/
```

3. Save the file.

Configuring portal-ext.properties for WAS 6.1 deployment

After you edit your `portal-ext.properties` file, there are two ways to deploy it in WAS 6.1.

- 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 on WAS 6.1

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-<version>.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 portal-ext.properties to Liferay Home on the server on WAS 6.1

After you create the `portal-ext.properties` file, you can manually upload it to WAS 6.1.

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.

Example settings for portal-ext.properties on WAS 6.1

The Endeca installation includes a default version of `portal-ext.properties`.

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 6.1. However, the following additional settings (which you must add to the file as described in the topic "Editing the portal-ext.properties file for WAS 6.1 deployment") are important for portlet deployment on WAS:

```
auto.deploy.dest.dir=/home/endeca/liferay/websphere-deploy
ehcache.portal.cache.manager.jmx.enabled=false
```

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 on WAS 6.1

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-<version>.zip` file you downloaded.

To deploy Endeca data source configuration:

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

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

Starting the application on WAS 6.1

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 > Enterprise Application** 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.
4. Restart WAS 6.1.

Deploying components and other plugins in WAS 6.1

This topic describes how to deploy components, themes, hooks, and other plugins in WAS 6.1.

These plugins are located in the `components-<version>.zip` package.

About Liferay component pre-processing on WAS 6.1

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.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 the Discovery Framework to load the newly deployed portlet.

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

Deploying components in WAS 6.1

Liferay must pre-process Discovery Framework components before they can be deployed in WAS 6.1.



Important: The Discovery Framework requires the Endeca Theme in order to start up. 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 Discovery Framework components in WAS 6.1:

1. Copy all component `.war` files 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 step 2 as modules 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 options appear in the following topics.

Deploying generated .war files on WAS 6.1 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.



Important: The procedure in this topic does not work in versions of WAS prior to 6.1.0.9, due to a known problem in WebSphere Application Server (for details, see <http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg1PK36365>). If you are using a service pack prior to 6.1.0.9, use `wsadmin` to deploy the generated .war file, as documented in the next topic.



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

To manually deploy a generated .war file:

1. Go to **Applications > 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 `/home/endeca/liferay/websphere-deploy/endeca-navigation-portlet.war`. Click **Ok**.
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 on WAS 6.1

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 `/home/endeca/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"
```


Installing the Liferay license

The Discovery Framework is built upon the Liferay Portal Enterprise Edition. Before you can start the Discovery Framework, you must install the Liferay license, which is available from the Endeca Developer Network (EDeN).

To install the license:

1. Download the license (Discovery Framework License) from the Discovery Framework section of EDeN (<http://eden.endeca.com>).
For details on navigating to this section, see the topic "Downloading the Endeca Discovery Framework software for the full installation."
2. Save the file to the `endeca-portal/deploy` directory of your Discovery Framework installation.

When you start the Discovery Framework, the license is installed.

Troubleshooting WAS 6.1 deployment

This topic discusses an issue to keep in mind when deploying the Discovery Framework on WAS 6.1.

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.

Installing the Discovery Framework on the WebSphere Application Server version 7

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

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. 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.



Note: Among the data sources in your Discovery Framework application, you 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. For details, see the topic "Specifying a default data source," located in the chapter "About data source configuration" in the *Discovery Framework Power User's Guide*.

High-level overview of WebSphere Application Server 7 deployment

This topic provides an overview of the steps you need to take to deploy the Discovery Framework on WAS 7.

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 and destination directories 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 other framework .war files.
7. Install the Liferay license.
The instructions for obtaining and installing the license are provided later in this section.
8. Start the Discovery Framework enterprise application.
9. Optionally, repeat step 6 for any additional plugins you want to add.

Deploying Discovery Framework dependency libraries on WAS 7

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-<version>.zip`.
2. Upload the following list of 32 .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
hsqldb.jar
jabsorb.jar
jackson-core-lgpl-1.7.2.jar
jackson-mapper-lgpl-1.7.2.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.

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 `endeca-portal-<version>.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 and then save it to the location of your choice.
3. Note the location of the `endeca-portal-<version>.war` file, as you will need to navigate to it in the next step.

Deploying the standalone portal WAR on WAS 7

After downloading and extracting 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.

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.

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 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 Endeca Discovery Framework WAR you downloaded earlier (`endeca-portal-<version>.war`), and then click **Next**.
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**.

- By default, the application name is **endeca-portal-*<version>*_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**.

- In **Map modules to servers**, accept the default settings and click **Next**.
- In **Map context roots for Web modules**, set the context root to the desired path for your Discovery Framework 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.

- In **Install New Application**, confirm that your settings are correct and then click **Finish**.
- Wait for installation and, if it is successful, click **Save directly to master configuration**.

Creating the Liferay Home directory on WAS 7

The remaining instructions in this section 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`

Editing the portal-ext.properties file for WAS 7 deployment

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

Endeca's default version of `portal-ext.properties` is included in the package `endeca-portal-dependencies-<version>.zip`.

- 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
#
# Set this to true to enable JMX integration in
# com.liferay.portal.cache.EhcachePortalCacheManager.
#
ehcache.portal.cache.manager.jmx.enabled=false
```



Note: 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.

- 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/
```

- Save the file.

Configuring portal-ext.properties for WAS 7 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 on WAS 7

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:

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

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

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.

Example settings for portal-ext.properties on WAS 7

Endeca's default version of `portal-ext.properties` is included in the package `endeca-portal-dependencies-<version>.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 7. However, the following additional settings (which you must add to the file as described in the topic "Editing the `portal-ext.properties` file for WAS 7 deployment") are important for portlet deployment on WAS:

```
auto.deploy.dest.dir=/home/endeca/liferay/websphere-deploy
ehcache.portal.cache.manager.jmx.enabled=false
```

Deploying Endeca data source configuration on WAS 7

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-<version>.zip` file you downloaded.

To deploy Endeca data source configuration:

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

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

Starting the application on WAS 7

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.
4. Restart WAS 7.

Deploying components and other plugins in WAS 7

This section describes how to deploy components, themes, hooks, and other plugins in WAS 7.

These plugins are located in the `components-<version>.zip` package.

About Liferay component pre-processing in WAS 7

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.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 the Discovery Framework to load the newly deployed portlet.

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

Deploying components in WAS 7

Liferay must pre-process Discovery Framework components before they can be deployed in WAS 7.



Important: The Discovery Framework requires the Endeca Theme in order to start up. 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 Discovery Framework components in WAS 7:

1. Copy all component `.war` files 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 step 2 as modules 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 options appear in the following topics.

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** 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 `/home/endeca/liferay/websphere-deploy/endeca-navigation-portlet.war`. Click **Ok**.

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 on WAS 7

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, where the command is executed from the Liferay deploy output directory (that is, the directory containing the `endeca-navigation-portlet.war` file):

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

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`.
- The command is executed in `/home/endeca/liferay/websphere-deploy/..`

Installing the Liferay license

The Discovery Framework is built upon the Liferay Portal Enterprise Edition. Before you can start the Discovery Framework, you must install the Liferay license, which is available from the Endeca Developer Network (EDeN).

To install the license:

1. Download the license (Discovery Framework License) from the Discovery Framework section of EDeN (<http://eden.endeca.com>).
For details on navigating to this section, see the topic "Downloading the Endeca Discovery Framework software for the full installation."
2. Save the file to the `endeca-portal/deploy` directory of your Discovery Framework installation.

When you start the Discovery Framework, the license is installed.

Troubleshooting WAS 7 deployment

This topic discusses an issue to keep in mind when deploying the Discovery Framework on WAS 7.

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.

Discovery Framework interaction with IAP 6.1.4

The Discovery Framework works with versions 6.1.3 and 6.1.4 of the MDEX Engine, but must be configured to take advantage of the new features in version 6.1.4.



Note: All MDEX Engines backing a specific installation should be either 6.1.3 or 6.1.4.

Configuring the Discovery Framework on Tomcat to work with IAP 6.1.4

This topic describes how to configure a Discovery Framework implementation running on Tomcat to work with IAP 6.1.4.

To configure the Discovery Framework on Tomcat to work with IAP 6.1.4:

Copy the 6.1.4 `endeca_navigation.jar` file to the `tomcat-<version>/lib/ext` directory. This overwrites the existing version of `endeca_navigation.jar`.

All 6.1.4 MDEX Engine features automatically work in the Discovery Framework. However, there is no built-in support for the 6.1.4 Presentation API features. To expose the new features, you have to add your own `QueryFunction` using the `QueryFunction` extension point, or manipulate the `ENEQuery` directly within the component.

Configuring the Discovery Framework on WAS to work with IAP 6.1.4

This topic describes how to configure a Discovery Framework implementation running on WebSphere Application Server to work with IAP 6.1.4.

To configure the Discovery Framework on WAS to work with IAP 6.1.4:

Copy the 6.1.4 `endeca_navigation.jar` file to the `WAS/AppServer/lib/ext` directory. This overwrites the existing version of `endeca_navigation.jar`.

All 6.1.4 MDEX Engine features automatically work in the Discovery Framework. However, there is no built-in support for the 6.1.4 Presentation API features. To expose the new features, you have to add your own `QueryFunction` using the `QueryFunction` extension point, or manipulate the `ENEQuery` directly within the component.

Downloading the Endeca Discovery Framework documentation

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

To download Discovery Framework documentation:

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 <version>**.

The **Product Download** page contains links to all available Discovery Framework packages.

- 3. Download the **Discovery Framework Documentation** package to your desired location.



Note: The Discovery Framework Documentation package is distributed as a zip file.

- 4. Extract the Discovery Framework Documentation zip file using a decompression utility. Optionally, you can extract the files to the `endeca-portal\doc` directory that was created when you installed the Discovery Framework.



Note: You can also view a searchable version of the Discovery Framework documentation online in the EDeN Knowledge Base.



Chapter 3

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`.

Getting started checklist

Keep the following points in mind when you start up the Discovery Framework:

- Before logging in to the Discovery Framework for the first time, make sure you have a data source in place. For more information, see the topic "Creating a `default.json` file" later in this section.
- Make sure to edit the host and port settings in `default.json` to match your configuration.

Creating a `default.json` file

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 in the **Framework Settings** component.

This task leads you through creating a `default.json` file based on the simplest possible model.

To create a `default.json` file:

1. From the `endeca-portal\data\endeca-data-sources` directory, open the `simple-data-source.json.sample` file in a text editor.
2. In the file, edit the server and port lines to match your data source. For example:

```
{  
  "server": "localhost",
```

```
"port": "8080"
}
```

3. Save the file as `default.json` in the `endeca-portal\data\endeca-data-sources` directory.

After you have established your `default.json` file, you can log on to the Discovery Framework and begin building your application by adding components.

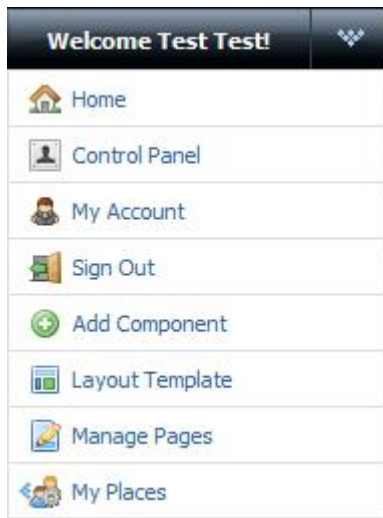
Accessing the Control Panel

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 Framework Settings

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

Configurable settings include the following:

- **df.cordalmageAuthEnabled:** Controls whether Corda chart image URLs are secured.



Note: The Corda image authorization feature is in Beta for this release and is not supported.

- **df.cordaServerExternalUrl** and **df.cordaServerInternalUrl**: The externally and internally accessible URLs of the Corda Server, which is used by the **Chart** component.
- **df.cordaServerRedirectorUrl**: The URL of the Corda Redirector, which can be used by the **Chart** component.
- **df.dataSourceDirectory**: The directory on disk from which to load data source definition files.
- **df.deepLinkPortletName**: The name of the deep link component.
- **df.defaultDataSource**: The name of the data source to use as the default.
- **df.defaultExporter**: The default exporter class.
- **df.exportPortletName**: The default name of the export portlet.
- **df.healthCheckTimeout**: The time (in milliseconds) for query timeout when checking data source availability on initialization.
- **df.maxExportAnalyticsRecords**: The maximum allowable number of Analytics records that can be exported.
- **df.maxExportBaseErrors**: The maximum allowable number of non-Analytics records that can be exported.
- **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.

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 Framework Settings

You modify **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 **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. In the **Discovery Framework** section of the **Control Panel Portal** menu, choose **Framework Settings**.

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.cordaimageAuthEnabled:	false	<i>Image Authorization (BETA): Whether to secure Corda chart image URLs so they're only accessible to the user who requested them</i>
df.cordaServerExternalUrl:	http://appdev-x2k8.ne.endeca.com:8080/corda/server	<i>The externally-accessible URL of the Corda server.</i>
df.cordaServerInternalUrl:	http://localhost:8080/corda/server	<i>The internally-accessible URL of the Corda server.</i>
df.cordaServerRedirectorUrl:	http://appdev-x2k8.ne.endeca.com:8080/endeca-corda-chart-portlet/ctRedirect	<i>The URL of the Corda Redirector.</i>
df.dataSourceDirectory:	\${liferay.home}/data/endeca-data-sources	<i>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.</i>
df.deepLinkPortletName:	endecadeeplinkportlet_WAR_endecadeeplinkportlet	<i>The name of the deep link portlet.</i>
df.defaultDataSource:	default	<i>The id of the data source to be used by default for new portlets.</i>
df.defaultExporter:	com.endeca.export.CSVExport	<i>The default exporter class.</i>
df.exportPortletName:	endeca-results-export-portlet_WAR_endeca-results-export-portlet	<i>This is the default name of the export portlet that can used with p_p_id.</i>
df.healthCheckTimeout:	5000	<i>The time, in milliseconds, query timeout when checking data source availability</i>

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 **Framework Settings** in the **Control Panel**, it probably means you did not install the `endeca-framework-settings-portlet-<version>.war` file. Please review your installation settings.

Adding Endeca standard components

The Discovery Framework contains several Endeca standard components. These components make it possible for you to add 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.



Chapter 4

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 Hypersonic (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 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 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-<version>.zip` and deploy the Corda Server servlet in that package.



Note: The Corda image authorization feature is in Beta for this release and is not supported.

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-<version>.zip` file.

About the Corda Server servlet

This topic describes the Corda Server 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 Corda charting module (also known as 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).

About the Discovery Framework implementation of Corda

All communication between the Discovery Framework and Corda occurs through the Corda Redirector. The location of the Corda Server servlet (or Corda Server) is configurable, allowing it to be deployed locally for development and remotely for production.

The Corda Redirector module is a proxy that allows the browser requests that embed generated chart images to be routed through the application server (in this case Liferay), rather than going directly to the Corda server. All requests for Corda images are made to the Corda Redirector module.

This new redirector-based deployment model ensures a greater level of control over network infrastructure. It enables system administrators to control network access, by allowing direct access *only* to the application server(s).

For more information about the Corda Redirector, see the [Corda documentation](#).

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.

Before deploying Corda, you must configure it. The following topics describe this process.



Note: Deploying the `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.

Extracting the corda-<version>.zip file

After you download the `corda-<version>.zip` file, you need to extract it to a temporary location in order to access the `corda.war` file.

To prepare the `corda.war` file for modifications:

1. Unzip the `corda-<version>.zip` file to your hard drive to make the `corda.war` file available.
2. Expand the `corda.war` file.

This Java archive can be expanded and re-packaged with Java's `jar` tool or with a zip utility.

Configuring Corda to allow connections from other hosts

The Corda Server servlet is set up by default to accept incoming requests from any host. You must configure it in order to allow only incoming requests from the application server that hosts the Discovery Framework.



Note: In the instructions below, the paths are relative to the location to which the `.war` was extracted.

To change the Corda configuration:

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="DefaultRead" Path=".*" Action="Load"/>
  <Map Name="DefaultSave" Path="./images/*" Action="Save"/>
  <Map Name="ValidDomain" Path="127.0.0.1" Action="allowDomain"/>
  <Map Name="ValidDomain" Path="localhost" Action="allowDomain"/>
  <Map Name="ValidDomain" Path="*" Action="allowDomain"/>
</PathMaps>
```

The last line, highlighted in bold, leaves Corda open to requests from anywhere. For this reason, you should replace "*" with the IP address (or range of addresses) allowed to access it (such as, for example, "192.168.*"). Specifically, you should add the location of the application server hosting the Discovery Framework.

Refer to [Corda's documentation](#) for details about this configuration file.

Adding or removing 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.



Note: In the instructions below, the paths are relative to the location to which the WAR was extracted.

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 the topic "Deploying the corda.war file" 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: Adding new PCXML chart templates requires updates to the Chart component to use the newly deployed PCXML files. For more information, see the topic "Configuring Chart to use PCXML templates."

Deploying the corda.war file

After configuring Corda, you must repackage the `.war` file and then deploy it.



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

Deploying on a Tomcat server:

1. Re-zip the modified `corda.war` file.
2. Deploy the file into Tomcat's `webapps` directory (such as `/path/to/tomcat-<version>/webapps`).
3. 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 further, 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. Re-zip the modified `corda.war` file.
2. Use the IBM Integrated Solutions Console, Deployment Manager, or `wsadmin` utility to deploy `corda.war`. The servlet should be deployed as an enterprise application with context root `/corda`.

Confirming the Corda Server servlet deployment

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

The server log should display messages similar to the following example when the Corda Server servlet starts successfully:

```
Corda Server (PopChart) Version 6.0.735
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
```

In addition, you can test the Corda Server servlet in a browser, as follows:

1. Access the following URL, using the name of your server rather than `server.example.com`:

```
http://server.example.com:8080/corda/server
```

A empty white box with a gray background displays.

2. Confirm that the title bar says **Corda Server (PopChart) Servlet Version 6.0.735**.
3. Access the following URL, using the name of your server rather than `server.example.com`:

```
http://server.example.com:8080/endeca-corda-chart-portlet/ctRedirect
```

The same empty white box and title bar mentioned above should appear.

Updating the Chart component with changes to the Corda Server servlet

By default, the `Chart` component is configured to look for the local instance of the Corda Server servlet, 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 servlet 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. Update the internal, external, and redirector URLs—`df.cordaServerInternalUrl`, `df.cordaServerExternalUrl`, and `df.cordaServerRedirectorUrl`—to point to the location (host and port) of the Corda Server servlet. For example:

```
df.cordaServerInternalUrl = http://server.example.com:8080/corda/server
```
5. Click **Update Settings**.
6. Restart the Discovery Framework.

Configuring Chart to use PCXML templates

During deployment, you added PCXML templates to your Corda implementation. Now you can configure the **Chart** component to use these templates.

The `endeca-corda-chart-portlet-<version>.war` archive file is included in `components-<version>.zip`.

To configure the **Chart** component's PCXML templates:

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



Note: This Java archive file can be expanded and re-packaged with Java's `jar` tool or with a zip utility.

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.CordaChartConfiguration">
  <property name="chartDisplayName" value="3D Pie" />
</bean>
```

```
<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

The following section provides troubleshooting information for Corda.

Change in Corda behavior in Discovery Framework

Starting in Discovery Framework version 1.3.1, the Corda splash screen no longer renders upon entering the base URL.

In addition, URLs such as `@_FILE` URL are now restricted, to prevent cross-site scripting vulnerabilities.

Problems rendering a chart

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. For details, see the topic "Configuring Corda to allow connections from other hosts," which appears earlier in this section.

Known issue with JVM crashes related to Corda Server servlet and JIT compilation

There is a known issue with JVM crashes that is related to the Corda Server servlet and JIT compilation on the latest version of Sun's JVM (specifically, version 1.6_21).

When this JVM crash occurs, the JVM error log typically notes that the active thread is a compiler thread, similar to the following:

```
Current thread (0x00002aab04077800):  JavaThread "CompilerThread1" daemon
[_thread_in_native, id=2353, stack(0x0000000041123000,0x0000000041224000)]
```

The compilation task being performed is related to the method `com.corda.b.dc.a()`, similar to the following:

```
Current CompileTask:
C2:4112      com.corda.b.dc.a(B)Z (559 bytes)
```

This issue can be resolved by disabling JIT compilation with the following JVM argument:

```
-XX:CompileCommand=exclude,com/corda/a/dc,a -XX:CompileCommand=ex-
clude,com/corda/b/dc,a
```

There is an expected, albeit minimal, performance degradation associated with disabling JIT compilation. You should be aware of the change and may want to perform additional testing to ensure adequate performance.



Note: This issue has not been reproduced on the latest version of Sun's 1.5 JVM.



Note: This issue is not known to exist on IBM's JVM and has not been reproduced on it.

Uninstalling the Discovery Framework

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

If your previous deployment was installed in an existing application server, follow the appropriate steps to stop and uninstall the earlier DF application from the application server before installing this version. For important details on Discovery Framework migration, see the topic "Upgrading from a previous version of the Discovery Framework," located in the *Discovery Framework Migration Guide*.



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

Index

6.1.4 IAP 41

- Tomcat configuration 41
- WAS configuration 41

A

- accessing
 - Control Panel 44
 - Liferay Portal documentation 12
- adding
 - Endeca components 46
 - PCXML templates 52

C

- Chart component
 - changing Corda Server source 53
 - updating with changes to Corda 53
 - using PCXML templates 54
- configuring
 - Corda server location 54
 - portal-ext.properties for WAS 6.1 29
 - portal-ext.properties for WAS 7 37
 - Tomcat 5.5 23
- confirming
 - Corda Server servlet deployment 53
- context root
 - changing for the Linux Tomcat bundle 20
 - changing for the Windows Tomcat bundle 15
- Control Panel
 - accessing 44
- Corda
 - about 50
 - adding PCXML templates 52
 - changing the source for a component 53
 - configuring Corda server location 54
 - configuring to allow connections from other hosts 51
 - confirming the servlet deployment 53
 - deploying corda.war 52
 - deploying the servlet 51
 - Discovery Framework implementation of 51
 - extracting the zip file 51
 - fixing connection problems 51, 55
 - image authorization Beta 50
 - installing 50
 - known issue with JVM crashes 55
 - obtaining 50
 - splash screen rendering changes 55
 - troubleshooting 51, 55
 - updating the Chart component with changes to 53
 - using with SSL 51

creating

- Liferay Home directory for WAS 6.1 28
- default.json 43
- Liferay home directory on WAS 7 36

D

- data source configuration
 - deploying on WAS 6.1 30
 - deploying on WAS 7 38
- database
 - changing from Hypersonic 49
 - process for switching 49
- default.json, creating 43
- deploying
 - components in WAS 6.1 31
 - components in WAS 7 38
 - components on WAS 6.1 31
 - components on WAS 7 39
 - Corda Server servlet 51
 - data source configuration on WAS 6.1 30
 - data source configuration on WAS 7 38
 - dependency libraries for Tomcat 5.5 22
 - Discovery Framework dependency libraries on WAS 6.1 26
 - Discovery Framework dependency libraries on WAS 7 34
 - Discovery Framework on Tomcat 5.5 24
 - generated .war files in WAS 6.1 using wsadmin 32
 - generated .war files in WAS 7 using wsadmin 40
 - generated .war files manually in WAS 6.1 32
 - generated .war files manually in WAS 7 39
 - standalone portal WAR on WAS 6.1 27
 - standalone portal WAR on WAS 7 35
- Discovery Framework
 - downloading 13
 - about settings for 44
 - deploying and starting on Tomcat 5.5 24
 - downloading documentation 41
 - installing Liferay license 16, 21, 25, 33, 40
 - installing on WAS 6.1 25
 - interaction with Liferay Portal 9
 - Linux installation steps 19
 - migrating from an earlier version 12
 - modifying settings 45
 - obtaining more information 12
 - options for installing 13
 - running as Windows service 16
 - starting 43
 - system requirements 10
 - uninstalling 56
 - upgrading 12
 - Windows installation steps 14

- documentation
 - Discovery Framework 41
 - Liferay Portal 12
- downloading
 - Corda software 50
 - Discovery Framework documentation 41
 - Discovery Framework software 13

E

- edit controls
 - turning on 44
- Endeca components
 - adding 46

H

- Hypersonic
 - replacing with another database 49

I

- IAP 6.1.4
 - Tomcat configuration 41
 - using the Discovery Framework with 41
 - WAS configuration 41
- installing
 - Corda 50
 - Discovery Framework on Linux 19
 - Discovery Framework on Tomcat 5.5 21
 - Discovery Framework on WAS 7 33
 - Discovery Framework on WebSphere Application Server 6.1 25
 - Discovery Framework on Windows 14
 - full Discovery Framework package 13
 - Liferay license 16, 21, 25, 33, 40
- Internet Explorer JavaScript time-out 11

J

- JavaScript time-out
 - on IE8 11
- JIT compilation, Corda JVM known problem 55
- JVM crashes in Corda 55

L

- license, installing Liferay 16, 21, 25, 33, 40
- Liferay Home directory
 - creating for WAS 6.1 28
 - creating for WAS 7 36
- Liferay Portal
 - accessing documentation for 12
 - component pre-processing on WAS 6.1 31
 - component pre-processing on WAS 7 39
 - installing the license 16, 21, 25, 33, 40
 - interaction with 9

- Linux Tomcat bundle
 - changing the context root 20
 - installing 19

M

- MDEX 6.1.4
 - using the Discovery Framework with 41
- migrating from an earlier version 12
- modifying
 - Discovery Framework settings 45
 - PCXML templates 52

O

- obtaining additional information 12
- obtaining the Discovery Framework software 13
- overview
 - Corda Server servlet 50
 - Discovery Framework package 9
 - switching databases 49
 - Tomcat 5.5 deployment 21
 - WAS 6.1 deployment 26
 - WAS 7 deployment 33

P

- package overview 9
- PCXML
 - adding templates 54
 - removing templates 54
- PCXML templates
 - adding 52
- portal-ext.properties
 - configuring for WAS 6.1 29
 - configuring for WAS 7 37
 - editing for WAS 6.1 29
 - editing for WAS 7 36
 - example for WAS 6.1 30
 - example for WAS 7 38
 - updating the WAS 6.1 application with 29
 - updating the WAS 7 application with 37
 - uploading to WAS 6.1 30
 - uploading to WAS 7 37
- prerequisites 10, 11

S

- settings
 - about 44
 - modifying 45
- standalone portal WAR
 - deploying on WAS 6.1 27
 - deploying on WAS 7 35
 - extracting on WAS 6.1 27
 - extracting on WAS 7 35
- starting the Discovery Framework 43
- system requirements 10, 11

T

- Tomcat 5.5
 - deploying and starting on 24
 - deploying the dependency libraries 22
 - high-level deployment overview 21
 - installing 22
 - installing Discovery Framework on 21
 - modifying configuration 23
- troubleshooting
 - Corda 51, 55
 - WAS 6.1 deployment 33
 - WAS 7 deployment 40

U

- uninstalling
 - Discovery Framework 56
- upgrading the Discovery Framework 12
- uploading
 - portal-ext.properties to WAS 6.1 30
 - portal-ext.properties to WAS 7 37

W

- WAS 6.1
 - configuring portal-ext.properties for 29
 - creating Liferay Home directory 28
 - deploying components 31
 - deploying components in 31
 - deploying data source configuration 30
 - deploying dependency libraries 26
 - deploying generated .war files manually 32
 - deploying generated .war files using wsadmin 32
 - deploying the standalone portal WAR 27
 - editing portal-ext.properties 29
 - example portal-ext.properties 30
 - extracting the standalone portal WAR 27
 - high-level deployment overview 26
 - installing Discovery Framework on 25
 - Liferay component pre-processing 31
 - starting the application 31

- WAS 6.1 (*continued*)
 - troubleshooting 33
 - updating to include portal-ext.properties 29
 - uploading portal-ext.properties to 30
- WAS 7
 - configuring portal-ext.properties for 37
 - creating Liferay Home directory 36
 - deploying components 39
 - deploying components in 38
 - deploying data source configuration 38
 - deploying dependency libraries 34
 - deploying generated .war files manually 39
 - deploying generated .war files using wsadmin 40
 - deploying the standalone portal WAR 35
 - editing portal-ext.properties 36
 - example portal-ext.properties 38
 - extracting the standalone portal WAR 35
 - high-level deployment overview 33
 - installing Discovery Framework on 33
 - Liferay component pre-processing 39
 - starting the application 38
 - troubleshooting 40
 - updating to include portal-ext.properties 37
 - uploading portal-ext.properties to 37
- WebSphere Application Server 6.1
 - installing Discovery Framework on 25
- WebSphere Application Server 7
 - installing Discovery Framework on 33
- Windows service
 - about 16
 - configuring 17
 - installing 18
 - installing Tomcat monitor 16
 - obtaining installer files 16
 - starting 18
 - troubleshooting 18
- Windows Tomcat bundle
 - changing the context root 15
 - installing 14
- wsadmin
 - using to deploy on WAS 6.1 32
 - using to deploy on WAS 7 40

