

Oracle® Endeca Server

Administrator's Guide

Version 7.5.1.1 • May 2013

Copyright and disclaimer

Copyright © 2003, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners. UNIX is a registered trademark of The Open Group.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Table of Contents

Copyright and disclaimer	2
Preface	5
About this guide	5
Who should use this guide	5
Conventions used in this guide	5
Contacting Oracle Customer Support	6
Chapter 1: Introduction	7
Taking ownership of your implementation	7
Overview of administrator tasks	8
Chapter 2: Oracle Endeca Server	9
Overview of Oracle Endeca Server application	9
Endeca Server configuration	10
Stopping Endeca Server	14
Starting Endeca Server	17
About the data domain and the Dgraph process	19
Data domain monitoring	20
Chapter 3: Endeca Server Command Reference	22
About the endeca-cmd utility	22
Data domain operations	23
Endeca Server node operations	29
Data domain profile operations	31
Command global options	35
Global options for host, port, and context root	36
Global options for SSL	37
Chapter 4: Dgraph Administrative Tasks	39
Updating spelling dictionaries	39
Checking the status of the Dgraph nodes	40
Tuning the Dgraph process cache	40
About connecting Web browsers to the data domain	41
Managing Dgraph core dump files	41
Managing Dgraph crash dump files on Windows	42
Managing Dgraph core dump files on Linux	42
Collecting debugging information	43
Running multiple Dgraphs on the same Windows machine	43
Identifying connection errors	43
Chapter 5: Administrative Operations	45
About administrative and configuration operations	45

List of administrative operations	46
help	47
flush	47
logroll	47
merge	47
ping	47
stats	48
statsreset	48
About logging variables for the Dgraph	48
Logging variable operation syntax	48
List of configuration operations	49
List of supported logging variables	49
log-enable	50
log-disable	50
log-status	50
help	51
Chapter 6: Managing the Merge Policy	52
Using a merge policy for incremental updates	52
Types of merge policies	52
Setting or changing the merge policy	53
Setting the merge policy with the Configuration Web Service	53
Retrieving the merge policy with the Configuration Web Service	53
Setting the merge policy with the Configuration Web Service	54
Changing the merge policy of a running data domain	54
Forcing a merge	55
Linux unlimit settings for merges	56
Chapter 7: Endeca Server Logging	57
Endeca Server logs	57
Log entry format	58
Setting log levels	61
Creating log handlers for debugging	62
Customizing the HTTP access log	63
Dgraph logs	65
Appendix A: Dgraph Flag Reference	68
Dgraph flags	68

Preface

Oracle® Endeca Server is the core search-analytical database. It organizes complex and varied data from disparate source systems into a faceted data model that is extremely flexible and reduces the need for up-front data modeling. This highly-scalable server enables users to explore data in an unconstrained and impromptu manner and to rapidly address new questions that inevitably follow every new insight.

About this guide

This guide describes the administrative tasks for the Oracle Endeca Server application running in a WebLogic Server.

Who should use this guide

This guide is intended for system administrators who administer and maintain an Oracle Endeca Server application and the data domains running in it.

This guide assumes that the Oracle Endeca Server software is already installed on a development server. It may be already installed in a production environment. It also assumes that you, or your Oracle Services representatives, have already configured the application on the development server.

You can choose to read specific topics from this guide individually as needed while maintaining your Oracle Endeca Server implementation after it has been initially deployed.

Conventions used in this guide

The following conventions are used in this document.

Typographic conventions

This table describes the typographic conventions used when formatting text in this document.

Typeface	Meaning
User Interface Elements	This formatting is used for graphical user interface elements such as pages, dialog boxes, buttons, and fields.
Code Sample	This formatting is used for sample code phrases within a paragraph.
<i>Variable</i>	This formatting is used for variable values. For variables within a code sample, the formatting is <i>Variable</i> .
File Path	This formatting is used for file names and paths.

Symbol conventions

This table describes the symbol conventions used in this document.

Symbol	Description	Example	Meaning
>	The right angle bracket, or greater-than sign, indicates menu item selections in a graphic user interface.	File > New > Project	From the File menu, choose New, then from the New submenu, choose Project.

Path variable conventions

This table describes the path variable conventions used in this document.

Path variable	Meaning
\$MW_HOME	Indicates the absolute path to your Oracle Middleware home directory, which is the root directory for your WebLogic installation.
\$DOMAIN_HOME	Indicates the absolute path to your WebLogic domain home directory. For example, if <code>endeca_server_domain</code> is the name of your WebLogic domain, then the <code>\$DOMAIN_HOME</code> value would be the <code>\$MW_HOME/user_projects/domains/endeca_server_domain</code> directory.
\$ENDECA_HOME	Indicates the absolute path to your Oracle Endeca Server home directory, which is the root directory for your Endeca Server installation.

Contacting Oracle 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>.



Chapter 1

Introduction

This section describes the stage at which you take control of the operation and maintenance of your application powered by the Oracle Endeca Server.

[Taking ownership of your implementation](#)

[Overview of administrator tasks](#)

Taking ownership of your implementation

As a system administrator, you take ownership of the Oracle Endeca Server implementation at a certain stage. This topic describes the context in which you will perform administrative tasks to maintain the stable operation of a properly functioning implementation.

This guide assumes that by this point in using the Oracle Endeca Server, you or your team have done the following:

- Planned and provisioned the hardware needed for the staging and production environments.
- Installed the WebLogic Server, the Application Developer Framework Runtime, and the Oracle Endeca Server.
- Read the *Oracle Endeca Information Discovery Getting Started Guide* (if you are planning to use Oracle Endeca Information Discovery Studio).
- Planned the user-facing details of your application, such as the attributes that will be displayed, the search interfaces, groups, and other features.

In addition, the guide assumes that you have performed the following application-building tasks:

- You have completed the process of extracting source information from your incoming data sources.
- You have completed the process of loading your configuration schema and your source data into the Oracle Endeca Server, thus creating the Oracle Endeca Server index.
- You have created a working prototype of your front-end application for your end users. This front-end application can be used to issue requests to the running Oracle Endeca Server in a production environment.
- You have deployed your application powered by the Oracle Endeca Server in a staging environment, and are either preparing to deploy it in production, or have already deployed it in production.

Overview of administrator tasks

This topic provides a brief overview of the administrator tasks described in this guide.

This guide assumes that you are performing administrator tasks on the Oracle Endeca Server. The types of tasks described in this guide are the following (as grouped by their section):

Chapter	Tasks
Endeca Server Administrative Tasks	<ul style="list-style-type: none"> • Start the Endeca Server application running in the WebLogic Server. • Stop the Endeca Server application running in the WebLogic Server. • Use <code>endeca-cmd</code> to: <ul style="list-style-type: none"> • Create, manage, and monitor the status of the Endeca data domains. • Manage and monitor the status of the Endeca Server nodes (if you are running more than one Endeca Server instance, such as in a cluster). • Create and manage data domain profiles and the Endeca Server node profile.
Dgraph Administrative Tasks	<ul style="list-style-type: none"> • Check the Dgraph process. • Manage Dgraph process core dump files. • Collect debugging information to help solve problems. • Troubleshoot Dgraph process socket and port errors. • Identify Dgraph process connection errors.
Administrative Operations and Logging Variables	<ul style="list-style-type: none"> • Flush the dynamic cache. • Force a query log roll. • Merge update generations and set the system's merge policy. • Check the Dgraph process Statistics page. • Modify the logging configuration for the Dgraph process.
Managing the Merge Policy	<ul style="list-style-type: none"> • Merge Dgraph index update generations. • Set and manage the merge policy for an Endeca data domain.



Chapter 2

Oracle Endeca Server

This section provides an overview of the Oracle Endeca Server Java application and describes administration and configuration tasks for it.

[Overview of Oracle Endeca Server application](#)

[Endeca Server configuration](#)

[Stopping Endeca Server](#)

[Starting Endeca Server](#)

[About the data domain and the Dgraph process](#)

[Data domain monitoring](#)

Overview of Oracle Endeca Server application

The Endeca Server is a Java application hosted in the WebLogic Server container that manages the Endeca data domains.



Note: It is useful to recognize that the term "Endeca Server" may refer to the Endeca Server software package, and to the Endeca Server Java application hosted in the WebLogic Server. Whenever this distinction is needed, the documentation refers to the software package as "the Endeca Server", and to the Java application as the "Endeca Server application".

You install the Endeca Server package on Linux or Windows machines that are running the WebLogic Server and that will host multiple Endeca data domains. Once the Endeca Server package is installed in the WebLogic Server, you use the WebLogic Server's Administration Console to manage the Endeca Server application.

The Endeca Server application manages the Endeca data domain clusters hosted in the Endeca Server cluster.



Note: On Windows, the Endeca Server application cannot be configured to run as a Windows service. However, a WebLogic Server instance can be configured to run as a Windows service. For details, see the WebLogic guide titled *Managing Server Startup and Shutdown* at: http://docs.oracle.com/cd/E23943_01/web.1111/e13708/winservice.htm#i1186180.

Endeca Server has a set of commands with which you create and control the data domains. For information, see [Endeca Server Command Reference on page 21](#).

Endeca Server configuration

A configuration file sets global parameters for Endeca Server, such as the default locations of mandatory files and directories.

The Endeca Server configuration file is named `EndecaServer.properties`, and is located in the `config` directory of your domain. For example, assuming that "endeca_domain" is the name of your WebLogic Server domain for the Endeca Server, the default location on Linux is:

```
$MW_HOME/user_projects/domains/endeca_domain/config/EndecaServer.properties
```

The path on Windows is similar.

The configuration includes parameters for the Endeca Server application (including the Cluster Coordinator). The default values in the file are set by the Endeca Server installer.

Note that the file's SSL-specific parameters (such as `endeca-ssl-keystore`) are empty at installation time. The values are filled in if you later run the `generate_ssl_keys` utility that creates the SSL keys and certificates.

Endeca Server parameters

The following configuration settings are specific to Endeca Server operations:

Endeca Server parameter	Meaning
<code>endeca-require-https</code>	If set to <code>true</code> , specifies that Endeca Server enforces the use of the HTTPS port. If set to <code>false</code> , uses the HTTP port.
<code>endeca-runtime-basedir</code>	The location of the Endeca Server's <code>endeca-server</code> directory. The default is the <code>\$ENDECA_HOME/endeca-server</code> location.
<code>endeca-data-dir</code>	The location where the data files for all the Endeca data domains are stored for this Endeca Server instance. The default location is the <code>\$DOMAIN_HOME/EndecaServer/data</code> directory. Note that you can specify only one directory (that is, specifying multiple directories is not allowed).
<code>endeca-offline-dir</code>	The location where the data files for an Endeca data domain are exported by the <code>export-dd endeca-cmd</code> . The default is the <code>\$DOMAIN_HOME/EndecaServer/offline</code> directory. Note that you can specify only one directory (that is, specifying multiple directories is not allowed).
<code>endeca-logs-dir</code>	The location of the Dgraph's standard out/err log and request log, as well as the PID file. The default is the <code>\$DOMAIN_HOME/EndecaServer/logs</code> directory. Note that you can specify only one directory (that is, specifying multiple directories is not allowed).

Endeca Server parameter	Meaning
endeca-dgraph-install	<p>The location of the Endeca Server's Dgraph application directory.</p> <p>The default is the <code>\$ENDECA_HOME/endeca-server/dgraph</code> directory.</p>
endeca-webserver-port	<p>The port on which the Endeca Server listens for requests.</p> <p>The default is 7001 (for non-SSL deployments) or 7002 (for SSL deployments).</p>
endeca-ds-port-min and endeca-ds-port-max	<p>The range of port numbers from which the Endeca Server selects the HTTP/HTTPS ports and the bulk-load ports for the Dgraph processes.</p> <p>Once a port is assigned to a Dgraph process, that port is not used for subsequent Dgraph port assignments (unless the data domain is disabled). This port assignment strategy prevents port collisions among the Dgraph processes.</p> <p>The defaults for <code>endeca-ds-port-min</code> are 7011 (for non-SSL deployments) or 7012 (for SSL deployments).</p> <p>The defaults for <code>endeca-ds-port-max</code> are 8011 (for non-SSL deployments) or 8012 (for SSL deployments).</p>
endeca-ds-cert-file	<p>The path of the SSL certificate file. This certificate file is used by all clients and servers to specify their identity when using SSL to connect to the Oracle Endeca Server.</p> <p>If the <code>generate_ssl_keys</code> utility was run, the default path is to the <code>\$DOMAIN_HOME/config/ssl/dgraphCert.pem</code> file. If the utility was not run, this setting is empty.</p>
endeca-ds-ca-file	<p>The path of the certificate authority file. This file is used by all clients and servers to authenticate the other endpoint of a communication channel.</p> <p>If the <code>generate_ssl_keys</code> utility was run, the default path is to the <code>\$DOMAIN_HOME/config/ssl/dgraphCA.pem</code> file. If the utility was not run, this setting is empty.</p>
endeca-secure-mode	<p>If set to <code>true</code>, specifies that Endeca Server is using SSL and that the Dgraphs should be started in SSL mode.</p> <p>If set to <code>false</code>, Endeca Server and the Dgraphs are not using SSL.</p>

Endeca Cluster Coordinator parameters

The following configuration settings are specific to the Endeca Cluster Coordinator:

Cluster Coordinator parameter	Meaning
<code>endeca-cluster-coordinator-dir</code>	The location of the Endeca Cluster Coordinator root directory. The default is the <code>\$ENDECA_HOME/endeca-server/cluster-coordinator</code> directory.
<code>endeca-cluster-coordinator-hosts</code>	The default is <code>localhost</code> (which should be used for a single-node installation). For a clustered deployment, this parameter is a comma-separated list of the machine names that are part of the Cluster Coordinator ensemble.
<code>endeca-cluster-coordinator-clientPort</code>	The port of the server on which the Cluster Coordinator is running. This the port at which the clients will connect. The default is 2181.
<code>endeca-cluster-coordinator-tickTime</code>	The length of a single tick, which is the basic time unit for the Cluster Coordinator. A tick is measured in milliseconds. It is used to regulate heartbeats and timeouts. The default is 2000 milliseconds.
<code>endeca-cluster-coordinator-initLimit</code>	The number of ticks that the initial synchronization phase can take. This number specifies the length of time the nodes have to connect to the leader node. The default is 10 ticks.
<code>endeca-cluster-coordinator-syncLimit</code>	The number of ticks that can take place between one node sending a request for an update and receiving an acknowledgment from the leader node. The default is 5 ticks.
<code>endeca-cluster-coordinator-dataDir</code>	The location of the directory where the in-memory database snapshots for the Cluster Coordinator and the transaction log of updates to its database are stored. The default is the <code>\$DOMAIN_HOME/EndecaServer/data</code> directory.
<code>endeca-cluster-coordinator-serverPort</code>	The port that follower nodes use to connect to the leader node. The default server port is 3181.

Cluster Coordinator parameter	Meaning
<code>endeca-cluster-coordinator-leaderPort</code>	The port that the Cluster Coordinator servers use to do leader election. The default leader port is 4181.
<code>endeca-cluster-coordinator-maxClientCnxns</code>	Limits the number of concurrent connections (at the socket level) that a single client, identified by IP address, may make to a single member of the cluster ensemble. This is used to prevent certain classes of DoS attacks, including file descriptor exhaustion. Setting this to 0 (the default) entirely removes the limit on concurrent connections.
<code>endeca-cluster-coordinator-minSessionTimeout</code>	Specifies the minimum session timeout in milliseconds that the server will allow the client to negotiate. The default is 4000 milliseconds. Note that this parameter is intended for use by Oracle Endeca Support only.
<code>endeca-cluster-coordinator-maxSessionTimeout</code>	Specifies the maximum session timeout in milliseconds that the server will allow the client to negotiate. The default is 180000 milliseconds. Note that this parameter is intended for use by Oracle Endeca Support only.



Note: Most of these parameters are used by the Endeca Server application and should not be modified. During the Endeca Server cluster deployment process, you may need to modify some of these parameters. For information on which parameters can be modified for the Endeca Server cluster deployment, see the *Oracle Endeca Server Installation Guide*.

Using a data directory on a different drive

The `endeca-data-dir` parameter allows you to specify a different drive for the data domain data files. (The default drive is the same one on which the Endeca Server application directory is installed.) On Windows, for example, your WebLogic Server and Endeca Server installations can be on drive C, while the data domain indexes can be created on drive D. They can also be placed on a shared file system, as in a clustered environment.

There is one caveat to keep in mind. If you have created data domains and later want to change the `endeca-data-dir` setting, any data domain created with the original `endeca-data-dir` setting cannot be started with the new (changed) `endeca-data-dir` setting (because its data files cannot be found at start-up). Therefore, to preserve the original data domains, do the following:

1. Use the `export-dd` command to export the data domains you want to keep.
2. Use the `delete-dd` command to delete all data domains (as they will not be able to be started).

3. Change the `endeca-data-dir` value to a different directory.
4. Use the `import-dd` command to import the data domains.

When making these changes, keep the following in mind:

- You can specify only one directory for the `endeca-data-dir` parameter. In other words, the data files for all of the data domains created by an Endeca Server instance must all be stored in the same directory.
- In a clustered environment, all the Endeca Server instances must use identical `EndecaServer.properties` files.
- You cannot mix Linux and Windows drives. For example, if the Endeca Server application directory is installed on a Linux drive, you cannot place the data files on a Windows drive.

You can also use the `endeca-logs-dir` parameter to store the Dgraph logs in a different directory or drive, and use the `endeca-offline-dir` parameter for exporting and importing data domains.

Stopping Endeca Server

Stopping the Oracle Endeca Server application makes it unavailable for WebLogic Server clients.

You can manually stop the Oracle Endeca Server application from the WebLogic Server Administration Console. Note that it is not necessary to stop Endeca Server in order to shut down WebLogic Server; in this case, WebLogic Server will stop Endeca Server as part of its shut-down procedure.

To stop the Oracle Endeca Server application:

1. Make sure that the Administration Server for the Endeca domain is running.
2. From your browser, access the Administration Server console using this syntax:

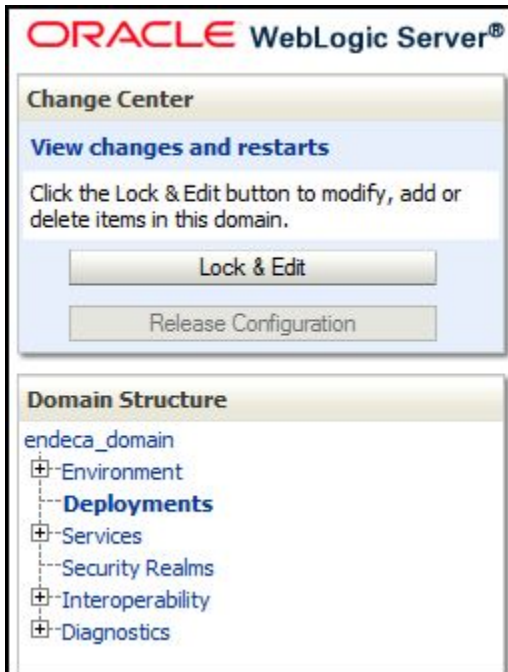
```
http://admin_server_host:admin_server_port/console
```

For example:

```
http://web007:7001/console
```

3. At the Administration Console login screen, log in with the administrator user name and password.

4. In the Domain Structure pane, click **Deployments** (in the `endeca_domain` tree).



- In the Deployments table, check the **oracle.endecaserver** Web application. Its State should be "Active" and its Health should be "OK", as in this abbreviated example:

Summary of Deployments

Control | Monitoring

This page displays a list of Java EE applications and stand-alone application modules that have been installed to this domain. Installed applications and modules can be started, stopped, updated (redeployed), or deleted from the domain by first selecting the application name and using the controls on this page.

To install a new application or module for deployment to targets in this domain, click the Install button.

[Customize this table](#)

Deployments

Install | Update | Delete | Start v | Stop v Showing 11 to 20 of 26 Previous | Next

<input type="checkbox"/>	Name ^	State	Health	Type	Deployment Order
<input type="checkbox"/>	oracle.adf.desktopintegration.model(1.0,11.1.1.2.0)	Active		Library	100
<input type="checkbox"/>	oracle.adf.management(1.0,11.1.1.2.0)	Active		Library	100
<input type="checkbox"/>	oracle.bi.adf.model.slib(1.0,11.1.1.2.0)	Active		Library	100
<input type="checkbox"/>	oracle.bi.adf.view.slib(1.0,11.1.1.2.0)	Active		Library	100
<input type="checkbox"/>	oracle.bi.adf.webcenter.slib(1.0,11.1.1.2.0)	Active		Library	100
<input type="checkbox"/>	oracle.bi.composer(11.1.1.0.1)	Active		Library	100
<input type="checkbox"/>	oracle.bi.jbips(11.1.1.0.1)	Active		Library	100
<input type="checkbox"/>	oracle.dconfig-infra(11,11.1.1.1.0)	Active		Library	100
<input checked="" type="checkbox"/>	oracle.endecaserver (1.0,7.5.1.0.0)	Active	OK	Enterprise Application	100
<input type="checkbox"/>	oracle.jrf.system.filter	Active		Library	100

Install | Update | Delete | Start v | Stop v Showing 11 to 20 of 26 Previous | Next

- In the Deployments table, click **Stop**, and select one of the stop options:
 - When work completes:** Specifies that WebLogic Server waits for the Endeca Server application to finish its work and for all currently connected users to disconnect.
 - Force Stop Now:** Specifies that WebLogic Server stops the Endeca Server application immediately, regardless of the work that is being performed and the users that are connected.
 - Stop, but continue servicing administration requests:** Specifies that WebLogic Server stops the Endeca Server application once all its work has finished, but then puts the application in Administration Mode so it can be accessed for administrative purposes.
- In the Stop Application Assistant, click **Yes**.

As a result, the Endeca Server application is stopped and its State now changes to "Prepared".



Note: If the Endeca Server application is in a "Prepared" (i.e., stopped) state when you shut down WebLogic Server, then the application is not automatically re-started when you start WebLogic Server. In this case, you must manually start Endeca Server.

Note that if you forcibly shut down any of the WebLogic Servers in the WebLogic domain that is hosting the Endeca Server cluster, there is a chance that the Dgraph processes or the Cluster Coordinator processes from the Endeca Server will be left running (this is because the Cluster Coordinator process runs in a separate JVM and the Dgraph process is not a Java process). Therefore, upon subsequent startup of the WebLogic Server, the Endeca Server software checks whether in this WebLogic domain any Dgraph or Cluster Coordinator processes are running and then shuts them down.

Starting Endeca Server

Starting the Oracle Endeca Server application makes it available for WebLogic Server clients.

When you start the WebLogic Server in which the Endeca Server application is deployed, it automatically starts the Endeca Server application. If the application was running when WebLogic Server was shut down, WebLogic Server automatically re-starts the application as part of its start-up procedure. Additionally, you can manually start the Oracle Endeca Server application from the WebLogic Server Administration Console.

To start a stopped Oracle Endeca Server application:

1. Make sure that the Administration Server for the Endeca domain is running.
2. From your browser, access the Administration Server console using this syntax:

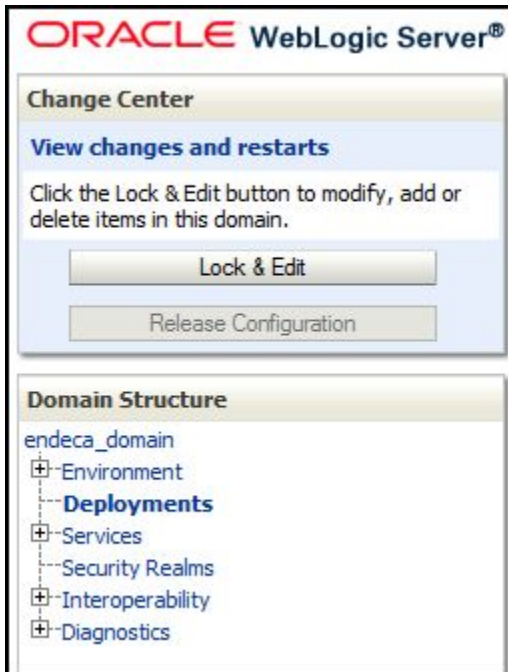
```
http://admin_server_host:admin_server_port/console
```

For example:

```
http://web007:7001/console
```

3. At the Administration Console login screen, log in with the administrator user name and password.

4. In the Domain Structure pane, click **Deployments** (in the `endeca_domain` tree).



- In the **Deployments** table, check the **oracle.endecaserver** Web application. Its State should be "Prepared" and its Health should be "OK", as in this abbreviated example:

Summary of Deployments

Control | Monitoring

This page displays a list of Java EE applications and stand-alone application modules that have been installed to this domain. Installed applications and modules can be started, stopped, updated (redeployed), or deleted from the domain by first selecting the application name and using the controls on this page.

To install a new application or module for deployment to targets in this domain, click the Install button.

[Customize this table](#)

Deployments

Install | Update | Delete | Start ▾ | Stop ▾ | Showing 11 to 20 of 26 | Previous | Next

<input type="checkbox"/>	Name ↕	State	Health	Type	Deployment Order
<input type="checkbox"/>	oracle.adf.desktopintegration.model(1.0,11.1.1.2.0)	Active		Library	100
<input type="checkbox"/>	oracle.adf.management(1.0,11.1.1.2.0)	Active		Library	100
<input type="checkbox"/>	oracle.bi.adf.model.slib(1.0,11.1.1.2.0)	Active		Library	100
<input type="checkbox"/>	oracle.bi.adf.view.slib(1.0,11.1.1.2.0)	Active		Library	100
<input type="checkbox"/>	oracle.bi.adf.webcenter.slib(1.0,11.1.1.2.0)	Active		Library	100
<input type="checkbox"/>	oracle.bi.composer(11.1.1.0.1)	Active		Library	100
<input type="checkbox"/>	oracle.bi.jbips(11.1.1.0.1)	Active		Library	100
<input type="checkbox"/>	oracle.dconfig-infra(11,11.1.1.1.0)	Active		Library	100
<input checked="" type="checkbox"/>	oracle.endecaserver (1.0,7.5.1.0.0)	Prepared	OK	Enterprise Application	100
<input type="checkbox"/>	oracle.jrf.system.filter	Active		Library	100

Install | Update | Delete | Start ▾ | Stop ▾ | Showing 11 to 20 of 26 | Previous | Next

- In the Deployments table, click **Start>Servicing all requests** (which makes the application immediately available to all WebLogic Server clients). You can also choose the **Servicing only administration requests** option, which makes the application available in Administration Mode only.
- In the Stop Application Assistant, click **Yes**.

As a result, the Endeca Server application is started and its State now changes to "Active".

About the data domain and the Dgraph process

The Endeca Server application services requests for one or more Endeca data domains.

An Endeca **data domain** is a set of one or more Dgraph processes that together handle end-user query requests. One of the Dgraph processes in a data domain is responsible for handling all write operations (updates, configuration changes), while the remaining Dgraph processes serve as read-only. All Dgraph nodes

in a given data domain typically utilize the same index residing on shared storage. Each data domain node is represented by the Dgraph process and is typically hosted on a separate Endeca Server instance application running in an Endeca Server cluster.

The Dgraph process for the data domain is referred to as the **data domain node**.

When you first install the Endeca Server software, it does not contain Endeca data domains. You create data domain profiles and use them to create named data domains.

If you configure a data domain to run on a single Endeca Server instance, it typically also requires a single node that is started once you start the Endeca Server application.

If you configure a data domain to run multiple Dgraph processes (such data domains are described as "data domains with multiple nodes"), then the Endeca Server cluster locates ports on which to start the Dgraph processes for the data domain and identifies which Dgraph configuration flags to use.

Once an Endeca data domain is created, you only need to use the name of the data domain to manage it. You do not need to know which port the Dgraph processes for the data domain are running on, as the Endeca Server keeps track of that information using its Cluster Coordinator services. This name-only reference to the data domains makes it much easier to enable and disable them and perform other data domain management operations.



Note: A data domain configured to run several Dgraph processes is typically deployed in an Endeca Server cluster. Such a data domain is also known as a **data domain cluster**. Both types of data domains — the one that runs a single Dgraph process and the one that runs multiple processes — are managed by the Endeca Server application. Thus, in cases when a multi-process data domain is implied, the term "data domain" may be used interchangeably with the term "data domain cluster".

For information on data domain clusters and the Endeca Server cluster, see the *Oracle Endeca Server Cluster Guide*.

About the Dgraph process

The Dgraph uses proprietary data structures and algorithms that allow it to provide real-time responses to client requests. It stores the index created from loading the data into it. After the index is stored, the Dgraph receives client requests via the application tier, queries the index, and returns the results. The communication between the Endeca Server and the Dgraph is secure by default.

The Dgraph is designed to be stateless. This design requires that a complete query be sent to it for each request. The stateless design facilitates the addition of Dgraph processes for load balancing and redundancy — any replica of a Dgraph can reply to queries independently of other replicas.

Data domain monitoring

The Endeca Server is responsible for keeping each Endeca data domain up and running, and therefore constantly monitors each data domain.

If an Endeca data domain crashes, the Endeca Server attempts to re-start it three times. If the re-start attempts are unsuccessful, the Endeca Server does not try to re-start it again.

As an administrator, you should periodically check the data domain logs to make sure the data domains are running properly. These logs are:

- `<dataDomain>.out` (for example, `bikes.out`) is the stdout/stderr log for a specific Endeca data domain.
- `<dataDomain>.reqlog` is the request log for a specific Endeca data domain.

By default, these logs are stored in the `$DOMAIN_HOME/EndecaServer/logs` directory.



Chapter 3

Endeca Server Command Reference

This section describes the commands of the `endeca-cmd` command-line interface.

[About the `endeca-cmd` utility](#)

[Data domain operations](#)

[Endeca Server node operations](#)

[Data domain profile operations](#)

[Command global options](#)

About the `endeca-cmd` utility

The Endeca Server has a command-line interface that lets you control nodes in the Endeca Server cluster, data domains hosted in the Endeca Server cluster, and the data domain nodes.

With the `endeca-cmd` commands, you can obtain a version of the Endeca Server, and perform operations on:

- Data domain nodes (also known as Dgraph nodes)
- Data domain profiles
- Endeca Server nodes

The `endeca-cmd` utility requires a Java run-time environment (JRE) to run. Therefore, verify that you have included the `bin` directory of the installed JDK at the beginning of the `PATH` variable definition on your system. Alternatively, check that you have correctly set the `JAVA_HOME` environment variable.

The syntax of the `endeca-cmd` command is:

```
endeca-cmd command [global options] [command options]
```

The `endeca-cmd` command uses two Web Service interfaces of the Endeca Server: the Cluster Web Service and the Manage Web Service. This means that all `endeca-cmd` commands can be issued programmatically, either using the Web Service requests directly (such as in `soapUI`), or by generating Java stubs from the Web Service requests and then using the Java methods for generating these requests. For information on the Cluster Web Service and Manage Web Service, see the *Oracle Endeca Server Cluster Guide*.

You can issue the `endeca-cmd` commands from the directory in which the `endeca-cmd` script is installed on any of the Endeca Server nodes. There are two versions of the `endeca-cmd` script: one for SSL use and one for non-SSL use.

Non-SSL version of `endeca-cmd`

The non-SSL version of the `endeca-cmd` script resides by default in the `endeca-cmd` directory in the root of the Endeca Server installation. For example, on Windows the default path is:

```
C:\Oracle\Middleware\EndecaServer7.5.1_1\endeca-cmd
```

The location on Linux is also in the Endeca Server installation directory.

Use this version of the script if your Endeca Server deployment is installed in a non-SSL environment.

SSL version of endeca-cmd

The SSL version of the `endeca-cmd` script resides by default in the `$DOMAIN_HOME/EndecaServer/bin` directory. For example, if `endeca_domain` is the name of your WebLogic Server domain, then the default path on Windows is:

```
C:\Oracle\Middleware\user_projects\domains\endeca_domain\EndecaServer\bin
```

This version allows you to issue `endeca-cmd` commands without having to specify the location of the SSL certificates with the `--keystore` and `--truststore` options. It does so by making use of the `EndecaCmd.properties` file, which is located by default in the `$DOMAIN_HOME/config` directory.

The parameters in the `EndecaCmd.properties` file are automatically set when you run the `generate_ssl_keys` utility to generate the certificates. The parameters in the file are:

- `host` — The name of the host machine on which the Endeca Server is running (same as the `--host` option).
- `port` — The port on the host machine on which the Endeca Server is listening (same as the `--port` option).
- `root` — The context root of the Endeca Server (same as the `--root` option).
- `keystore` — The location of the keystore (same as the `--keystore` option).
- `truststore` — The location of the truststore (same as the `--truststore` option).
- `ssl` — A Boolean setting that indicates whether SSL is being used (same as the `--ssl` option).

The following is an example of a `EndecaCmd.properties` file on a Windows server:

```
host=Web007
port=7002
root=/endeca-server
keystore=C:\Oracle\Middleware\user_projects\domains\endeca_domain\config\ssl
/endecaServerClientCert.ks
truststore=C:\Oracle\Middleware\user_projects\domains\endeca_domain\config\ssl
/endecaServerTrustStore.ks
ssl=true
```

By default, you are prompted for the keystore password whenever you issue a command. This is the password that you specified with the `--sslPassphrase` flag of the `generate_ssl_keys` utility. Note that you can override the prompt by providing the password with the `--password` option.

Data domain operations

These commands operate on Endeca data domains. For example, they allow you to create data domains and return information about their status.

You can use the data domain operations only after you have used `endeca-cmd` to configure Endeca Server node profile and data domain profiles.

The commands for managing data domains are:

- [create-dd on page 24](#)
- [clone-dd on page 25](#)
- [delete-dd on page 25](#)
- [disable-dd on page 26](#)
- [enable-dd on page 26](#)
- [export-dd on page 26](#)
- [import-dd on page 27](#)
- [get-dd on page 27](#)
- [get-dd-health on page 28](#)
- [get-dd-status on page 28](#)
- [rescale-dd on page 28](#)
- [allocate-bulk-load-port on page 28](#)
- [list-dd on page 29](#)
- [update-spelling-dictionaries on page 29](#)
- [version on page 29](#)

create-dd

The `create-dd` command creates, registers, and starts a data domain with the specified name, using the configuration settings from either the default data domain profile or the specified data domain profile.

The syntax for this command is:

```
endeca-cmd create-dd <new-data-domain> [global-options] [create-options]
```

where *new-data-domain* is mandatory and is the name of the new Endeca data domain. The name follows these rules:

- The name must be unique among any other Endeca data domains in this Endeca Server instance (or Endeca Server cluster).
- The name cannot contain these characters: & (ampersand), | (pipe), ; (semicolon), \ (back slash), / (forward slash).
- The name cannot start with "." (a period).
- The name must be enclosed in double quotes if it contains spaces. Note that after being created, the name must be referenced within double quotes in subsequent commands.

The data files for the data domain are put in the `$DOMAIN_HOME/EndecaServer/data` directory by default. You can change this location by specifying a different directory for the `endeca-data-dir` parameter in the Endeca Server configuration file `EndecaServer.properties`.

The following additional command options can be used to change the created data domain's configuration (defaults are used otherwise):

Create Option	Meaning
<code>--dd-profile-name <i>name</i></code>	Specifies the name of the data domain profile name to be used. If not specified, defaults to using the default data domain profile.
<code>--is-enabled <i>boolean</i></code>	If set to <code>true</code> , indicates that the new data domain is enabled. If this option is not specified, the data domain is enabled. If you want to create and register but not start the data domain, specify <code>--is-enabled false</code> .

Example 1:

```
endeca-cmd create-dd MyDD
```

creates an Endeca data domain named **MyDD** using the default data domain profile. The data domain will be enabled.

Example 2:

```
endeca-cmd create-dd MyDD2 --dd-profile-name MyProfile --is-enabled false
```

creates a **MyDD2** data domain using the data domain profile named **MyProfile**. The data domain is created and registered, but not started.

clone-dd

The `clone-dd` command creates a data domain based on the existing data domain. The syntax for this command is:

```
endeca-cmd clone-dd <cloned-domain> --source-name <source-domain> [global-options] [clone-option]
```

where *cloned-domain* is the unique name of the new Endeca data domain, as copied from *source-domain*. The source data domain must be enabled before you run this command. The new data domain name follows the same naming rules as the `create-dd` command.

The `--is-enabled` option, if set to `true`, indicates that the new data domain is enabled. If this option is not specified, the data domain is enabled. If you want to clone but not start the data domain, specify `--is-enabled false`.

This example:

```
endeca-cmd clone-dd MyDD3 --source-name MyDDMaster --is-enabled false
```

creates a **MyDD3** data domain by cloning the **MyDDMaster** data domain. The data domain is created and registered, but not started.

delete-dd

The `delete-dd` command deletes a data domain with the specified name. This de-registers the data domain from the Endeca Server cluster, shuts down the Dgraph nodes serving this data domain, and deletes the index files for this data domain.

The syntax for this command is:

```
endeca-cmd delete-dd <data-domain> [global-options]
```

disable-dd

The `disable-dd` command stops the specified enabled data domain but continues to allocate resources to the Dgraph nodes.

The syntax for this command is:

```
endeca-cmd disable-dd <data-domain> [global-options]
```

where *data-domain* is the name of the disabled data domain that will be stopped.

The data domain must be enabled before it can answer queries.

enable-dd

The `enable-dd` command starts the specified disabled data domain. The syntax for this command is:

```
endeca-cmd enable-dd <data-domain> [global-options]
```

where *data-domain* is the name of the disabled data domain that will be started.

export-dd

The `export-dd` command exports the index of the specified data domain by taking a snapshot of the index files and copying them into the offline directory under another name. A snapshot represents a copy of the index files only, and does not capture any other characteristics of the data domain.

The running state of the data domain to be exported depends on your operating system:

- Linux: Both disabled data domains and enabled data domains can be exported.
- Windows: Disabled data domains can be exported, but enabled data domains cannot be exported.

The syntax for this command is:

```
endeca-cmd export-dd <data-domain> [--offline-name <exported-domain>] [global-options]
```

where *data-domain* is the name of the existing Endeca data domain to be exported.

The `--offline-name` option specifies the name to give to the exported data domain. This name must be unique. If this option is not specified, the exported name is assigned automatically, by appending the date to the original data domain name, using this format:

```
name_MMMMM-dd-yyyy-hh-mm
```

The command returns the resulting name used for the exported index.



Important: Keep track of the value of `--export-name`, because you will need it later for importing this index.

The location of the offline directory is specified by the `endeca-offline-dir` parameter in the Endeca Server configuration file. By default, this is the `$DOMAIN_HOME/EndecaServer/offline` directory.

This example:

```
endeca-cmd export-dd MyDD --offline-name MyDD_offline
```

exports the **MyDD** data domain to the offline directory under the name **MyDD_offline**.

import-dd

The `import-dd` command creates a new data domain with the specified *name* using previously exported index files. The syntax for this command is:

```
endeca-cmd import-dd <new-data-domain> --offline-name <exported-domain> [global-options]
[import-options]
```

where *new-data-domain* is the name of the new data domain that will be created from the exported data domain specified by the `--offline-name` flag.

Using this operation assumes that you have previously created a data domain whose index is currently exported. It also assumes that you kept track of the `--export-name` you used in the `export-dd` command, or know the name that was automatically assigned.

The command has these options:

Import Option	Meaning
<code>--dd-profile-name</code>	The name of the data domain profile to use (if not specified, defaults to the default data domain profile).
<code>--is-enabled</code>	<p>If set to <code>true</code>, indicates that the new data domain is enabled after it is imported.</p> <p>If this option is not specified, the imported data domain is enabled (this is the default).</p> <p>If you want to import but not start the data domain, specify <code>--is-enabled false</code>.</p>

This example:

```
endeca-cmd import-dd MyDD --offline-name MyDD_offline --dd-profile-name MyDDProfile --is-enabled
false
```

imports the index **MyDD_offline** from the offline directory into the new data domain **MyDD_offline**, which is created with the **MyDDProfile** data domain profile and is not enabled.

get-dd

The `get-dd` command returns the characteristics of the specified data domain.

The syntax for this command is:

```
endeca-cmd get-dd <data-domain> [global-options]
```

where *data-domain* is the name of the data domain for which to return information.

The returned information includes whether the data domain is enabled, the number of follower nodes, the number of query processing threads, and the list of arguments sent to the Dgraph processes for this data domain.

get-dd-health

The `get-dd-health` command returns information about a data domain's health in the Endeca Server cluster, with the specified name.

The syntax for this command is:

```
endeca-cmd get-dd-health <data-domain> [global-options]
```

where *data-domain* is the name of the data domain for which to return health information. The data domain can be enabled or disabled.

The returned information lists the status of the Dgraph nodes, including the leader node and follower nodes.

get-dd-status

The `get-dd-status` command returns runtime statistics about the specified data domain.

The syntax for this command is:

```
endeca-cmd get-dd-status <data-domain> [global-options]
```

where *data-domain* is the name of the data domain for which to return statistics. The data domain must be enabled for statistics to be returned.

The response includes the following information:

- The size of the index records (in MB)
- The number of source records in the data domain (this number excludes non-data, or system records).
- The Dgraph node statistics for each running Dgraph node in the data domain, including the Dgraph startup time, last index creation time, and path.

rescale-dd

The `rescale-dd` command adds a specified number of follower Dgraph nodes to the data domain and starts these nodes.

The syntax for this command is:

```
endeca-cmd rescale-dd <data-domain> [--num-followers <integer>] [global-options]
```

where *data-domain* is the name of the existing Endeca data domain to export.

The `--num-followers` option specifies the number of nodes to add. This name must be unique. If this option is not specified, the number defaults to 1. Note that the specified number cannot exceed the configured maximum allowable number of over-subscribed nodes.

This example:

```
endeca-cmd rescale-dd MyDD --num-followers 4
```

adds four follower Dgraph notes to the **MyDD** data domain and starts them.

allocate-bulk-load-port

The `allocate-bulk-load-port` command returns a host name for the leader node and the port used for Bulk Load Interface, for a specified data domain. The syntax for this command is:

```
endeca-cmd allocate-bulk-load-port [global-options]
```

This is a read-write operation — if the current leader node is available, it verifies the current Dgraph leader node and reports it along with the port used for Bulk Load. If the current leader node is not available, it appoints a new leader node and a new bulk load port and reports them.

Note that when you create a data domain profile, you can specify a port for Bulk Load, using one of the following two options. Use the `putDataDomainProfile` operation of the Cluster Web Service, which lets you specify `args`, one of which is the port for Bulk Load. Alternatively, use `endeca-cmd put-dd-profile -args --bulk_load_port`. Because this port must be unique on this machine, it is not guaranteed that the Endeca Server cluster will actually use the port you assign — it may assign its own port. You can always use `endeca-cmd allocate-bulk-load-port name` to find out which port is used for Bulk Load.

list-dd

The `list-dd` command lists all existing data domains and shows their status (enabled or disabled).

The syntax for this command is:

```
endeca-cmd list-dd [--verbose] [global-options]
```

The `--verbose` option displays additional status information for each data domain: the name, description, the number of nodes, the number of query processing threads, and the list of arguments sent to the Dgraph processes for this data domain.

update-spelling-dictionaries

The `update-spelling-dictionaries` command updates the spelling dictionaries in the index for the specified data domain.

The syntax for this command is:

```
endeca-cmd update-spelling-dictionaries <data-domain> [global-options]
```

This operation enables spelling correction and allows you to rebuild the dictionaries for spelling correction from the data corpus while continuing to issue queries and updates to the data domain and without stopping and restarting its Dgraph nodes.

version

The `version` command lists the version of the Oracle Endeca Server and the version of the Dgraph process powering the data domains (if the Dgraph processes are currently running).

The syntax for this command is:

```
endeca-cmd version [global-options]
```

Endeca Server node operations

These commands operate on Endeca Server nodes, by allowing you to specify the node profile that will be used by each Endeca Server node in the cluster, obtain the node profile that is currently configured, delete an Endeca Server node, list nodes, and obtain the status of the cluster and its nodes.

The commands for Endeca Server nodes are:

- [set-node-profile on page 30](#)
- [get-node-profile on page 30](#)

- [get-node-health on page 30](#)
- [get-cluster-health on page 31](#)
- [delete-node on page 31](#)
- [list-nodes on page 31](#)

set-node-profile

The `set-node-profile` command sets the hardware profile for the Endeca Server nodes.

The syntax for this command is:

```
endeca-cmd set-node-profile [global-options] [set-options]
```

The command has these options:

Set Option	Meaning
<code>--num-cpu-cores <int></code>	The number of CPU cores on each Endeca Server machine. If not specified, defaults to the number of cores allocated to this Endeca Server when it is deployed in WebLogic Server.
<code>--ram-size-mb <int></code>	The size of the main cache on the Endeca Server node. If not specified, defaults to 1024 MB.

This example:

```
endeca-cmd set-node-profile --num-cpu-cores 8 --ram-size-mb 2048
```

sets the number of CPU cores to 8 and the size of the main cache to 2048.

get-node-profile

The `get-node-profile` command lists the hardware profile used for all of the Endeca Server nodes that are currently running. The profile information is the number of CPU cores and the size of RAM (MB).

The syntax for this command is:

```
endeca-cmd get-node-profile [global-options]
```

The command has no options.

get-node-health

The `get-node-health` command lists the health information for the specified Endeca Server node. The command returns the Endeca Server node's host name and port, the protocol, and the status of the node. Additionally, it lists the status of the Dgraph nodes hosted on this Endeca Server node (these Dgraph nodes belong to different data domain clusters).

The syntax for this command is:

```
endeca-cmd get-node-health <node-hostname> [global-options]
```

where *node-hostname* is the name of an Endeca Server node (it should be one of the names returned by the `list-nodes` command).

This command has no options.

get-cluster-health

The `get-cluster-health` command reports an overall status of the Endeca Server cluster. The command lists the host names and ports of the nodes running the Cluster Coordinator and their status, the Endeca Server nodes and their status, and the data domains, indicating whether they are enabled.

The syntax for this command is:

```
endeca-cmd get-cluster-health [global-options]
```

The command has no options.

delete-node

The `delete-node` command deletes an Endeca Server node after checking that no data domains are allocated on this node. If the Endeca Server node that you would like to delete is hosting any Dgraph nodes serving any data domains, the Endeca Server attempts to migrate them to other Endeca Server cluster nodes, before removing this node. If the Endeca Server cluster has no sufficient resources for moving data domains to other nodes, it prevents you from deleting this node.

If you use this command to delete a node and the node is deleted successfully, make sure you also shut down the WebLogic Server hosting this Endeca Server instance.

The syntax for this command is:

```
endeca-cmd delete-node <node-hostname> [global-options]
```

where *node-hostname* is the name of an Endeca Server node (it should be one of the names returned by the `list-nodes` command).

This command has no options.

list-nodes

The `list-nodes` command lists the host names of the active Endeca Server nodes, without providing any additional information.

The syntax for this command is:

```
endeca-cmd list-nodes [global-options]
```

The command has no options.

Data domain profile operations

These commands operate on Endeca data domain profiles.

A data domain profile is a named template that provides configuration settings to be used for the creation of Endeca data domains. The configuration settings for new data domains include:

- The number of follower nodes required
- Whether the leader is dedicated to updating requests or is sharing a regular query load
- Whether the Endeca Server can oversubscribe its nodes while sharing them with other data domains

- Whether the Dgraph processes should be read-only
- Session affinity configuration
- The hardware characteristics of Dgraph processes (the number of threads and the cache size)
- The Dgraph flags to use to start up the Dgraph process

The commands described in this topic create and delete data domain profiles, as well as return information about them.



Important: Once you create a data domain profile, you cannot change its configuration.

The commands for managing data domain profiles are:

- [put-dd-profile on page 32](#)
- [get-dd-profile on page 34](#)
- [list-dd-profiles on page 34](#)
- [delete-dd-profile on page 35](#)

put-dd-profile

The `put-dd-profile` command creates a new data domain profile with the specified name. Note that a default data domain profile (named `default`) always exists in the Endeca Server and does not need to be created with this command.

The syntax for this command is:

```
endeca-cmd put-dd-profile <new-profile-name> [global-options] [create-options]
```

where *new-profile-name* is the name of the new data domain.

The following additional command options can be used to change the created data domain profile's configuration (defaults are used otherwise):

Create Option	Description
<code>--description <text></code>	A full description of the data domain profile. If the description contains spaces, it must be enclosed within double quotes. If this option is not specified, the description defaults to an empty string.
<code>--query-leader <bool></code>	Whether the leader node can accept read-only queries. If not specified, defaults to <code>true</code> .
<code>--num-followers <int></code>	How many follower nodes (Dgraph processes) should be configured in the data domain. If not specified, defaults to 0.

Create Option	Description
<code>--oversubscribe <bool></code>	<p>Whether a data domain (or any of its nodes) can be placed on an oversubscribed Endeca Server instance.</p> <p>If not specified, defaults to <code>true</code>.</p>
<code>--num-compute-threads <int></code>	<p>Specifies the number of computational threads in the Dgraph node's threading pool.</p> <p>The value must be a positive integer that is equal or greater than 2.</p> <p>The recommended number of computational threads for the Dgraph process is typically equal to the number of CPU cores on the host Endeca Server node.</p> <p>If no value is specified or 0 is specified, this defaults to the larger of 2 or the number of available processors.</p>
<code>--compute-cache-size <int></code>	<p>Specifies an absolute value in MB for the Dgraph process cache, for each Dgraph node in the data domain.</p> <p>If no value is specified or 0 is specified, this defaults to the Dgraph default cache size computed as 10% of the amount of RAM available on the Endeca Server node hosting the Dgraph node.</p>
<code>--startup-timeout <int></code>	<p>Specifies the maximum length of time (in seconds) that is allowed for the data domain's Dgraph process to start up.</p> <p>Default is 600 seconds.</p>
<code>--shutdown-timeout <int></code>	<p>Specifies the maximum length of time (in seconds) that is allowed for the data domain's Dgraph process to shut down.</p> <p>Default is 30 seconds.</p>
<code>--read-only <bool></code>	<p>Whether a data domain should be created as read-only and not accept updating requests, including data loading requests.</p> <p>If not specified, defaults to <code>false</code>.</p>
<code>--session-id-type <type></code>	<p>The method to use for establishing session affinity.</p> <p>The available options are: <code>header</code> (for HTTP headers), <code>parameter</code> (for URL parameters), or <code>cookie</code>.</p> <p>The default method is <code>header</code>.</p>
<code>--session-id-key <name></code>	<p>The name of the key to use for maintaining affinity.</p> <p>The default name is <code>X-Endeca-Session-ID</code>.</p>

Create Option	Description
<code>--args <dgraph-flags></code>	Specifies a list of the Dgraph flags that will be used for the data domain's Dgraph process. The <code>--args</code> flag must be the last flag on the command line, as all of its arguments are passed on to the Dgraph process.
<code>--args --usage</code>	Provides a list of the available Dgraph process flags.

This example:

```
endeca-cmd put-dd-profile MyProfile --description "group profile" --oversubscribe false --args
--net-timeout 60
```

creates the data domain profile named **MyProfile**, which cannot be placed on an oversubscribed Endeca Server instance, and starts the Dgraph with a network timeout value of 60 seconds. **MyProfile** uses the default values for the other configuration settings.

get-dd-profile

The `get-dd-profile` command lists the characteristics of the data domain profile with the specified name.

The syntax for this command is:

```
endeca-cmd get-dd-profile <profile-name> [global-options]
```

where *profile-name* is the name of an existing data domain profile. This command has no options.

This example:

```
endeca-cmd get-dd-profile MyProfile
```

returns the following details for the data domain profile named **MyProfile**:

```
MyProfile
Description: group profile
AllowQueriesOnLeader: true
AllowOverSubscribe: false
NumFollowers: 0
ReadOnly: false
NumComputeThreads: 2
ComputeCacheSizeMB: 0
StartupTimeoutSeconds: 600
ShutdownTimeoutSeconds: 30
SessionIdType: HEADER
SessionIdKey: X-Endeca-Session-ID
Args: [--net-timeout, 60]
```

list-dd-profiles

The `list-dd-profiles` command returns a list of data domain profiles configured in the Endeca Server cluster. For each data domain profile, this command returns its name, description, and other characteristics, such as the number of nodes, the number of query processing threads, and the list of arguments (if any) that are sent to the Dgraph processes for this profile.

The syntax for this command is:

```
endeca-cmd list-dd-profiles [--verbose] [global-options]
```

The `--verbose` option includes additional status information for each data domain.

delete-dd-profile

The `delete-dd-profile` command deletes the data domain profile with the specified name. This command does not affect any data domains that may be using this profile.

The syntax for this command is:

```
endeca-cmd delete-dd-profile <profile-name> [global-options]
```

where *profile-name* is the name of an existing data domain profile. This command has no options.

This example:

```
endeca-cmd delete-dd-profile MyProfile
```

deletes the data domain profile named **MyProfile**.

Note that you cannot delete the default data domain profile (named `default`). If you attempt to do so, the command fails with this error:

```
OES-000107: Cannot delete the default data domain profile.
```

You can delete a data domain profile that was used to create an existing data domain — when the data domain is created, the configuration information from the profile is copied into the data domain's state. Therefore, the data domain does not use the profile after creation.

Command global options

The Endeca Server command interface has several global options that allow you to specify the host, port, and context root of the Oracle Endeca Server, and enable the interface for SSL.

The global options are:

- `--host`
- `--port`
- `--root`
- `--help`
- `--keystore`
- `--truststore`
- `--ssl`
- `--password`
- `--keystore-password`
- `--truststore-password`
- `--key-password`

Getting online help

The `--help` option provides usage help for the Endeca Server commands.

The syntax for obtaining general help is:

```
endeca-cmd --help
```

The syntax for obtaining help on a specific command is:

```
endeca-cmd <command> --help
```

This example displays usage help for the `create-dd` command:

```
endeca-cmd create-dd --help
```

[Global options for host, port, and context root](#)

[Global options for SSL](#)

Global options for host, port, and context root

Three global options allow you to specify the host name, port number, and context root information for the Endeca Server.

--host option

You use the `--host` option when you want to run a command on an Endeca Server that is running on a remote machine. The `--host` argument can be either the full name of the remote machine or its IP address.

The following example illustrates the `--host` global option:

```
endeca-cmd create-dd bikes --host web7.example.com
```

The command tells the Endeca Server running on the **web7.example.com** remote machine (and listening on its default port) to create an Endeca data domain named **bikes** in the default location on that remote machine.

--port option

The `--port` option is used whenever the Endeca Server is not running on its default port, regardless of whether the Endeca Server is running locally or on a remote machine. If you do not specify `--port`, the default port is used for the command.

The default ports for the Endeca Server are:

- 7001 is the default HTTP port in the WebLogic Server on which the Endeca Server application is listening. This port is used if the Endeca Server is running in a non-SSL mode.
- 7002 is the default HTTPS secure port in the WebLogic Server on which the Endeca Server application is listening, if the Endeca Server is configured over SSL.

The following example illustrates both the host and port global options:

```
endeca-cmd get-dd bikes --host web7.example.com --port 9090
```

The command tells the Endeca Server running on the **web7.example.com** remote machine (and listening on a non-default port 9090) to return the status of the Endeca data domain named **bikes**.

--root option

The Endeca Server application uses **/endeca-server** as the default name of its context root when running in WebLogic Server. The `--root` option is used to specify this context-root name. If you do not specify `--port`, the default **/endeca-server** context root is used for the command.

Global options for SSL

These global options are used to support SSL-enabled communications with an Oracle Endeca Server running over SSL.

You are required to use these options if you have enabled the Oracle Endeca Server to run only over SSL. Before using these options, you must run the `generate_ssl_keys` utility to generate the required keystore and truststore certificates:



- `endecaServerClientCert.ks` keystore
- `endecaServerTrustStore.ks` truststore

These files are placed in the `$DOMAIN_HOME/config/ssl` directory. For details on this utility, see the *Oracle Endeca Server Installation Guide*.



Note: This topic assumes that you are using the non-SSL version of the `endeca-cmd` utility. The SSL version references a configuration file that provides the certificate locations, so you do not have to use these options. For details on the SSL version of the `endeca-cmd` utility, see [SSL version of endeca-cmd on page 23](#).

The following global options in `endeca-cmd` are provided to enable SSL support:

Option	Description
<code>--keystore</code>	<p>Specifies the location of a keystore file needed for authentication to the Oracle Endeca Server. For example, the <code>endecaServerClientCert.ks</code> keystore can be used.</p> <p>If you use this option, it implies that SSL should be used for communication between <code>endeca-cmd</code> and the Oracle Endeca Server. This means that you don't need to use the <code>--ssl</code> option.</p> <p> Note: If you specify a keystore, this causes <code>endeca-cmd</code> to prompt for a password. Therefore, if you use the <code>--keystore</code> option, you cannot run <code>endeca-cmd</code> as part of a script.</p>
<code>--truststore</code>	<p>Specifies the location of a truststore file needed for verifying the authenticated connection to the Oracle Endeca Server. For example, the <code>endecaServerTrustStore.ks</code> truststore can be used.</p> <p>If you use this option, it implies that SSL should be used for communication between <code>endeca-cmd</code> and the Oracle Endeca Server. This means that you don't need to use the <code>--ssl</code> option.</p> <p> Note: If you specify a truststore, this causes <code>endeca-cmd</code> to prompt for a password. Therefore, if you use the <code>--truststore</code> option, you cannot run <code>endeca-cmd</code> as part of a script.</p>

Option	Description
--ssl	<p>Specifies whether to use an authenticated SSL connection to the Oracle Endeca Server.</p> <p>If you use either the --keystore or --truststore option, then you don't need to use the --ssl option, as the authenticated SSL connection is implied by specifying the keystore or the truststore file.</p> <p>If you use --ssl without either a --keystore or --truststore option, you can use an empty keystore and the cacerts file as your truststore. Alternatively, you can use the standard approaches for creating keystore and truststore files in the WebLogic Server.</p>
--password	<p>Specifies a password for accessing the keystore and its key, and the truststore. The keystore and truststore are both assumed to use this password.</p> <p>The password used for the key in the keystore will be this password with "clientkey" appended (and in the case of failure, this password will also be tried for the key).</p> <p>This option is intended for use in scripts.</p>
--keystore-password	<p>Specifies a password for accessing the keystore (overrides the --password option).</p> <p>This option is intended for use in scripts.</p>
--truststore-password	<p>Specifies a password for accessing the truststore (overrides the --password option).</p> <p>This option is intended for use in scripts.</p>
--key-password	<p>Specifies a password for accessing the key within the keystore (overrides the --password option)</p> <p>This option is intended for use in scripts.</p>



Chapter 4

Dgraph Administrative Tasks

This section describes some basic administrative tasks for the Dgraph, contains Dgraph troubleshooting tips, and describes the Dgraph logs.

[Updating spelling dictionaries](#)

[Checking the status of the Dgraph nodes](#)

[Tuning the Dgraph process cache](#)

[About connecting Web browsers to the data domain](#)

[Managing Dgraph core dump files](#)

[Collecting debugging information](#)

[Running multiple Dgraphs on the same Windows machine](#)

[Identifying connection errors](#)

Updating spelling dictionaries

Use the `endeca-cmd update-spelling-dictionaries data_domain` command to update the spelling dictionaries in the data domain. Run this operation after you have added data records to the data domain, to enable spelling correction in the Dgraph.

During the data ingest process, you can run this operation periodically to update the spelling dictionary used by the Dgraph for Automatic Spelling Correction and Did You Mean (DYM).

In a cluster of Dgraph nodes, this command is routed to the leader node.

The `update-spelling-dictionaries` option of the `endeca-cmd` command performs the following actions:

- Crawls the text search portion of the index for the data domain for all terms that meet the constraint settings.

The constraint settings include minimum word occurrences and maximum and minimum number of characters, for records and attribute values. The Dgraph nodes in the data domain use these constraints to update the spelling dictionaries. You can change them in the Global Configuration Record.

- Updates the spelling dictionaries stored in the Dgraph for processing of all queries arriving after this update to the index files. The Dgraph uses these updated dictionaries when processing all future queries.



Note: Because of the nature of continuous query, once the Dgraph processes this request, it will start using the updated spelling dictionary after a certain point in its processing, and all newly incoming queries will be answered against the updated spelling dictionary.

The Dgraph applies the updated settings while continuing to run queries and without needing to restart.

Checking the status of the Dgraph nodes

A quick way of checking the health of a Dgraph node is to get its status from the Endeca Server.

To check the aliveness of a Dgraph:

1. From a command prompt, navigate to the `endeca-cmd` directory in the Endeca Server root installation.
2. Issue the `get-dd-status` command, specifying a particular Endeca data domain, as in this example:

```
endeca-cmd get-dd-status books
```

Alternatively, you can also use the `get-dd-health` command, to obtain information about the specific data domain's health, including the status of its Dgraph nodes.

Tuning the Dgraph process cache

In the Endeca Server, caching of results for the Dgraph process is used extensively, for many types of queries. You can use the following tips for tuning it.

One of the patterns of caching utilized by the Endeca Server Dgraph process involves caching of intermediate results during parallel query processing. Such a caching strategy allows the computing resources to be utilized better, and therefore, improves throughput of the Endeca Server.

Use the following recommendations when tuning performance of your applications running on top of the Endeca Server:

- Check the current configuration setting for the Dgraph cache by using this command: `endeca-cmd get-dd-profile <data_domain_profile_name>`, where `<data_domain_profile_name>` is the name of the profile used by your data domain.

This command returns the setting for the `--compute-cache-size`, for all Dgraph processes serving your data domain. This is the amount of RAM, in MB, allocated to the result cache on each Dgraph node in the data domain.

The default is 0, which is interpreted as follows: When an absolute value is set to 0, the default Dgraph cache size is computed as 10% of the amount of RAM available on the Endeca Server node hosting the Dgraph node.

- Obtain the Dgraph stats page, for each Dgraph process running for your data domain, by using the following URL: `http://<host>:<port>/endeca-server/admin/<data_domain>?op=stats`, where `host` and `port` refer to the host name and port of the Endeca Server, and `data domain` is the name of your data domain.

Note that if your data domain runs on more than one Endeca Server instance, you should issue this command on each machine hosting the Endeca Server for this data domain.

- On the Dgraph stats page, examine the Dgraph cache, in particular the number of evictions. If this number is high, it may mean that the default value is insufficient for your data domain.

Experiment with gradually increasing the Dgraph cache size. For example, you may start by changing the cache size to 20% of memory instead of leaving it at the default value.

Note that to specify the new value, you would have to calculate what 20% of memory means in MB, based on the size of RAM on the machine hosting the Endeca Server.

- Be careful with increasing the cache size too significantly (for example, to 50% of memory), because the downside of having a large cache size is that memory used by the cache is not available for query evaluation (and is not released back from the cache).

To specify a new Dgraph cache size for the data domain, use this command:

```
endeca-cmd put-dd-profile
<data_domain_profile_name> --compute-cache-size <value>
```

It creates a new data domain profile with the specified setting for the cache size. (In order to have a usable data domain profile, specify other options for the new data domain profile, to mimic an existing data domain profile. These options are not included in this example.) In the new data domain profile, all Dgraph processes will use the specified Dgraph cache size. The existing data domain would need to be recreated utilizing this profile.

Alternatively, instead of using `endeca-cmd`, you can set the cache size using the `putDataDomainProfile` operation of the Cluster Web Service, with a value for `computeCacheSizeMB`, as in this abbreviated example:

```
<ns1:putDataDomainProfile xmlns:ns1="http://www.endeca.com/endeca-server/cluster/1/0">
  <ns1:dataDomainProfile>
    ...
    <ns2:computeCacheSizeMB xmlns:ns2="http://www.endeca.com/endeca-server/types/1/0">1<
/ns2:computeCacheSizeMB>
  </ns1:dataDomainProfile>
</ns1:putDataDomainProfile>
```

For additional information on `endeca-cmd`, see this guide. For full information on managing data domains, see the *Oracle Endeca Server Cluster Guide*.

About connecting Web browsers to the data domain

For security reasons, you should never allow user Web browsers to connect directly to the machine hosting the Endeca Server and the Endeca data domains.

Browsers started by non-administrators should always connect to your application through an application server.

IPv4 and IPv6 address support

The Oracle Endeca Server and the Endeca Dgraph process support both IPv4 (Internet Protocol Version 4) and IPv6 (Internet Protocol Version 6) addressing schemes for connections. This IPv4 and IPv6 addressing support is configured automatically in the Oracle Endeca Server and the Dgraph, so there is no need for the administrator to do any explicit addressing configuration.

Managing Dgraph core dump files

In the rare case of a Dgraph crash, the Dgraph writes its core dump files on disk.

When the Dgraph runs on a very large data set, the size of its index files stored in-memory may exceed the size of the physical RAM. If such a Dgraph process fails, it may need to write out potentially very large core dump files on disk. The core files are written to the Endeca Server's `logs` directory.

To troubleshoot the Dgraph, it is often useful to preserve the entire set of core files written out as a result of such failures. When there is not enough disk space, only a portion of the files is written to disk until this

process stops. Since the most valuable troubleshooting information is contained in the last portion of core files, to make these files meaningful for troubleshooting purposes, it is important to provision enough disk space to capture the files in their entirety.

Two situations are possible, depending on your goal:

- To troubleshoot a Dgraph crash, provision enough disk space to capture the entire set of core files. In this case, the files are saved at the expense of potentially filling up the disk.
- To prevent filling up the disk, you can limit the size of these files on the operating system level. In this case, with large Dgraph applications, only a portion of core files is saved on disk. This may limit their usefulness for debugging purposes.

Keep in mind that the Endeca Server attempts to restart the Dgraph node when it crashes. If the start-up retry fails, the Endeca Server will retry the start-up only one more time.

[Managing Dgraph crash dump files on Windows](#)

[Managing Dgraph core dump files on Linux](#)

Managing Dgraph crash dump files on Windows

On Windows, all Dgraph crash dump files are saved on disk by default.

The Dgraph uses the `MiniDump` function from the Microsoft `DbgHelp` library.

Provision enough disk space to accommodate core files based on this estimate:

- The projected upper limit for the size of these files is equal, at a maximum, to the size of the physical memory used by the data files for all data domains hosted on the Oracle Endeca Server. Often the files take up less space than that.

Managing Dgraph core dump files on Linux

It is recommended to use the `ulimit -c unlimited` setting for the Dgraph process core dump files. Non-limited core files contain all Dgraph data that is resident in memory (RSS of the Dgraph process).

Because large applications powered by the Oracle Endeca Server may take up the entire amount of available RAM, the core dump files can also grow large and take up the space equal to the size of the physical RAM on disk plus the size of the server data files in memory.

Provision enough disk space to accommodate core files based on this estimate:

- The projected upper limit for the size of these files should be equal, at a maximum, to the size of the physical RAM. Often the files take up less space than that.



Note: If you are not setting `ulimit -c unlimited`, you could be seeing the Dgraph process crashes that do not write any core files to disk, since on some Linux installations the default for `ulimit -c` is set to 0.

Alternatively, it is possible to limit the size of core files with the `ulimit -c <size>` command, although this is not recommended. If you set the limit size in this way, the core files cannot be used for debugging, although their presence will confirm that the Dgraph had crashed.

To be able to troubleshoot the crash, change this setting to `ulimit -c unlimited`, and reproduce the crash while capturing the entire core file. Similarly, to enable support to troubleshoot the crash, you will need to reproduce the crash while capturing the full core file.

Collecting debugging information

Before attempting to debug an issue with the data domain, collect the following information.

- Hardware specifications and configuration.
- Description of the Oracle Endeca Server topology (number and names of servers, number of Dgraphs and their data domains).
- The data from the Dgraph Server Statistics page.
- Dgraph input.
- Description of which Dgraphs are affected.

You can use the `endeca-cmd` commands to gather this information. For example, you can list the data domains and obtain their status with `endeca-cmd list-dd --verbose`, and also use `endeca-cmd get-dd-health` and `endeca-cmd get-dd-status`.

Running multiple Dgraphs on the same Windows machine

If you have more than one Endeca data domain running on a single Windows machine, the Endeca Server assigns each Dgraph a port that does not conflict with other Dgraph ports on that machine. (These Dgraph processes serve different data domains.)

This prevents multiple Dgraphs running on a single machine from presenting inconsistent behavior.

Identifying connection errors

If the Dgraph standard out log contains `connection broken` messages, although it may look like the problem occurred with the Dgraph, the actual cause of the problem is usually a broken connection between the server that hosts the front-end application and the server that hosts the Dgraph.

In the case of connection errors, various parts of the implementation issue the following error and warning messages:

- The Dgraph standard out log contains warnings similar to the following:

```
WARN [DATE TIME] UTC (1239830549803)
DGRAPH {dgraph}: Aborting request: connection broken: client 10.10.21.21
```

- The Dgraph request log contains an abnormal `status 0` message similar to the following:

```
1239830549803 10.6.35.35 - 349 0 19.35 0.00 0 - 0 0 - -
```

Typically, the `connection broken` message means that the Dgraph encountered an unexpected failure in the connection between the client and the Dgraph. This type of error may occur outside the Dgraph, such as in the network, or be caused by the timeout of the client application session.

Investigate the connection between the client and the Dgraph. For example, to prevent timeouts of the client application sessions, you may decide to implement front-end application retries.



Chapter 5

Administrative Operations

The Oracle Endeca Server supports many Dgraph administrative and configuration operations that you can access through simple URLs. You can use these operations to control the behavior of the Dgraph processes that are running under the Oracle Endeca Server.

[About administrative and configuration operations](#)

[About logging variables for the Dgraph](#)

About administrative and configuration operations

Administrative and configuration operations make it possible to check Dgraph statistics, and enable or disable diagnostic flags, without having to stop a running Dgraph. This section lists the supported administrative and configuration URLs, describes the functions of each URL, and defines the syntax of those URLs.

Syntax of administrative and configuration operations

In the following listings:

- `<host>` refers to the hostname or IP address of the Oracle Endeca Server.
- `<port>` refers to the port on which the Oracle Endeca Server is listening.
- `endeca-server` is the context root of the Oracle Endeca Server application.
- `<data_domain>` refers to the name of the Endeca data domain on which the command will operate.

Queries to these URLs are handled in the Dgraph's request queue like any other request—that is, they are handled on a first-come, first-served basis. They are also reported in the Dgraph request log like any other request.

For administrative operations, the syntax is:

```
http://<host>:<port>/endeca-server/admin/<data_domain>?op=<supported-operation>
```

For example:

```
http://localhost:7001/endeca-server/admin/books?op=merge
```

For configuration operations, the syntax is:

```
http://<host>:<port>/endeca-server/config/<data_domain>?op=<supported-operation>
```

For example:

```
http://localhost:7001/endeca-server/config/books?op=log-status
```



Note: If you are using HTTPS mode, use `https` in the URL.

[List of administrative operations](#)

List of administrative operations

Administrative (or admin) operations listed in this topic allow you to control the behavior of the Dgraph processes in a data domain.

The Dgraph recognizes the following admin operations:

Admin operation	Description
<code>/admin/datadomain?op=help</code>	Returns the usage page for all of the admin operations.
<code>/admin/datadomain?op=flush</code>	Specifies when the Dgraph should flush its dynamic cache.
<code>/admin/datadomain?op=logroll</code>	Forces a query log roll.
<code>/admin/datadomain?op=merge</code>	Merges update generations in the index and sets the system's merge policy.
<code>/admin/datadomain?op=ping</code>	Checks the aliveness of a Dgraph and returns a lightweight message.
<code>/admin/datadomain?op=stats</code>	<p>Returns the Dgraph Statistics page, for the specific Dgraph on which this command runs.</p> <p>In a data domain cluster, if you issue this command, it will run on any Dgraph process that it locates. Therefore, In an Endeca Server cluster hosting one or more data domain clusters, to obtain the Dgraph node statistics for the nodes in the data domain, run <code>endeca-cmd get-dd-stats</code>, which returns the Dgraph statistics for all Dgraph nodes in the data domain.</p>
<code>/admin/datadomain?op=statsreset</code>	Resets the Dgraph Statistics page for one of the Dgraphs in the data domain.

[help](#)[flush](#)[logroll](#)[merge](#)[ping](#)[stats](#)[statsreset](#)

help

`/admin/datadomain?op=help` returns the usage page for all of the administrative operations.

flush

`/admin/datadomain?op=flush` flushes the Dgraph cache.

The `flush` operation clears all entries from the Dgraph cache. It returns the following message:

```
flushing cache...
```

If you are debugging query problems, you can approximate cold-start or post-update performance by clearing the Dgraph cache prior to running a request.

logroll

`/admin/datadomain?op=logroll` forces a query log roll.

The `logroll` command returns a message similar to the following:

```
rolling log... Successfully rolled log file.
```

merge

`/admin/datadomain?op=merge` forces a merge, and (optionally) changes the merge policy of a running Dgraph. In a data domain cluster of Dgraph nodes, this command is routed to the leader node of the data domain cluster where it is processed.

[Managing the Merge Policy](#)

ping

`/admin/datadomain?op=ping` checks the aliveness of a Dgraph and returns a lightweight message.

You can view the Dgraph Statistics page to check whether the Dgraph is running and accepting queries, but that comes with some overhead. A quicker way to check the aliveness of a Dgraph is by running the `ping` command.

Because ping requests are given the highest priority and are processed synchronously (as they are received), a ping response time is independent of the number of outstanding requests in the Dgraph.

The `ping` command returns a lightweight page that lists the Dgraph process and the current date and time, such as the following:

```
dgraph Web07:5556 responding at Wed Mar 27 15:35:27 2013
```

You can use this operation to monitor the health or heartbeat of the Dgraph, and as a health check for load balancers.

stats

`/admin/datadomain?op=stats` returns the Dgraph Statistics page.

The Dgraph Statistics page provides a detailed breakdown of what the Dgraph process is doing, and is a useful source of information about your Endeca implementation's configuration and performance. It provides information such as startup time, last data files creation time, and path. This lets you focus your tuning and load-balancing efforts. By examining this page, you can see where the Dgraph is spending its time.



Important: In an Endeca Server cluster hosting more than one data domain cluster, to obtain the Dgraph statistics for the nodes in the particular data domain, run the Endeca Server command `get-dd-status` on the host and port for the Endeca Server cluster. This command is automatically routed to one of the Dgraph nodes for the requested data domain, and returns the Dgraph statistics for all Dgraph nodes in this data domain.

statsreset

`/admin/datadomain?op=statsreset` resets the Dgraph Statistics page.

The `statsreset` operation returns the following message:

```
resetting server stats...
```

About logging variables for the Dgraph

You can use logging variables with config operations. This lets you obtain detailed information about Dgraph processing, to help diagnose unexpected application behavior or performance problems, without stopping and restarting the Dgraph or requiring a configuration update.

Although you can also specify general verbose logging at the Dgraph command line with the `-v` flag, it requires a Dgraph restart to take effect.

[Logging variable operation syntax](#)

[List of configuration operations](#)

[List of supported logging variables](#)

Logging variable operation syntax

Dgraph logging variables are toggled using the `/config/ddomain?op=log-enable&name=<variable-name>` and `/config/ddomain?op=log-disable&name=<variable-name>` operations.

You can include multiple logging variables in a single request. Unrecognized logging variables generate warnings.

For example, this operation:

```
/config/wine?op=log-enable&name=requestverbose
```

turns on verbose logging for queries, while this operation:

```
config/wine?op=log-enable&name=textsearchrelrankverbose&name=textsearchspellverbose
```

turns on verbose logging for both the text search relevance ranking and spelling features.

However, this operation:

```
config/wine?op=log-enable&name=allmylogs
```

returns an unsupported logging setting message.

In addition, the following operations are supported:

- `/config/ddomain?op=log-status` returns a list of all logging variables with their values (true or false).
- The special name `all` can be used with `/config/ddomain?op=log-enable` or `/config/?dstoreop=log-disable` to set all logging variables.

List of configuration operations

Configuration (or config) operations listed in this topic allow you to modify configuration and logging information for the Dgraph from within the system.

The Dgraph recognizes the following config operations:

Config operation	Description
<code>/config/ddomain?op=help</code>	Returns the usage page for all of the config operations.
<code>/config/ddomain?op=log-enable</code>	Enables verbose logging for one or more specified variables.
<code>/config/ddomain?op=log-disable</code>	Disables verbose logging for one or more specified variables.
<code>/config/ddomain?op=log-status</code>	Returns verbose logging status.

List of supported logging variables

The following table describes the supported logging variables that you can use with related config operations to toggle logging verbosity for specified features.

Logging variable names are not case sensitive.

Variable	Description
<code>verbose</code>	Enables verbose mode.
<code>requestverbose</code>	Prints information about each request to <code>stdout</code> .
<code>textsearchrelrankverbose</code>	Enables verbose information about relevance ranking during search query processing.
<code>textsearchspellverbose</code>	Enables verbose output for spelling correction features.
<code>dgraphperfverbose</code>	Enables verbose performance debugging messages during core Dgraph navigation computations.

Variable	Description
dgraphrefinementgroupverbose	Enables refinement verbose/debugging messages.

[log-enable](#)

[log-disable](#)

[log-status](#)

[help](#)

log-enable

The `log-enable` operation turns on verbose logging.

You can include multiple logging variables in a single request. Unrecognized logging variables generate warnings.

For example, this operation:

```
/config/wine?op=log-enable&name=requestverbose
```

turns on verbose logging for queries, while this operation:

```
config/wine?op=log-enable&name=textsearchrelrankverbose&name=textsearchspellverbose
```

turns on verbose logging for both the text search relevance ranking and spelling features.

However, this operation:

```
config/wine?op=log-enable&name=allmylogs
```

returns an "Unsupported logging setting" message.

log-disable

The `log-disable` operation turns off verbose logging.

`/config/dstore?op=log-disable` with no arguments returns the same output as `log-status`.

log-status

The `log-status` operation returns a list of all logging variables with their values (true or false).

For example, if you have enabled verbose logging on two of the features, you would see a message similar to the following:

```
Logging settings:

verbose - FALSE
requestverbose - TRUE
updateverbose - FALSE
recordfilterperfverbose - FALSE
textsearchrelrankverbose - TRUE
textsearchspellverbose - FALSE
dgraphperfverbose - FALSE
dgraphrefinementgroupverbose - FALSE
```

help

`/config/dstore?op=help` returns the usage page for all of the config operations.



This section describes how to set and manage a merge policy for each Endeca data domain.

[Using a merge policy for incremental updates](#)

[Types of merge policies](#)

[Setting or changing the merge policy](#)

[Changing the merge policy of a running data domain](#)

[Forcing a merge](#)

[Linux unlimit settings for merges](#)

Using a merge policy for incremental updates

A merge policy determines how frequently the Dgraph merges incremental update generations in its index files, for a specific data domain.

The data layer stores the index files of an Endeca data domain as a series of internal files with versions. As a result:

- Old versions can be accessed while new versions are created.
- Old versions are garbage-collected when no longer needed.

A version is persisted as a sequence of generation files. A new version appends a new generation file to the sequence. Query latency depends, in part, on the number and size of generation files used to store the index files.

Generation files are combined through a process called *merging*. Merging is a background task that does not affect the Endeca data domain processing, but may affect its performance. Because of this, you can set a *merge policy* that dictates the aggressiveness of the merges. In a clustered environment, a request to set the merge policy is routed to the leader Dgraph node.

Types of merge policies

You can set the merge policy to one of two settings: balanced or aggressive.

- **Balanced:** This policy strikes a balance between low latency and high throughput. This is the default policy of an Endeca data domain.
- **Aggressive:** This policy merges frequently and completely to keep query latency low at the expense of average throughput.

The balanced policy is recommended for the majority of applications. However, aggressive merging may help those applications that meet the following criteria:

- Query latency is the primary concern.
- A large fraction of the records (for example, 20%) are either modified or deleted by incremental updates before re-baselines.
- The time to perform an aggressive merge is less than the time between incremental updates.



Note: Under normal conditions, you do not need to change the default balanced policy.

Setting or changing the merge policy

The `mdex-config_MergePolicy` attribute in the Global Configuration Record (or GCR) sets the merge policy for the Endeca data domain.

You can set the merge policy by sending a request to the Configuration Web Service to change the settings for the GCR, utilizing any Web services tool, such as soapUI.

Alternatively, you can use the URL `/admin/dataDomainName?op=merge` command to change the merge policy of a running data domain, or to force a merge.

If you are running a cluster of nodes, this request is routed to the leader node and processed by it for the data domain's index.

Setting the merge policy with the Configuration Web Service

Setting the merge policy with the Configuration Web Service

You can use the Configuration Web Service to programmatically retrieve and set the merge policy in the GCR.

Retrieving the merge policy with the Configuration Web Service

You can retrieve the Global Configuration Record to see the current setting for the merge policy.

To retrieve the Global Configuration Record:

1. In a tool such as soapUI, access the Configuration Web Service on the Oracle Endeca Server for the data domain, as in this example for the "books" data domain:

```
http://web007:7001/endeca-server/ws/config/books?wsdl
```

2. Use the `getGlobalConfigRecord` operation to retrieve the Global Configuration Record via the Configuration Web Service, as in this example:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <config:configTransaction
      xmlns:config="http://www.endeca.com/MDEX/config/services/types/2/0"
      xmlns:mdex="http://www.endeca.com/MDEX/config/XQuery/2009/09">
      <config:getGlobalConfigRecord />
    </config:configTransaction>
  </soap:Body>
</soap:Envelope>
```

The results response from the Configuration Web Service should look like this example (the SOAP elements have been removed):

```
<config-service:results
  xmlns:config-service="http://www.endeca.com/MDEX/config/services/types/2/0">
  <mdex:globalConfigRecord xmlns:mdex="http://www.endeca.com/MDEX/config/XQuery/2009/09">
    <mdex:record xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      ...
      <mdex-config_Key type="mdex:string">global</mdex-config_Key>
      <mdex-config_MergePolicy type="mdex:string">balanced</mdex-config_MergePolicy>
      ...
    </mdex:record>
  </mdex:globalConfigRecord>
</config-service:results>
```

In this example, the merge policy is set to `balanced` for the Dgraph process of the Endeca data domain.

Setting the merge policy with the Configuration Web Service

You can programmatically set the merge policy for the data domain by updating the Global Configuration Record.

To set the merge policy in the Global Configuration Record:

1. In a tool such as SOAP UI, access the Configuration Web Service on the Oracle Endeca Server for the data domain, using this syntax:

```
http://host:port/endeca-server/ws/config/dataDomain?wsdl
```

host and *port* specify the host name and port on which the Endeca Server is running and *dataDomain* is the name of the Endeca data domain to be merged.

2. Use the `putGlobalConfigRecord` operation to set the value of the `mdex-config_MergePolicy` attribute in the Global Configuration Record, as in this example that changes the merge policy to `aggressive` (note that all attributes must be put, but the example omits most of them for the sake of clarity):

```
<config:configTransaction
  xmlns:config="http://www.endeca.com/MDEX/config/services/types/2/0"
  xmlns:mdex="http://www.endeca.com/MDEX/config/XQuery/2009/09">
  <config:putGlobalConfigRecord>
    <mdex:record xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      ...
      <mdex-config_Key type="mdex:string">global</mdex-config_Key>
      <mdex-config_MergePolicy type="mdex:string">aggressive</mdex-config_MergePolicy>
      ...
    </mdex:record>
  </config:putGlobalConfigRecord>
</config:configTransaction>
```

Changing the merge policy of a running data domain

The URL `/admin/dataDomainName?op=merge` command can be used to change the merge policy of a running Endeca data domain.

The sticky version of the `merge` command is intended to change the merge policy of a running data domain. The duration of the policy change is for the life of the current Dgraph process (that is, until the Dgraph process is restarted), or until another sticky change is performed during the current Dgraph process.

The format of the sticky version of the command is:

```
/admin/dataDomainName?op=merge&mergepolicy=<policy>&stickymergepolicy
```

where *policy* is either *balanced* or *aggressive*, and *dataDomainName* is the name of the Endeca data domain for which you are changing the policy.

The command also performs a merge operation if warranted.

This example:

```
http://web007:7001/endeca-server/admin/books?op=merge&mergepolicy=aggressive&stickymergepolicy
```

forces a merge operation (if one is needed) and changes the current merge policy to an aggressive policy for the "books" data domain. In this example, the Endeca Server is running on the machine named "web007" on port 7001.

Forcing a merge

The URL `/admin/dataDomainName?op=merge` command can also be used to force a merge.

Manually forcing a merge is considered a one-time version, because after the merge operation is performed (via a temporary *aggressive* change to the merge policy), the merge policy reverts to its previous setting.

The one-time version of the `merge` command is used to perform a complete merge of all generations without making a change to the default merge policy.

Forcing a merge implies starting a full merge of all generations of index files. When running this command, be aware of the following considerations:

- **Memory requirements.** Forcing a complete merge utilizes the server's memory. If the amount of memory reaches the amount of RAM that is available, the merge operation will continue to work, but could run substantially slower and have a higher impact on query performance.
- **Disk space requirements.** Forcing a merge requires provisioning three times the amount of disk space as the current size of the index files for the particular data domain. If not enough disk space is provisioned, it could be disruptive to force a complete merge. This consideration is especially important for running this command on the Oracle Endeca Server in a production environment.

In an Endeca Server cluster hosting a data domain cluster, when you issue this command for the data domain, it is routed to the leader Dgraph node for the data domain.

The format of the one-time version of the command is:

```
/admin/dataDomainName?op=merge&mergepolicy=<version>
```

The following example assumes that the Dgraph process of the data domain is using a balanced merge policy, and that you want to temporarily apply an aggressive policy so that the merging can be performed:

```
http://web007:7001/endeca-server/admin/books?op=merge&mergepolicy=aggressive
```

When you issue the command, the Dgraph process starts a manual merge of its index files. After the merging is performed, the merge policy reverts to its previous setting.

Linux unlimited settings for merges

For purposes of generation merging, it is recommended that you set the `ulimit -v` and `-m` parameters to unlimited.

An unlimited setting for the `-v` option sets no limit on the maximum amount of virtual memory available to a process, and for the `-m` option sets no limit on the maximum resident set size. These unlimited settings can help prevent problems when the Dgraph is merging the generation files.

An example of a merge problem due to insufficient disk space and memory resources is the following Dgraph error:

```
ERROR 04/03/13 05:24:35.668 UTC (1364966675668) DGRAPH {dgraph} BackgroundMergeTask:
exception thrown: Can't parse generation file Endeca.generation.v2-v446.dat, caused by
I/O Exception: While mapping file, caused by mmap failure
(file=/data/mdx/workspace/test_indexes/generations/Endeca.generation.v2-v446.dat
offset=0 length=28728057528): Cannot allocate memory
```

In this case, the problem was caused because the Dgraph could not allocate enough virtual memory for its merging task.



Chapter 7

Endeca Server Logging

This section describes how logging is managed by the Oracle Endeca Server and by the Dgraph.

[Endeca Server logs](#)

[Dgraph logs](#)

Endeca Server logs

Endeca Server uses Oracle Diagnostic Logging (ODL) for logging and its messages are written to WebLogic Server logs.

Each WebLogic Server instance maintains these log files:

- The `AdminServer-diagnostic.log` contains ODL messages from WebLogic applications. The Endeca Server standard output (stdout) and standard error (stderr) messages are written to this file.
- The `<domain-name>.log` file (for example, `endeca_server_domain.log`) receives a sub-set of the ODL messages. This domain log is intended to provide a central location from which to view the overall status of the domain.
- The `AdminServer.log` contains messages from its subsystems and applications. Although this log contains messages from the Endeca Server application (and is therefore useful to debug Endeca Server application problems), it does not contain Endeca Server ODL messages.

By default, these log files are located in the `$DOMAIN_HOME/servers/AdminServer/logs` directory. For example, if `endeca_domain` is the name of your domain, then the default path on Windows is:

```
C:\Oracle\Middleware\user_projects\domains\endeca_server_domain\servers\AdminServer\logs
```

All Endeca Server ODL log entries are prefixed with `OES` followed by the number and text of the message, as in this example:

```
OES-000135: Endeca Server has successfully initialized
```

Because all logs are text files, you can view their contents with a text editor. You can also view entries from the WebLogic Administration Console.

[Log entry format](#)

[Setting log levels](#)

[Creating log handlers for debugging](#)

[Customizing the HTTP access log](#)

Log entry format

This topic describes the format of Endeca Server log entries, including their message types and log levels.

As an example of an error message, assume that you use the `disable-dd` command on a nonexistent data domain (such as misspelling "test" as "tesxt"). The command fails and the resulting error in the command window is:

```
C:\Oracle\Middleware\EndecaServer7.5.1_1\endeca-cmd>endeca-cmd disable-dd text
SEVERE: Error while invoking endpoint "http://localhost:7001/endeca-server/ws/manage"
from client endeca-cmd encountered a problem.
caused by:
The Endeca Server returned an error: OES-000095: No such data domain: text
caused by:
OES-000095: No such data domain: text
  OES-000095: No such data domain: text
```

In addition, the full error (including exception information) is also logged in the `AdminServer-diagnostic.log` file in this format:

```
[2013-01-11T14:47:10.259-05:00] [AdminServer] [ERROR] [OES-000066]
[com.endeca.opmodel.ws.ManagePortImpl]
[tid: [ACTIVE].ExecuteThread: '90' for queue: 'weblogic.kernel.Default (self-tuning)'] [userId:
<anonymous>]
[ecid: 0000Jkf9fj191jZ_Tlo2yclGw6ko000001,0:1] [WEBSERVICE_PORT.name: managePort]
[APP: oracle.endecaserver#1.0
@7.5.1.0.0] [J2EE_MODULE.name: endeca-server.war] [WEBSERVICE.name: manage]
[J2EE_APP.name: oracle.endecaserver_1.0@7.5.0.0.0] OES-000066: Service error:
com.endeca.cluster.ClusterException$NoSuchDataDomainException:
OES-000095: No such data domain: tesxt
com.endeca.cluster.ClusterException$NoSuchDataDomainException:
OES-000095: No such data domain: tesxt[[
  at com.endeca.cluster.Cluster.setDataDomainEnabled(Cluster.java:903)
  at com.endeca.opmodel.ws.ManagePortImpl.disableDataDomain(ManagePortImpl.java:352)
  at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
  at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
  at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
  at java.lang.reflect.Method.invoke(Method.java:601)
...
  at weblogic.servlet.internal.ServletRequestImpl.run(ServletRequestImpl.java:1490)
  at weblogic.work.ExecuteThread.execute(ExecuteThread.java:256)
  at weblogic.work.ExecuteThread.run(ExecuteThread.java:221)
]]
```

In the message, the fields map to the following attributes:

Log entry field	Meaning
[2013-01-11T14:47:10.259-05:00]	The timestamp, which is the date and time when the message was generated.
[AdminServer]	The organization ID for the originating component, which is the WebLogic Admin Server.
[ERROR]	The ODL message type. See the table below for possible values.

Log entry field	Meaning
[OES-000066]	The message ID that uniquely identifies the message within the component. The ID consists of the prefix OES (representing the component), followed by a dash, then a number.
[com.endeca.opmodel.ws.ManagePortImpl]	The component ID, which identifies the Endeca Server component that originated the message.
[tid: [ACTIVE].ExecuteThread: '90' for queue: 'weblogic.kernel.Default (self-tuning)']	The ID of the thread that generated the message.
[userId: <anonymous>]	The name of the user whose execution context generated the message.
[ecid: 0000Jkf9fj191jz_Tlo2yclGw6ko000001, 0:1]	The Execution Context ID (ECID), which is a global unique identifier of the execution of a particular request in which the originating component participates.
[WEBSERVICE_PORT.name: managePort]	Supplemental Attribute that contains a component-specific attribute about the event.
[APP: oracle.endecaserver#1.0@7.5.1.0.0]	Application ID for the Endeca Server application.
[J2EE_MODULE.name: endeca-server.war]	Supplemental Attribute that contains a component-specific attribute about the event.
[WEBSERVICE.name: manage]	Supplemental Attribute that contains a component-specific attribute about the event.
[J2EE_APP.name: oracle.endecaserver_1.0@7.5.0.0.0]	Supplemental Attribute that contains a component-specific attribute about the event.
[OES-000066: Service error: ...]	Message Text. The text of the error message.

Message types and levels

The WebLogic loggers for Endeca Server are configured with the amount and type of information written to log files, by specifying the message type and level. For each message type, possible values for message level are from 1 (highest severity) through 32 (lowest severity). Generally, you need to specify only the type (that is, you do not need to specify the level).

When you specify the type, WebLogic returns all messages of that type, as well as the messages that have a higher severity. For example, if you set the message type to `WARNING`, WebLogic also returns messages of type `ERROR`.

The following shows the message types and the most common levels for each type, as well as the ODL log-level mapping to the Java equivalent.

ODL message type	Message level	Java equivalent	Description
ERROR	1	SEVERE	A serious problem that requires immediate attention from the administrator and is not caused by a bug in the product.
WARNING	1	WARNING	A potential problem that should be reviewed by the administrator.
NOTIFICATION	1	INFO	A major lifecycle event such as the activation or deactivation of a primary sub-component or feature. Level 1 is the default level for NOTIFICATION.
NOTIFICATION	16	CONFIG	A finer level of granularity for reporting normal events.
TRACE	1	FINE	Trace or debug information for events that are meaningful to administrators, such as public API entry or exit points.
TRACE	16	FINER	Detailed trace or debug information that can help Oracle Support diagnose problems with a particular subsystem.
TRACE	32	FINEST	Very detailed trace or debug information that can help Oracle Support diagnose problems with a particular subsystem.

The ERROR, WARNING, and NOTIFICATION with level 1 types have no performance impact. The performance impact on the other types and levels are as follows:

- NOTIFICATION with level 16: Minimal performance impact.
- TRACE with level 1: Small performance impact. You can enable this level occasionally on a production environment to debug problems.
- TRACE with level 16: High performance impact. This level should not be enabled on a production environment, except on special situations to debug problems.
- TRACE with level 32: Very high performance impact. This level should not be enabled in a production environment. It is intended to be used to debug the product on a test or development environment.

Setting log levels

The WebLogic Scripting Tool allows you to change log levels on the Endeca Server loggers.

You can use the WebLogic Scripting Tool (WLST) to change the default logging levels of the loggers used by the Endeca Server components. The WLST script is located in the `$MW_HOME/oracle_common/common/bin` directory.



Note: This topic explains how to modify the default Endeca Server loggers. For information on creating a new log handler for Endeca Server (which uses a separate log file), see [Creating log handlers for debugging on page 62](#).

Full documentation for WLST is provided in the *WebLogic Scripting Tool Command Reference*, which is available online at: http://docs.oracle.com/cd/E29597_01/web.1111/e13813/toc.htm

You can use the WLST `listLoggers()` command to list the Endeca Server loggers:

```
listLoggers(pattern="com.endeca.*")
```

The Endeca Server loggers are:

```
com.endeca.cluster.Cluster
com.endeca.features.TunnelingServlet
com.endeca.features.ws.ControlServletContextListener
com.endeca.opmodel.control.ClusterCoordinator
com.endeca.opmodel.control.DGraph
com.endeca.opmodel.control.DGraphStateMachine
com.endeca.opmodel.control.DGraphs
com.endeca.opmodel.opSupport.ClusterCoordinatorRunner
com.endeca.opmodel.opSupport.DGraphRunner
com.endeca.opmodel.opSupport.RealProcessHandler
com.endeca.opmodel.testhelp.Gatekeeper
com.endeca.opmodel.testhelp.Thrower
com.endeca.router.RoutingFilter
com.endeca.tunneling.util.StreamUtil
com.endeca.util.JRFUtil
com.endeca.util.PropertyUtil
com.endeca.util.TimingFilter
```

You can then use the WLST `setLogLevel()` command to set a new log level for a specific logger. The syntax is shown in this example:

```
setLogLevel(logger="com.endeca.cluster.Cluster", level="TRACE:1", persist=1)
```

where:

- `logger` is a mandatory parameter that specifies the full name of the logger to be modified. Note that the logger name is case-sensitive (which also means that it cannot use a wildcard).
- `level` is a mandatory parameter that specifies the log level. You can specify either an ODL level (such as `TRACE:16`) or a Java level (such as `FINER`). For a list of the levels, see [Message types and levels on page 59](#).
- `persist` is an optional parameter that specifies whether the level should be saved to the configuration file. The values are 0 (do not save the level) or 1 (save the level, which will persist across WebLogic restarts). The default is 1.

To set a log level for an Endeca Server logger:

1. Make sure that WebLogic Server is running.
2. From a command prompt, change to the WLST directory.

3. Start the WLST utility with the script for your operating system:
 - Linux: `./wlst.sh`
 - Windows: `wlst.cmd`
4. From within WLST, connect to WebLogic Server with your administrator username and password.

For example:

```
connect('weblogic', 'welcome1')
```

5. Use the `setLogLevel()` command to set a new log level for a specific Endeca Server logger.

For example:

```
setLogLevel(logger="com.endeca.opmodel.control.DGraph", level="TRACE:1")
```

6. Use the `listLoggers` command to verify that the log level was changed.

For example:

```
listLoggers(pattern="com.endeca.opmodel.control.DGraph")
```

7. When you have finished, use the `exit()` command to exit the WLST utility.

If you have an Endeca Server cluster, you can set the log level on a specific machine by adding a `target` option (at Step 5 above).

Keep in mind that the logger will continue to write to the same log file, which by default is `$DOMAIN_HOME/servers/AdminServer/logs/AdminServer-diagnostic.log`.

Creating log handlers for debugging

When you are debugging an Endeca Server problem, it is useful to create a log handler that only handles Endeca Server loggers and writes to an individual log file.

The WLST `configureLogHandler()` command can add a log handler with a variety of options. The basic syntax (used in this topic) is shown by this example:

```
configureLogHandler(name="endeca-handler", addToLogger="com.endeca", path="c:/EndecaServer.log",
  handlerType="oracle.core.ojdl.logging.ODLHandlerFactory", addHandler=1)
```

where:

- `name` specifies the name of the new log handler.
- `addToLogger` specifies the name of an existing logger to be added to this log handler.
- `path` specifies the absolute path and name of the log file.
- `handlerType` specifies the name of the Java class that provides the handler implementation. It must be an instance of `java.util.logging.Handler` or `oracle.core.ojdl.logging.HandlerFactory`.
- `addHandler` is a Boolean value (1=true, 0=false). If the value is true, then the named handler will be added.

If you have an Endeca Server cluster, you can create the log handler on a specific machine by adding a `target` option to the command.

For more `configureLogHandler()` command options, see the *WebLogic Scripting Tool Command Reference*.

To add a new log handler:

1. Make sure that WebLogic Server is running.
2. From a command prompt, navigate to the `$MW_HOME/oracle_common/common/bin` directory.
3. Start the WLST utility with the script for your operating system:
 - Linux: `./wlst.sh`
 - Windows: `wlst.cmd`
4. From within WLST, connect to WebLogic Server with your administrator username and password.

For example:

```
connect("weblogic", "welcome1")
```

5. Use the `setLogLevel()` command to add a logger and set a log level for it.

For example:

```
setLogLevel(logger="com.endeca", level="FINER", addLogger=1)
```

6. Use the `configureLogHandler()` command to create the log handler.

See the example above for the syntax.

7. Use the `listLogHandlers` command to verify that the log handler was created.

For example:

```
listLogHandlers(name="endeca-debug-handler")
```

The command output should look similar to this:

```
Handler Name: endeca-debug-handler
type: ODL
path: c:/temp/endeca-debug.log
```

8. When you have finished, use the `exit()` command to exit the WLST utility.

If you later want to delete the debug log handler, use this WLST command:

```
configureLogHandler(name="endeca-debug-handler", removeFromLogger="com.endeca", removeHandler=1)
```

Note that because you cannot delete a log level, you should set the log level to a very coarse level. For example:

```
setLogLevel(logger="com.endeca", level="SEVERE")
```

Customizing the HTTP access log

You can customize the format of the default HTTP access log.

By default, WebLogic Server keeps a log of all HTTP transactions in a text file. The file is named `access.log` and is located in the `$DOMAIN_HOME/servers/AdminServer/logs` directory.

The log provides true timing information from WebLogic, in terms of how long each individual Endeca Server request takes. This timing information can be important in troubleshooting a slow system.

Note that this setup needs to be done on a per-server basis. That is, in a clustered environment, this has to be done for the Admin Server and for every Managed Server. This is because the clone operation (done when installing a clustered environment) carries over SSL configuration, but does not carry over access log configuration.

The default format for the file is the common log format, but you can change it to the extended log format, which allows you to specify the type and order of information recorded about each HTTP communication. This topic describes how to add the following identifiers to the file:

- `date` — Date on which transaction completed, field has type `<date>`, as defined in the W3C specification.
- `time` — Time at which transaction completed, field has type `<time>`, as defined in the W3C specification.
- `time-taken` — Time taken for transaction to complete in seconds, field has type `<fixed>`, as defined in the W3C specification.
- `cs-method` — The request method, for example GET or POST. This field has type `<name>`, as defined in the W3C specification.
- `cs-uri` — The full requested URI. This field has type `<uri>`, as defined in the W3C specification.
- `sc-status` — Status code of the response, for example (404) indicating a "File not found" status. This field has type `<integer>`, as defined in the W3C specification.

To customize the HTTP access log:

1. Log into the Administration Server console.
2. In the Change Center of the Administration Console, click **Lock & Edit**.
3. In the left pane of the Console, expand **Environment** and select **Servers**.
4. In the Servers table, click the **AdminServer** name.
5. In the Settings for AdminServer page, select **Logging>HTTP**.
6. On the Logging > HTTP page, make sure that the **HTTP access log file enabled** checkbox is checked.
7. Click **Advanced**.
8. In the Advanced pane:
 - (a) In the Format list box, select **Extended**.
 - (b) In the Extended Logging Format Fields, enter this space-delimited string:


```
date time time-taken cs-method cs-uri sc-status
```
9. Click **Save**.
10. In the Change Center of the Administration Console, click **Activate Changes**.
11. Stop and then re-start WebLogic Server.

The following is an example of the configured HTTP access log with several log entries:

```
#Version:      1.0
#Fields:      date time time-taken cs-method cs-uri sc-status
#Software:    WebLogic
#Start-Date:  2013-03-22      15:23:40
2013-03-22   15:27:07      0.967   POST    /endeca-server/ws/manage      200
2013-03-22   15:27:32      7.301   POST    /endeca-server/es/control     200
2013-03-22   15:27:32      7.567   POST    /endeca-server/ws/manage      200
2013-03-22   16:23:35      0.219   GET     /endeca-server/ws/conversation/sh?wsdl  200
2013-03-22   16:23:35      0.0     GET     /favicon.ico                   404
```


2013-03-22	16:24:14	0.031	GET	/endeca-server/ws/conversation/sh?wsdl	200
2013-03-22	16:24:14	0.031	GET	/endeca-server/ws/conversation/sh?XSD	
				=lql_parser_types.xsd	200
2013-03-22	16:27:15	5.692	POST	/endeca-server/ws/conversation/sh	200
2013-03-22	16:27:50	0.624	POST	/endeca-server/ws/conversation/sh	200

Note that all the queries were successful (status code of 200), except for the one with the 404 status code.

Dgraph logs

Any Dgraph node in a data domain creates two logs.

You can use these Dgraph logs to troubleshoot queries, or to track performance of particular queries or updates.

By default, the Dgraph logs are stored in the `$DOMAIN_HOME/EndecaServer/logs` directory. For example, if your domain name is `endeca_server_domain`, the pathname on Windows is:

```
C:\Oracle\Middleware\user_projects\domains\endeca_server_domain\EndecaServer\logs
```

You can specify another location for the logs by changing the path in the `endeca-logs-dir` parameter in the `EndecaServer.properties` configuration file.

Because the Dgraph logs are text files, you can use a text editor to view them.

Dgraph request log

The Dgraph request log (also called the query log) contains one entry for each request processed. The requests are sorted by their timestamp.

The default name of the Dgraph request log is:

```
dataDomainName.reqlog
```

where `dataDomainName` is the name of the Endeca data domain that the Dgraph node is servicing.

Also, the `/admin/dataDomainName?op=logroll` command forces a query log roll, with the side effect of remapping the request log.

The format of the Dgraph request log consists of fourteen fields, which contain the following information:

- Field 1: Timestamp (UNIX Time with milliseconds)
- Field 2: Client IP Address
- Field 3: Outer Transaction ID, if defined
- Field 4: Request ID
- Field 5: Response Size (bytes)
- Field 6: Total Time (fractional milliseconds)
- Field 7: Processing Time (fractional milliseconds)
- Field 8: HTTP Response Code (0 on client disconnect)
- Field 9: - (unused)
- Field 10: Queue Status (on request arrival, number of requests in queue if positive, or number of available slots at the same priority if negative)

- Field 11: Thread ID
- Field 12: HTTP URL (URL encoded)
- Field 13: HTTP POST Body (URL encoded; truncated to 64KBytes, by default; - if empty)
- Field 14: HTTP Headers (URL encoded)

Note that a dash (-) is entered for any field for which information is not available or pertinent.

By default, the Dgraph truncates the contents of the body for POST requests at 64K. This default setting saves disk space in the log, especially during the process of adding large numbers of records to the data domain. If you need to review the log for the full contents of the POST request body, contact Oracle support.

Dgraph stdout/stderr log

The default name of the Dgraph stdout/stderr log is:

```
dataDomainName.out
```

The Dgraph stdout/stderr log includes startup messages as well as warning and error messages. You can increase the verbosity of the log via the Dgraph `-v` flag. You can also set the logging variables to toggle logging verbosity for specified features, which are described in [List of supported logging variables on page 49](#).

Note that the Dgraph stdout/stderr log reports startup and shutdown times (and other informational messages) using the system's local time zone, with no zone label displayed, but displays warning and error messages in UTC.

Using grep on the Dgraph request log

When diagnosing performance issues, you can use `grep` with a distinctive string to find individual requests in the Dgraph request log. For example, you can use the string:

```
value%3D%22RefreshDate
```

If you have Studio, it is more useful to find the `X-Endeca-Portlet-Id` HTTP Header for the portlet sending the request, and `grep` for that. This is something like:

```
X-Endeca-Portlet-Id:
endecareultslistportlet_WAR_endecareultslistportlet_INSTANCE_5RKp_LAYOUT_11601
```

As an example, if you set:

```
PORTLET=endecareultslistportlet_WAR_endecareultslistportlet_INSTANCE_5RKp_LAYOUT_11601
```

then you can look at the times and response codes for the last ten requests from that portlet with a command such as:

```
grep $PORTLET Discovery.reqlog | tail -10 | cut -d ' ' -f 6,7,8
```

The command produces output similar to:

```
20.61 20.04 200
80.24 79.43 200
19.87 18.06 200
79.97 79.24 200
35.18 24.36 200
87.52 86.74 200
26.65 21.52 200
81.64 80.89 200
28.47 17.66 200
82.29 81.53 200
```

There are some other HTTP headers that can help tie requests together:

- `X-Endeca-Portlet-Id` — The unique ID of the portlet in the application.
- `X-Endeca-Session-Id` — The ID of the user session.
- `X-Endeca-Gesture-Id` — The ID of the end-user action (not filled in unless Studio has CLIENT logging enabled).
- `X-Endeca-Server-Query-Id` — If multiple dgraph requests are sent for a single Endeca Server request, they will all have the same `X-Endeca-Server-Query-Id`.

For information on enabling Studio logging, see the *Oracle Endeca Information Discovery Studio Administration and Customization Guide*.



Dgraph Flag Reference

This appendix provides a description of the flags (options) used by the Dgraph process.

Dgraph flags

Dgraph flags

The Oracle Endeca Server starts the Dgraph process for each data domain node in the data domain cluster.

When you create a data domain profile, you can optionally specify that its Dgraph processes start with any of the flags in the following table. When the data domain is created based on this data domain profile, all Dgraph nodes in this data domain use the Dgraph flags that you specified for this data domain profile.

To obtain a listing of the Dgraph process arguments, at the Oracle Endeca Server host and port, issue the `put-dd-profile` command with the `--args --usage` flag:

```
endeca-cmd put-dd-profile --args --usage
```

The following Dgraph flags allow you to adjust its configuration:

Flag	Description
?	Print the help message and exit.
-v	Verbose mode. Print information about each request to <code>stdout</code> .
--ancestor_counts	Compute counts for root managed attribute values and any intermediate managed attribute value selections. By default, the Dgraph only computes refinement counts for proper refinements (in other words, for actual managed attribute values). It does not compute counts for root managed attribute values or for any intermediate managed attribute value selections.
--backlog-timeout <seconds>	Specify the wait limit (in seconds) for a query that has been read and queued for processing. This is the maximum number of seconds that a query is allowed to spend waiting in the processing queue before the Dgraph responds with a timeout message. The default value is 0 seconds.

Flag	Description
<code>--esampmin <num></code>	<p>Specify the minimum number of records to sample during refinement computation. The default is 0.</p> <p>Tuning recommendations:</p> <ul style="list-style-type: none"> • For most applications, larger values reduce performance without improving dynamic refinement ranking quality. • For some applications with extremely large, non-hierarchical managed attributes (if they cannot be avoided), larger values can meaningfully improve dynamic refinement ranking quality with minor performance cost.
<code>--implicit_exact</code>	<p>Disable approximate computation of implicit refinements.</p> <p>Use of this option is not recommended.</p> <p>If this option is not enabled, managed attribute values without full coverage of the current result record set may sometimes be returned as implicit refinements, although the probability of such "false" implicit refinements is minuscule.</p>
<code>--implicit_sample <num></code>	<p>Set the maximum number of records to sample when computing implicit refinements (which are a performance tuning parameter).</p> <p>The default value is 1024.</p>
<code>--net-timeout <num></code>	<p>Specify the maximum number of seconds the Dgraph waits for the client to download data from queries across the network.</p> <p>The default network timeout value is 30 seconds.</p>
<code>--pidfile <pidfile-path></code>	<p>Specify the file to which to write the process ID (PID).</p> <p>The default PID file is named <code>datadomain.pid</code> and is located in the <code>logs</code> directory of the Endeca Server.</p>
<code>--search_max <num></code>	<p>Specify the maximum number of terms for text search.</p> <p>The default value is 10.</p>
<code>--snip_cutoff <num></code>	<p>Limit the number of words in an attribute that the Dgraph evaluates to identify the snippet.</p> <p>If a match is not found within <code><num></code> words, the Dgraph does not return a snippet, even if a match occurs later in the attribute value.</p> <p>If the flag is not specified, or <code><num></code> is not specified, the default is 500.</p>
<code>--snip_disable</code>	<p>Globally disable snippeting.</p>

Flag	Description
<code>--sslcafile <CA-certfile-path></code>	Specify the path of the <code>eneCA.pem</code> Certificate Authority file that the Dgraph will use to authenticate SSL communications with other components that must communicate with the Dgraph. If not given, SSL mutual authentication is not performed.
<code>--sslcertfile <certfile-path></code>	Specify the path of the <code>eneCert.pem</code> certificate file that the Dgraph will use to present to any client for SSL communications. If not given, SSL is not enabled for Dgraph communications.
<code>--sslcipher <cipher-list></code>	Set one or more cipher names (such as RC4-SHA) that specify the minimum cryptographic algorithm that the Dgraph will use during the SSL negotiation. If multiple ciphers are specified, the names must be separated by colons.
<code>--stat-all</code>	Enable all available dynamic attribute value characteristics. Note that this option has performance implications and is not intended for production use.
<code>--stat-brel</code>	Create dynamic record attributes indicating the relevance rank assigned to full-text search result records.
<code>--syslog</code>	Direct all output to syslog.
<code>--unctret</code>	Specify to the Dgraph not to compute implicit managed attributes, and to only compute and present explicitly specified managed attributes, when displaying refinements in navigation results. Specifying this flag does not reduce the size of the resulting record set that is being displayed; however, it improves run-time performance of the Dgraph process. Be aware that if you use this flag, in order to receive meaningful navigation refinements, you need to make top-level precedence rules work for ALL outbound queries.
<code>--validate_data</code>	Validate that all processed data loads and then exit.
<code>--wildcard_max <count></code>	Specify the maximum number of terms that can match a wildcard term in a wildcard query that contains punctuation, such as <code>ab*c.def*</code> . The default is 100.

Index

A

- administrative tasks
 - admin and config operations 45
 - overview 8
- admin operations
 - flush 47
 - help 47
 - list of 46
 - logroll 47
 - merge 47
 - ping 47
 - stats 48
 - statsreset 48
- aggressive merge policy 52
- allocate-bulk-load-port command, Endeca Server 28

B

- balanced merge policy 52

C

- cache tuning for the Dgraph 40
- changing the merge policy of a data domain 54
- clone-dd command, Endeca Server 25
- config operations
 - about 45
 - for logging verbosity 48
 - help 51
 - list of 49
 - log-disable 50
 - log-enable 50
 - logging variables 49
 - log-status 51
- connecting a Web browser to the Oracle Endeca Server 41
- connection errors, Dgraph and client 43
- core dump files
 - in the Dgraph 42
 - managing 41
- create-dd command, Endeca Server 24

D

- delete-dd command, Endeca Server 25
- delete-dd-profile command, Endeca Server 35
- delete-node command, Endeca Server 31
- Dgraph 48
 - about 19, 20
 - admin operations 46
 - cache tuning 40
 - checking aliveness of 40

- config operations 49
- crash dump files on Linux 42
- crash dump files on Windows 42
- flags 68
- identifying connection errors 43
- logging variables for 48
- logs 65
- running multiple instances on a single machine 43
- what to collect for debugging 43
- dictionaries, updating 29
- disable-dd command, Endeca Server 26

E

- enable-dd command, Endeca Server 26
- EndecaCmd.properties file 23
- Endeca data domain
 - adding nodes 28
 - cloning 25
 - creating 24
 - creating profiles 32
 - deleting 25
 - disabling 26
 - displaying version 29
 - enabling 26
 - exporting 26
 - get health information about 28
 - get information about 27
 - get statistics about 28
 - importing 27
 - listing 29
 - logs 20
 - updating dictionaries 29
- Endeca Server
 - command interface, global options 35
 - configuration file 10
 - data domain logs 20
 - host option 36
 - logs 57
 - overview 9
 - port option 36
 - root option 36
 - starting 17
 - stopping 14
- Endeca Server commands
 - allocate-bulk-load-port 28
 - clone-dd 25
 - create-dd 24
 - delete-dd 25
 - delete-dd-profile 35
 - delete-node 31
 - disable-dd 26
 - enable-dd 26
 - export-dd 26

- get-cluster-health 31
- get-dd 27
- get-dd-health 28
- get-dd-profile 34
- get-dd-status 28
- get-node-profile 30
- global options 35
- import-dd 27
- list-dd-profiles 34
- list-dd-status 29
- list-nodes 31
- options for SSL 37
- put-dd-profile 32
- rescale-dd 28
- set-node-profile 30
- update-spelling-dictionaries 29
- version 29

- EndecaServer.properties file 10
- export-dd command, Endeca Server 26

F

- flags, Dgraph 68
- flush admin operation 47
- forcing a merge 55

G

- get-cluster-health command, Endeca Server 31
- get-dd command, Endeca Server 27
- get-dd-health command, Endeca Server 28
- get-dd-profile command, Endeca Server 34
- get-dd-status command, Endeca Server 28
- get-node-profile command, Endeca Server 30
- Global Configuration Record
 - retrieving with API 53
 - setting merge policy 54

H

- help admin operation 47
- help config operation 51
- HTTP access log 63

I

- import-dd command, Endeca Server 27
- incremental updates, merge policy for 52
- IPv4 and IPv6 address support in Endeca Server 41

L

- list-dd command, Endeca Server 29
- list-dd-profiles command, Endeca Server 34
- list-nodes command, Endeca Server 31
- log-disable config operation 50
- log-enable config operation 50

- logging variables 48
 - operation syntax 48, 50
 - supported variables for 49

- logroll admin operation 47

logs

- Dgraph 65
- Endeca Server 57
- message types 59
- setting Endeca Server log levels 61
- WebLogic HTTP access log 63

- log-status config operation 51

M

- merge admin operation 47

merge policy

- changing 53
- changing in a running data domain 54
- forcing a merge 55
- for incremental updates 52
- getting programmatically 53
- setting programmatically 54
- types of 52

N

- nodes, Endeca Server
 - deleting 31
 - getting cluster health 31
 - getting node profile 30
 - listing nodes 31
 - setting node profile 30

O

- operation syntax for Dgraph logging variables 48

Oracle Endeca Server

- connecting Web browsers to 41
- IPv4 and IPv6 address support 41

P

- ping admin operation 47
- profile, data domain
 - deleting 35
 - displaying profile settings 34
 - listing profiles 34
- put-dd-profile command, Endeca Server 32

R

- rescale-dd command, Endeca Server 28

S

- set-node-profile command, Endeca Server 30
- spelling, enabling, for the data domain 39
- SSL options, endeca-cmd 37
- stats admin operation 48

statsreset admin operation 48

U

update spelling dictionaries 39

update-spelling-dictionaries command, Endeca Server 29

URL operations, about 45

V

variables for Dgraph logging 49

version command, Endeca Server 29

W

WebLogic logs

AdminServer 57

HTTP access log 63

Who should use this guide 5

WLST logging commands 61