

Oracle® Endeca Server

Administrator's Guide

Version 7.7.0 • January 2016

Copyright and disclaimer

Copyright © 2003, 2016, 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	6
About this guide	6
Who should use this guide	6
Conventions used in this guide	6
Contacting Oracle Customer Support	7
Chapter 1: Introduction	8
Taking ownership of your implementation	8
Overview of administrator tasks	8
Chapter 2: Oracle Endeca Server as a Java Application	10
Overview of Oracle Endeca Server application	10
The EndecaServer.properties file	11
Using a data directory on a different drive	17
Starting Endeca Server	18
Stopping Endeca Server	20
Restarting WebLogic Server without stopping Endeca Server	23
Changing the network interface	24
Chapter 3: Endeca Server Command Reference	25
About the endeca-cmd utility	25
Command global options	26
Global options for host, port, and context root	27
Global options for SSL	28
Endeca Server node operations	30
Data domain profile operations	32
Data domain operations	38
Chapter 4: Managing Data Domains	47
The Endeca data domain and the Dgraph process	47
Adding, removing, changing and updating data domains	48
Adding a new data domain	49
Removing a data domain	51
Cloning an existing data domain	51
Adding nodes to a data domain	52
Updating a data domain	53
Importing, exporting, enabling, or disabling data domains	55
Exporting and importing a data domain	55
Enabling a data domain	57
Disabling a data domain	58

Monitoring and backing up data domains	58
Obtaining information about data domains	58
Monitoring the data domain health	60
Checking the status of the data domain nodes	60
Backing up the index	61
Collecting debugging information	61
Other tasks and characteristics of data domains	62
Memory consumption in the Endeca Server	62
About using control groups (cgroups) for data domains	63
Enabling cgroups for the data domain	64
Read-only data domains	66
Auto-idling and oversubscribing behavior of data domains	67
Issuing cache warming requests to the Dgraph	68
Configuring session affinity for data domains	69
About connecting Web browsers to data domains	70
Chapter 5: Dgraph Administrative Tasks	71
Dgraph flags	71
Administrative operations for the Dgraph	73
Syntax of administrative operations	73
List of administrative operations	74
help	75
flush	75
logroll	75
merge	75
ping	75
stats	76
statsreset	76
Managing the index merge policy for the data domain	76
Using a merge policy for incremental index updates	76
Types of merge policies	77
Setting the merge policy	77
Retrieving the merge policy	78
Changing the merge policy	78
Forcing a merge	79
Linux ulimit command for merges	80
Managing Dgraph core dump files	80
Managing Dgraph crash dump files on Windows	80
Managing Dgraph core dump files on Linux	81
Chapter 6: Endeca Server Logging	82
Endeca Server logs	82
Log entry format	83
Logging properties files	84
Customizing the HTTP access log	87
Endeca data domain logs	88
Dgraph request log and stdout/sterr log	89

Identifying Dgraph connection errors	91
List and syntax of configuration operations	91
Logging variables for the Dgraph process	92
Logging variable operation syntax	92
List of supported logging variables	93
log-enable	93
log-disable	94
log-status	94
help	94
Cluster Coordinator log	94
Gathering information for diagnosing problems	96

Preface

Oracle® Endeca Server is a hybrid search-analytical engine that organizes complex and varied data from disparate sources. At the core of Endeca Information Discovery, the unique NoSQL-like data model and in-memory architecture of the Endeca Server create an extremely agile framework for handling complex data combinations, eliminating the need for complex up-front modeling and offering extreme performance at scale. Endeca Server also supports 35 distinct languages.

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 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):

Section	Tasks
Oracle Endeca Server Server Commands	Operations for managing the Endeca Server application, and <code>endeca-cmd</code> Endeca Server command reference: <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.
Managing Data Domains	Operations for managing Endeca data domains: <ul style="list-style-type: none"> • Add, delete, rescale, import, export, enable, disable, update, or list Endeca data domains. Also, warm the Dgraph process cache for data domains. • Check the health of the data domain and the status of the Dgraph processes for the data domain.
Dgraph Administrative Tasks	<ul style="list-style-type: none"> • Use Dgraph flags. • Manage Dgraph process core dump files. • Troubleshoot Dgraph process socket and port errors. • Identify Dgraph process connection errors. • Use Dgraph administrative operations: <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. • Use the Dgraph index merge policy: <ul style="list-style-type: none"> • Merge data domain's index update generations. • Set and manage the merge policy for an Endeca data domain.
Endeca Server Logging	<ul style="list-style-type: none"> • Endeca Server logs, their format, log levels, and customization options. • Data domain logs • Dgraph logs, Dgraph configuration operations for setting log levels, and logging variables.



Chapter 2

Oracle Endeca Server as a Java Application

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](#)

[The EndecaServer.properties file](#)

[Using a data directory on a different drive](#)

[Starting Endeca Server](#)

[Stopping Endeca Server](#)

[Restarting WebLogic Server without stopping Endeca Server](#)

[Changing the network interface](#)

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: <https://docs.oracle.com/middleware/1213/wls/START/winservice.htm#START143>.

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 24](#).

The EndecaServer.properties file

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

The `EndecaServer.properties` file is located in the `config` directory of your domain. For example, assuming that "endeca_server_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_server_domain/config/EndecaServer.properties
```

The path on Windows is similar. The configuration file includes:

- [Endeca Server parameters on page 11](#)
- [Cluster Coordinator parameters on page 14](#)
- [Data Enrichment parameters on page 16](#)

The default values in the file are set by the Endeca Server installer.




Note: Most of these parameters are used by the Endeca Server application and should not be modified. If you do need to modify some of them, stop the Endeca Server on the machine on which you are modifying the parameter and restart the Endeca Server. If your Endeca Server is deployed as the Endeca Server cluster, make the changes to the `EndecaServer.properties` on all machines and then restart the Endeca Server instances — the Endeca Server relies on this file being the same on all of its instances.


The SSL-specific parameters, such as `endeca-ssl-keystore`, in the `Endeca.Server.properties` are empty at installation time. These 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 in `EndecaServer.properties` are specific to Endeca Server operations:

Endeca Server parameter	Description
<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).

Endeca Server parameter	Description
<code>endeca-offline-dir</code>	The location where the data files for an Endeca data domain are exported by the <code>export-dd</code> of <code>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 process standard out/err log and request log, as well as the PID file. The default is the <code>\$DOMAIN_HOME/EndecaServer/logs</code> directory. You can specify only one directory (that is, specifying multiple directories is not allowed).
<code>endeca-dgraph-install</code>	The location of the Endeca Server Dgraph application directory. The default is the <code>\$ENDECA_HOME/endeca-server/dgraph</code> directory.
<code>endeca-disable-entity-validation-refresh</code>	<p>(Optional). Specifies whether entity validation is enabled in the Endeca Server. If not present in the <code>EndecaServer.properties</code> file, validation happens (this is the default). Validation is skipped (disabled) only if you include this flag and set it to <code>true</code>. To change the value, edit the file and restart the Endeca Server.</p> <p> Note: This flag is needed for performance improvement in cases when you are adding large numbers of entities to the data domain, during an update or a data loading phase. This flag should not be used otherwise.</p>
<code>endeca-webserver-port</code>	The port on which the Endeca Server listens for requests. The default is 7001 (for non-SSL deployments), or 7002 (for SSL deployments).
<code>endeca-ds-port-min</code> and <code>endeca-ds-port-max</code>	<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. 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>
<code>endeca-ds-pin-timeout-min</code>	The default minimum value is 60000 milliseconds (ms). This is the lowest timeout value the Dgraph process can use when <code>PinDataVersion</code> request is issued. This parameter, as well as <code>endeca-ds-pin-timeout-max</code> and <code>endeca-ds-pin-timeout-default</code> are used in the Endeca Server to enable index version pinning requests. For information about index data version pinning, see the <i>Oracle Endeca Server Developer's Guide</i> .

Endeca Server parameter	Description
<code>endeca-ds-pin-timeout-max</code>	The default maximum value is 300000 ms. This is the highest timeout value the Dgraph process can use when <code>PinDataVersion</code> request is issued.
<code>endeca-ds-pin-timeout-default</code>	The default value for the timeout is 120000 ms. This is the default timeout value used by the Dgraph if you do not specify the value in the <code>PinDataVersion</code> request. (You typically use this request of the Conversation Web Service to pin a data version).
<code>endeca-ds-cert-file</code>	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. 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.
<code>endeca-ds-ca-file</code>	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. 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.
<code>endeca-secure-mode</code>	If set to <code>true</code> , specifies that Endeca Server is using SSL and that the Dgraphs should be started in SSL mode. If set to <code>false</code> , Endeca Server and the Dgraphs are not using SSL.
<code>endeca-memory-to-index-size-ratio</code>	<p>Specifies the ratio of virtual memory allocated for a data domain to the index size. The default is 2.0. This setting is used by the Endeca Server for data domain allocation on its nodes. It affects how the Endeca Server calculates whether it has sufficient amount of memory to allocate memory on its nodes to the newly-created data domains. For example, if the index size is 40MB and the default ratio of 2 is used, the Endeca Server will attempt to allocate 80 MB for the data domain.</p> <p> Note: This setting should only be modified by the system administrator of the overall Endeca Server deployment (such as a cluster), and should not be modified by the system administrators of specific data domains hosted in the Endeca Server. Before modifying this setting, learn more about how this setting is used and assess how it applies to your own deployment. For more information on how this setting is used, see the topic on data domain allocation in the <i>Oracle Endeca Server Cluster Guide</i>.</p>

Endeca Server parameter	Description
<code>endeca-threads-allowed-per-core</code>	<p>Specifies how many threads are actually allowed on a single core, on each Endeca Server node. The default is 1. This setting is used by the Endeca Server for data domain allocation. It affects how the Endeca Server calculates whether it has sufficient number of processing threads to allocate on its nodes to the newly-created data domains.</p> <p> Note: This setting should only be modified by the system administrator of the overall Endeca Server deployment (such as a cluster), and should not be modified by the system administrators of specific data domains hosted in the Endeca Server. For more information on this setting, see the topic on data domain allocation in the <i>Oracle Endeca Server Cluster Guide</i>.</p>
<code>endeca-cgroups-enabled</code> , <code>endeca-cgroups-specified-by-percentage</code> , <code>endeca-cgroups-reserved-ram-mb</code> , <code>endeca-cgroups-reserved-swap-mb</code> , <code>endeca-cgroups-reserved-ram-percentage</code> , <code>endeca-cgroups-reserved-swap-percentage</code>	<p>These settings are only applicable to the Endeca Server deployment on Linux 6 (Red Hat Enterprise Linux 6). These settings control how the Endeca Server uses the cgroups feature in Linux 6. For information, see About using control groups (cgroups) for data domains on page 63.</p>

Cluster Coordinator parameters

The following configuration settings in `EndecaServer.properties` are specific to the Endeca Server Cluster Coordinator:

Cluster Coordinator parameter	Description
<code>endeca-cluster-coordinator-dir</code>	<p>The location of the Endeca Cluster Coordinator root directory.</p> <p>The default is the <code>\$ENDECA_HOME/endeca-server/cluster-coordinator</code> directory.</p>
<code>endeca-cluster-coordinator-hosts</code>	<p>The default is <code>localhost</code> (which should be used for a single-node installation). For an Endeca Server cluster deployment, this parameter must contain comma-separated names of all Managed Servers that are part of the Cluster Coordinator ensemble. Always use the top-level domain names (or the first portions of the FQDN), for this field, such as <code>host1</code>, <code>host2</code>, <code>host3</code>. Do not use the full versions of Fully Qualified Domain Names.</p>
<code>endeca-cluster-coordinator-clientPort</code>	<p>The port of the server on which the Cluster Coordinator is running. This is the port at which the clients will connect.</p> <p>The default is 2181.</p>

Cluster Coordinator parameter	Description
<code>endeca-cluster-coordinator-tickTime</code>	<p>The length of a single tick, which is the basic time unit for the Cluster Coordinator.</p> <p>A tick is measured in milliseconds. It is used to regulate heartbeats and timeouts.</p> <p>The default is 2000 milliseconds.</p>
<code>endeca-cluster-coordinator-initLimit</code>	<p>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.</p> <p>The default is 10 ticks.</p>
<code>endeca-cluster-coordinator-syncLimit</code>	<p>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.</p> <p>The default is 5 ticks.</p>
<code>endeca-cluster-coordinator-dataDir</code>	<p>Sets the location of the directory where the Cluster Coordinator will store the in-memory database snapshots and, unless otherwise specified, the transaction log of updates to its database.</p> <p>The default is the <code>\$DOMAIN_HOME/EndecaServer/data</code> directory.</p>
<code>endeca-cluster-coordinator-dataLogDir</code>	<p>Sets the location of the directory where the Cluster Coordinator will store the transaction logs. This property allows a dedicated log device to be used, and helps avoid competition between logging and snapshots.</p> <p>The default is the <code>\$DOMAIN_HOME/EndecaServer/data</code> directory.</p> <p>Note that this property does not affect the location of the Cluster Coordinator main log (named <code>coordinator.log</code>), which by default is stored in the <code>\$DOMAIN_HOME/EndecaServer/logs</code> directory.</p>
<code>endeca-cluster-coordinator-auto-purge-snap-retain-count</code>	<p>Sets the number of snapshots to retain in the Cluster Coordinator data directory.</p> <p>The default is 3 snapshots.</p>
<code>endeca-cluster-coordinator-auto-purge-interval</code>	<p>Sets the purge task interval in hours.</p> <p>The default is 1 hour. Set to 0 (zero) to disable the auto-purge feature.</p>
<code>endeca-cluster-coordinator-logLevel</code>	<p>Sets the log level for entries in the Cluster Coordinator main log (named <code>coordinator.log</code>).</p> <p>By default, this property is not present in the file, and the default log level is <code>WARN</code>. However, this property can be added to change the log level for use in diagnosing problems.</p>

Cluster Coordinator parameter	Description
<code>endeca-cluster-coordinator-serverPort</code>	The port that follower nodes use to connect to the leader node. The default server port is 3181.
<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.

Data Enrichment parameters

The following configuration settings in `EndecaServer.properties` are specific to the Endeca Server Data Enrichment module. The Data Enrichment module is used for enrichments by Studio.

Data Enrichment parameters	Description
<code>endeca-data-enrichment-pluginsDir</code> , <code>endeca-data-enrichment-resourceDir</code> , <code>endeca-data-enrichment-collectionBatchSize</code> , <code>endeca-data-enrichment-collectionMatchRetries</code> , <code>endeca-data-enrichment-taskHistorySize</code>	<p>These properties define the configuration for the Data Enrichment module. The Data Enrichment module is packaged with the Endeca Server and is used with Studio.</p> <p>If you are not using Data Enrichment module in Studio, ignore these parameters. If you are using the Data Enrichment module in the context of Studio, you can use the defaults for these parameters without changing them. Note the following about these parameters:</p> <ul style="list-style-type: none"> • The <code>*-pluginsDir</code> and <code>*-resourcesDir</code> directories should point to the location on the shared file system (this is especially true if your deployment is hosted in the Endeca Server cluster with more than one Endeca Server machine). • <code>*-collectionBatchSize</code> defines the batch size for loading the data into the data enrichment modules. The default should be acceptable in most cases, but changing it may be required, depending on the characteristics of the data. Tweaking this value can have an impact on performance. • <code>*-collectionMaxRetries</code> defines the number of retries the Data Enrichment pipeline performs if it cannot obtain the current index version while performing a read operation from the index of the data domain in the Endeca Server. The default is 3. • <code>*-taskHistorySize</code> determines how many previous runs (and information about them) are stored, for each pipeline. The default is 10.

Using a data directory on a different drive

The `endeca-data-dir` parameter in the `EndecaServer.properties` file allows you to specify a different drive for the data domain's index. 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. The default drive is the same one on which the Endeca Server application directory is installed. You can also change the data directory for existing data domains.

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 index cannot be found at start-up). Therefore, use the following procedure to change the data directory.

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 indexes 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.

To change the data directory for existing data domains:

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.

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.

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:

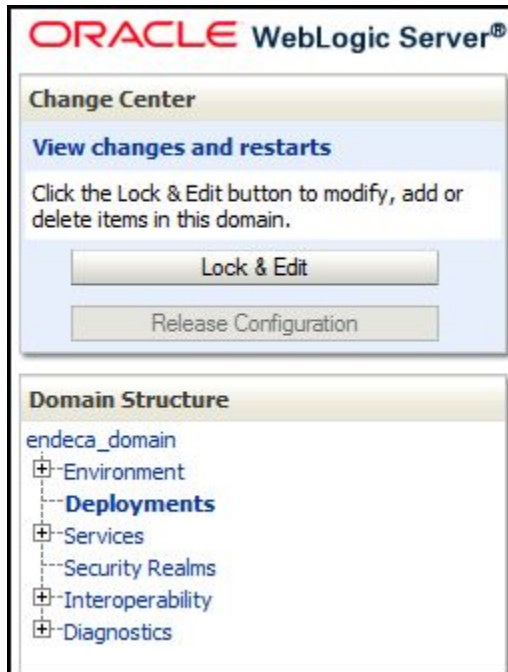
```
http://admin_server_host:admin_server_port/console
```

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).



5. 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 21 to 28 of 28 [Previous](#) | [Next](#)

<input type="checkbox"/>	Name	State	Health	Type	Deployment Order
<input checked="" type="checkbox"/>	 oracle.endecaserver (1.0,7.6.0.0.0)	Prepared	✓ OK	Enterprise Application	100
<input type="checkbox"/>	 oracle.jrf.system.filter	Active		Library	100
<input type="checkbox"/>	 oracle.jsp.next(11.1.1,11.1.1)	Active		Library	100
<input type="checkbox"/>	 oracle.pwdgen(11.1.1,11.1.1.2.0)	Active		Library	100
<input type="checkbox"/>	 oracle.wsm.seedpolicies(11.1.1,11.1.1)	Active		Library	100
<input type="checkbox"/>	 orai18n-adf(11.1.1.1.1.0)	Active		Library	100
<input type="checkbox"/>	 UIX(11,11.1.1.1.0)	Active		Library	100
<input type="checkbox"/>	 wsil-wls	Active	✓ OK	Enterprise Application	5

Install | Update | Delete | Start ▾ | Stop ▾ | Showing 21 to 28 of 28 [Previous](#) | [Next](#)

6. 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.

7. In the Stop Application Assistant, click **Yes**.

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

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:

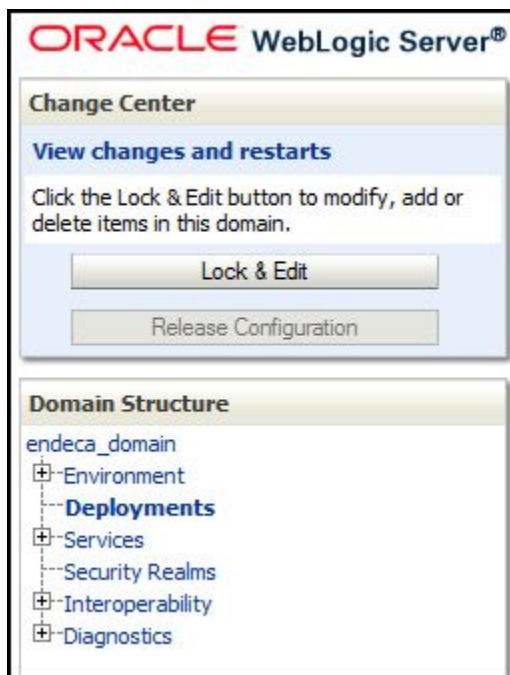
1. Make sure that the Administration Server for the Endeca domain is running.
2. From your browser, access the Administration Server console using this syntax:

```
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).



5. 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 ▾ | Stop ▾ | Showing 21 to 28 of 28 [Previous](#) | [Next](#)

<input type="checkbox"/>	Name	State	Health	Type	Deployment Order
<input checked="" type="checkbox"/>	oracle.endecaserver (1.0,7.6.0.0.0)	Active	✔ OK	Enterprise Application	100
<input type="checkbox"/>	oracle.jrf.system.filter	Active		Library	100
<input type="checkbox"/>	oracle.jsp.next(11.1.1,11.1.1)	Active		Library	100
<input type="checkbox"/>	oracle.pwdgen(11.1.1,11.1.1.2.0)	Active		Library	100
<input type="checkbox"/>	oracle.wsm.seedpolicies(11.1.1,11.1.1)	Active		Library	100
<input type="checkbox"/>	orai18n-adf(11.1.1.1.1.0)	Active		Library	100
<input type="checkbox"/>	UIX(11.1.1.1.1.0)	Active		Library	100
<input type="checkbox"/>	wsil-wls	Active	✔ OK	Enterprise Application	5

Install | Update | Delete | Start ▾ | Stop ▾ | Showing 21 to 28 of 28 [Previous](#) | [Next](#)

6. 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.
7. 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.

Restarting WebLogic Server without stopping Endeca Server

If you shut down WebLogic Server while the Endeca Server is running, when you restart WebLogic Server, two situations are possible. When Endeca Server is deployed on a single node only (in either cluster mode or non-cluster mode), if you shut down WebLogic Server, Endeca Server starts without delay. If you installed Endeca Server in a cluster mode and deployed it on more than one node, Endeca Server needs a limited time period to resolve its startup process.



Note: The remainder of this topic is only relevant if you installed in a cluster mode and deployed Endeca Server on more than one node. If you deployed Endeca Server on a single node (non-cluster or cluster mode), Endeca Server starts without delay.

When Endeca Server is installed in a cluster mode and deployed on more than one node, if you shut down WebLogic Server while the Endeca Server is running, when you restart WebLogic Server, the Endeca Server goes into a transient state. The state resolves automatically after a timeout. To avoid the transient state of the Endeca Server, before you stop the WebLogic Server, stop the Endeca Server application running in it.

The following statements describe this use case in greater detail:

- Shutting down and restarting the WebLogic Server without stopping the Endeca Server leads to the Endeca Server entering a transient state. This process continues for a configured period of time until it resolves itself.

The length of the transient state is controlled with the `endeca-cluster-coordinator-maxSessionTimeout` setting in the `EndecaServer.properties` file. This setting is balanced for optimal performance of data domains consisting of more than one Endeca Server instance. Oracle does not recommend changing this setting (for example, do not change this setting to 0). If the Endeca Server cluster administrators are considering changing this setting because of their specific network or cluster topology, they should contact Oracle Endeca Support.

- While in the transient state, the Endeca Server selects a new leader Dgraph node for the data domain. During this time, its data domain behaves as read-only — it accepts non-updating end-user queries and does not process updating requests. Once the transient state resolves itself, the new leader Dgraph node is elected and the data domain starts accepting updating queries. While the Endeca Server is in the transient state, an informational message is issued. If the transient state cannot be resolved for any reason, an error is issued. If you receive an error, restart the WebLogic Server domain.

For additional information about the Endeca Server cluster behavior (such as, for a description of how enhanced availability is achieved for data domains), see the *Oracle Endeca Server Cluster Guide*.

Changing the network interface

The network interface that the Endeca Server is bound to is defined by WebLogic Server's Listen Address setting. You can change this setting in the Administration Console.

To change the Endeca Server's network interface:

1. Log in to the Administration Console.
2. In the Change Center, click **Lock & Edit**.
3. Under **Environment**, click **Servers**.
4. Select the server's name from the **Servers** table.
The settings menu opens.
5. Enter the new address in the **Listen Address** field.
6. Click **Save**, then click **Activate Changes**.
7. Restart WebLogic Server.



Chapter 3

Endeca Server Command Reference

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

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

[Command global options](#)

[Endeca Server node operations](#)

[Data domain profile operations](#)

[Data domain operations](#)

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\EndecaServer7.7.0\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.

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`

Do not forget to specify global options with `endeca-cmd`. If you do not specify them, `endeca-cmd` assumes that the defaults are used for the Endeca Server (such as the default port and host). For example, assume you have configured the Endeca Server application in WebLogic domain to use a port that is different from a default port of 7001 (in a non-SSL installation). In this case, in order for `endeca-cmd` to find the correct port, you should list it explicitly, as one of the global options. For example, this command creates a data domain `my_test_dd` in the Endeca Server running on port 9001 (the data domain is disabled when created):

```
endeca-cmd create-dd my_test_dd --port 9001 --is-enabled false
```

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 26](#).

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>
<code>--ssl</code>	<p>Specifies whether to use an authenticated SSL connection to the Oracle Endeca Server.</p> <p>If you use either the <code>--keystore</code> or <code>--truststore</code> option, then you don't need to use the <code>--ssl</code> option, as the authenticated SSL connection is implied by specifying the keystore or the truststore file.</p> <p>If you use <code>--ssl</code> without either a <code>--keystore</code> or <code>--truststore</code> option, you can use an empty keystore and the <code>cacerts</code> file as your truststore. Alternatively, you can use the standard approaches for creating keystore and truststore files in the WebLogic Server.</p>
<code>--password</code>	<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>
<code>--keystore-password</code>	<p>Specifies a password for accessing the keystore (overrides the <code>--password</code> option).</p> <p>This option is intended for use in scripts.</p>

Option	Description
<code>--truststore-password</code>	Specifies a password for accessing the truststore (overrides the <code>--password</code> option). This option is intended for use in scripts.
<code>--key-password</code>	Specifies a password for accessing the key within the keystore (overrides the <code>--password</code> option). This option is intended for use in scripts.

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 31](#)
- [get-node-health on page 31](#)
- [get-cluster-health on page 31](#)
- [delete-node on page 31](#)
- [list-nodes on page 32](#)

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 amount of virtual memory on the Endeca Server node to allocate to the Endeca Server application. 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).
- A set of Dgraph flags to use to start up the Dgraph processes. These flags help fine-tune performance of search and other computations.
- Additional, low-level Dgraph flags.

When you create a data domain, Endeca Server creates an internal `.profile` file associated with each new data domain and stores it in the index of the data domain. The file contains the configuration of the profile with which the data domain was created. If you later update the data domain, its `.profile` file is updated. When you export the data domain, the `.profile` file is exported, and when you import the data domain, this file is used as the data domain's profile for the new data domain. You do not need to specify this `.profile` file when updating, exporting, or importing a data domain. It is used automatically.

The `.profile` file contains the data domain profile info, which is applied to the data domain. Its contents is similar to the following:

```
computeCacheSizeMB=0
sessionIdType=header
numFollowers=0
numComputeThreads=4
allowOversubscribe=true
shutdownTimeoutSeconds=30
startupTimeoutSeconds=600
args=
idleTimeoutMinutes=15
autoIdle=false
readOnly=false
description=this a customized data domain profile
allowQueriesOnLeader=true
sessionIdKey=X-Endeca-Session-ID
```


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.

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. However, you can create a different data domain profile, and use `endeca-cmd update-dd` command to update the existing data domain with this profile.

The commands for managing data domain profiles are:

- [put-dd-profile on page 33](#)
- [get-dd-profile on page 37](#)
- [list-dd-profiles on page 37](#)
- [delete-dd-profile on page 37](#)

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):

Option for put-dd-profile	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.
<code>--read-only <bool></code>	Whether a data domain should be created as read-only and not accept updating requests, including data loading requests. If not specified, defaults to <code>false</code> .

Option for put-dd-profile	Description
<code>--oversubscribe <bool></code>	Whether a data domain (or any of its nodes) can be placed on an oversubscribed Endeca Server instance. If not specified, defaults to <code>true</code> .
<code>--num-compute-threads <int></code>	Specifies the number of computational threads in the Dgraph node's threading pool. The value must be a positive integer that is equal or greater than 4. 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. If no value is specified or 0 is specified, this defaults to the larger of 4 or the number of available processors.
<code>--compute-cache-size <int></code>	Specifies an absolute value in MB for the Dgraph process cache, for each Dgraph node in the data domain. 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.
<code>--startup-timeout <int></code>	Specifies the maximum length of time (in seconds) that is allowed for the data domain's Dgraph process to start up. Default is 600 seconds.
<code>--shutdown-timeout <int></code>	Specifies the maximum length of time (in seconds) that is allowed for the data domain's Dgraph process to shut down. Default is 30 seconds.
<code>--ancestor-counts <bool></code>	Optional. If enabled, computes counts for root managed attribute values and any intermediate managed attribute value selections. The default is <code>false</code> . If this option is not enabled, the Dgraph only computes refinement counts for actual managed attribute values. It does not compute counts for root managed attribute values, or for any intermediate managed attribute value selections.
<code>--backlog-timeout <int></code>	Optional. 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.
<code>--refinement-sampling-min <int></code>	Optional. The minimum number of records to sample during refinement computation. The default is 0. 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 <bool></code>	Optional. Disables approximate computation of implicit refinements. Use of this option is not recommended. If this option is not enabled (this is the default), 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.

Option for put-dd-profile	Description
<code>--implicit-sample <int></code>	Optional. Sets the maximum number of records to sample when computing implicit refinements (which are a performance tuning parameter). The default value is 1024.
<code>--net-timeout <int></code>	Optional. Specifies the maximum number of seconds the Dgraph waits for the client to download data from queries across the network. The default network timeout value is 30 seconds.
<code>--search-max <int></code>	Optional. Sets the maximum number of terms for text search. The default value is 10.
<code>--search-char-limit <int></code>	Optional. Sets the maximum length (in characters) of a search term for record and value searches. The default is 132 characters. Any term exceeding this length will not be indexed, and thus will not be found in record and value searches.
<code>--snippet-cutoff <int></code>	Optional. Limits the number of words in an attribute that the Dgraph evaluates to identify the snippet. If a match is not found within <i><num></i> words, the Dgraph does not return a snippet, even if a match occurs later in the attribute value. If the flag is not specified, or <i><num></i> is not specified, the default is 500.
<code>--snippet-disable <bool></code>	Optional. Globally disables snippeting. The default is <i>false</i> , meaning snippeting is enabled.
<code>--dynamic-category-enable <bool></code>	Optional. Enable all available dynamic attribute value characteristics. The default is <i>false</i> . Note that this option has performance implications and is not intended for production use.
<code>--contraction-disable <bool></code>	Optional. Specifies 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. The default is <i>false</i> . 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>--wildcard-max <int></code>	Optional. Specifies the maximum number of terms that can match a wildcard term in a wildcard query that contains punctuation, such as <i>ab*c.def*</i> . The default is 100.

Option for put-dd-profile	Description
<code>--auto-idle <bool></code>	Specifies whether the Endeca Server should turn this data domain idle, if the data domain receives no queries during the specified <code>--idle-timeout</code> . If <code>auto-idle</code> is set to <code>false</code> , the data domain never turns idle. If set to <code>true</code> , the data domain turns idle after <code>idle-timeout</code> expires and if no queries arrive during the timeout period. The data domain is considered idle if it is set to <code>auto-idle</code> in its data domain profile, and if Endeca Server stops its Dgraph process. An idle data domain automatically activates if it receives a query. The timer for the timeout is reset once the idle data domain is activated.
<code>--idle-timeout <int></code>	Optional. Specifies the timeout period, in minutes, after which the data domain that is configured to <code>auto-idle</code> is turned idle by the Endeca Server, if the data domain receives no queries during the timeout period. The default is 15 minutes, and it is used if the timeout is not specified. If the timeout you specify is less than 15 minutes, Endeca Server issues a warning to the log file and resets the timeout to 15 minutes. 15 minutes is the lowest idle timeout value you can set. This timeout allows for long-running queries to complete successfully before the data domain is turned idle. If you have any queries running longer than 15 minutes, increase the idle timeout. This setting is not used for data domains for which <code>auto-idle</code> is set to <code>false</code> .
<code>--session-id-type <type></code>	The method to use for establishing session affinity. The available options are: <code>header</code> (for HTTP headers), <code>parameter</code> (for URL parameters), or <code>cookie</code> . The default method is <code>header</code> .
<code>--session-id-key <name></code>	The name of the key to use for maintaining affinity. The default name is <code>X-Endeca-Session-ID</code> .
<code>--args <dgraph-flags></code>	Specifies a list of the additional 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. See Dgraph flags on page 71 .

This example:

```
endeca-cmd put-dd-profile MyProfile --description "group profile" --oversubscribe false
--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: 4
ComputeCacheSizeMB: 0
StartupTimeoutSeconds: 600
ShutdownTimeoutSeconds: 30
AutoIdle: false
SessionIdType: HEADER
SessionIdKey: X-Endeca-Session-ID
Args: []
```

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.

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 38](#)
- [clone-dd on page 39](#)
- [delete-dd on page 40](#)
- [disable-dd on page 40](#)
- [enable-dd on page 40](#)
- [export-dd on page 40](#)
- [import-dd on page 41](#)
- [get-dd on page 42](#)
- [get-dd-health on page 42](#)
- [get-dd-status on page 42](#)
- [rescale-dd on page 42](#)
- [allocate-bulk-load-port on page 43](#)
- [list-dd on page 43](#)
- [update-dd on page 43](#)
- [warm-cache-dd on page 43](#)
- [version on page 44](#)
- [get-logs on page 44](#)

create-dd

The `create-dd` command creates, registers, and starts a new 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 index for the data domain is created 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`. Inside this index, an internal `.profile` file is created. This file is then used when you export the data domain (by exporting its index). This file is also used as the new data domain's profile, when you create a new data domain by importing a previously exported index. If you update a data domain profile, Endeca Server automatically updates the `.profile` in the data domain's index.



Note: You cannot use `create-dd` command to restore a previously created data domain based on its index. Use `import-dd` and `export-dd` for this purpose.

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

Create Option	Description
<code>--dd-profile-name</code> <i>name</i>	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</code> <i>boolean</i>	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 (including the `.profile` file 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 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	Description
<code>--dd-profile-name</code>	(Optional). The name of the data domain profile to use. Here is what happens if it is not specified: If you are importing a data domain that was exported from an Endeca Server before version 7.6.1.13, and do not specify a data domain profile, or if it does not exist in your system, Endeca Server uses the default data domain profile when creating a new data domain. If you are importing an index from the data domain created in Endeca Server 7.6.1.13 or greater, then you do not need to specify the data domain profile because the system automatically uses the <code>.profile</code> stored in the imported index.
<code>--is-enabled</code>	If set to <code>true</code> , indicates that the new data domain is enabled after it is imported. If this option is not specified, the imported data domain is enabled (this is the default). If you want to import but not start the data domain, specify <code>--is-enabled false</code> .

This example:

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

imports the index **MyDD** from the offline directory into the new data domain **NewDD**, 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. (This information is intended only for Oracle Endeca Support.)

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 <data-domain> [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: If a data domain is idle, issuing this command activates the data domain (by restarting its Dgraph process).

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-dd

The `update-dd` command updates the specified data domain with a new data domain profile. You can run this command only on a disabled data domain. The data domain remains disabled after running this command, and you must enable it.

The syntax for this command is:

```
endeca-cmd update-dd <data-domain> --dd-profile-name <profile-name> [global-options]
```

The `--dd-profile-name` option is required. It specifies the name of the data domain profile you want to use for this data domain. Thus before using this command, ensure that the data domain profile you want to use is already created.

This command changes all parameters of the data domain that are specified in the data domain profile, and automatically reallocates the data domain on the nodes in the Endeca Server cluster.

warm-cache-dd

The `warm-cache-dd` command warms the underlying Dgraph process cache for the specified data domain.

The syntax for this command is:

```
endeca-cmd warm-cache-dd <data-domain> [--time-limit-sec <sec>] [global-options]
```

The `--time-limit-sec` parameter is optional. It specifies the time limit during which you would like the warming script to run. If you do not specify the timeout, the default value of 1800 seconds (30 minutes) is used.

This command causes the Dgraph process to warm up its cache without the need to create a custom warm-up script. The command takes into account current Dgraph process usage pattern for selecting the set of queries for replay for a specified period of time. It allows the Dgraph to reuse its computation results across queries, and thus helps reduce the user-observable latencies in query processing and improves performance.

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]
```

get-logs

The `get-logs` command obtains WebLogic, Dgraph, and Cluster Coordinator (ZooKeeper) log files and compresses them to a single zip file. A configuration file specifies the settings needed by the command, such as the destination path for the zip file and whether the installation is a single-node or clustered environment.

Windows prerequisites:

- PowerShell 2.0 (or later) must be installed.
- The Java SDK `jar` command must be a recognized CMD command.
- In a clustered environment, a credential that can be used on all nodes is required. The `get-logs` will ask for the Remote Server username and password once. The input will be used as the credential for all nodes.

There are no special prerequisites for Linux.

Command syntax: The `get-logs` syntax is:

```
endeca-cmd get-logs <conf-file>
```

where *conf-file* is the name (including the path) of the configuration file for harvesting the logs.

Configuration file: The `$ENDECA_HOME/endeca-cmd/log-harvest` directory contains two versions of the `get-logs` configuration file: `getLogs_linux.prop` for Linux and `getLogs_windows.prop` for Windows.

Property	Platform	Meaning
CLUSTER_MODE	Both	Whether this is a cluster deployment: <ul style="list-style-type: none"> • If set to TRUE, it is a cluster deployment. • If set to FALSE, it is a single-node deployment.

Property	Platform	Meaning
ORACLE_HOME	Both	<p>The path to the WebLogic root directory. Examples:</p> <pre>// Windows: C:\WebLogic\Oracle\Middleware // Linux: /localdisk/WebLogic/Oracle/Middleware</pre>
ENDECA_SERVER_DOMAIN_NAME	Both	<p>The name of the WebLogic domain for Endeca Server. Example:</p> <pre>ENDECA_SERVER_DOMAIN_NAME=endeca_domain</pre>
DEST_PATH_SINGLE	Windows	<p>For single-node environments, specifies the name of the folder where the log packages will be stored. Example:</p> <pre>C:\tmp</pre> <p>This property is ignored if <code>CLUSTER_MODE=TRUE</code>.</p>
TARGET_SERVERS	Both	<p>For cluster environments, specifies a comma-separated list of servers to get logs from. The servers must be specified by short names. For example, if the server names are <code>web01.example.com</code>, <code>web02.example.com</code>, and <code>web03.example.com</code>, you use:</p> <pre>TARGET_SERVERS=web01,web02,web03</pre> <p>This property is ignored if <code>CLUSTER_MODE=FALSE</code>, in which case <code>get-logs</code> will run only on localhost.</p>
DEST_PATH	Linux	<p>The name of the directory where the log packages will be stored:</p> <ul style="list-style-type: none"> For cluster environments, the directory must be a shared directory. For single-node environments, a local path can be used.
SHARE_FOLDER	Windows	<p>For cluster environments, specifies the name of a directory shared by all the machines. For example:</p> <pre>\\server\share</pre> <p>This property is ignored if <code>CLUSTER_MODE=FALSE</code>.</p>
DRIVE_NAME	Windows	<p>For cluster environments, specifies the drive mapped to the share folder (the <code>SHARE_FOLDER</code> property). For example:</p> <pre>Z:</pre> <p>This property is ignored if <code>CLUSTER_MODE=FALSE</code>.</p>

Property	Platform	Meaning
DEST_PATH_CLUSTER	Windows	<p>For cluster environments, specifies the name of the folder where the log packages will be stored. This will be created under the share folder (the SHARE_FOLDER property). Example:</p> <pre>endecaserver\log-harvest</pre> <p>This property is ignored if CLUSTER_MODE=FALSE.</p>

Returned log files: The command produces a ZIP file containing the gathered logs. The package name is:

```
ES7.7-log-package_<timestamp>.zip
```

where the format of *timestamp* is: YYYYMMDDHHMMSSSS

When unzipped on a single-node environment, two directories are extracted:

- One directory contains the Dgraph logs and the Cluster Coordinator log.
- The other directory contains the WebLogic logs, plus the EndecaServer.properties configuration file.

The cluster version will include a set of Dgraph and WebLogic logs for each node.

Note that unzipping any log package creates a directory named ES7.7-log-packages. the ES7.7-log-packages directory is used as a temp folder. Therefore, unzipping subsequent log package ZIPs may overwrite the ES7.7-log-packages directory, which should not matter as the original ZIP file contents are not changed.

Windows example of running get-logs on a single-node environment: The command line is:

```
endeca-cmd get-logs .\log-harvest\getLogs_windows.prop
```

The command response is:

```
[fcal-pc] Creating temp folder ES7.7-log-packages under C:\tmp.....Success
[fcal-pc] Collecting WebLogic logs.....Success
[fcal-pc] Collecting Dgraph and ZooKeeper logs.....Success
[fcal-pc] Compressing all logs under C:\tmp\ES7.7-log-packages.....Success

*****
Generated log package C:\tmp\ES7.7-log-package-2015111813383322.zip
*****
```



Chapter 4

Managing Data Domains

This section describes how to add or delete data domains, change the number of data domain nodes, and do other maintenance tasks, such as copying, importing, enabling, and monitoring a data domain.

The Endeca data domain and the Dgraph process

Adding, removing, changing and updating data domains

Importing, exporting, enabling, or disabling data domains

Monitoring and backing up data domains

Other tasks and characteristics of data domains

The Endeca 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.

When you create a data domain, Endeca Server creates an internal `.profile` file associated with each new data domain and stores it in the index of the data domain. The file contains the configuration of the profile with which the data domain was created. If you later update the data domain, its `.profile` file is updated. When you export the data domain, the `.profile` file is exported, and when you import the data domain, this file is used as the data domain's profile for the new data domain.

The `.profile` file contains the data domain profile info, which is applied to the data domain. Its contents is similar to the following:

```
computeCacheSizeMB=0
sessionIdType=header
numFollowers=0
```

```
numComputeThreads=4
allowOversubscribe=true
shutdownTimeoutSeconds=30
startupTimeoutSeconds=600
args=
idleTimeoutMinutes=15
autoIdle=false
readOnly=false
description=this a customized data domain profile
allowQueriesOnLeader=true
sessionIdKey=X-Endeca-Session-ID
```

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.

Adding, removing, changing and updating data domains

You can create a new data domain, clone an existing data domain, change the number of data domain nodes, and update the data domain to use another data domain profile.

[Adding a new data domain](#)

[Removing a data domain](#)

[Cloning an existing data domain](#)

[Adding nodes to a data domain](#)

[Updating a data domain](#)

Adding a new data domain

You add a data domain with the `endeca-cmd create-dd name` command, where *name* is the name of your data domain. Running this command is equivalent to using the `createDataDomain` operation in the Manage Web Service request.

Before you can create a data domain on multiple Endeca Server nodes, the following conditions must be met:

- The Endeca Server software must be installed and deployed as an Endeca Server cluster on multiple WebLogic Server machines. Additionally, the Cluster Coordinator services of the Endeca Server should be running on an odd number of Endeca Server nodes.



Note: In a development environment, you can install and deploy an Endeca Server cluster and its Cluster Coordinator service on a single machine. Such a cluster will not be highly available, but it will still allow you to create a data domain that could be used for testing purposes. To run a data domain in production, you should recreate the data domain in the Endeca Server cluster deployed on multiple WebLogic Server machines. For information on installation and deployment in a cluster, see the *Oracle Endeca Server Installation Guide*.

- An administrator responsible for the Endeca Server cluster deployment has configured a node profile for the Endeca Server nodes, informing you, as the data domain cluster administrator, of the available capacity on the Endeca Server nodes, and the host and port of the Endeca Server cluster.
- An administrator of the Endeca Server cluster has created data domain profiles you can use. When you create a data domain, you specify one of these data domain profiles. Optionally, you can use the `default` data domain profile, which always exists in the system. For information on data domain profiles, including the default profile, see the *Oracle Endeca Server Cluster Guide*.



Note: While you can use `endeca-cmd create-dd` command to create a new data domain, you cannot use this command for restoring a previously created data domain. To restore a previously created data domain based on its index, use the `import-dd` and `export-dd` commands.

To create a data domain:

1. Use a command-line window (for example, open a Command Prompt in Windows) and navigate to the `endeca-cmd` directory.
2. Use `endeca-cmd list-dd-profiles --verbose` or the equivalent operation in the Manage Web Service (`listDataDomainProfiles`), to obtain a list of all defined data domain profiles.

The Endeca Server returns a list of all defined data domain profiles, including their characteristics, such as name, description, the number of data domain nodes, the number of processing threads for the Dgraph processes, the cache size, whether the data domain requires a dedicated leader node, and others. This list also includes the default data domain profile.

3. Select the data domain profile from this list. You will use its name to create a data domain. You can also select `default`, which is the default data domain profile that always exists.
4. Create a data domain using the command similar to the following example:

```
endeca-cmd create-dd MyDD
--dd-profile-name test
--is-enabled true
```

where:

- `MyDD` is the name of the data domain you are creating.

- `test` is the name of the data domain profile that will be used.
- `--is-enabled true` indicates that this data domain should be enabled once created.

The name of the data domain 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.

If the data domain profile exists (the default data domain profile always exists, but you may have created your own data domain profile), and the Endeca Server cluster has sufficient resources to host the new data domain, it is created.

Alternatively, you can issue the request similar to the following example, with the Manage Web Service:

```
<ns1:createDataDomain xmlns:ns1="http://www.endeca.com/endeca-server/manage/2/0">
  <ns1:name>MyDD</ns1:name>
  <ns1:ddProfileName>default</ns1:ddProfileName>
  <ns1:enabled>true</ns1:enabled>
</ns1:createDataDomain>
```

This example uses the default data domain profile that always exists in the Endeca Server cluster.

As a result of creating a data domain, Endeca Server creates a `.profile` file associated with each new data domain and stores it in the index of the data domain. The file contains the configuration of the profile with which the data domain was created. If you later update the data domain, its `.profile` file is updated. When you export the data domain, the `.profile` file is exported. You do not need to specify this `.profile` file when updating, exporting, or importing. It is used automatically.

5. To verify the data domain has been created successfully, issue one of these commands: `endeca-cmd list-dd`, or `endeca-cmd get-dd name`

Alternatively, you can issue `listDataDomains`, or `getDataDomain` requests, using the Manage Web Service.

The following example illustrates the result of the `getDataDomain` operation. It lists the details of the successfully created data domain `MyDD`. This data domain uses the data domain profile for which additional arguments have been specified (as compared with the default data domain profile):

```
<ns3:getDataDomainResponse
  xmlns:ns2="http://www.endeca.com/endeca-server/types/2/0"
  xmlns:ns3="http://www.endeca.com/endeca-server/manage/2/0">
  <ns3:dataDomainDetail>
    <ns2:name>MyDD</ns2:name>
    <ns2:allowOversubscribe>true</ns2:allowOversubscribe>
    <ns2:allowQueriesOnLeader>true</ns2:allowQueriesOnLeader>
    <ns2:numFollowers>0</ns2:numFollowers>
    <ns2:readOnly>false</ns2:readOnly>
    <ns2:enabled>true</ns2:enabled>
    <ns2:numComputeThreads>4</ns2:numComputeThreads>
    <ns2:computeCacheSizeMB>0</ns2:computeCacheSizeMB>
    <ns2:startupTimeoutSeconds>600</ns2:startupTimeoutSeconds>
    <ns2:shutdownTimeoutSeconds>30</ns2:shutdownTimeoutSeconds>
    <ns2:sessionIdType>header</ns2:sessionIdType>
```

```
<ns2:sessionIdKey>X-Endeca-Session-ID</ns2:sessionIdKey>
</ns3:dataDomainDetail>
</ns3:getDataDomainResponse>
```

Removing a data domain

You remove a data domain with `endeca-cmd delete-dd name`, or with the `deleteDataDomain` request of the Manage Web Service.

When the data domain is removed, the Endeca Server removes all its associated resources, which include source records, its `.profile` file, its configuration, and Endeca Server index. No backup of any sort is performed. When a data domain is deleted it cannot be restored.

To delete a data domain:

1. Use a command-line window (for example, open a Command Prompt in Windows) and navigate to the `endeca-cmd` directory.

Alternatively, access the Manage Web Service of the Endeca Server cluster at the URL, similar to the following example: `http://my-endeca-server:7001/endeca-server/ws/manage`. In this example, `my-endeca-server` is the host name of the Endeca Server cluster, `7001` is the port, and `endeca-server` is the root context.

2. Depending on the method you chose, use one of these options:

Option	Description
<code>deleteDataDomain</code> request	This request in the Manage Web Service deletes a data domain if you specify its name.
<code>endeca-cmd delete-dd name</code>	This command lets you delete the data domain if you specify its name.

3. To verify that the data domain has been deleted, list data domains and review the list.

Cloning an existing data domain

You can add a new data domain that replicates an existing one with the `endeca-cmd clone-dd name` command, or with the `cloneDataDomain` operation of the Manage Web Service.

Before you can clone a data domain with this command, these tasks should be completed:

- The Endeca Server node profiles are defined.
- The Endeca Server nodes are started in the Endeca Server cluster.
- The data domain for which you want to create a copy has been created.



Important: This data domain must be enabled for the clone command to succeed.

The name of the new data domain should not be already in use by any other data domain in the Oracle Endeca Server.

To clone a data domain:

1. Use a command-line window (for example, open a Command Prompt in Windows) and navigate to the `endeca-cmd` directory. Alternatively, you can access the Manage Web Service at the host and port of your Endeca Server cluster.
2. Depending on the method you choose, issue the `endeca-cmd clone-dd` command, or the `cloneDataDomain` operation with the Manage Web Service.

Specify the name for the new data domain on the command line, the name of the source data domain with `--source-name`, and (optionally) whether the domain should be enabled. If you don't specify the value for enabling, the new data domain is enabled by default.

The following example illustrates the `endeca-cmd` command:

```
endeca-cmd clone-dd MyNewDD --source-name MyExistingDD --is-enabled true
```

This example creates a copy of the `MyExistingDD` data domain, gives it a name `MyNewDD`, and enables this new data domain after it has been created.

Alternatively, you can issue a web service request similar to this example, to the Manage Web Service:

```
<ns1:cloneDataDomain xmlns:ns1="http://www.endeca.com/endeca-server/manage/2/0">
  <ns1:name>MyNewDD</ns1:name>
  <ns1:sourceName>MyExistingDD</ns1:sourceName>
  <ns1:enabled>true</ns1:enabled>
</ns1:cloneDataDomain>
```

If the Endeca Server cluster has sufficient capacity, the new data domain is created as a copy of the existing data domain.

3. Verify the creation of the cloned data domain by listing all existing data domains in the Endeca Server with `endeca-cmd list-dd`.

Adding nodes to a data domain

To rescale, or add Dgraph nodes to a running data domain, use the `endeca-cmd rescale-dd` command, or the `rescaleDataDomain` operation of the Manage Web Service.

Each data domain cluster consists of zero or one leader node and zero or more follower nodes. When you create a data domain, you use the data domain profile that determines the total number of Dgraph nodes, whether the data domain is read-only (and thus does not have a leader node), and the number of follower nodes.

You can add Dgraph follower nodes to a data domain that is either enabled or disabled. Adding follower nodes does not affect the Endeca Server query processing for the data domain hosted on these nodes.

When you add follower nodes, this changes the profile of the specific data domain. However, the domain profile that was used to initially create this data domain does not change.

Before you add follower nodes:

- Verify that the Endeca Server cluster contains a sufficient number of Endeca Server nodes (Remember that each data domain node must be hosted on a separate Endeca Server node). Issue the `endeca-cmd list-nodes` and `endeca-cmd get-node-profile` commands, or their equivalent operations in the Cluster Web Service. This should provide you with information about the capacity of the Endeca Server cluster.

- Check the current number of nodes in the data domain. Issue the `endeca-cmd get-dd my_dd` command (or an equivalent Manage Web Service operation, `getDataDomain`), to obtain information about the current number of follower Dgraph nodes (specified with `--num-followers` or its equivalent in the Manage Web Service).

If the number of follower nodes is zero, this indicates that the data domain consists of one leader node that is also configured to handle read-only queries, in addition to handling update requests. If the number of follower nodes is 1, this means that the total number of nodes in the data domain is 2.

To add follower nodes to a data domain:

1. Use a command-line window (for example, open a Command Prompt in Windows) and navigate to the `endeca-cmd` directory.

Alternatively, access the Endeca Server cluster's Manage Web Service at a URL similar to the following: `http(s)://host:port/endeca-server/ws/manage`, where the protocol depends on whether the Endeca Server configuration is secure or not, `host` and `port` represent the host name and port of the Endeca Server, and `endeca-server` is the default context root.

2. Use the `rescaleDataDomain` operation of the Manage Web Service, as in the following abbreviated example:

```
<ns1:rescaleDataDomain xmlns:ns1="http://www.endeca.com/endeca-server/manage/2/0">
  <ns1:name>my_data_domain</ns1:name>
  <ns1:numFollowers>1</ns1:numFollowers>
</ns1:rescaleDataDomain>
```

Alternatively, you can use: `endeca-cmd rescale-dd my_data_domain --num-followers 1`.

This operation asks the Endeca Server cluster to start another Dgraph follower node for the data domain `my_data_domain`.

If the Endeca Server cluster that hosts this data domain has a sufficient number of nodes, the operation succeeds and adds an additional Dgraph follower node.

If the number of Endeca Server nodes available to host additional data domain nodes is not sufficient, an error message is issued.

Updating a data domain

You update a data domain with the `endeca-cmd update-dd name --dd-profile-name` command, where `name` is the name of your data domain, and the `--dd-profile-name` option specifies the data domain profile you want to use (this option is required). Running this command is equivalent to using the `updateDataDomain` operation in the Manage Web Service request.

Using the `update-dd` command is useful when, for example, you would like to troubleshoot the Dgraph process for the data domain. In this case, you can create a data domain profile that is equivalent to the used profile in all respects, but with additional flags, and then update the data domain to use this profile. Also, updating a data domain is useful when you want to change any characteristics of the data domain specified in its profile, (such as, whether the leader node should process queries, or the cache size for the data domain).

If you compare the `update-dd` command to `rescale-dd`, in `rescale-dd`, you can only add more nodes to a running data domain. By comparison, `update-dd` lets you change any parameter of a data domain that is controlled by its data domain profile, but, unlike `rescale-dd`, the `update-dd` command only works on an offline data domain: you disable the data domain, update it, and then re-enable it.

Before you update a data domain in the Endeca Server, the following conditions must be met:

- The data domain must be disabled.
- Another data domain profile (different from the profile used by your data domain) must exist. You specify it when you update a data domain. For information on data domain profiles, see the *Oracle Endeca Server Cluster Guide*.
- The Endeca Server must have sufficient resources to utilize the new data domain profile.

To update a data domain:

1. Disable the data domain: `endeca-cmd disable-dd name`.
2. Use a command-line window (for example, open a Command Prompt in Windows) and navigate to the `endeca-cmd` directory.
3. Use `endeca-cmd list-dd-profiles --verbose` or the equivalent operation in the Manage Web Service (`listDataDomainProfiles`), to obtain a list of all defined data domain profiles.

The Endeca Server returns a list of all defined data domain profiles, including their characteristics, such as name, description, the number of data domain nodes, the number of processing threads for the Dgraph processes, the cache size, whether the data domain requires a dedicated leader node, and others. This list also includes the default data domain profile.

4. Select the data domain profile you would like to use from this list. You will use its name to update a data domain.
5. Update a data domain to use the new data domain profile, using the command similar to the following example:

```
endeca-cmd update-dd MyDD --dd-profile-name test
```

where:

- `MyDD` is the name of the data domain you are updating.
- `test` is the name of the data domain profile that will be used (this option is required).



Note: The name of the data domain can include spaces (for example, if it consists of two words). In this case, the name should be enclosed in double quotes, as in this example: `endeca-cmd update-dd "My data" --dd-profile-name test`. Thereafter, the name should also be enclosed in double quotes when used in other Endeca Server commands.

Alternatively, you can issue the request similar to the following example, with the Manage Web Service:

```
<ns1:updateDataDomain xmlns:ns1="http://www.endeca.com/endeca-server/manage/2/0">
  <ns1:name>MyDD</ns1:name>
  <ns1:ddProfileName>test</ns1:ddProfileName>
</ns1:updateDataDomain>
```

This example uses the data domain profile named `test`.

As a result of this command, the Endeca Server changes those parameters of the data domain that are specified in the data domain profile, and automatically reallocates the data domain on the nodes in the Endeca Server cluster. If the data domain profile exists, and the Endeca Server cluster has sufficient resources, the data domain is updated.

- To verify the data domain has been updated successfully, issue one of these commands: `endeca-cmd list-dd`, or `endeca-cmd get-dd name`

Alternatively, you can issue `listDataDomains`, or `getDataDomain` requests, using the Manage Web Service.

You can also now enable the data domain.

The following example illustrates the result of the `getDataDomain` operation. It lists the details of the successfully updated data domain `MyDD`. This data domain is not yet enabled, and it uses the data domain profile with additional arguments, (as compared with the default data domain profile):

```
<ns3:getDataDomainResponse xmlns:ns2="http://www.endeca.com/endeca-server/types/2/0"
  xmlns:ns3="http://www.endeca.com/endeca-server/manage/2/0">
  <ns3:dataDomainDetail>
    <ns2:name>MyDD</ns2:name>
    <ns2:allowOversubscribe>true</ns2:allowOversubscribe>
    <ns2:allowQueriesOnLeader>true</ns2:allowQueriesOnLeader>
    <ns2:numFollowers>0</ns2:numFollowers>
    <ns2:readOnly>false</ns2:readOnly>
    <ns2:enabled>false</ns2:enabled>
    <ns2:numComputeThreads>4</ns2:numComputeThreads>
    <ns2:computeCacheSizeMB>0</ns2:computeCacheSizeMB>
    <ns2:startupTimeoutSeconds>600</ns2:startupTimeoutSeconds>
    <ns2:shutdownTimeoutSeconds>30</ns2:shutdownTimeoutSeconds>
    <ns2:sessionIdType>header</ns2:sessionIdType>
    <ns2:sessionIdKey>X-Endeca-Session-ID</ns2:sessionIdKey>
  </ns3:dataDomainDetail>
</ns3:getDataDomainResponse>
```

Importing, exporting, enabling, or disabling data domains

You can import, export, enable, and disable data domains.

[Exporting and importing a data domain](#)

[Enabling a data domain](#)

[Disabling a data domain](#)

Exporting and importing a data domain

Exporting a data domain takes a snapshot of the data domain's index files, which is useful for backups. Importing a data domain lets you create a new data domain based on an exported index.

When you export a data domain, the Endeca Server takes a snapshot of its index files and stores it in the offline directory on a shared file system. While the data domain's index is exported, the data domain continues to run. If you need to export the same data domain again, use a different name for its exported index.

A **snapshot** represents a copy of the index files only, and does not capture any other characteristics of the data domain, such as its profile. This means that when you subsequently import the data domain from the snapshot, the Endeca Server creates a new data domain based on the same index, using the data domain's internal `.profile` file, from the previously exported data domain. If you are importing a data domain index from a version that is earlier than Endeca Server 7.6.1.13, the data domain's `.profile` does not exist in the index of such a data domain. Therefore, when you import it, you must specify that data domain's profile name (if you have it). If you don't have the data domain's profile name, then the new data domain will be created based on the imported index based on the default data domain profile.

When you import a data domain, the previously taken snapshot of the index files is copied from the specified index label and a new data domain is created with the specified data domain profile. You can optionally specify whether the new data domain should be enabled. The new data domain will use the index files copied from the snapshot.

To export and import the data domain:

1. Use a command-line window and navigate to the `endeca-cmd` directory.
2. Use the options as shown in the following examples:

- To export, use:

```
endeca-cmd export-dd my_dd --offline-name my_dd_offline
```

This example exports the data domain's index named `my_dd` to the index file named `my_dd_offline`. (You can provide your own names).

The exported index is stored in the offline directory. The name you specify for `--offline-name` must be unique, or you can omit it. If you omit it, it is assigned automatically, by appending the date to the original data domain name, as in this format: `name_MMMMM-dd-yyyy-hh-mm`. The successful response to this request returns the resulting name used for the exported index.



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

- To import, use:

```
endeca-cmd import-dd new_dd --offline-name my_dd_offline --dd-profile-name --is-enabled false
```

This example imports the index `my_dd_offline` into the new data domain `new_dd`, which is created with the `.profile` data domain profile (stored in the index of the previously exported data domain). Observe that because `--is-enabled false`, the data domain is not enabled after it is created.

Importing assumes that you have previously created a data domain whose index is currently exported, and kept track of the `--offline-name` value you used when exporting, or know the name that was automatically assigned.



Note: Starting with Endeca Server version 7.6.1.13, each previously exported data domain contains a `.profile` internal file, as part of its index. This file is used as a data domain profile when you import the index and create a new data domain based on this index. However, if you are importing a data domain from a version before Endeca Server 7.6.1.13, then its index does not have the `.profile` file. (You are required to know the data domain profile name, and also to have this profile created in your Endeca Server deployment.) When you import this index, you can then either specify the data domain profile name (if you have it and if this data domain profile exists in your cluster), or if you don't have the data domain profile for the domain you are trying to import, then Endeca Server imports the index (which includes `.profile`) and creates the data domain based on the `.profile`.

Note that in step 2, if you are not using `endeca-cmd`, you can use the `importDataDomain` and `exportDataDomain` operations in the requests of the Manage Web Service, as in these abbreviated examples.

Here is an example of the export operation that exports the index for data domain `my_dd`:


```
<ns1:exportDataDomain xmlns:ns1="http://www.endeca.com/endeca-server/manage/2/0">
  <ns1:name>my_dd</ns1:name>
  <ns1:nameOfOfflineDataDomain>my_offline_index</ns1:nameOfOfflineDataDomain>
</ns1:exportDataDomain>
```

It produces the following response:

```
<ns3:exportDataDomainResponse
  xmlns:ns2="http://www.endeca.com/endeca-server/types/2/0"
  xmlns:ns3="http://www.endeca.com/endeca-server/manage/2/0">
  <ns3:exportName>my_offline_index</ns3:exportName>
</ns3:exportDataDomainResponse>
```

Where `my_offline_index` is the name of the exported index files.

Here is an example of the import operation that creates a new data domain `new_dd`. This data domain is enabled once it is created:

```
<ns1:importDataDomain xmlns:ns1="http://www.endeca.com/endeca-server/manage/2/0">
  <ns1:name>new_dd</ns1:name>
  <ns1:nameOfOfflineDataDomain>my_offline_index</ns1:nameOfOfflineDataDomain>
  <ns1:enabled>true</ns1:enabled>
</ns1:importDataDomain>
```

If the operation runs successfully, an empty `importDataDomainResponse` is returned.

Enabling a data domain

Enabling a data domain is a means of starting all Dgraph processes serving this data domain.

When you initially create a data domain, or clone an existing data domain, and don't specify that it should be disabled, the data domain is enabled by default — the Endeca Server creates the data domain and enables it. A data domain is enabled when the Endeca Server starts the Dgraph processes serving this domain.

When you disable a data domain, it remains registered with the Endeca Server, but its Dgraph processes are stopped. The resources the Endeca Server reserves for this data domain remain allocated to it — when you subsequently enable the data domain, it continues to use these resources.

To enable a data domain:

1. Use one of these options:

Option	Description
<code>enableDataDomain</code> of the Manage Web Service, or <code>endeca-cmd enable-dd name</code>	Use this option to enable a previously disabled data domain.
<code>createDataDomain</code> of the Manage Web Service, or <code>endeca-cmd create-dd --is-enabled true</code>	When you create a new data domain, you can enable it. If you don't explicitly specify whether the data domain should be enabled, it is enabled by default.
<code>cloneDataDomain</code> of the Manage Web Service, or <code>endeca-cmd clone-dd --is-enabled true</code>	When you clone an existing data domain, you can enable it. If you don't explicitly specify whether the data domain should be enabled, it is enabled by default.

Disabling a data domain

Disabling a data domain is a means of stopping the data domain nodes (the Dgraph processes) serving this data domain.

When you initially create a data domain, or clone an existing data domain, you can optionally set `is-enabled` to `false` in `endeca-cmd create--dd`, thus making the data domain disabled when it is created.

A data domain is disabled when its nodes are not running. When you disable a data domain, it remains registered with the Endeca Server, but its Dgraph processes are stopped. The resources the Endeca Server reserves for this data domain remain allocated to it — when you subsequently enable the data domain, it continues to use these resources.

To disable a data domain:

1. Use one of these options from the Manage Web Service or from the `endeca-cmd`:

Option	Description
<code>disableDataDomain</code> or <code>endeca-cmd disable-dd name</code>	This command of the Manage Web Service disables a data domain.
<code>createDataDomain</code> or <code>endeca-cmd create-dd --is-enabled false</code>	When you create a new data domain, you can specify <code>false</code> for <code>is-enabled</code> . In this case, the created data domain is disabled (its processes are not running).
<code>cloneDataDomain</code> or <code>endeca-cmd clone-dd --is-enabled false</code>	When you clone an existing data domain, if you specify <code>false</code> for <code>is-enabled</code> , the new data domain is disabled after it is created.

Monitoring and backing up data domains

You can list data domains, their host names and ports, monitor their status, and obtain information about the data domain's index, number of records, and other statistics. You can also back up an index file for a running data domain.

[Obtaining information about data domains](#)

[Monitoring the data domain health](#)

[Checking the status of the data domain nodes](#)

[Backing up the index](#)

[Collecting debugging information](#)

Obtaining information about data domains

You can list one or more data domains using `endeca-cmd list-dd`, or the operations from the Manage Web Service.

For each data domain, the Endeca Server returns its name, description, whether it is enabled, and other characteristics.

To list data domains:

1. Use a command-line window (for example, open a Command Prompt in Windows) and navigate to the `endeca-cmd` directory or access the Manage Web Service.
2. Use one of these options, depending on which list you would like to receive:

Option	Description
<code>listDataDomains</code> in the Manage Web Service, or the equivalent <code>endeca-cmd list-dd --verbose</code> command.	Returns information about all data domains registered with the Endeca Server, along with their characteristics.
<code>getDataDomain</code> in the Manage Web Service, or the equivalent <code>endeca-cmd get-dd name</code> command.	Returns information about a single data domain with the specified <code>name</code> .



Note: The Bulk Load port and host are determined dynamically and allocated by the Endeca Server cluster, for each data domain. To obtain information about the Bulk Load host and port, use the `allocateBulkLoadPortname` operation of the Manage Web Service, or `endeca-cmd allocate-bulk-load-port <data-domain> [global-options]`.

Example

Here is an example of the response from the `listDataDomains` operation. In this example, two data domains are listed, `test1` and `test2`:

```
<ns3:listDataDomainsResponse
  xmlns:ns2="http://www.endeca.com/endeca-server/types/2/0"
  xmlns:ns3="http://www.endeca.com/endeca-server/manage/2/0">
  <ns3:dataDomainDetail>
    <ns2:name>test1</ns2:name>
    <ns2:allowOversubscribe>true</ns2:allowOversubscribe>
    <ns2:allowQueriesOnLeader>true</ns2:allowQueriesOnLeader>
    <ns2:numFollowers>0</ns2:numFollowers>
    <ns2:readOnly>false</ns2:readOnly>
    <ns2:enabled>true</ns2:enabled>
    <ns2:idle>false</ns2:idle>
    <ns2:numComputeThreads>4</ns2:numComputeThreads>
    <ns2:computeCacheSizeMB>0</ns2:computeCacheSizeMB>
    <ns2:startupTimeoutSeconds>600</ns2:startupTimeoutSeconds>
    <ns2:shutdownTimeoutSeconds>30</ns2:shutdownTimeoutSeconds>
    <ns2:sessionIdType>header</ns2:sessionIdType>
    <ns2:sessionIdKey>X-Endeca-Session-ID</ns2:sessionIdKey>
    <ns2:autoIdle>false</ns2:autoIdle>
    <ns2:idleTimeoutMinutes>10</ns2:idleTimeoutMinutes>
  </ns3:dataDomainDetail>

  <ns3:dataDomainDetail>
    <ns2:name>test2</ns2:name>
    <ns2:allowOversubscribe>true</ns2:allowOversubscribe>
    <ns2:allowQueriesOnLeader>true</ns2:allowQueriesOnLeader>
    <ns2:numFollowers>0</ns2:numFollowers>
    <ns2:readOnly>false</ns2:readOnly>
    <ns2:enabled>true</ns2:enabled>
    <ns2:idle>false</ns2:idle>
    <ns2:numComputeThreads>2</ns2:numComputeThreads>
    <ns2:computeCacheSizeMB>0</ns2:computeCacheSizeMB>
    <ns2:startupTimeoutSeconds>600</ns2:startupTimeoutSeconds>
    <ns2:shutdownTimeoutSeconds>30</ns2:shutdownTimeoutSeconds>
```

```
<ns2:sessionIdType>header</ns2:sessionIdType>
<ns2:sessionIdKey>X-Endeca-Session-ID</ns2:sessionIdKey>
<ns2:autoIdle>true</ns2:autoIdle>
<ns2:idleTimeoutMinutes>1</ns2:idleTimeoutMinutes>
</ns3:dataDomainDetail>
</ns3:listDataDomainsResponse>
```

Monitoring the data domain health

Use the `endeca-cmd get-dd-health` to check the health of the data domain and list the status of the Dgraph nodes, including details about each Dgraph process that is not running. Alternatively, you can use the `getDataDomainHealth` request of the Cluster Web Service.

Before running this command, obtain the data domain name using `endeca-cmd list-dd` or the Manage Web Service.

To obtain the health of the data domain:

1. Issue a command similar to the following, specifying the data domain name:

```
endeca-cmd get-dd-health MyDataDomain
```

Alternatively, you can issue the `getDataDomainHealth` request with the Cluster Web Service:

```
<ns1:getDataDomainHealth xmlns:ns1="http://www.endeca.com/endeca-server/cluster/2/0">
  <ns1:name>MyDataDomain</ns1:name>
</ns1:getDataDomainHealth>
```

The command returns the data domain status, and lists which Dgraph nodes are not running (if any).

Checking the status of the data domain nodes

A quick way of checking the health of a Dgraph node in the data domain is to get its status from the Endeca Server. Use the `endeca-cmd get-dd-status` command, or the `getDataDomainStats` from the Manage Web Service, to obtain information about the size of the data domain's index, the number of indexed records, and the Dgraph statistics page.

Before you run this command, ensure that the data domain is enabled.

To obtain the data domain statistics information:

1. Issue this command:

```
endeca-cmd get-dd-status <dd_name>
```

where `dd_name` is the name of the data domain. Alternatively, you can issue the request to the Manage Web Service, as in this abbreviated example for `MyDataDomain`:

```
<ns1:getDataDomainStats xmlns:ns1="http://www.endeca.com/endeca-server/manage/2/0">
  <ns1:name>MyDataDomain</ns1:name>
</ns1:getDataDomainStats>
```

The request returns:

- The host name, the port, and the protocol used for the Endeca Server hosting this data domain
- The size of the index for the data domain (in MB)
- The number of records in the index for the data domain

- The Dgraph statistics pages in XML.



Note: The Dgraph statistics pages represent statistics information for each of the Dgraph nodes in the data domain. These pages are intended for Oracle Endeca Support.

Backing up the index

To back up a data domain's index, take a snapshot of the index files by exporting them. You can later restore the index to this particular state.

This procedure discusses how to create a backup copy of the data domain's index, so that in the event of a failure, the data domain can be restored based on this index.

You can safely run an export command on a running data domain that is in the process of loading data or processing end-user queries. This is because the export command takes a snapshot of the data domain's index files based on a pinned version of the index (the index version is pinned automatically by the Endeca Server, for the purpose of taking a snapshot).

To back up the data domain's index:

1. On any node in the Endeca Server, export the data domain by taking a snapshot of its index files. See [Exporting and importing a data domain on page 55](#).
2. Save the exported files to an offline location on another machine.



Note: When you restore the data domain based on the previously-taken snapshot of its index, you will still need to recreate the deployment. This includes recreating the Endeca Server cluster, restoring the `EndecaServer.properties` file on all cluster nodes, and recreating the data domain profile, including the options with which the data domain was started.

3. Once you have recreated the data domain profile in a different location, import the data domain.

Backing up the data domain's index is only a part of backing up the data domain. To back up the data domain, in addition to backing up the index, back up the `EndecaServer.properties` file.

If you are backing up a data domain that is deployed in a cluster, or backing up an entire Endeca Server cluster hosting more than one data domain, it is also helpful to record the topology of the Endeca Server nodes and data domains hosted on them (this information is captured in the WebLogic Server configuration, visible through the WebLogic Console), and the allocation of Cluster Coordinator services on these nodes and their ports (this information is captured in `EndecaServer.properties`).

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 in the Endeca Server cluster, number of Dgraph nodes and their data domains).
- The data from the Dgraph Server Statistics page.
- Dgraph input.
- Description of which Endeca Server nodes (and which hosted Dgraph nodes) 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`.

Other tasks and characteristics of data domains

You can issue cache warming operations to the data domain's Dgraph processes, and configure session affinity for data domains. You can also learn how to connect to the data domains, how to load data, and how different types of data domains behave, such as read-only, idle, or oversubscribed data domains.

[*Memory consumption in the Endeca Server*](#)

[*About using control groups \(cgroups\) for data domains*](#)

[*Read-only data domains*](#)

[*Auto-idling and oversubscribing behavior of data domains*](#)

[*Issuing cache warming requests to the Dgraph*](#)

[*Configuring session affinity for data domains*](#)

[*About connecting Web browsers to data domains*](#)

Memory consumption in the Endeca Server

The Endeca Server query performance is dependent on many characteristics of your specific deployment, such as query workload, query complexity, data domain configuration, and the characteristics of the loaded records, as well as the size of the data domain's index. In view of these characteristics, a hardware sizing must be performed prior to deployment, to assess memory consumption and other hardware needs of your deployment.

One of the characteristics that affect the sizing is estimating the projected memory consumption by the Endeca Server. The following sections in this topic describe different aspects of memory consumption by the Endeca Server:

- [*Memory consumption by the Dgraph process on page 62*](#)
- [*Memory consumption by the Dgraph internal cache on page 63*](#)
- [*Memory consumption by the Data Enrichment plugins on page 63*](#)

Memory consumption by the Dgraph process

Summary: Initial memory consumption by the Dgraph is not indicative or predictive estimate for the number of data domains that can be provisioned.

When the Endeca Server Java application is started, its Dgraph process initially claims a significant amount of RAM on the system for its use. The Dgraph process allocates considerable amounts of virtual memory while ingesting data or executing complex queries. This is observable if you run operating system diagnostic tools. However, the usage of physical memory by the Dgraph also depends on the demands of other processes on the system. When other memory-intensive processes, including other Dgraph processes, are present in the operating system, the Dgraph releases a significant portion of its physical memory quickly. Without such pressure, it may retain the physical memory indefinitely. This is an expected behavior.

To conclude, measurements of physical memory usage on a machine with few data domains are not a predictive estimate of the memory requirements for a larger number of data domains. You should not rely on these estimates for predicting how much memory the Dgraph actually requires to run to support multiple configured data domains.

Memory consumption by the Dgraph internal cache

The Dgraph cache size should be configured to be large enough to allow Endeca Server to operate smoothly under normal query load. You configure the Dgraph cache size through the data domain profile's `--compute-cache-size` parameter of `endeca-cmd`. For information, see [Data domain profile operations on page 32](#).

While the Dgraph typically operates within the limits of its configured Dgraph cache size, it is possible for the cache to become over-subscribed for short periods of time. During such periods, the Dgraph may use up to 1.5 times more cache than it has configured. It is important to note that Endeca Server does not expect to routinely reach an increase in its configured Dgraph cache usage. When the cache size reaches the 1.5 times threshold, the Dgraph starts to more aggressively evict processes that consume its cache, so that the cache memory usage can be reduced to its configured limits.

Memory consumption by the Data Enrichment plugins

Data Enrichment plugins (used via Studio as Enrichments), require adding memory on each machine hosting Endeca Server.

If you are planning to use data enrichment plugins (such as term extraction) in Studio, consider adding additional memory of about 10GB per each instance of Data Enrichment plugin that is expected to run concurrently in the data domain. In other words, if users in the data domain plan to run term extraction, for each such process, additional memory should be provisioned on all Endeca Server machines hosting this data domain.

About using control groups (cgroups) for data domains

On Oracle Enterprise Linux 6 (Red Hat Enterprise Linux Server release 6), Endeca Server has a new option to use cgroups to limit the total memory usage of Dgraph processes hosted on the machine. Control groups (also known as cgroups) is a kernel resource-controlling feature of the Linux 6 operating system. They provide a way to define and allocate the system resources to one or more specific processes, while controlling, at a high-level, the utilization of these resources, and ensuring that the processes do not consume excessive resources. In this way, cgroups help avoid situations where the hosting machines run out of memory for their functioning and are forced to shut down due to their hosted applications taking over all available resources on the machine.

For more information on cgroups, see the section on using cgroups in the *Red Hat Enterprise Linux 6 Resource Management Guide*, available from the *Oracle Linux Documentation*: http://linux.oracle.com/documentation/OL6/Red_Hat_Enterprise_Linux-6-Resource_Management_Guide-en-US.pdf.

In the context of the Endeca Server, cgroups can be utilized on the hosting machines, if the Endeca Server is deployed on Linux 6 and if you enable cgroups in the Endeca Server. Once you enable cgroups after the installation, the Endeca Server relies on this mechanism to allocate resources to all data domains it is hosting. The following statements describe the overall strategy Endeca Server utilizes to allocate its resources (with and without cgroups).

The limit on memory and threads consumption in the Endeca Server is enforced with the following methods:

- For both Windows and Linux, Endeca Server allows data domain creation only if system resources allow it (this approach is applicable for both Windows and Linux and does not involve relying in cgroups). By using this strategy, Endeca Server limits the number of Dgraph processes that can be started on each Endeca Server node based on the internal calculations on the amount of available memory and processing threads.



Note: For detailed information on the logic behind the Endeca Server data domain allocation process, see the *Oracle Endeca Server Cluster Guide*. The data domain allocation strategy assumes that you, as an administrator of the Endeca Server cluster, have made configuration decisions for the types of data domains the Endeca Server is allowed to host, such as, you have created data domain profiles that allow auto-idling of data domains, and you have made calculations for the number of compute threads and the amount of memory to allocate for each data domain profile you plan to use.

- Additionally, if deployed on Oracle Linux 6, or Red Hat Enterprise Linux 6, Endeca Server limits consumption of system's resources by the Dgraph processes for each data domain through the OS-level configuration of cgroups (this assumes that you have enabled cgroups in the Endeca Server by first adjusting `EndecaServer.properties`, and running the `setup_cgroups.sh`). When you run the script, Endeca Server creates and uses a single cgroup for all Dgraph processes hosted on this machine. Once cgroups are used in the Endeca Server, then even if all of its resources are used for hosting data domains, cgroups ensure that the machine hosting the Endeca Server can continue to operate and log messages, and can be accessed by the system administrator for maintenance, checking its logs, and troubleshooting. If cgroup is enabled, Endeca Servers uses the cgroup limit of total virtual memory instead of the total virtual memory of the machine to allocate data domains.



Note: If cgroups are not used or not enabled (either because you are not running on Linux 6, or because you didn't enable them), the strategy does not rely on them.

- Endeca Server further attempts to shut down unused data domains to make way for active ones, based on the configuration for auto-idling in the data domain profiles. Endeca Server only enables a data domain that was previously idle if the calculated amount of available resources on the Endeca Server node allows it.



Important: If you are planning to deploy a large number of self-service applications in the Endeca Server, to increase the allocation guarantees and ensure that Endeca Server nodes continue to operate even when many data domains are provisioned, you are strongly encouraged to deploy Endeca Servers on Oracle Linux 6 or RHEL 6, which both allow Endeca Server to utilize cgroups.

Enabling cgroups for the data domain

If the Endeca Server is deployed on Linux 6, you can enable utilization of cgroups for the Endeca Server application and its processes, by making the required changes in the `EndecaServer.properties` configuration file, on each Endeca Server node hosting a data domain, and then running the specially-provided `setup_cgroups.sh` script located in `$DOMAIN_HOME/EndecaServer/bin` directory of your Endeca Server deployment.

Before you start, ensure that the following requirements are met:

- You have deployed the Endeca Server on the machine(s) running Oracle Linux 6 or Red Hat Enterprise Linux 6. (If you are running an Endeca Server cluster, all Endeca Server nodes must be running on the same hardware and operating system levels).

- You have root access to the machine on which the Endeca Server is deployed, or you have `sudo` access.
- Verify that your Linux 6 installation contains an `/etc/redhat-release` directory with the phrase: Red Hat Enterprise Linux Server release 6.
- Ensure the `libcgroup` package is installed on your system by running, as root:

```
# yum install libcgroup
```

You should receive an output similar to the following abbreviated example:

```
...
Installed Packages
libcgroup...
```

To enable the `cgroups` feature in the Endeca Server:

1. Stop the Endeca Server using the WebLogic Server Admin Console or the script.
2. Navigate to `$DOMAIN_HOME/config`, open the `EndecaServer.properties` file, open it in any text editor and examine the following lines:

```
endeca-cgroups-enabled=false
endeca-cgroups-specified-by-percentage=true
endeca-cgroups-reserved-ram-mb=5120
endeca-cgroups-reserved-swap-mb=5120
endeca-cgroups-reserved-ram-percentage=10.0
endeca-cgroups-reserved-swap-percentage=10.0
```

Notice that once `cgroups` are enabled in this file, you can specify the memory limit as either percentage or actual size in megabytes.

3. Modify the `endeca-cgroups-enabled` setting to specify `true`.
4. Do one of the following:
 - If you leave `endeca-cgroups-specified-by-percentage=true`, then the following parameters will be used by the `cgroup` created for the Endeca Server by the `setup_cgroups.sh` script:

```
endeca-cgroups-reserved-ram-percentage=10.0
endeca-cgroups-reserved-swap-percentage=10.0
```

These settings indicate the percentage of RAM and available swap memory size you want to always remain available on the machine and reserved for other processes, and not used by the `cgroup` serving a particular data domain. You can change these settings, or let the system use the defaults from this file.

- If you specify `false` in `endeca-cgroups-specified-by-percentage=`, then the following parameters will be used by the `cgroup` created for the Endeca Server by the `setup_cgroups.sh` script:

```
endeca-cgroups-reserved-ram-mb=5120
endeca-cgroups-reserved-swap-mb=5120
```

These settings indicate, in megabytes, the exact amount of RAM and available swap memory size you want to always remain available on the machine and reserved for other processes, and not used by the `cgroup` serving the data domain. You can change these settings, or let the system use the defaults from this file.

5. Save the `EndecaServer.properties` file.



Note: If you are running an Endeca Server cluster, modify the `EndecaServer.properties` file in the same way on all machines on which the Endeca Server is deployed.

- Go to `$DOMAIN_HOME/EndecaServer/bin`, locate the `setup_cgroups.sh` and run it as a root user as follows:

```
setup_cgroups.sh <user> <endeca_server_domain>
```

where `<user>` indicates your user name on this machine, and `<endeca_server_domain>` indicates the name of the WebLogic Server domain created for the Endeca Server Java application.

The script checks that you are running Red Hat Enterprise Linux 6, and then runs the `cgconfig` utility that checks for the presence of `libcgroup` package on the machine, and creates a cgroup with the name `<endeca_server_domain>`.

If the script creates the cgroup, it issues a message:

```
Finished successfully. Note: Endeca Server will need to be restarted to see these changes.
```



Note: If you are running an Endeca Server cluster, run the `setup_cgroups.sh` in the same way on all machines on which the Endeca Server is deployed.

- Start the Endeca Server machine(s).

Upon startup, Endeca Server checks the settings in the `EndecaServer.properties` file, and if cgroups are enabled (set to `true`), utilizes the cgroup that was created for the Endeca Server with `setup_cgroups.sh` for all Dgraph processes that will be started on this Endeca Server node.

Endeca Server uses a single group to control all Dgraph processes on the same node. If the total used physical/virtual memory size of Dgraph processes exceeds the limit, the operating system will start terminating processes.



Note: If later you want to disable the use of cgroups in the Endeca Server (for any reason), stop the Endeca Server node(s), modify `EndecaServer.properties` to indicate `endeca-cgroups-enabled=false`, and restart the node(s). In this case, the Endeca Server will not rely on cgroups after startup.

Read-only data domains

When defining a data domain profile, you can specify whether the data domain should be created as read-only. Having a read-only data domain is useful in the development environment or for demonstration purposes — a read-only data domain does not allow changes to its index, but lets the application users issue read-type queries, such as regular search and navigation queries.

To create a read-only data domain, you can export an existing data domain that is not read-only and then import its index using a read-only data domain profile. This way, an imported data domain will have an index with the same data in it, but its Dgraph nodes will be read-only (follower nodes), thus preventing end-users from modifying its configuration or index in any way.

Note that when you initially create a new data domain that is empty of source data, its profile should not be configured as read-only, because its index needs to be populated with data.

A read-only data domain has these characteristics:

- It is created with the data domain profile with the read-only parameter: `endeca-cmd put-dd-profile -read-only true`, or `readOnly` in the `putDataDomainProfile` of the Cluster Web Service.
- It processes only read requests to its index. This means that the Endeca Server responds to end-user queries from the Conversation Web Service, for this data domain.

- It rejects all data loading requests (from the Bulk Load interface and from the Data Ingest Web Service), and all Configuration Web Service requests, as they are updating (read-write) requests. To let you issue "listing-type" operations to a read-only data domain, the Conversation Web Service provides equivalent read-only operations:
 - `AttributeGroupListConfig` retrieves a list of attribute groups in a data domain.
 - `PropertyListConfig` returns all the attributes in a data domain.
 - `AvailableSearchKeysConfig` retrieves a list of the searchable attributes and search interfaces available in the data domain.
- It processes read-only requests from the Entity and Collection Configuration Web Service (`sconfig`), and rejects updating requests from this web service. This means that operations for listing entities (`listEntities`), listing collections (`listCollections`) and listing filter rules (`listFilterRules`) are allowed on a read-only data domain.

For more information on the Conversation Web Service, and Entity and Collection Configuration Web Service requests and how to construct them, see the *Oracle Endeca Server Developer's Guide*.

Auto-idling and oversubscribing behavior of data domains

Data domain profiles define two important aspects of data domain behavior — automatic idling and oversubscribing.

Automatic idling of data domains

A data domain profile may contain two settings:

- `auto-idle`, which determines whether the data domain should be automatically turned to idle if it receives no queries during the timeout period. If set to `false`, the data domain is never turned idle. If set to `true`, the data domain turns idle if the `idleTimeoutMinutes` period expires. The data domain is said to be idle when the Endeca Server stops the Dgraph process for it.
- `idle-timeout`, which determines the optional timeout for automatically idling a data domain. If not specified, the default timeout of 15 minutes is used. If the timeout you specify is less than 15 minutes, Endeca Server issues a warning and resets the timeout to 15 minutes, because 15 minutes is the lowest idle timeout value you can set. This timeout allows for long-running queries to complete successfully before the data domain is turned idle. If you have any queries running longer than 15 minutes, increase the idle timeout.

The data domain that is set to automatically idle will be turned idle if, during this timeout period, it does not receive queries that can activate it.

You can use these data domain profile settings through the `endeca-cmd put-dd-profile --auto-idle true/false --idle-timeout <min>` command. Alternatively, you can use `putDataDomainProfile` operation, and its options, of the Cluster Web Service.

For detailed information on how auto-idling works, see the topic "How idling affects data domain behavior", in the *Endeca Server Cluster Guide*.

Oversubscribing behavior of data domains

A data domain profile may contain a setting, `oversubscribe` that allows Endeca Server to oversubscribe resources while sharing them between this data domain and other data domains hosted by the Endeca Server.

This data domain profile setting is available through the `endeca-cmd put-dd-profile --oversubscribe` command. Alternatively, you can use the `putDataDomainProfile` operation of the Cluster Web Service.

Typically, the system administrators of the Endeca Server cluster make decisions about which data domain profiles should be available for your use when you create data domains. For information on these settings and how they affect the behavior of data domains, see the *Oracle Endeca Server Cluster Guide* (which addresses the Endeca Server cluster administrators).

Issuing cache warming requests to the Dgraph

In a running data domain, you can issue a request to warm up the cache of its Dgraph processes.

To issue the cache-warming request, use the `warmCache` operation of the Manage Web Service or this Endeca Server command:

```
endeca-cmd warm-cache-dd <name> --time-limit-sec <sec>
```

This operation improves performance of the Endeca Server and is useful to run, for instance, after an update for loading data records.

Caching of end-user queries plays an important role in the query performance of the Dgraph process, as it allows the Dgraph to reuse its computation and thus help reduce the user-observable latencies. However, each time there is an update to the index, most of the cache entries become outdated since they depend upon the index version. The Endeca Server provides a cache warming operation to warm the Dgraph cache after updates. The operation obviates the need to create custom warm-up scripts.

You must explicitly issue the cache warming request as it does not run automatically. The only parameter for the operation is the time limit for which the cache warming job is allowed to run. The cache warming operation asks the Dgraph cache to generate a list of representative queries and plays them back to the Endeca Server. It thus warms the Dgraph cache, but does not take more than the specified time. A successful invocation of cache warming operation returns immediately with an empty response and starts the cache warming job in the background. Once the time limit is reached, the cache warming job stops. If during this time you issue any other requests to the Endeca Server, they take priority over cache warming job. If you issue a cache warming operation in an Endeca Server cluster hosting a data domain, the operation runs on all Dgraph nodes serving this data domain.

The cache warming operation takes into account the current Dgraph process usage pattern for selecting the set of representative queries for replay. Note that the existing cache may also contain queries that won't run after the index had changed, for example, because the records schema had changed after an update. The cache warming operation ignores errors from such queries (if they are selected for replay), and proceeds to run other queries in its list. The actual queries replayed by the cache warming operation do not appear in the request log.

To issue a request to warm up the Dgraph cache:

1. Use a command-line window (for example, open a Command Prompt in Windows) and navigate to the `endeca-cmd` directory or access the Manage Web Service.
2. Issue a command to warm the Dgraph cache, similar to the following example:

```
endeca-cmd warm-cache-dd MyDD --time-limit-sec 2400
```

where:

- `MyDD` is the name of the data domain you are creating.
- `--time-limit-sec <sec>` is an optional time limit, in seconds. The default time limit is 1800 seconds (30 minutes). If you do not specify the time limit, the cache warming operation uses the default.



Note: The name of the data domain can include spaces (for example, if it consists of two words). In this case, the name should be enclosed in double quotes. Thereafter, the name should also be enclosed in double quotes when used in other Endeca Server commands.

If the data domain exists, the Endeca Server starts the Dgraph warming job in the background and runs the list of queries for the specified time. Once the time limit is reached, the cache warming job stops. If, during this time you issue any requests to the Endeca Server, they take priority and are processed before any cache warming queries are processed.

Alternatively, you can issue the request similar to the following example, with the Manage Web Service:

```
<ns1:warmCache xmlns:ns1="http://www.endeca.com/endeca-server/manage/2/0">
  <ns1:name>MyDD</ns1:name>
  <ns1:timeLimitSec>2400</ns1:timeLimitSec>
</ns1:warmCache>
```

This example issues an operation that warms the Dgraph cache for the data domain `MyDD` and runs for 40 minutes. A successful operation returns an empty response, and the Endeca Server launches the Dgraph warming script in the background.

Configuring session affinity for data domains

When receiving client requests, the Endeca Server cluster routes them to the arbitrary Endeca Server nodes hosting the appropriate data domains. Optionally, you can configure your data domain profile to utilize session affinity for request routing.

Configuring session affinity in the Endeca Server guarantees that queries with the same session ID are routed to the same Dgraph nodes in the data domain. This improves query processing performance by efficiently utilizing the Dgraph process cache, and improves performance of caching entities (known in Studio as views).

When configuring the data domain profile with `endeca-cmd`, you can specify:

endeca-cmd parameter	Description
<code>endeca-cmd put-dd-profile --session-id-type <i>method</i></code>	<p>The <i>method</i> that will be used 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>. The values are not case-sensitive.</p>
<code>endeca-cmd put-dd-profile --session-id-key <i>name</i></code>	<p>The name of the object that will be checked by the specified method. The default name is <code>X-Endeca-Session-ID</code>.</p>

When you create a new data domain using this profile, the Endeca Server utilizes the values you have specified.

For all three methods (HTTP header, URL parameter, or cookie), the Endeca Server attempts to locate the session ID based on the method and name you specified. If the Endeca Server does not locate a specific session ID in the request, it arbitrarily selects an ID and attempts to route the request to the appropriate Endeca Server node using round-robin distribution.

The Endeca Server expects that the front-end client application that will be issuing requests to the hosted data domains will explicitly set the HTTP headers, the URL parameters, and cookies for its requests.



Note: The Endeca Server returns a cookie with any query-type request to a data domain using the cookie method which does not have the cookie set. Therefore, if you select the cookie method, the cookie returned by the Endeca Server in the previous request should be used for routing the subsequent request.

To configure the data domain for using a particular method of session affinity:

1. Use a command-line window (for example, open a Command Prompt in Windows) and navigate to the `endeca-cmd` directory.
2. When configuring a data domain profile, use the command as in the following example:

```
endeca-cmd put-dd-profile MyProfile --session-id-type header --session-id-key
x-endeca-session-id
```

Alternatively, you can use the following syntax in the Cluster Web Service request (this example is abbreviated as it lists only the parameters specific to session affinity configuration, and omits other data domain parameters):

```
<ns1:putDataDomainProfile xmlns:ns1="http://www.endeca.com/endeca-server/cluster/2/0">
  <ns1:dataDomainProfile>
    ...
    <ns10:sessionIdType xmlns:ns10="http://www.endeca.com/endeca-server/types/2/0">header<
/ns10:sessionIdType>
    <ns11:sessionIdKey xmlns:ns11="http://www.endeca.com/endeca-server/types/2
/0">X-Endeca-Session-ID</ns11:sessionIdKey>
  </ns1:dataDomainProfile>
</ns1:putDataDomainProfile>
```

About connecting Web browsers to data domains

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. If you use Studio with the Endeca Server, this requirement is satisfied by user authentication and security features in Studio.

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.



Chapter 5

Dgraph Administrative Tasks

This section describes the Dgraph flags, and the administrative operations for the Dgraph process. It also describes the Dgraph index merge policy and managing the Dgraph core dump files.

[Dgraph flags](#)

[Administrative operations for the Dgraph](#)

[Managing the index merge policy for the data domain](#)

[Managing Dgraph core dump files](#)

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. For example, you can specify:

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

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 full listing of all 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, using this syntax `endeca-cmd put-dd-profile --args --<flag>`

Dgraph Flag that must be specified only with <code>endeca-cmd put-dd-profile --args --<Dgraph_flag></code>	Description
<code>?</code>	Print the help message and exit.
<code>-v</code>	Verbose mode. Print information about each request to <code>stdout</code> .
<code>--pidfile <pidfile-path></code>	Specify the file to which to write the process ID (PID). The default PID file is named <code>datadomain.pid</code> and is located in the <code>logs</code> directory of the Endeca Server.

Dgraph Flag that must be specified only with <code>endeca-cmd put-dd-profile --args --<Dgraph_flag></code>	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>--syslog</code>	Direct all output to syslog.
<code>--validate_data</code>	Validate that all processed data loads and then exit.

Other flags returned in the list (when you issue `endeca-cmd put-dd-profile --args --usage`), can be specified in two ways:

- Using the `endeca-cmd put-dd-profile --args --<Dgraph_flag>` option (as already discussed), or
- Using the `endeca-cmd put-dd-profile <--Dgraph_flag>`. Notice that this method does not include `--args`. This method is recommended. For a description of those flags that can be specified this way, see [Data domain profile operations on page 32](#), specifically, a description of the `put-dd-profile` command.

Mapping between flags

In addition to a subset of flags that can only be specified with `--args`, there is a list of flags that you can specify either with or without `--args`.



Note: If a flag can be specified with and without `--args`, it is recommended to specify it without the `--args`.

The following table provides a mapping between the syntax of the Dgraph flags that can be specified both ways (with `--args` and without `--args`):

Syntax of the flag when you specify it without the <code>--args</code> , as in: <code>endeca-cmd put-dd-profile --<Dgraph_flag></code>	Syntax of the flag when you specify it with the <code>--args</code> , as in: <code>endeca-cmd put-dd-profile --args --<Dgraph_flag></code>
<code>--ancestor-counts</code>	<code>--ancestor_counts</code>
<code>--backlog-timeout</code>	<code>--backlog_timeout</code>
<code>--refinement-sampling-min</code>	<code>--esampmin</code>
<code>--implicit-exact</code>	<code>--implicit_exact</code>

Syntax of the flag when you specify it without the --args, as in: <code>endeca-cmd put-dd-profile --<Dgraph_flag></code>	Syntax of the flag when you specify it with the --args, as in: <code>endeca-cmd put-dd-profile --args --<Dgraph_flag></code>
<code>--implicit-sample</code>	<code>--implicit_sample</code>
<code>--net-timeout</code>	<code>--net-timeout</code>
<code>--search-char-limit</code>	<code>--search_char_limit</code>
<code>--search-max</code>	<code>--search_max</code>
<code>--snippet-cutoff</code>	<code>--snip_cutoff</code>
<code>--snippet-disable</code>	<code>--snip_disable</code>
<code>--dynamic-category-enable</code>	<code>--stat-all</code>
<code>--contraction-disable</code>	<code>--unctrct</code>
<code>--wildcard-max</code>	<code>--wildcard_max</code>

Administrative operations for the Dgraph

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

Syntax of administrative operations

List of administrative operations

help

flush

logroll

merge

ping

stats

statsreset

Syntax of administrative operations

Administrative operations use the following syntax.

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

Where:

- `<host>` refers to the hostname or IP address of the Oracle Endeca Server. For the `ping` and `help` operations, the `<host>` can be any hostname or IP address that is accessible on your network. For security reasons, all other administrative operations must be run locally from the machine running the Oracle Endeca Server. In this scenario, specify a `<host>` value of `127.0.0.1` (i.e. the localhost).
- `<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 operates.

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 example:

```
http://127.0.0.1:7001/endeca-server/admin/books?op=merge
```



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

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 `ping` and `help` operations can be run from any hostname or IP address that is accessible on your network. For security reasons, all other administrative operations must be run locally from the machine running the Oracle Endeca Server. In this scenario, specify a `<host>` value of `127.0.0.1` (i.e. the localhost).

The Dgraph supports 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>	Returns the Dgraph Server Statistics page, for the specific Dgraph on which this command runs. This page is intended for use by Oracle Endeca Support only.
<code>/admin/datadomain?op=statsreset</code>	Resets the Dgraph Statistics page for one of the Dgraphs in the data domain.

help

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

flush

`/admin/datadomain?op=flush` flushes the Dgraph cache. This operation must be run locally from the machine running the Oracle Endeca Server.

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. This operation must be run locally from the machine running the Oracle Endeca Server.

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. This operation must be run locally from the machine running the Oracle Endeca Server.

ping

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

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 does not activate an idle data domain.

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 Server Statistics page. Note that this page is intended for use by Oracle Support only. This operation must be run locally from the machine running the Oracle Endeca Server.

statsreset

`/admin/datadomain?op=statsreset` resets the Dgraph Server Statistics page. This operation must be run locally from the machine running the Oracle Endeca Server.

The `statsreset` operation returns the following message:

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

Note that the Dgraph Server Statistics page is intended for use by Oracle Support only.

Managing the index merge policy for the data domain

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

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

[Types of merge policies](#)

[Setting the merge policy](#)

[Retrieving the merge policy](#)

[Changing the merge policy](#)

[Forcing a merge](#)

[Linux ulimit command for merges](#)

Using a merge policy for incremental index updates

A merge policy determines how frequently the Dgraph process of the Endeca data domain 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.

You can set the merge policy by sending a request to the Configuration Web Service to change the settings for the Global Configuration Record(GCR), utilizing any Web services tool, such as soapUI. Specifically, the `mdex-config_MergePolicy` attribute in the Global Configuration Record sets the merge policy for the Endeca data domain.

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.

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 the merge policy

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>
```

Retrieving the merge policy

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.

Changing the merge policy

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 `ulimit` command 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.

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 indexes of 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.



Chapter 6

Endeca Server Logging

This section describes four categories of logging in the Endeca Server: the logging of the Endeca Server process in the WebLogic Server domain, the Endeca data domain logging, the Dgraph process logs, and the Cluster Coordinator log.

[Endeca Server logs](#)

[Endeca data domain logs](#)

[Dgraph request log and stdout/stderr log](#)

[Cluster Coordinator log](#)

[Gathering information for diagnosing problems](#)

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.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.
- 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. This log can be considered an application log as is it contains messages generated from the Endeca data domains.
- The `DataEnrichment.log` contains messages from the Endeca Data Enrichment subsystem. This log does not use ODL for its messages.
- The WebLogic domain log for Endeca Server is named `<domain-name>.log` (for example, `endeca_server_domain.log`) and 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.

By default, these log files are located in the `$DOMAIN_HOME/servers/AdminServer/logs` directory. For example, if `endeca_server_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](#)

[Logging properties files](#)

[Customizing the HTTP access log](#)

Log entry format

This topic describes the format of Endeca Server log entries.

The following is an example of a Notification message that is generated by disabling an Endeca data domain:

```
[2016-01-11T10:38:33.470-05:00] [EndecaServer] [NOTIFICATION] [OES-000045]
[com.endeca.opmodel.control.DGraph] [host: fcal-pc] [nwaddr: 10.137.179.59]
[tid: [ACTIVE].ExecuteThread: '6' for queue: 'weblogic.kernel.Default (self-tuning)'] [userId: fcal]
[ecid: 0000L81LfHp9_aZ_Tl4EyclM^vs6000003,0] OES-000045: Stopped data store "bikes"
```

The format of the Endeca Server log fields (using the above example) and their descriptions are as follows:

Log entry field	Description	Example
Timestamp	The date and time when the message was generated. This reflects the local time zone.	[2016-01-11T10:38:33.470-05:00]
Component ID	The ID of the component that originated the message. "EndecaServer" is hard-coded for the Endeca Server messages.	[EndecaServer]
Message Type	The type of message (log level): <ul style="list-style-type: none"> INCIDENT_ERROR ERROR WARNING NOTIFICATION TRACE UNKNOWN 	[NOTIFICATION]
Message ID	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.	[OES-000045]
Module ID	The Java class that prints the message entry.	[com.endeca.opmodel.control.DGraph]
Host name	The name of the host where the message originated.	[host: fcal-pc]
Host address	The network address of the host where the message originated	[nwaddr: 10.137.179.59]

Log entry field	Description	Example
Thread ID	The ID of the thread that generated the message.	[tid: [ACTIVE].ExecuteThread: '6' for queue: 'weblogic.kernel.Default (self-tuning)']
User ID	The name of the user whose execution context generated the message.	[userId: fcal]
ECID	The Execution Context ID (ECID), which is a global unique identifier of the execution of a particular request in which the originating component participates.	[ecid: 0000L8lLfHp9_aZ_Tl4Eyc1M^vs6000003,0]
Message Text	The text of the log message.	OES-000045: Stopped data store "bikes"

Logging properties files

Endeca Server and Data Enrichment have default Log4j configuration files that set their logging properties.

Both files are located in the \$DOMAIN_HOME/config directory.

Endeca Server logging properties

The Endeca Server properties file is named `EndecaServerLog4j.properties`. The default version of the file is as follows:

```
log4j.rootLogger=WARN, stdout, ODL

# Console Appender
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d [%p] [%c] %L - %m%n

# ODL-format Log Appender
log4j.appender.ODL=com.endeca.logging.ODLAppender
log4j.appender.ODL.MaxSize=104857600
log4j.appender.ODL.MaxSegmentSize=10485760
log4j.appender.ODL.encoding=UTF-8

# Log level per packages
log4j.logger.com.endeca=INFO
log4j.logger.org.apache.zookeeper=WARN
```

The file defines two appenders (stdout and ODL) for the root logger and also sets log levels for two packages.

The file has the following properties:

Logging property	Description
<code>log4j.rootLogger=WARN, stdout, ODL</code>	The level of the root logger is defined as WARN and attaches the Console Appender (stdout) and ODL-format Log Appender (ODL) to it.
<code>log4j.appender.stdout=org.apache.log4j.ConsoleAppender</code>	Defines stdout as a Log4j ConsoleAppender
<code>log4j.appender.stdout.layout.ConversionPattern</code>	Sets the <code>PatternLayout</code> class for the stdout layout.
<code>log4j.appender.stdout.layout.ConversionPattern</code>	<p>Defines the log entry conversion pattern as:</p> <ul style="list-style-type: none"> • %d is the date of the logging event. • %p outputs the priority of the logging event. • %c outputs the category of the logging event. • %L outputs the line number from where the logging request was issued. • %m outputs the application-supplied message associated with the logging event while %n is the platform-dependent line separator character. <p>For other conversion characters, see: https://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/PatternLayout.html</p>
<code>log4j.appender.ODL=com.endeca.util.ODLAppender</code>	Defines ODL as an ODL Appender. ODL (Oracle Diagnostics Logging) is the logging format for Oracle applications.
<code>log4j.appender.ODL.MaxSize</code>	Sets the maximum amount of disk space to be used by the <code>AdminServer-diagnostics.log</code> file and the logging rollover files. The default is 1048576000 (about 1GB). Older log files are deleted to keep the total log size under the given limit.
<code>log4j.appender.ODL.MaxSegmentSize</code>	Sets the maximum size (in bytes) of the log file. When the log file reaches this size, a rollover file is created. The default is 104857600 (about 10 MB).

Logging property	Description
<code>log4j.appender.ODL.encoding</code>	Sets character encoding the log file. The default UTF-8 value prints out UTF-8 characters in the file.
<code>log4j.logger.com.endeca</code>	Sets the default log level for the Endeca Server log messages. INFO is the default log level.
<code>log4j.logger.org.apache.zookeeper</code>	Sets the default log level for the Cluster Coordinator (ZooKeeper) client logger. WARN is the default log level.

Data Enrichment logging properties

The Data Enrichment properties file is named `DataEnrichmentLog4j.properties`. It does not use ODL, but is otherwise very similar to the Endeca Server Log4j configuration file.

The file sets INFO as the default log level for the `log4j.logger.oracle.endeca.enrichment` package.

Logging levels

The WebLogic logger for Endeca Server is configured with the type of information written to log files, by specifying the log 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 WARN, WebLogic also returns messages of type FATAL and ERROR.

The log levels (in decreasing order of severity) are:

Java Log Level	ODL Log Level	Meaning
OFF	N/A	Has the highest possible rank and is used to turn off logging.
FATAL	INCIDENT_ERROR	Indicates a serious problem that may be caused by a bug in the product and that should be reported to Oracle Support. In general, these messages describe events that are of considerable importance and which will prevent normal program execution.
ERROR	ERROR	Indicates a serious problem that requires immediate attention from the administrator and is not caused by a bug in the product.
WARN	WARNING	Indicates a potential problem that should be reviewed by the administrator.

Java Log Level	ODL Log Level	Meaning
INFO	NOTIFICATION	A message level for informational messages. This level typically indicates a major lifecycle event such as the activation or deactivation of a primary sub-component or feature. This is the default level.
DEBUG	TRACE	Debug information for events that are meaningful to administrators, such as public API entry or exit points.

These levels allow you to monitor events of interest at the appropriate granularity without being overwhelmed by messages that are not relevant. When you are initially setting up your application in a development environment, you might want to use the `INFO` level to get most of the messages, and change to a less verbose level in production.

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:  2016-01-22      15:23:40
2016-01-22    15:27:07      0.967      POST      /endeca-server/ws/manage      200
2016-01-22    15:27:32      7.301      POST      /endeca-server/es/control      200
2016-01-22    15:27:32      7.567      POST      /endeca-server/ws/manage      200
2016-01-22    16:23:35      0.219      GET       /endeca-server/ws/conversation/sh?wsdl      200
2016-01-22    16:23:35      0.0       GET       /favicon.ico      404
2016-01-22    16:24:14      0.031      GET       /endeca-server/ws/conversation/sh?wsdl      200
2016-01-22    16:24:14      0.031      GET       /endeca-server/ws/conversation/sh?XSD
=lql_parser_types.xsd      200
2016-01-22    16:27:15      5.692      POST      /endeca-server/ws/conversation/sh      200
2016-01-22    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.

Endeca data domain logs

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.

Dgraph request log and stdout/sterr log

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 93](#).

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:
endecareresultslistportlet_WAR_endecareresultslistportlet_INSTANCE_5RKp_LAYOUT_11601
```

As an example, if you set:

```
PORTLET=endecareresultslistportlet_WAR_endecareresultslistportlet_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*.

[Identifying Dgraph connection errors](#)

[List and syntax of configuration operations](#)

[Logging variables for the Dgraph process](#)

Identifying Dgraph 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.

List and syntax of configuration operations

Use configuration (or config) operations to modify logging information for the Dgraph within the Endeca Server.

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.

The configuration operations utilize the following syntax:

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

Where:

- `<host>` refers to the hostname or IP address of the Oracle Endeca Server. For the `help` operation, the `<host>` can be any hostname or IP address that is accessible on your network. For security reasons, all other configuration operations must be run locally from the machine running the Oracle Endeca Server. In this scenario, specify a `<host>` value of `127.0.0.1` (i.e. the localhost).
- `<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 example:

```
http://127.0.0.1:7001/endeca-server/config/books?op=log-status
```



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

Logging variables for the Dgraph process

You can use logging variables with configuration 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

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=textsearchrelevanceverbose&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 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.
<code>dgraphrefinementgroupverbose</code>	Enables refinement verbose/debugging messages.

[log-enable](#)

[log-disable](#)

[log-status](#)

[help](#)

log-enable

The `log-enable` operation turns on verbose logging. This operation must be run locally from the machine running the Oracle Endeca Server.

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. This operation must be run locally from the machine running the Oracle Endeca Server.

`/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). This operation must be run locally from the machine running the Oracle Endeca Server.

For example, if you enable 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.

Cluster Coordinator log

The Cluster Coordinator maintains a log of its operations.

This log facilitates diagnosing Endeca Server problems related to the Cluster Coordinator.

The Cluster Coordinator log is named `coordinator.log` and by default is 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\coordinator.log
```

You can specify another location for the log by changing the path in the `endeca-cluster-coordinator-dataLogDir` parameter in the `EndecaServer.properties` configuration file.

Changing the log level

The Cluster Coordinator log is enabled by default with a log level of `WARN`.

You can change the log level by adding the `endeca-cluster-coordinator-logLevel` parameter to the `EndecaServer.properties` configuration file. Note that this property is not present in the file by default.

The following example sets the log level to `INFO`:

```
endeca-cluster-coordinator-logLevel=INFO
```

The log levels that can be set (in order of higher to lower severity) are:

- `FATAL` level designates very severe error events that will presumably lead the application to abort.
- `ERROR` level designates error events that might still allow the application to continue running.
- `WARN` level designates potentially harmful situations.
- `INFO` level designates informational messages that highlight the progress of the application at coarse-grained level.
- `DEBUG` Level designates fine-grained informational events that are most useful to debug an application.
- `TRACE` Level designates finer-grained informational events than the `DEBUG` level.

Unless you are diagnosing problems, it is recommended that you use a level of `WARN` (or higher) to prevent the log file from growing too large.

Configuration for auto-purge feature

Besides its main log described above, the Cluster Coordinator maintains files which are a persistent copy of the znodes stored by a particular serving ensemble. These are the snapshot and transactional log files, and are stored in the Cluster Coordinator data directory. As changes are made to the znodes, these changes are appended to a transaction log. Occasionally, when a log grows large, a snapshot of the current state of all znodes is written to disk. This snapshot supercedes all previous logs.

The Cluster Coordinator has an automatic purge feature for the transaction logs and snapshots. This auto-purge feature is enabled by default.

To prevent these logs from taking up too much disk space, the `EndecaServer.properties` file has two properties that are used to configure the auto-purge feature:

- - `endeca-cluster-coordinator-auto-purge-snap-retain-count` sets the number of snapshots to retain in the Cluster Coordinator data directory. The most recent snapshots and the corresponding transaction logs are retained and the rest are deleted. It is set to 3 as the default.
 - `endeca-cluster-coordinator-auto-purge-interval` sets the purge task interval in hours, and is set to 1 as the default. Set to 0 (zero) to disable the auto-purge feature.

For example:

```
endeca-cluster-coordinator-auto-purge-snap-retain-count=3
endeca-cluster-coordinator-auto-purge-interval=1
```

For more information on this file, see [The `EndecaServer.properties` file on page 11](#).

Gathering information for diagnosing problems

This section describes the information that you need to gather in the event of a problem with Endeca Server or the Dgraph.

When you report the problem to Oracle Support, you may be asked to supply this information to help Oracle engineers diagnose and fix the problem.

Endeca Server information

There are four areas of information to gather for Endeca Server problems.

1: WebLogic standard Logs

Get the full contents of the `$DOMAIN_HOME/servers/AdminServer/logs` directory. For the name of the logs, see [Endeca Server logs on page 82](#). These logs can be retrieved with the Endeca Server `get-logs` command (see [get-logs on page 44](#)).

2: Config files

These files in the `$DOMAIN_HOME/config` directory:

- `config.xml`
- `EndecaServer.properties`

3: Thread dumps

The first step in to obtain a thread dump is to get the JVM process PID for WebLogic Server and Cluster Coordinator (ZooKeeper). The **jps** tool (which is available on both Linux and Windows) can provide the PIDs you need.

The **jps -mlv** command lists all running JVMs. On Linux, you can use one of these formats to obtain the WebLogic or Cluster Coordinator PIDs:

- `jps -mlv | fgrep weblogic`
- `jps -mlv | fgrep zookeeper`

The following example shows the **jps** output for the WebLogic process:

```
jps -mlv | fgrep weblogic
18329 weblogic.Server -Xms1024m -Xmx4096m -XX:MaxPermSize=1024m
-Dweblogic.Name=AdminServer
...
```

In the example, 18329 is the WebLogic JVM PID.

After you have obtained the PIDs, use the **jstack** tool to generate thread dumps and save them in a file, using this syntax:

```
jstack -l <pid> <filename>
```

For example:

```
jstack -l 18329 jstack.weblogic.outIt
```

If the JVM is not responsive, add the **-F** flag:

```
jstack -F -l <pid> <filename>
```

It is very helpful to have a couple of thread dumps a few minutes apart, with filenames indicating the order.

Note that both the **jps** and **jstack** tools are in the `JAVA_HOME/bin` directory.

4: Heap dumps

Use the **jmap** tool to generate heap dumps. As with **jstack**, you must first get the PID with the **jps** command.

You then run **jmap** with this syntax:

```
jmap -dump:format-b,file=<filename>.hprof <pid>
```

Again, if the JVM is not responsive, add the `-F` flag.

Note that the **jmap** tool is in the `JAVA_HOME/bin` directory.

Dgraph

There are different sets of logs that are needed, depending upon whether it is a performance issue or a crash. You may also need to send Cluster Coordinator (ZooKeeper) logs.

1: Logs for performance issues

Collect the following information:

- `<data-domain-name>.reqlog` from the `$DOMAIN_HOME/EndecaServer/logs` directory
- Dgraph Statistics page
- WebLogic `access.log`
- output from: `endeca-cmd version`
- output from: `endeca-cmd get-dd <data-domain-name>`

You can get the Dgraph Statistics page results by running the URL stats command in your browser, as in this example that returns statistics for the "bikes" data domain:

```
http://localhost:7001/endeca-server/admin/bikes?op=stats
```

You can also pipe the results of the Endeca Command `get-dd-status` into a file, as in this example using the same bikes data domain:

```
endeca-cmd get-dd-status bikes > bikes_stats.txt
```

2: Logs for Dgraph crashes

In order to diagnose a Dgraph crash, collect the following information:

- `<data-domain-name>.out` log from the `$DOMAIN_HOME/EndecaServer/logs` directory
- Dgraph core dump file

For information on managing Dgraph core dump files, see [Managing Dgraph core dump files on page 80](#).

3: Logs for other correctness issues

For investigating correctness issues that do not involve a Dgraph crash (unexpected SOAP fault, query returning incorrect results, etc.), collect the following data:

- Dgraph indexes for the Endeca data domains
- Dgraph `<data-domain-name>.reqlog`
- Dgraph `<data-domain-name>.out`
- output from: `endeca-cmd version`

The .reqlog and .out files are in the \$DOMAIN_HOME/EndecaServer/logs directory. The Dgraph indexes are in the \$DOMAIN_HOME/EndecaServer/data directory.

4: Cluster Coordinator log

The Cluster Coordinator (ZooKeeper) log is described in [Cluster Coordinator log on page 94](#). This log is retrieved as part of the Endeca Server `get-logs` command operation.

5: Retrieving schemas for Endeca data domains

If you are asked to provide the schema for an Endeca data domain, you can use the `export` operation of the Configuration Web Service. This operation returns the attribute information for a data domain, as well as its groups, configuration documents (including search interfaces), and the Global Configuration Record.

You can use the `export` operation with an Integrator ETL graph or a tool such as soapUI. The following is a soapUI sample request:

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns="http://www.endeca.com/MDEX/config/services/types/3/0"
  xmlns:ns1="http://www.endeca.com/MDEX/config/XQuery/2009/09">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:configTransaction>
      <ns:export/>
    </ns:configTransaction>
  </soapenv:Body>
</soapenv:Envelope>
```

You can use soapUI's Dump File request property to write the results to a disk file.

6: Changing Dgraph flags

You may be asked to add flags to the Dgraph to generate more complete log entries. For example, a command flag is the `-v` flag, which produces more verbose entries (note that the flag has only one dash instead of the usual two).

You cannot change a data domain profile after creation, nor can you change the configuration of an existing data domain. For these reasons, the only way to change the Dgraph flags for a data domain is to create a new data domain profile (with the appropriate flags) and then update the specific data domain with the new profile. You will be using Endeca Server commands (`endeca-cmd`) for these changes.

To change a data domain's Dgraph flags:

1. Disable the data domain:

```
endeca-cmd disable-dd bikes
```

2. Create a new data domain profile, adding the Dgraph flags with the `--args` option (note that `--args` must be the last option on the command line):

```
endeca-cmd put-dd-profile MyProfile --args -v
```

3. Apply the new profile to the data domain:

```
endeca-cmd update-dd bikes --dd-profile-name MyProfile
```

4. Enable the data domain:

```
endeca-cmd enable-dd bikes
```

After re-enabling the data domain, it will start the Dgraph using the flags specified in the `--args` option in Step 2.

Later, you can use the `update-dd` command to re-apply the previous profile (typically the default profile) to the data domain.

Topology information

In addition to the log files and system information listed above, you should also provide information about the topology of your EID deployment. Such information includes the following:

- Hardware specifications and configuration of the machines.
- Description of the Endeca Server topology (number and names of servers in the Endeca Server cluster, number of Dgraph nodes and their data domains).
- Description of which Endeca Server nodes (and which hosted Dgraph nodes) are affected.
- Network topology.

Index

A

- adding
 - data domains 49
 - nodes, to data domain 52
- administrative tasks
 - admin and config operations 74
 - overview 9
- admin operations
 - flush 75
 - help 75
 - list of 74
 - logroll 75
 - merge 75
 - ping 75
 - statsreset 76
- aggressive merge policy 77
- allocate-bulk-load-port command, Endeca Server 43
- Automatic idling of data domains 67
- auto-purge feature, Cluster Coordinator 95

B

- backup, of data domain index 61
- balanced merge policy 77

C

- cache warming for data domains 68
- cgroups
 - about 64
 - enabling 64
- changing the merge policy of a data domain 78
- clone-dd command, Endeca Server 39
- cloning data domains 51
- Cluster Coordinator
 - auto-purge feature 95
 - main log 94
- config operations
 - about 74, 92
 - for logging verbosity 92
 - help 94
 - list of 91
 - log-disable 94
 - log-enable 93
 - logging variables 93
 - log-status 94
- configuring session affinity 69
- connecting a Web browser to the Oracle Endeca Server 70
- connection errors, Dgraph and client 91
- core dump files

- in the Dgraph 81
- managing 80

create-dd command, Endeca Server 38

D

- data directory, changing for data domain 17
- data domain
 - adding 49
 - adding Dgraph nodes 52
 - backing up the index 61
 - cache warming 68
 - cgroups, enabling 64
 - checking health 60
 - configuring session affinity 69
 - copying 51
 - deleting 51
 - disabling 58
 - enabling 57
 - idling 67
 - importing and exporting 55
 - number of records 60
 - obtaining information about 58
 - oversubscribing 67
 - read-only 66
 - size of index 60
 - status information 60
 - updating 53
 - using a data directory on a different drive 17
- delete-dd command, Endeca Server 40
- delete-dd-profile command, Endeca Server 37
- delete-node command, Endeca Server 31
- Dgraph 92
 - about 47, 48
 - admin operations 74
 - cache warming 68
 - checking aliveness of 60
 - config operations 91
 - crash dump files on Linux 81
 - crash dump files on Windows 80
 - flags 71
 - identifying connection errors 91
 - logging variables for 92
 - logs 89
 - what to collect for debugging 61
- disable-dd command, Endeca Server 40
- disabling data domain 58

E

- enable-dd command, Endeca Server 40
- enabling data domain 57
- EndecaCmd.properties file 26

- Endeca data domain
 - adding nodes 42
 - cloning 39
 - creating 38
 - creating profiles 33
 - deleting 40
 - disabling 40
 - displaying version 44
 - enabling 40
 - exporting 40
 - get health information about 42
 - get information about 42
 - get statistics about 42
 - importing 41
 - listing 43
 - logs 88
 - obtaining logs 44
 - updating 43
 - warming Dgraph cache 43

- Endeca Server
 - Cluster Coordinator log 94
 - command interface, global options 26
 - configuration file 11
 - data domain logs 88
 - host option 27
 - logging configuration 84
 - logs 82
 - memory consumption 62
 - overview 10
 - port option 28
 - root option 28
 - starting 18
 - stopping 20
 - transient state 23

- Endeca Server commands
 - allocate-bulk-load-port 43
 - clone-dd 39
 - create-dd 38
 - delete-dd 40
 - delete-dd-profile 37
 - delete-node 31
 - disable-dd 40
 - enable-dd 40
 - export-dd 40
 - get-cluster-health 31
 - get-dd 42
 - get-dd-health 42
 - get-dd-profile 37
 - get-dd-status 42
 - get-logs 44
 - get-node-profile 31
 - global options 26
 - import-dd 41
 - list-dd-profiles 37
 - list-dd-status 43
 - list-nodes 32
 - options for SSL 28
 - put-dd-profile 33
 - rescale-dd 42
 - set-node-profile 30
 - update-dd 43

- version 44
 - warm-cache-dd 43

- EndecaServer.properties file 11
- export-dd command, Endeca Server 40
- exporting data domain 55

F

- flags, Dgraph 71
- flush admin operation 75
- forcing a merge 79

G

- gathering information for diagnosing problems 96
- get-cluster-health command, Endeca Server 31
- get-dd command, Endeca Server 42
- get-dd-health command, Endeca Server 42
- get-dd-profile command, Endeca Server 37
- get-dd-status command, Endeca Server 42
- get-logs command, Endeca Server 44
- get-node-profile command, Endeca Server 31
- Global Configuration Record
 - retrieving with API 78
 - setting merge policy 77

H

- help admin operation 75
- help config operation 94
- HTTP access log 87

I

- idling of data domains 67
- import-dd command, Endeca Server 41
- importing data domain 55
- incremental updates, merge policy for 76
- index backing up, for a data domain 55
- IPv4 and IPv6 address support in Endeca Server 70

L

- list-dd command, Endeca Server 43
- list-dd-profiles command, Endeca Server 37
- listing data domains 58
- list-nodes command, Endeca Server 32
- load balancer configuration for session affinity 69
- log-disable config operation 94
- log-enable config operation 93
- logging variables 92
 - operation syntax 92, 93
 - supported variables for 93
- logroll admin operation 75

- logs
 - Dgraph 89
 - Endeca Server 82
 - WebLogic HTTP access log 87
- log-status config operation 94

M

- merge admin operation 75
- merge policy
 - changing 77
 - changing in a running data domain 78
 - forcing a merge 79
 - for incremental updates 76
 - getting programmatically 78
 - setting programmatically 77
 - types of 77
- monitoring data domain status 60

N

- network interface, changing 24
- nodes, Endeca Server
 - deleting 31
 - getting cluster health 31
 - getting node profile 31
 - listing nodes 32
 - setting node profile 30

O

- operation syntax for Dgraph logging variables 92
- Oracle Endeca Server
 - connecting Web browsers to 70
 - IPv4 and IPv6 address support 70
- oversubscribing data domain behavior 67

P

- ping admin operation 75
- profile
 - for the data domain 32, 47
- profile, data domain
 - deleting 37

- displaying profile settings 37
 - listing profiles 37

- .profile file 32, 47

- put-dd-profile command, Endeca Server 33

R

- read-only data domain 66
- rescale-dd command, Endeca Server 42

S

- session affinity configuring in data domain profile 69
- set-node-profile command, Endeca Server 30
- size of index, data domain 60
- snapshot, Dgraph 55
- SSL options, endeca-cmd 28
- statsreset admin operation 76

T

- transient state, Endeca Server 23

U

- update-dd command, Endeca Server 43
- updating data domains 53
- URL operations, about 74

V

- variables for Dgraph logging 93
- version command, Endeca Server 44

W

- warm-cache-dd command, Endeca Server 43
- WebLogic logs
 - AdminServer 82
 - HTTP access log 87
 - log entry format 83
- Who should use this guide 6