# Oracle Endeca Guided Search

## Getting Started Guide

## Version 6.2.2 • March 2012

# Contents

# Copyright and disclaimer

# Preface

Oracle Endeca's Web commerce solution enables your company to deliver a personalized, consistent customer buying experience across all channels — online, in-store, mobile, or social. Whenever and wherever customers engage with your business, the Oracle Endeca Web commerce solution delivers, analyzes, and targets just the right content to just the right customer to encourage clicks and drive business results.

Oracle Endeca Guided Search is the most effective way for your customers to dynamically explore your storefront and find relevant and desired items quickly. An industry-leading faceted search and Guided Navigation solution, Oracle Endeca Guided Search enables businesses to help guide and influence customers in each step of their search experience. At the core of Oracle Endeca Guided Search is the MDEX Engine,™ a hybrid search-analytical database specifically designed for high-performance exploration and discovery. The Endeca Content Acquisition System provides a set of extensible mechanisms to bring both structured data and unstructured content into the MDEX Engine from a variety of source systems. Endeca Assembler dynamically assembles content from any resource and seamlessly combines it with results from the MDEX Engine.

Oracle Endeca Experience Manager is a single, flexible solution that enables you to create, deliver, and manage content-rich, cross-channel customer experiences. It also enables non-technical business users to deliver targeted, user-centric online experiences in a scalable way — creating always-relevant customer interactions that increase conversion rates and accelerate cross-channel sales. Non-technical users can control how, where, when, and what type of content is presented in response to any search, category selection, or facet refinement.

These components — along with additional modules for SEO, Social, and Mobile channel support — make up the core of Oracle Endeca Experience Manager, a customer experience management platform focused on delivering the most relevant, targeted, and optimized experience for every customer, at every step, across all customer touch points.

## About this guide

This guide walks you through the process of setting up your Endeca implementation, based on a sample wine application.

**Important:** If you have purchased the Oracle Endeca Experience Manager, please read the *Oracle Endeca Experience Manager Getting Started Guide*.

The guide describes high-level tasks involved in installing the core packages that comprise Oracle Endeca Guided Search, provisioning and configuring components in the system, and using the Endeca Deployment Template to perform operational tasks, such as running updates.

This guide assumes that you have a basic understanding of Oracle Endeca Guided Search products and are familiar with basic Endeca concepts. For more information, see the *Oracle Endeca Guided Search Concepts Guide* and the *Endeca Glossary*.

Use the *Getting Started Guide* to get started with an Endeca project:

1. Read about the core Oracle Endeca Guided Search packages and the Endeca Deployment Template, and learn how to download and install them.

2. Next, run the Deployment Template scripts to provision and initialize a sample application on a single development server.
3. Then run the baseline update script and use the JSP reference implementation to explore the sample wine application.

This guide also contains information about additional Endeca packages, default Endeca variables and ports, and running the reference implementations for Java or ASP.NET.

For detailed installation instructions for each component, see the individual installation guides.

# Who should use this guide

This guide is for application developers who are building Endeca applications using Oracle Endeca Guided Search.

This guide is intended to help Endeca users through the process of downloading and installing their Endeca components. It provides a walk-through on how to set up a development environment and run the sample application.

# 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 Oracle Endeca Customer Support

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

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

Part 1

# Introduction

Chapter 1

# Introduction

This section provides an overview of Endeca and its components.

## Core installation packages

Oracle® Endeca® Guided Search consists of several core packages and several optional components. This guide focuses on working with the core packages and the Deployment Template.

Oracle Endeca Guided Search is comprised of the following core packages:

- Endeca MDEX Engine
- Endeca Platform Services
- Endeca Presentation API
- Oracle Endeca Workbench

Endeca includes many additional components, but this guide is an introduction to setting up the three core packages and using the Deployment Template to manage them.

### MDEX Engine overview

The Endeca MDEX Engine is the indexing and query engine that provides the backbone for all Endeca solutions.

The MDEX Engine uses proprietary data structures and algorithms that allow it to provide real-time responses to client requests. The MDEX Engine stores the indices that were created by the Endeca Information Transformation Layer (ITL). After the indices are stored, the MDEX Engine receives client requests via the application tier, queries the indices, and then returns the results.


The MDEX Engine is designed to be stateless. This design requires that a complete query be sent to the MDEX Engine for each request. The stateless design of the MDEX Engine facilitates the addition of MDEX Engine servers for load balancing and redundancy. Because the MDEX Engine is stateless, any replica of an MDEX Engine on one server can reply to queries independently of a replica on other MDEX Engine servers.

Consequently, adding replicas of MDEX Engines on additional servers provides redundancy and improved query response time. That is, if any one particular server goes down, a replica of an MDEX Engine provides redundancy by allowing other servers in the implementation to continue to reply to queries. In addition, total response time is improved by using load balancers to distribute queries to a replica MDEX Engine on any of the additional servers.

The MDEX Engine package contains the following components:

| MDEX Engine Component | Description |
|---|---|
| Dgraph | The Dgraph is the name of the process for the MDEX Engine.<br>A typical Endeca implementation includes one or more Dgraphs. Optionally, it can include an Agraph that manages a number of Dgraphs. |
| Agraph | The Agraph is the name of the program that runs in a distributed configuration in addition to the Dgraph. The Agraph typically resides on a separate machine.<br>The Agraph program is responsible for receiving requests from clients, forwarding the requests to the distributed Dgraphs, and coordinating the results. From the perspective of the Endeca Presentation API, the Agraph program behaves similarly to the Dgraph program.<br>Agraph-based implementations allow parallelization of query processing. The implementation of this parallelization results from partitioning the set of records into two or more disjoint subsets of records and then assigning each subset to its own Dgraph.<br>  **Note:** Starting with the MDEX Engine version 6.0, (namely, with installations on the 64-bit platforms) a more powerful Dgraph can accommodate much larger data sets without the need to implement an Agraph. |
| Dgidx | Dgidx is the indexing program that reads the tagged Endeca records that were prepared by Forge and creates the proprietary indices for the Endeca MDEX Engine. |
| Agidx | Agidx is the program that creates a set of Agidx indices which support the Agraph program in a distributed environment. |
| dgwordlist | The `dgwordlist` utility is used to manually compile the text-based `worddat` dictionary into the binary `spelldat` dictionary. This enables use of the Aspell dictionary module in the MDEX Engine. |
| enecerts | The Endeca `enecerts` utility creates the SSL certificates. |

## Platform Services overview

The Endeca Platform Services package consists of a number of components that are used to build Endeca applications in support of the Endeca MDEX Engine.

Two of the major components of the Endeca Platform Services package are the Endeca Information Transformation Layer (which includes Forge and other Data Foundry components) and the Endeca Application Controller (EAC). The following table lists the components that are available in the Platform Services installation package.

| Platform Services Component | Description |
|---|---|
| Endeca Application Controller (EAC) | The EAC components consist of the EAC Central Server (which coordinates the command, control, and monitoring of all Agents in an Endeca implementation), the EAC Agent (which controls the work of an Endeca implementation on a single host machine) and the EAC command-line utility, `eaccmd`. |
| Data Foundry | Consists of the Forge program and its related components, such as record adapters, record manipulators, dimension servers, property mappers, and so on. The Content Adapter Development Kit (CADK) is also installed. Note that the Dgidx program is not part of this package, but is available in the MDEX Engine installation package. |
| Logging and Reporting System | The Log Server and Report Generator, which (together with the Logging API) make up the Endeca Logging and Reporting System. |
| Reference Implementations | Sample Endeca applications that include a sample Developer Studio project (including source data and instance configuration files), as well as JSP and .NET user interface (front-end) applications. |
| emgr_update | A utility that lets you upload the instance configuration to Endeca Workbench and download it from Endeca Workbench. |
| Endeca Control System | The Endeca Job Control Daemon (JCD) and the Control Interpreter. These components control and administer Oracle Endeca Guided Search running on one or multiple host machines. The Endeca Control System should be installed on the machine that hosts the Endeca Platform Services. Note that the Control System is deprecated, and is not installed by default. |

## Presentation API overview

The Endeca Presentation API provides interfaces to the Endeca MDEX Engine and Log Server. The Endeca Presentation API must be installed on the machine that hosts the Web application server.

The API is available in two packages:

- The Presentation API for UNIX, which includes the Java version of the API only (JAR files, Javadoc, and Installation and Release Notes file.
- The Presentation API for Windows, which includes both the Java version of the API (JAR files, Javadoc, and Installation and Release Notes file) and also the .NET version of the API (DLL files, CHM Help, and Installation and Release Notes file).

## Oracle Endeca Workbench overview

Oracle Endeca Workbench is a suite of tools that brings together best-in-class Web-site management capabilities including merchandising, Content Spotlighting, search configuration, and usage reporting.

In addition to these powerful tools for business users, Workbench provides features for system administrators to configure the resources used by an Endeca implementation, monitor its status, start and stop system processes, and download an implementation's instance configuration for debugging and troubleshooting purposes.

The Oracle Endeca Workbench package contains the following components:

- Endeca Tools Service
- Oracle Endeca Workbench

In addition, the installation includes a version of the Endeca JSP reference application, which serves as the default preview application in Oracle Endeca Workbench.

# About the Deployment Template

The Deployment Template provides a collection of operational components that serve as a starting point for development and application deployment.

The template includes the complete directory structure required for deployment, including Endeca Application Controller (EAC) scripts, configuration files, and batch files or shell scripts that wrap common script functionality.

The Deployment Template is the recommended method for building your application deployment environment.

Chapter 2

# Installing Oracle Endeca Guided Search

This section provides prerequisite and instructional information about installing Oracle Endeca Guided Search.

## Version compatibilities

To determine the compatibility of components in Oracle Endeca Guided Search, see the *Oracle Endeca Guided Search Compatibility Matrix* available on the Oracle Technology Network.

Core packages comprising the Endeca Access Platform:

- MDEX Engine
- Presentation API
- Platform Services
- Oracle Endeca Workbench

Optional packages:

- Developer Studio
- Endeca Deployment Template

  **Note:** This guide assumes that you have downloaded and installed the Endeca Deployment Template

- Content Acquisition System

## Installer file names

Endeca installation packages and executables are named according to a common convention.

The installer file names follow the format:

```
componentname_version_arch-OS
```

For example:

```
mdex_622_x86_64pc-linux.sh
```

The *componentname* is the component identifier for the component being installed. In the example installer, `mdex` is the identifier for Endeca MDEX Engine.

The *version* is the release version, without periods. In the example installer, `622` identifies Endeca MDEX Engine version 6.2.2.

The *arch-OS* is the architecture and operating system identifier for the component being installed. In the example installer, `x86_64pc-linux` identifies the file as an installer for the 64-bit Linux platform. The following table lists the *arch-OS* identifiers and their platforms:

| arch-OS identifier | Installation platform |
|---|---|
| x86_64pc-linux | Linux running on 64-bit Intel processors |
| x86_64pc-win32 | Windows running on 64-bit Intel processors |

# Preparing for installation

Before you install or upgrade any Endeca components, make sure to read installation and migration requirements as they apply to your scenario.

The following documentation is available on the Oracle Technology Network:

| Component | Related Documentation |
|---|---|
| Endeca MDEX Engine | See the *Endeca MDEX Engine Installation Guide* and the *Endeca MDEX Engine Migration Guide*. |
| Endeca Presentation API | See the *Installation instructions and release notes* file. |
| Oracle Endeca Workbench | See the *Oracle Endeca Workbench Installation Guide* and the *Oracle Endeca Workbench Migration Guide*. |
| Endeca Platform Services | See the *Endeca Platform Services Installation Guide* and the *Endeca Platform Services Migration Guide*. |
| Endeca Content Acquisition System | See the *Endeca Content Acquisition System Installation Guide* and the *Endeca Content Acquisition System Migration Guide*. |

# Installing the core Endeca components

While you can install the Endeca packages in any order, Oracle recommends that you install them in the following order:

1. The MDEX Engine package.
2. An Endeca Presentation API package (as appropriate for Window or UNIX).
3. The Platform Services package.
4. Oracle Endeca Workbench package.
5. Additional installation packages, such as the Endeca Deployment Template, Developer Studio, and Content Acquisition System (CAS) on those servers that require them.
6. Separately licensed packages, such as Relationship Discovery, Analytics, or CMS connectors.

For installation instructions, see the appropriate installation guide.

**Note:** If you are upgrading from previous releases, read the *Migration Guide* and follow guidance on how to prepare your implementation for migration. Next, proceed with downloading and installing the software packages in the order listed in this topic.

# Downloading and installing the Endeca Deployment Template

You can download and run the Deployment Template on a single machine that serves as your Endeca development server, or on several machines running on the same operating system (Windows or UNIX).

Before downloading the Deployment Template, decide on and provision the hardware that you will use in your development environment. For detailed information about the Endeca Deployment Template, see the *Oracle Endeca Deployment Template Usage Guide*.

To download and install the Endeca Deployment Template:

1. Download the latest available version of the Deployment Template from the Oracle Software Delivery Cloud.
2. Unzip the Deployment Template into `C:\` if you are running on Windows or a directory such as `/localdisk/` on UNIX.
   The package creates a directory structure under `C:\Endeca\Solutions` on Windows and `/localdisk/Endeca/Solutions` on UNIX.
3. Create a directory for deploying your project, for example, create `C:\Endeca\apps` on Windows or `/localdisk/apps` on UNIX.

You have installed the Deployment Template and are prepared to run it.

Next, you run the Deployment Template to establish the Endeca project based on the sample wine application, and run a baseline update script in this project.

# Setting the Endeca environment variables

Having the environment variables properly set ensures that the different Endeca components can communicate with each other.

To set the Endeca environment variables:

Do the following:

| Option | Description |
|---|---|
| On Windows | To set the environment variables for Platform Services and Oracle Endeca Workbench, run the installation process for these packages. This properly sets up the environment variables for them.<br>To set the environment variables for the MDEX Engine (in particular, to set the `ENDECA_MDEX_ROOT`), run the `\Endeca\MDEX\<version>\mdex_set¬up.bat` script. |
| On UNIX | Depending on the package and your platform, use the `source` command to run the scripts that set the variables. For example, in your Endeca installation directories, run:<br>• `source /endeca/MDEX/<version>/mdex_setup_sh.ini`. The `mdex_setup` script sets up the MDEX Engine variables.<br>• `source /endeca/PlatformServices/workspace/setup/in¬staller_sh.ini`. This script sets up the Platform Services variables.<br>• `source /endeca/Workbench/workspace/setup/in¬staller_sh.ini`. This script sets up the Endeca Workbench variables. |

For information on setting environment variables required by other Endeca packages, refer to the installation guides for each package.

# Starting the Endeca HTTP and Tools services

If you have multiple servers, the Endeca HTTP service must be running on all the machines in your Endeca environment, except the Application server. When the Endeca HTTP service is running this means that the Endeca Application Controller (EAC) is running. The Endeca Tools Service must be running on the Tools server.

Before starting the Endeca HTTP and Tools services, verify that you have:

- Installed the MDEX Engine, Platform Services and Workbench.
- Set the environment variables for the MDEX Engine and Platform Services.

To start the Endeca HTTP service and the Endeca Tools Service:

Do the following:

| Option | Description |
|---|---|
| On Windows | Go to **Start** > **Control Panel** > **Administrative Tools** > **Services**, select the Endeca HTTP service and the Endeca Tools Service and click **Start**. |

| Option | Description |
|---|---|
| |  **Note:** On the servers on which you have installed the Platform Services and Endeca Workbench packages, the Endeca HTTP and Tools services are started automatically when you reboot the machines. |
| On UNIX | To start the Endeca HTTP service, run `$ENDECA_ROOT/tools/server/bin/startup.sh`<br>To start the Endeca Tools Service, run `$ENDECA_TOOLS_ROOT/server/bin/startup.sh` |

Part 2

# Working with the sample application

Chapter 3

# Running the Sample Application

To configure a sample project on a single development server, install all the required Endeca packages and the Endeca Deployment Template on this server, and run the Deployment Template scripts to create, provision, and initialize the Endeca application and run the baseline update.

## Configuring the sample application on a single development server

To configure an application on a single development server, run the Deployment Template `deploy` script and accept the defaults.

Before running the Deployment Template, verify that:

- You have installed the MDEX Engine, Platform Services (including the EAC Central Server and Agent), and Oracle Endeca Workbench on the same machine.
- The Endeca HTTP and Tools services are running on this server. (When the Endeca HTTP service is running, the EAC is running.)
- You have downloaded the Deployment Template on this server, and set up a directory for your deployment, such as `C:\Endeca\apps` on Windows or `/localdisk/apps` on UNIX.

To configure the application on a single development server:

1. Open a command prompt window and navigate to the `C:\Endeca\Solutions\deploymentTemplate-`*`version`*`\bin` directory on Windows or `/usr/local/Endeca/Solutions/deploymentTemplate-`*`version`*`/bin` on UNIX.
2. Run the `deploy.bat` or `deploy.sh` script.

   This script creates the project directories and configuration files.
3. Enter information as prompted, or accept the defaults.
4. Confirm the correct version of the Platform Services installation package (the template verifies the `ENDECA_ROOT` variable), and answer `Yes` to proceed.
5. Select the deployment type, `Dgraph`.
6. Specify the name of the application: `MyApp` and the location of the application directory: `C:\Endeca\apps` on Windows or `/localdisk/apps` on UNIX.

**Note:** In this guide, the directory for each of your applications is referred to by the *`[appDir]`* abbreviation. With the paths above, this is equal to `C:\Endeca\apps\MyApp` on Windows and `/localdisk/apps/MyApp` on UNIX.

7. Specify the EAC port (the Endeca HTTP service port) or accept the default port: `8888`
8. For **Enable Workbench integration**, specify `Yes`.

**Note:** This configuration also applies to any Oracle Endeca Workbench edition.

9. Specify Oracle Endeca Guided Search Workbench port (this is the Endeca Tools Service port for your Oracle Endeca Workbench edition) or accept the default port: `8006`.
10. Specify other necessary ports:
    a) For the Dgraph1, specify the Dgraph1 user query port or accept the default: `15000`
    b) For the Dgraph2, specify the Dgraph2 user query port or accept the default: `15001`
    c) For the Endeca Logging and Reporting Server, specify the server port or accept the default: `15010`

**Note:** The Logging Server port number can be no larger than `32767`. If you plan to use the reference implementation and verify the Logging Server, you can set the Logging Server to run on port 15002 (for Dgraph1) or on port 15003 (for Dgraph2), and the reference implementation will work by default when connected to an MDEX Engine running on ports 15000 and 15001, respectively. These settings assume that the Logging Server runs on the same machine as the MDEX Engines. If you are using a different port for your Dgraph with the JSP reference implementation, specify a port equal to `Dgraph_port_number + 2`. This is because the Logging Server for the JSP reference implementation submits log entries to a port 2 above the Dgraph port.

Now you have provisioned the directories for the application and need to initialize it.

By default, the Deployment Template provisions a project in which two Dgraphs run on the same MDEX Engine server host. If you prefer to configure only one Dgraph, edit the *[appDir]*`/config/script/AppConfig.xml` file to delete `Dgraph2` entries.

**Example of the `AppConfig.xml` file**

The following example shows an abbreviated version of the `AppConfig.xml` file that is created when you run the `deploy` script for a single server in your development environment. This example lists two Dgraphs.

You can remove the second Dgraph, if needed:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  ##############################################################################

  # This file contains settings for an EAC application.
  ...
    ##############################################################################

    # Global variables
    #
  -->
  <app appName="MyApp" eacHost="DevServer.myCompany.com" eacPort="8888"
  ...
  </app>

  <!--
    ##############################################################################
```

```
    # Servers/hosts
    ...
  -->
  <host id="ITLHost" hostName="DevServer.MyCompany.com" port="8888" />
  <host id="MDEXHost" hostName="DevServer.MyCompany.com" port="8888" />
  <host id="webstudio" hostName="DevServer.MyCompany.com" port="8888" >
    <directories>
      <directory name="webstudio-report-dir">./reports</directory>
    </directories>
  </host>
.....
  <!--
    ###########################################################################

    # Config Manager.
    ...
  -->
  <custom-component id="ConfigManager" host-id="ITLHost"
class="com.endeca.soleng.eac.toolkit.component.ConfigManagerComponent">
    <properties>
      <property name="webStudioEnabled" value="true" />
      <property name="webStudioHost" value="DevServer.MyCompany.com" />
      <property name="webStudioPort" value="8006" />
     ....
    </properties>
    ....
  <!--
    ###########################################################################

    # Forge
    #
  -->
  <forge id="Forge" host-id="ITLHost">
    ...
  </forge>
....

<!--
    ###########################################################################

    # Dgidx
    #
  -->
  <dgidx id="Dgidx" host-id="ITLHost">
    ....
  </dgidx>

  <!--
    ###########################################################################

    # Dgraph Cluster
    #
  -->
  <dgraph-cluster id="DgraphCluster" getDataInParallel="true">
    <dgraph ref="Dgraph1" />
    <dgraph ref="Dgraph2" />
  </dgraph-cluster>

  ....
```

```
  <!--
    ############################################################################

     # Dgraphs
     #
  -->
  <dgraph id="Dgraph1" host-id="MDEXHost" port="15000">
    ....
  </dgraph>

  <dgraph id="Dgraph2" host-id="MDEXHost" port="15001">
    ....
  </dgraph>

....
  <!--
    ############################################################################

     # LogServer
     #
  -->
  <logserver id="LogServer" host-id="ITLHost" port="15010">
     ....

  </logserver>

  ....
```

# Initializing the application

To initialize the application, run the `initialize_services` script from the Endeca Deployment Template.

It is assumed that you have run the Deployment Template `deploy` script to create the directory structure, configuration files and scripts for the application.

To initialize the application:

On the development server, or on a Data Processing (ITL) server in your environment, run *[appDir]*`\control\initialize_services.bat` on Windows or *[appDir]*`/control/initialize_services.sh` on UNIX.
This script initializes the sample wine application.

After you have provisioned and initialized the application, you can run the baseline update script using the Deployment Template and also access Endeca Workbench to check the status of the running components.

# Running the baseline update script

The baseline update script runs the MDEX Engine (the indexer and the Dgraph) to index the records and to update the MDEX Engine with the indexed data.

Before running the baseline update script, ensure that you have provisioned the sample wine reference implementation with the Deployment Template, by running its *[appDir]*`\control\initialize_services.bat` or *[appDir]*`/control/initialize_services.sh` script.

To run a baseline update script on the Data Processing (ITL) server:

1. Run *[appDir]*`\control\load_baseline_test_data.bat` or *[appDir]*`/control/load_baseline_test_data.sh`
   This script uploads the reference implementation data into the locations expected by the Deployment Template workflow, and communicates to the EAC that the data is ready for processing.
2. Run *[appDir]*`\control\baseline_update.bat` or *[appDir]*`/control/baseline_update.sh` script.
   This script takes a few moments to complete.
3. Log in to Oracle Endeca Workbench as an administrator, and open the EAC Admin Console. Verify that the application is provisioned correctly with all Endeca components running on the hosts and ports that exist in your configuration.

**Note:** The user name for the predefined Oracle Endeca Workbench administrator is `admin` and the default password is `admin`. After logging in as the admin user, you can modify the password.

In addition to running the baseline update script, you can use the Deployment Template to run a partial update script, a configuration update script , and Log Server scripts for obtaining daily log reports. For information about performing these tasks and about customizing the Deployment Template for your own data and server topology, see the *Oracle Endeca Deployment Template Usage Guide*.

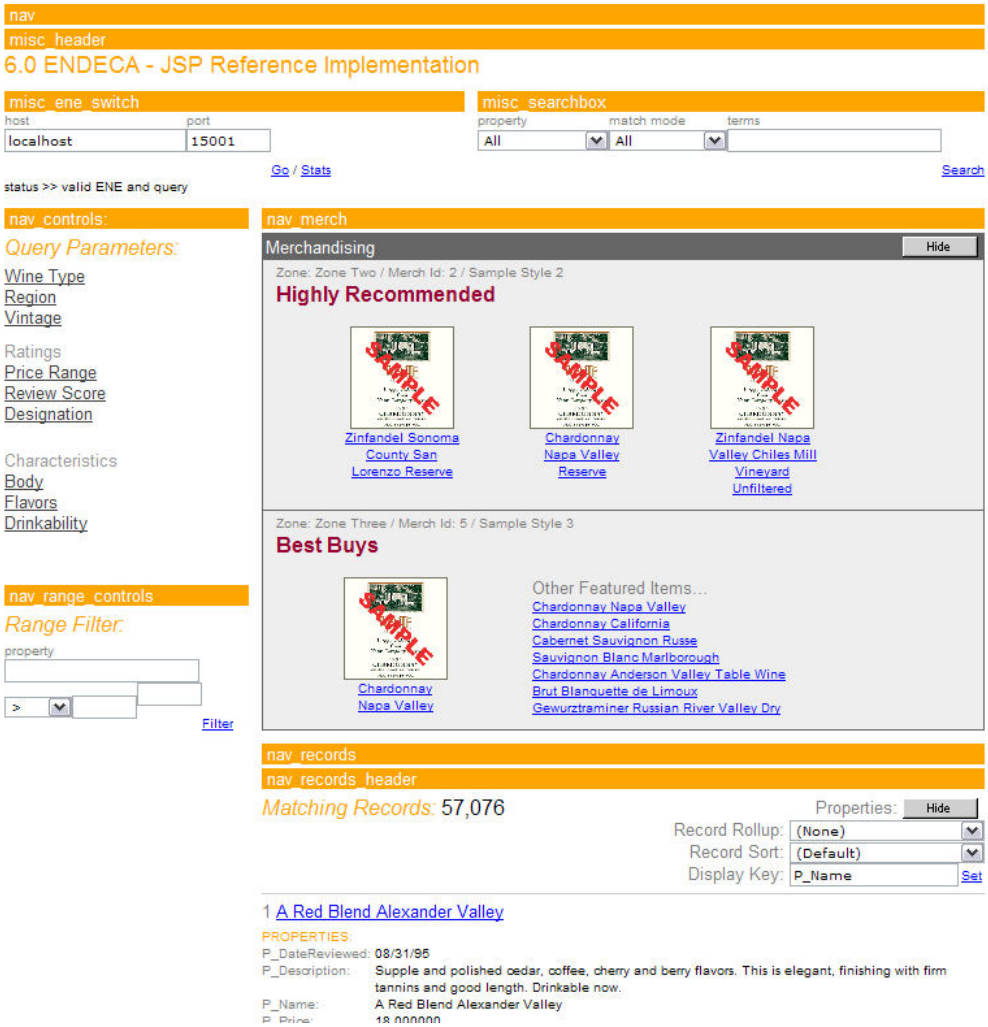
# Verifying your installation with the JSP reference application

After you have successfully run a baseline update and started the Endeca components, you can use the JSP reference implementation to navigate and search your data.

The JSP reference application is installed as part of Oracle Endeca Workbench installation and runs in the Endeca Tools Service.

To verify an Endeca setup with the internal Endeca JSP reference application:

1. Open Internet Explorer.
2. In the Address box, enter the following URL:
   `http://`*`WorkbenchHost`*`:8006/endeca_jspref`
   Replace *WorkbenchHost* with the name of the machine that is running Oracle Endeca Workbench. If you used a different port when you configured Oracle Endeca Workbench, substitute that port for `8006`.
   This URL brings you to a page with a link called **ENDECA-JSP Reference Implementation**.
3. Click the **ENDECA-JSP Reference Implementation** link.
4. Enter the host name and port of the machine that the MDEX Engine is running on. For example, enter `localhost` and `15000`. Click **Go**.
   You should see the reference implementation displaying the sample wine data.


nav
misc_header
6.0 ENDECA - JSP Reference Implementation
misc_ene_switch
host
port
localhost
15001
misc_searchbox
property
match mode
terms
All
All
Go / Stats
Search
status >> valid ENE and query
nav_controls:
Query Parameters:
Wine Type
Region
Vintage
Ratings
Price Range
Review Score
Designation
Characteristics
Body
Flavors
Drinkability
nav_range_controls
Range Filter:
property
>
Filter
nav_merch
Merchandising
Hide
Zone: Zone Two / Merch Id: 2 / Sample Style 2
Highly Recommended
Zinfandel Sonoma County San Lorenzo Reserve
Chardonnay Napa Valley Reserve
Zinfandel Napa Valley Chiles Mill Vineyard Unfiltered
Zone: Zone Three / Merch Id: 5 / Sample Style 3
Best Buys
Chardonnay Napa Valley
Other Featured Items...
Chardonnay Napa Valley
Chardonnay California
Cabernet Sauvignon Russe
Sauvignon Blanc Marlborough
Chardonnay Anderson Valley Table Wine
Brut Blanquette de Limoux
Gewurztraminer Russian River Valley Dry
nav_records
nav_records_header
Matching Records: 57,076
Properties:
Hide
Record Rollup:
(None)
Record Sort:
(Default)
Display Key:
P_Name
Set
1 A Red Blend Alexander Valley
PROPERTIES:
P_DateReviewed: 08/31/95
P_Description: Supple and polished cedar, coffee, cherry and berry flavors. This is elegant, finishing with firm tannins and good length. Drinkable now.
P_Name: A Red Blend Alexander Valley

Chapter 4

# Running the Reference Implementations

You can use an Endeca reference implementation (a sample Endeca Web application) to verify that your Endeca components are installed and working properly. The reference applications are included as part of the Platform Services package. Updated APIs for the reference applications are distributed with the Endeca Presentation API.

## Running the JSP reference implementation

The JSP reference application can be installed in an application server with J2EE support such as Apache Tomcat. This section differs from the "Verifying your installation with the JSP reference application" section in that here we assume that you are installing the JSP reference implementation to a standalone version of the Tomcat Web server.

If you are running the JSP reference implementation to test an Endeca Analytics installation, you must first follow the instructions in *Enabling Endeca Analytics*.

### Setting up the JSP reference implementation on Windows

While this section assumes that you use the Tomcat server, you can use other application servers.

The JSP reference implementation depends on several paths related to the Tomcat Web server and Java SDK. This section assumes the following paths in your environment:

| | |
|---|---|
| The location of the Tomcat installation | `C:\jakarta-tomcat-`*`version`* |
| The location of the Java SDK installation | `C:\j2sdk-`*`version`* |

In the following procedures, adjust the paths as needed for your environment.

To set up the JSP reference implementation:

1. Copy the reference implementation user interface directory `%ENDECA_REFERENCE_DIR%\endeca_jspref` into the `C:\jakarta-tomcat-`*`version`*`\webapps` directory.

   The `%ENDECA_REFERENCE_DIR%` variable is set as part of the Platform Services installation.

2. (Optional.) Navigate to `C:\jakarta-tomcat-`*`version`*`\conf` and open the `server.xml` file in a text editor. You can modify the file as follows:

a) Change the port that Tomcat listens on for a shutdown command from its default of 8005:

```
<Server port="8005" shutdown="SHUTDOWN">
```

b) Change the Tomcat HTTP listening port from its default of 8080:

```
<!-- Define a non-SSL Coyote HTTP/1.1 Connector on port 8080 -->
<Connector port="8080" ...
```

c) Save and close the `server.xml` file.

3. If your version of Java requires it, make sure that the `JAVA_HOME` environment variable is set to the location of the Java SDK directory. For example, the location might be `C:\j2sdk-`*`version`*.

**Note:** See the Tomcat documentation for more information about your version of the Tomcat server to check if it requires a `JAVA_HOME` environment variable.

To set the `JAVA_HOME` environment variable:

a) From the Windows Control Panel, select **System**.
b) Go to the **Advanced** tab and select **Environment Variables**.
c) In the **System Properties** section, locate and select `JAVA_HOME`.

If `JAVA_HOME` does not exist, select **New**, and then in the **Variable Name** field, enter `JAVA_HOME`

d) In the **Variable Value** field, enter the path of the Java SDK directory and click **OK**.
e) Click **OK** to close the **Environment Variables** window.
f) Click **OK** to close the **System Properties** window.

4. Copy the following files from the `PresentationAPI\<version>\java\lib` directory to `C:\jakarta-tomcat-`*`version`*`\webapps\endeca_jspref\WEB-INF\lib`:
   - `bcprov-jdk-`*`version`*`.jar` (Bouncy Castle encryption implementation)
   - `endeca_logging.jar` (Endeca Logging API)
   - `endeca_navigation.jar` (Endeca Presentation API)
5. Copy the following Endeca Report Generator file from the `%ENDECA_ROOT%\lib\java` directory to `C:\jakarta-tomcat-`*`version`*`\webapps\endeca_jspref\WEB-INF\lib`:
   - `rg.jar`
6. Start the Tomcat server. See the Tomcat documentation for specific instructions.

The JSP reference implementation is set up and you can now test your Endeca installation with it.

## Setting up the JSP reference implementation on UNIX

While this section assumes that you use the Tomcat server, you can use other application servers.

The JSP reference implementation depends on several paths related to the Tomcat Web server and Java SDK. This section assumes the following path names:

| | |
|---|---|
| The location of the Tomcat installation | `/usr/local/tomcat-`*`version`* |
| The location of the Java SDK installation | `/usr/local/j2sdk-`*`version`* |

**Note:** The Java SDK installation must consist of the entire JDK, and not just the location of a copied or linked Java binary.

To set up the JSP reference implementation:

1. Copy the reference implementation from `$ENDECA_REFERENCE_DIR/endeca_jspref` to the Tomcat `/webapps` directory (for example, `/usr/local/tomcat-`*`version`*`/webapps`).

   The `$ENDECA_REFERENCE_DIR` variable is set as part of the Platform Services installation.
2. (Optional.) Go to the `/usr/local/tomcat-`*`version`*`/conf` directory and open the `server.xml` file in a text editor. You can modify the file as follows:
   a) Change the port that Tomcat listens on for a shutdown command from its default of 8005:
   ```
   <Server port="8005" shutdown="SHUTDOWN">
   ```
   b) Change the Tomcat HTTP listening port from its default of 8080:
   ```
   <!-- Define a non-SSL Coyote HTTP/1.1 Connector on port 8080 -->
   <Connector port="8080" ...
   ```
   c) Save and close the `server.xml` file.
3. Set the appropriate Tomcat environment variables.
   - For `csh` and similar shells, set:
   ```
   setenv JAVA_HOME /usr/local/j2sdk-version
   setenv CATALINA_BASE /usr/local/tomcat-version
   ```
   - For `bash`, set:
   ```
   export JAVA_HOME=/usr/local/j2sdk-version
   export CATALINA_BASE=/usr/local/tomcat-version
   ```
   Generally these commands should be placed in a script run at the startup of the shell so that the variables are set for future use.
4. Copy the following Endeca files from the `PresentationAPI/<version>/java/lib` directory to `/usr/local/tomcat-`*`version`*`/webapps/endeca_jspref/WEB-INF/lib`:
   - `bcprov-jdk-`*`version`*`.jar` (Bouncy Castle encryption implementation)
   - `endeca_logging.jar` (Endeca Logging API)
   - `endeca_navigation.jar` (Endeca Presentation API)

   This enables Tomcat to access these files.
5. Copy the following Endeca Report Generator file from the `$ENDECA_ROOT/lib/java` directory to `/usr/local/tomcat-`*`version`*`/webapps/endeca_jspref/WEB-INF/lib`:
   - `rg.jar`
6. Start the Tomcat server.

The JSP reference implementation is set up and you can now test your Endeca installation with it.

## Enabling the Analytics controls in the JSP reference implementation

The Endeca JSP reference implementation includes a set of Analytics controls that are not displayed by default. These controls are useful for learning about, developing, and debugging Analytics statements.

These instructions pertain to the Endeca JSP reference implementation that runs under the Endeca Tools Service. If your Endeca JSP reference is running on a standalone Tomcat, use the same instructions, substituting the path names in your Tomcat installation for the ones below

To enable the Analytics controls in the Endeca JSP reference implementation:

1. After installing Oracle Endeca Workbench package, place `CordaEmbedder.jar` in this directory:
   - Windows: `%ENDECA_TOOLS_ROOT%\server\webapps\endeca_jspref\WEB-INF\lib`
   - UNIX: `$ENDECA_TOOLS_ROOT/server/webapps/endeca_jspref/WEB-INF/lib`

   **Note:** This file is available as part of the Corda Server installation package and is required by the reference implementation even if you do not intend to use charts.

2. Edit the `web.xml` file (which is in the WEB-INF directory from step 1) and add the definition of the `eneAnalyticsEnabled` parameter, as in this example:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- This file identifies these directories as containing
a Web application. -->
<!DOCTYPE web-app
     PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
     "http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
   <context-param>
      <param-name>eneAnalyticsEnabled</param-name>
      <param-value>1</param-value>
     <description>Flag to enable Endeca Analytics controls</description>

   </context-param>
</web-app>
```

3. Restart the Endeca Tools Service.
4. In a Web browser, navigate to the JSP reference implementation. The Analytics controls should be visible.

## Testing your Endeca installation with the JSP reference implementation

Once you have set up the JSP reference implementation, you can test your Endeca installation with it.

To test the Endeca installation with the JSP reference implementation:

1. Open Internet Explorer and enter the following URL:
   `http://EndecaServerNameorIP:PortNumber/endeca_jspref`, where the `EndecaServerNameorIP` is the machine on which you set up the reference application, and the `PortNumber` is the port on which the Tomcat server is listening.
   For example, enter: `http://localhost:8080/endeca_jspref`
2. Click the **ENDECA-JSP Reference Implementation** link to launch the JSP reference implementation.
3. Enter the host name as the server name or IP of the machine on which you installed the Endeca MDEX Engine.
4. Enter the port number you specified for the MDEX server in the Deployment Template `AppConfig.xml` or in the `remote_index.script` control script. This is the port on which the MDEX Engine accepts queries.

5. Click **Go**.

   The JSP reference implementation opens.

# Running the ASP.NET reference implementation

The ASP.NET reference implementation runs in IIS 6.0 on Windows Server 2003 64-bit systems, and requires some configuration before you deploy the application.

## Configuring the 64-bit version of ASP.NET

Before you set up the reference application, make sure you have enabled the 64-bit version of ASP.NET.

The ASP.NET reference implementation supports versions 2.0 SP1, 3.0, and 3.5 of ASP.NET.

To install the 64-bit version of ASP.NET:

1. From a command prompt, issue the following command to disable 32-bit mode:
   ```
   cscript %SYSTEMDRIVE%\inetpub\adminscripts\adsutil.vbs SET W3SVC/App¬
   Pools/Enable32bitAppOnWin64 0
   ```
2. Issue the following command to install the 64-bit version of ASP.NET 2.0 and to install the script maps at the IIS root:
   ```
   %SYSTEMROOT%\Microsoft.NET\Framework64\v2.0.50727\aspnet_regiis.exe -i
   ```

**Note:** The .NET DLLs packaged with this release are compiled using the 64-bit version of the .NET Framework. They should be compatible with .NET Frameworks 2.0 SP1, 3.0, and 3.5.

## Enabling ASP pages in IIS on Windows 2003

On Windows 2003, Microsoft IIS does not have ASP pages enabled as a Web server extension by default. You must enable them in the IIS Manager.

To enable ASP pages in IIS:

1. Go to **My computer** > **Manage** > **Services and Applications**.
2. Open the IIS Manager, and select **Web Service Extensions**.
3. Right-click **Allow all Web service extensions for a specific application** and choose **ASP**.

## Setting up the ASP.NET reference implementation

In this section we assume that you are using IIS 6.0 and .NET 2.0. The reference implementation supports versions 2.0 SP1, 3.0, and 3.5 of ASP.NET.

You must make sure that the 64-bit version of ASP.NET is configured and that you have enabled the ASP pages as an extension in the Microsoft IIS before proceeding with setup of the ASP.NET reference implementation.

To set up the ASP.NET reference implementation:

1. Copy all the `Endeca.*.dll` files from `PresentationAPI\<version>\dotNet\lib` to: `C:\Endeca\PlatformServices\reference\endeca_ASP.NETref\bin`.
2. Modify the following IIS settings:
   a) From the Windows Control Panel, select **Administrative Tools** > **Internet Information Services**.
   b) In the **Internet Information Services** tree pane, expand the machine icon for the local machine.
   c) Right-click **Default Website**.
   d) Select **New** > **Virtual Directory**.

   **Note:** If you are using IIS 7, you should create an **Application** rather than a **Virtual Directory**.

   e) Fill in the following fields in the **Virtual Directory Creation** wizard as follows:

| Field | Value |
|---|---|
| **Virtual Directory Alias** | `endeca_ASP.NETref` |
| **Website Content Directory** | Browse to the location of the ASP.NET reference implementation. The default location is: `c:\Endeca\PlatformServices\reference\endeca_ASP.NETref` |
| **Access Permissions** | Leave the default settings in place. |

   The Virtual Directory Creation wizard opens.
   f) Click **Next**, then click **Finish**.
   g) In the IIS Manager MMC snap-in, to set the virtual directory name as an application name, right-click the virtual directory, and select **Virtual Directory** > **Application settings** > **Create**. The application name can be set to any name, and you can use the alias you used for the virtual directory as an example. Set **Execute Permissions** to `Scripts Only`.
   h) Close the Internet Information Services window.

The ASP.NET reference implementation is set up and you can now test your Endeca installation with it.

## Testing your Endeca installation with the ASP.NET reference implementation

Once you have set up the ASP.NET reference implementation, you can test your Endeca installation with it.

To test the Endeca installation with the ASP.NET reference implementation:

1. Open Internet Explorer.
2. Navigate to the following location: `http://EndecaServerNameorIP/endeca_ASP.NETref`
   `EndecaServerNameorIP` refers to the machine on which you set up the reference application.
   For example, assuming that you use the default IIS port of 80:
   `http://localhost/endeca_ASP.NETref`
3. From here, click **Endeca .NET Reference Implementation** to launch the Endeca ASP.NET Reference Implementation.

The Endeca ASP.NET Reference Implementation asks you for a host and port of the MDEX Engine server.

4. Enter the host name as the server name or IP of the machine on which you installed the Endeca MDEX Engine.
5. Enter the port number you specified for the MDEX server in the Deployment Template `AppConfig.xml` or in the `remote_index.script` control script. This is the port on which the MDEX Engine accepts queries.
6. Click **Go**.
   The ASP.NET reference implementation opens.

Part 3

# What's Next

Chapter 5

# Guide to Endeca Documentation

This section provides information about the documentation required for configuring various aspects of an Endeca implementation.

## Where to find relevant documentation

This topic lists Oracle Endeca Guided Search documentation relevant to each major implementation task.

| For information about | See this documentation |
|---|---|
| Basic Endeca concepts | • *Oracle Endeca Guided Search Concepts Guide*<br>• *Endeca Glossary* |
| Data ingest | The CAS documentation set, especially:<br>• *CAS Quick Start Guide*<br>• *CAS Developer's Guide*<br>• *CAS Console for Oracle Endeca Workbench Help* |
| Pipeline creation | • *Forge Guide*<br>• *Oracle Endeca Developer Studio Help*<br>• *Partial Updates Guide* |
| Application development | For information about Endeca features and details about the Endeca Presentation API:<br>• *Basic Development Guide*<br>• *Advanced Development Guide*<br>For information about working with Web services and XQuery for Endeca:<br>• *Web Services and XQuery Developer's Guide*<br>For information about building applications using the RAD Toolkit for ASP.NET: |

| For information about | See this documentation |
|---|---|
| | • *RAD Toolkit for ASP.NET Developer's Guide* |
| Deployment and operational tasks | • *Oracle Endeca Guided Search Administrator's Guide*<br>• *Oracle Endeca Deployment Template Usage Guide* |

Chapter 6

# Additional Installation Packages

The following packages are highly recommended, although not required to follow the procedures in this guide. They can be installed and integrated into your implementation later. Access to these packages is included with all licenses of Oracle Endeca Guided Search.

## About Developer Studio

Developer Studio is a Windows application that you use to define all aspects of your instance configuration including pipeline components, Endeca properties and dimensions, precedence rules, dynamic business rules, and user profiles.

With Developer Studio, you can define:

- Pipeline components for tasks such as loading, standardizing, joining, mapping, and exporting data.
- Endeca properties and property attributes such as sort and rollup.
- Dimensions and dimension values, including dimension hierarchy.
- Precedence rules among dimensions that provide better control over your implementation's navigation flow.
- Search configurations, including which properties and dimensions are available for search.
- Dynamic business rules that allow you to promote certain records on your Web site using data-driven business logic. Dynamic business rules are used to implement merchandising and content spotlighting.
- User profiles that tailor the content returned to an end-user based upon pre-configured rules.

Developer Studio uses a project file, with an `.esp` extension, that contains pointers to the XML files that support an instance configuration.

## About the Content Acquisition System (CAS)

The Content Acquisition System (CAS) provides components that manage all file system and CMS crawls, as well as all Web crawls.

The CAS package includes:

- Endeca CAS Server
- Endeca CAS Console
- Endeca CAS API

- Endeca Web Crawler.

**Note:** Connectors to a variety of content management systems (CMSs) are available as separately licensed packages.

# About the RAD Toolkit

The Rapid Application Development (RAD) Toolkit provides controls and components to build Endeca applications and also provides a simplified interface to the Endeca Presentation API. The RAD Toolkit is available for ASP.NET.

The RAD Toolkit for ASP.NET contains the following components:

| RAD Toolkit for ASP.NET component | Description |
|---|---|
| RAD API for .NET | Provides a simplified interface to the Endeca MDEX Engine and makes programming more friendly to the typical .NET developer. |
| Visual Studio server controls, including Endeca data source controls, and Endeca user interface controls | These controls help developers quickly build Endeca applications and also provide a simple interface to the Endeca Presentation API for ASP.NET.<br>The controls participate in ASP.NET declarative data binding and include an Endeca-specific data source control to easily set host, port, and query-specific information. |
| Reference application | Like other Endeca reference applications, the RAD Toolkit reference application provides a simple front-end interface that allows you to connect to an MDEX Engine and examine a record set.<br>This reference application can be run in Postback mode, URL mode, RAD Toolkit Server Controls mode, or RAD Toolkit Server Controls URL mode. |

Appendix A

# Full List of Documentation Resources

This section describes the documentation related to each platform component. Only essential documentation is included with the product installation, but all Endeca documentation is available on the Oracle Technology Network for browsing or download, either individually or as part of an overall Documentation package.

## General Endeca documentation

The following table lists the documentation that applies across multiple Endeca packages.

| Title | Description |
|---|---|
| *Oracle Endeca Guided Search Getting Started Guide* | Overview of Endeca components including information about configuration scenarios. |
| *Oracle Endeca Guided Search Compatibility Matrix* | Summary of version compatibility information for Endeca components. |
| *Oracle Endeca Guided Search Concepts Guide* | Introduction to Oracle Endeca Guided Search. Covers the key concepts underlying Endeca applications. |
| *Oracle Endeca Guided Search Administrator's Guide* | Describes tasks involved in administering and maintaining applications built upon the Oracle Endeca Guided Search. It bridges the gap between the work performed by the Endeca Services team and the issues that system administrators encounter when maintaining the system. |
| *Oracle Endeca Glossary* | A reference for Endeca terms and definitions. |
| *Oracle Endeca Guided Search Third-Party Software Usage and Licenses* | Provides copyright, license agreement, and/or disclaimer of warranty information for the third-party software packages that Endeca incorporates. |

## MDEX Engine documentation

The following table lists the documentation related to the MDEX Engine package.

| Title | Description |
|---|---|
| *Analytics Guide* | Provides an overview of Endeca Analytics and describes the Analytics and Charting APIs, date and time properties, and key properties. |
| *Basic Development Guide* | Provides information about working with records, dimensions, and basic search features. |
| *Advanced Development Guide* | Covers such topics as Endeca Query Language (EQL), record filters, bulk export, spelling correction, phrasing, relevance ranking, and dynamic business rules. |
| *Oracle Endeca MDEX Engine Installation Guide* | Provides a brief overview of the Endeca MDEX Engine, details installation procedures, and describes how to configure the licensing keys for the Language Pack. Covers both Windows and Linux/UNIX system requirements and installation procedures. |
| *Oracle Endeca MDEX Engine Migration Guide* | Provides information on migrating from previous versions of Endeca software. |
| *Partial Updates Guide* | A guide to preparing and running partial updates in your Endeca application. |
| *Performance Tuning Guide* | Provides guidelines on monitoring and tuning the performance of the Endeca MDEX Engine. Contains tips on resolving associated operational issues. |
| *Web Services and XQuery Developer's Guide* | Describes how to use Web services and XQuery for Endeca. Web services and XQuery for Endeca provides Endeca application developers with a flexible, extensible, and standards-compliant query processing solution. |
| *Oracle Endeca MDEX Engine Release Notes* | Details the changes specific to this release, including bug fixes and new features. |

# Presentation API documentation

The following table lists the documentation for the Presentation API package.

| Title | Description |
|---|---|
| *Presentation API for Java Reference (Javadoc)* | The Java reference documentation for the Endeca Presentation, Analytics, and Charting APIs. |
| *Presentation API for .NET Reference* | The .NET reference documentation for the Endeca Presentation, Analytics, and Charting APIs. |
| *Logging API for Java Reference (Javadoc)* | The Java reference documentation for the Endeca Logging API. |
| *Logging API for .NET Reference* | The .NET reference documentation for the Endeca Logging API. |

# Platform Services documentation

The following table lists the documentation related to the Platform Services package.

| Title | Description |
|---|---|
| *Content Adapter Developer's Guide* | Describes how to write Java manipulators and content adapters using the Endeca Content Adapter Development Kit. |
| *Control System Guide* | Provides information on using the Endeca Control System, including communicating with the JCD service and running control scripts. |
| *Oracle Endeca Application Controller Guide* | Describes the tasks involved in managing implementations using the Endeca Application Controller. |
| *Forge Guide* | The essential reference for developers of the back-end of Endeca applications (the instance configuration), including Forge pipeline-related tasks. |
| *Log Server and Report Generator Guide* | Describes how to configure and run the Endeca Log Server and the Report Generator. |
| *Platform Services Installation Guide* | Describes how to install the Endeca Platform Services software and the Endeca Document Conversion Module. Covers both Windows and Linux/UNIX system requirements and installation procedures. |
| *Platform Services Migration Guide* | Provides information on migrating from previous versions of Endeca software. |
| *Relationship Discovery Guide* | Describes the tasks involved in creating an Endeca Relationship Discovery application. Relationship Discovery is a separately licensed module. |
| *Security Guide* | Describes how to implement user authentication and how to structure your data to limit access to only those users with the correct permissions. |
| *Data Foundry Expression Reference* | Describes the Data Foundry expression language, used in record manipulators in Developer Studio. |
| *Oracle Endeca Developer Studio Help* | Help (including context-sensitive help) for using Endeca Developer Studio to define all aspects of your instance configuration, including properties, dimensions, and pipelines. |
| *Forge API Guide for Perl* | Describes the classes and methods you can incorporate into Perl manipulators in Developer Studio. You can use Perl manipulators in pipelines to manipulate records. |
| *XML Reference* | Describes the XML elements contained in the XML and DTD files of the Endeca Information Transformation Layer. |
| *API reference documentation (Javadoc and .NET API reference)* | The reference documentation for the Endeca Presentation, Logging, Analytics, and Charting APIs. |

# Oracle Endeca Workbench documentation

The following table lists the documentation related to the Endeca Workbench package.

| Title | Description |
|---|---|
| *Oracle Endeca Workbench Administrator's Guide* | The essential guide for administrators of Endeca implementations and application developers who maintain and customize Workbench instances. |
| *Oracle Endeca Workbench User's Guide* | The essential guide for business users of Endeca Workbench. Describes enhancements business users can make to Endeca implementations with a focus on working with dynamic business rules, search configuration, and reports. |
| *Oracle Endeca Workbench Installation Guide* | Describes how to install the Endeca Workbench software. Covers both Windows and Linux/UNIX system requirements and installation procedures. |
| *Oracle Endeca Workbench Migration Guide* | Provides information on migrating from previous versions of Endeca software. |
| *Oracle Endeca Workbench Help* | Help (including context-sensitive help) for using Endeca Workbench to perform business-user tasks and administer an Endeca implementation. There are versions for each Workbench edition. |
| *Oracle Endeca Workbench Release Notes* | Details the changes specific to this release, including bug fixes and new features. |

# Content Assembler API documentation

The following table lists the documentation related to the Content Assembler API, used in conjunction with the Experience Manager component of Endeca Workbench.

| Title | Description |
|---|---|
| *Experience Manager Developer's Guide* | Describes the process of developing templates and other supporting tasks to enable content administrators to configure dynamic landing pages using the Endeca Experience Manager. Also describes extending Experience Manager functionality with community editors. |
| *Content Assembler API Developer's Guide* | Describes the process of developing applications with cartridges (for use with the Endeca Experience Manager), including usage of the Content Assembler API and an overview of the reference applications. Also describes extending Content Assembler functionality with community tag handlers. There are versions for Java and .NET. |
| *API reference documentation (Javadoc and .NET API reference)* | The reference documentation for the Endeca Content Assembler APIs. |
| *Experience Manager Editor API reference* | The reference documentation for the Experience Manager Editor API, part of the Experience Manager Editor SDK. |

| Title | Description |
|---|---|
| *Content Assembler Release Notes* | Details the changes specific to this release, including bug fixes and new features. |

# Content Acquisition System (CAS) documentation

The following table lists the documentation related to the Content Acquisition System (CAS) package.

| Title | Description |
|---|---|
| *CAS Quick Start Guide* | Describes the basics of the Endeca Content Acquisition System (CAS) and then walks you through the high-level process of installing Endeca with CAS, adding manipulators, crawling data sources, and processing the Endeca records in a Forge pipeline. |
| *CAS Console Help* | Describes the tasks involved in managing various data sources including file systems, Documentum repositories, and other CMS repositories using the CAS Console for Endeca Workbench. |
| *CAS Developer's Guide* | Provides an overview of the Endeca Content Acquisition System, including the Endeca CAS Server, the Component Instance Manager, and the Record Store. The guide also explains how to create a Forge pipeline that utilizes the source data gathered from file system and CMS crawls. |
| *CAS Extension API Guide* | Describes how to implement, test, and package CAS extensions using the CAS Extension API. |
| *CAS Installation Guide* | Describes how to install the Endeca CAS software. Covers both Windows and Linux/UNIX system requirements and installation procedures. |
| *CAS Migration Guide* | Describes the major migration tasks for the suite of CAS components. |
| *CAS API Guide* | Provides reference information about the Endeca CAS Server API, the Component Instance Manager API, and the Record Store API. |
| *CMS Connector Guides* | Describe the tasks involved in enabling and configuring the various CMS connectors for use with the CAS Server. |
| *Web Crawler Guide* | Describes the major tasks involved in configuring the Endeca Web Crawler and using it to run crawls that gather source data from Web sites. |
| *CAS Release Announcement* | Describes the major new features in this release. |
| *CAS Release Notes* | Details the changes specific to this release, including bug fixes and new features. |

# Rapid Application Development (RAD) Toolkit documentation

The following table lists the documentation related to the Rapid Application Development (RAD) Toolkit for ASP.NET.

| Title | Description |
|---|---|
| *RAD Toolkit Developer's Guide* | The essential guide for developers of the front-end of Endeca applications (primarily API-related tasks). Also includes information about installation tasks. |
| *RAD Toolkit Release Announcement* | Describes the major new features in this release. |
| *RAD Toolkit Release Notes* | Details the changes specific to this release, including bug fixes and new features. |
| *API reference documentation (.NET API reference)* | The reference documentation for the Endeca RAD API. See also the Input Types and Output Types diagrams for additional information about the API. |

# Documentation for other packages

The following table lists the documentation related to other Endeca packages.

### Deployment Template

| Title | Description |
|---|---|
| *Oracle Endeca Deployment Template Usage Guide* | Describes the Deployment Template directories and script functionality, and identifies touch-points where developers may need to configure or extend the template for their projects. |
| *Release Notes (README)* | Details the changes specific to this release, including bug fixes and new features. |

### Developer Studio

| Title | Description |
|---|---|
| *Oracle Endeca Developer Studio Installation Guide* | Provides an overview of Developer Studio and describes system requirements and installation procedures. |
| *Oracle Endeca Developer Studio Help* | Help (including context-sensitive help) for using Endeca Developer Studio to define all aspects of your instance configuration, including properties, dimensions, and pipelines. |
| *Data Foundry Expression Reference* | Describes the Data Foundry expression language, used in record manipulators in Developer Studio. |
| *Forge API Guide for Perl* | Describes the classes and methods you can incorporate into Perl manipulators in Developer Studio. You can use Perl manipulators in pipelines to manipulate records. |

| Title | Description |
|---|---|
| *XML Reference* | Describes the XML elements contained in the XML and DTD files of the Endeca Information Transformation Layer. |

**Search Engine Optimization Module**

| Title | Description |
|---|---|
| *Sitemap Generator Developer's Guide* | Describes the Endeca Sitemap Generator and provides instructions for using it to generate sitemaps for an Endeca application. |
| *URL Optimization API Developer's Guide* | Describes the major tasks involved in developing an application that utilizes the Endeca URL Optimization API. There are versions for Java, the Presentation API for ASP.NET, and the RAD Toolkit for ASP.NET. |
| *API reference documentation (Javadoc and .NET API reference)* | The reference documentation for the URL Optimization APIs. |

Appendix B

# Endeca Environment Variables and Port Usage

This section lists all the environment variables and ports used by the Endeca software. Depending on which components you have installed, not all of them may apply to your implementation.

## Endeca environment variables

The Endeca installation programs create several environment variables.

For each variable, the first value listed is the path if you accept the default installation path on Windows (under `C:\Endeca\`*`product`* ) and use a per-machine installation. The default paths for a per-user installation will be rooted in the `%USERPROFILE%` directory.

The second value is the path within your installation directory on UNIX. For example, if you install Endeca to `/usr/local/`, the full path of `ENDECA_ROOT` would be `/usr/local/endeca/Platform¬ Services/`*`version`* in your environment.

In addition to creating the variables below, the installation may add Endeca directories to the `PATH` variable.

**Note:** For the MDEX Engine installation, environment and `PATH` variables are set by running the `mdex_setup` scripts provided by the installation. See the *Oracle Endeca MDEX Engine Installation Guide* for more information.

### MDEX Engine variables

The following variable is used by the MDEX Engine:

| Variable | Description | Default value |
|---|---|---|
| `ENDECA_MDEX_ROOT` | Specifies the path of the MDEX Engine root directory. | • `C:\Endeca\MDEX\`*`version`*<br>• `endeca/MDEX/`*`version`* |

### Platform Services variables

The following variables are used by the Platform Services:

| Variable | Description | Default value |
|---|---|---|
| ENDECA_ROOT | Specifies the path of the Platform Services root directory. | • C:\Endeca\PlatformSer¬ vices*version*<br>• endeca/PlatformServices/*ver¬ sion* |
| ENDECA_REFERENCE_DIR | Specifies the path of the directory that contains the Endeca reference implementations, such as the sample wine project and the JSP and .NET UI references. | • C:\Endeca\PlatformSer¬ vices\reference<br>• endeca/PlatformServices/ref¬ erence |
| ENDECA_CONF | Specifies the path of the workspace directory for the Endeca HTTP service, which contains configuration files, logs, and temporary storage directories. | • C:\Endeca\PlatformSer¬ vices\workspace<br>• endeca/PlatformSer¬ vices/workspace |
| PERLLIB | Specifies the path of the perl root directory and its directory of libraries. | • %ENDECA_ROOT%\perl and %ENDE¬ CA_ROOT%\perl\5.8.3\lib<br>• $ENDECA_ROOT/lib/perl:$ENDE¬ CA_ROOT/lib/perl/Control:$EN¬ DECA_ROOT/perl/lib:$ENDE¬ CA_ROOT/perl/lib/site_perl |
| PERL5LIB | Same as the PERLLIB variable. | Same as the PERLLIB variable. |
| UnixUtils | Specifies the path of the utilities directory, which contains Windows versions of some UNIX common utilities. | • %ENDECA_ROOT%\utilities<br>• not available on UNIX |

### Endeca Workbench variables

The following variables are used by the Endeca Workbench:

| Variable | Description | Default value |
|---|---|---|
| ENDE¬ CA_TOOLS_ROOT | Specifies the path of the Endeca Workbench root directory. | • C:\Endeca\Workbench*version*<br>• endeca/Workbench/*version* |
| ENDE¬ CA_TOOLS_CONF | Specifies the path of the workspace directory for the Endeca Tools Service, which | • C:\Endeca\Workbench\workspace<br>• endeca/Workbench/workspace |

| Variable | Description | Default value |
|---|---|---|
| | contains configuration files, logs, and temporary storage directories. | |

### Other variables

Other variables used by Endeca include the following:

| Variable | Description | Default value |
|---|---|---|
| `ENDECA_PROJECT_DIR` | Specifies the path of the deployed application. This variable is set and used by the Endeca Deployment Template. | Value is taken from user input at installation time. |
| `ENDECA_PROJECT_NAME` | Specifies the project name that is used, for example, as the JCD job prefix for jobs defined in the project's Job Control Daemon. This variable is set and used by the Endeca Deployment Template. | Value is taken from user input at installation time. |

# Endeca ports

This topic describes the ports used by the Endeca packages and their default port numbers.

You can replace any of the default port numbers with numbers of your own, as long as they do not conflict with an existing port on your machine. Port numbers can be no larger than `32767`.

### Service ports

| Port | Default |
|---|---|
| Endeca Tools Service port | `8006` |
| Endeca Tools Service SSL port | `8446` |
| Endeca Tools Service shutdown port | `8084` |
| CAS Service port | `8500` |
| CAS Service shutdown port | `8506` |
| Endeca HTTP Service port | `8888` |
| Endeca HTTP Service SSL port | `8443` |
| Endeca HTTP Service shutdown port | `8090` |
| Endeca Control System JCD port | `8088` |

| Port | Default |
|---|---|
| **Note:** The JCD is deprecated. | |

### Deployment Template ports

These are the port numbers suggested by the Deployment Template installation, but you can specify any other port when you deploy your application.

| Port | Default |
|---|---|
| Dgraph1 user query port | `15000` |
| Dgraph2 user query port | `15001` |
| Agraph1 user query port (Agraph deployments only) | `14000` |
| Agraph2 user query port (Agraph deployments only) | `14001` |
| Forge server (Agraph deployments with Parallel Forge only) | `14099` |
| Endeca Logging and Reporting Server port<br>**Note:** The Logging Server port number can be no larger than `32767`. | `15010` |

### Reference implementation ports

These port numbers are used in the configuration files that ship with the reference implementation (`sample_wine_data`).

| Port | Default |
|---|---|
| Endeca MDEX Engine user query port | `8000` |
| Endeca Logging and Reporting Server port<br>**Note:** The Logging Server port number can be no larger than `32767`. In the JSP reference implementation, the default Logging server port number is larger by 2 than the corresponding Dgraph port number. For example, for the Dgraph port 15000, the default port for the Logging Server in the reference implementation is 15002. For the Dgraph port 15001, the default port for the Logging Server in the reference implementation is 15003. (This assumes that the Logging Server is running on the same host as the MDEX Engine.) | `8002` |

# Index

## R



## S



## T



## V