Oracle® Endeca Server

Administrator's Guide

Version 7.6.1 • December 2013

Copyright and disclaimer

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

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.
Variable	This formatting is used for variable values. For variables within a code sample, the formatting is *`Variable`*.
`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 `endeca_server_domain` is the name of your WebLogic domain, then the `$DOMAIN_HOME` value would be the `$MW_HOME/user_projects/domains/endeca_server_domain` directory.
`$ENDECA_HOME`	Indicates the absolute path to your Oracle Endeca Server home directory, which is the root directory for your Endeca Server installation.

Contacting Oracle Customer Support

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

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

Chapter 1

Introduction

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

Taking ownership of your implementation

Overview of administrator tasks

Taking ownership of your implementation

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

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

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

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

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

Overview of administrator tasks

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

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

Section	Tasks
Oracle Endeca Server Server Commands	Operations for managing the Endeca Server application, and `endeca-cmd` Endeca Server command reference: • Start the Endeca Server application running in the WebLogic Server. • Stop the Endeca Server application running in the WebLogic Server. • Use `endeca-cmd` to: • 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: • 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	• 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: • 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: • Merge data domain's index update generations. • Set and manage the merge policy for an Endeca data domain.

Section	Tasks
Endeca Server Logging	• 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 Endeca Server is a Java application hosted in the WebLogic Server container that manages the Endeca data domains.

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

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

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

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

Endeca Server has a set of commands with which you create and control the data domains. For information, see *Endeca Server Command Reference on page 25*.

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 12*
- *Cluster Coordinator parameters on page 15*
- *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
`endeca-require-https`	If set to `true`, specifies that Endeca Server enforces the use of the HTTPS port. If set to `false`, uses the HTTP port.
`endeca-runtime-basedir`	The location of the Endeca Server's `endeca-server` directory. The default is the `$ENDECA_HOME/endeca-server` location.
`endeca-data-dir`	The location where the data files for all the Endeca data domains are stored for this Endeca Server instance. The default location is the `$DOMAIN_HOME/EndecaServer/data` directory. Note that you can specify only one directory (that is, specifying multiple directories is not allowed).

Endeca Server parameter	Description
endeca-offline-dir	The location where the data files for an Endeca data domain are exported by the export-dd of endeca-cmd. The default is the $DOMAIN_HOME/EndecaServer/offline directory. Note that you can specify only one directory (that is, specifying multiple directories is not allowed).
endeca-logs-dir	The location of the Dgraph process standard out/err log and request log, as well as the PID file. The default is the $DOMAIN_HOME/EndecaServer/logs directory. You can specify only one directory (that is, specifying multiple directories is not allowed).
endeca-dgraph-install	The location of the Endeca Server Dgraph application directory. The default is the $ENDECA_HOME/endeca-server/dgraph directory.
endeca-webserver-port	The port on which the Endeca Server listens for requests. The default is 7001 (for non-SSL deployments), or 7002 (for SSL deployments).
endeca-ds-port-min and endeca-ds-port-max	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. The defaults for endeca-ds-port-min are 7011 (for non-SSL deployments) or 7012 (for SSL deployments). The defaults for endeca-ds-port-max are 8011 (for non-SSL deployments) or 8012 (for SSL deployments).
endeca-ds-pin-timeout-min	The default minimum value is 60000 milliseconds (ms). This is the lowest timeout value the Dgraph process can use when PinDataVersion request is issued. This parameter, as well as endeca-ds-pin-timeout-max and endeca-ds-pin-timeout-default are used in the Endeca Server to enable index version pinning requests. For information about index data version pinning, see the *Oracle Endeca Server Developer's Guide*.
endeca-ds-pin-timeout-max	The default maximum value is 300000 ms. This is the highest timeout value the Dgraph process can use when PinDataVersion request is issued.
endeca-ds-pin-timeout-default	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 PinDataVersion request. (You typically use this request of the Conversation Web Service to pin a data version).

Endeca Server parameter	Description
`endeca-ds-cert-file`	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 `generate_ssl_keys` utility was run, the default path is to the `$DOMAIN_HOME/config/ssl/dgraphCert.pem` file. If the utility was not run, this setting is empty.
`endeca-ds-ca-file`	The path of the certificate authority file. This file is used by all clients and servers to authenticate the other endpoint of a communication channel. If the `generate_ssl_keys` utility was run, the default path is to the `$DOMAIN_HOME/config/ssl/dgraphCA.pem` file. If the utility was not run, this setting is empty.
`endeca-secure-mode`	If set to `true`, specifies that Endeca Server is using SSL and that the Dgraphs should be started in SSL mode. If set to `false`, Endeca Server and the Dgraphs are not using SSL.
`endeca-memory-to-index-size-ratio`	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. **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 *Oracle Endeca Server Cluster Guide*.
`endeca-threads-allowed-per-core`	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. **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 *Oracle Endeca Server Cluster Guide*.

Endeca Server parameter	Description
endeca-cgroups-enabled, endeca-cgroups-specified-by-percentage, endeca-cgroups-reserved-ram-mb, endeca-cgroups-reserved-swap-mb, endeca-cgroups-reserved-ram-percentage, endeca-cgroups-reserved-swap-percentage	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 61*.

Cluster Coordinator parameters

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

Cluster Coordinator parameter	Description
endeca-cluster-coordinator-dir	The location of the Endeca Cluster Coordinator root directory. The default is the `$ENDECA_HOME/endeca-server/cluster-coordinator` directory.
endeca-cluster-coordinator-hosts	The default is `localhost` (which should be used for a single-node installation). For an Endeca Server cluster deployment, this parameter must contain comma-separated `localhost` names of all Managed Servers that are part of the Cluster Coordinator ensemble. Always use the `localhost` names, also known as the top-level domain names, for this field, such as `host1`, `host2`, `host3`. Do not use the Fully Qualified Domain Names.
endeca-cluster-coordinator-clientPort	The port of the server on which the Cluster Coordinator is running. This the port at which the clients will connect. The default is 2181.
endeca-cluster-coordinator-tickTime	The length of a single tick, which is the basic time unit for the Cluster Coordinator. A tick is measured in milliseconds. It is used to regulate heartbeats and timeouts. The default is 2000 milliseconds.
endeca-cluster-coordinator-initLimit	The number of ticks that the initial synchronization phase can take. This number specifies the length of time the nodes have to connect to the leader node. The default is 10 ticks.

Cluster Coordinator parameter	Description
`endeca-cluster-coordinator-syncLimit`	The number of ticks that can take place between one node sending a request for an update and receiving an acknowledgment from the leader node. The default is 5 ticks.
`endeca-cluster-coordinator-dataDir`	The location of the directory where the in-memory database snapshots for the Cluster Coordinator and the transaction log of updates to its database are stored. The default is the `$DOMAIN_HOME/EndecaServer/data` directory.
`endeca-cluster-coordinator-serverPort`	The port that follower nodes use to connect to the leader node. The default server port is 3181.
`endeca-cluster-coordinator-leaderPort`	The port that the Cluster Coordinator servers use to do leader election. The default leader port is 4181.
`endeca-cluster-coordinator-maxClientCnxns`	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.
`endeca-cluster-coordinator-minSessionTimeout`	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.
`endeca-cluster-coordinator-maxSessionTimeout`	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
endeca-data-enrichment-pluginsDir, endeca-data-enrichment-resourceDir, endeca-data-enrichment-collectionBatchSize, endeca-data-enrichment-collectionMatchRetries, endeca-data-enrichment-taskHistorySize	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. 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: • The *-pluginsDir and *-resourcesDir 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. • *-collectionBatchSize 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. • *-collectionMaxRetries 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. • *-taskHistorySize 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:

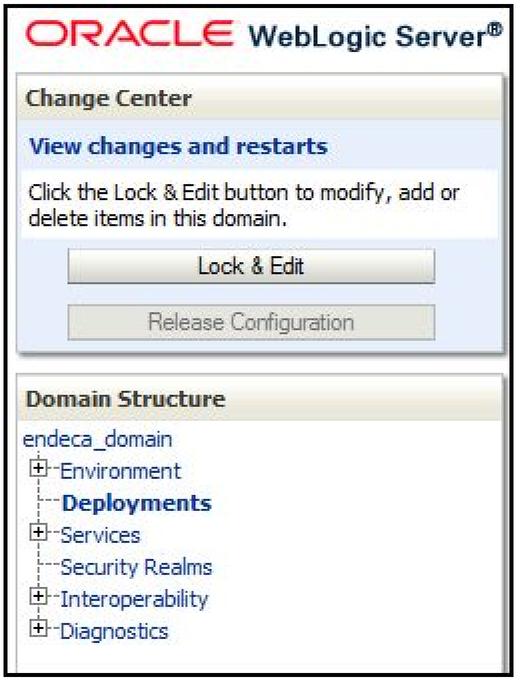
   ```
   http://admin_server_host:admin_server_port/console
   ```

   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).



ORACLE WebLogic Server®

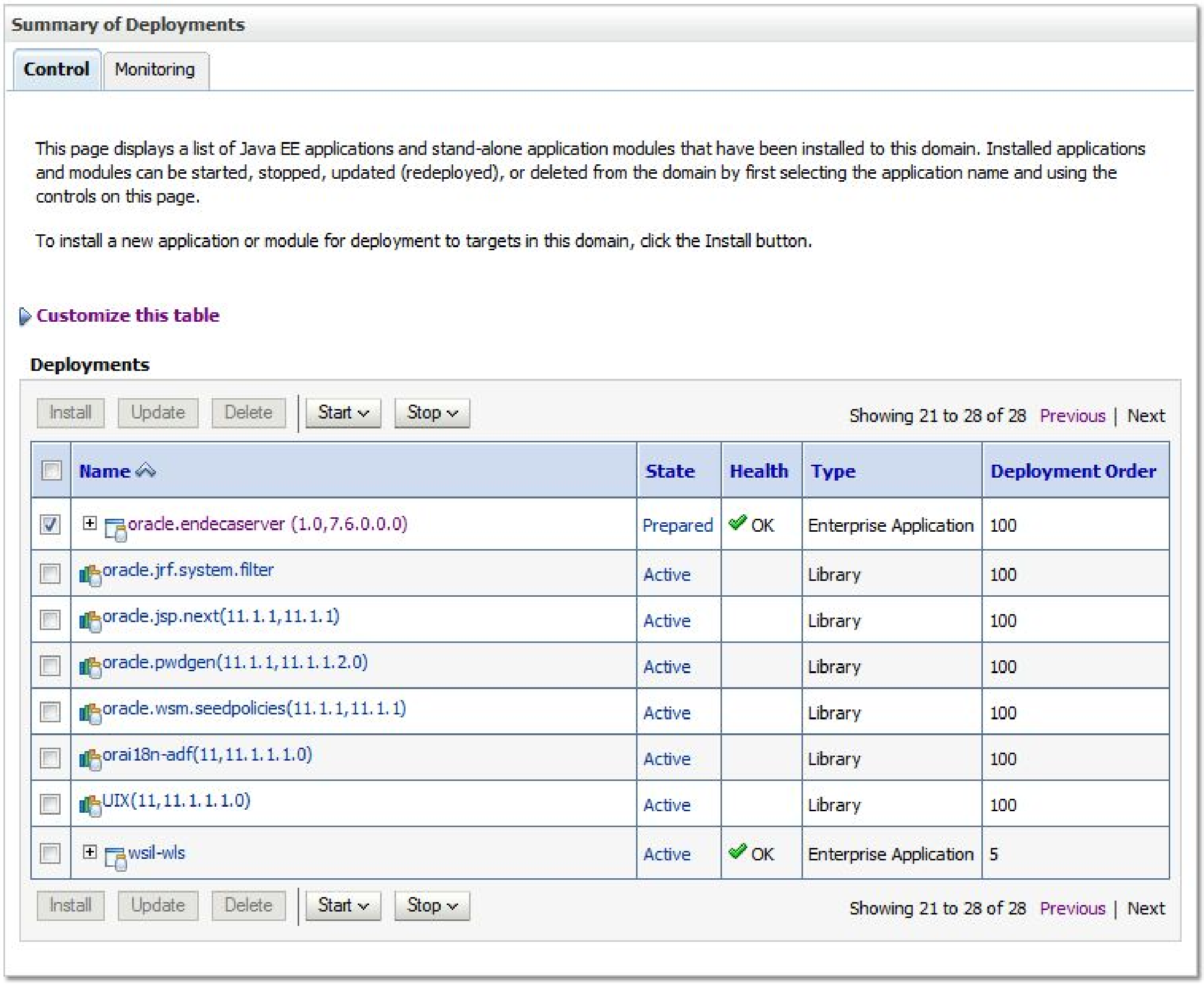
Change Center

View changes and restarts

Click the Lock & Edit button to modify, add or delete items in this domain.

Domain Structure

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

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.

Deployments

□	Name	State	Health	Type	Deployment Order
☑	⊞ oracle.endecaserver (1.0,7.6.0.0.0)	Prepared	✓ OK	Enterprise Application	100
□	oracle.jrf.system.filter	Active		Library	100
□	oracle.jsp.next(11.1.1,11.1.1)	Active		Library	100
□	oracle.pwdgen(11.1.1,11.1.1.2.0)	Active		Library	100
□	oracle.wsm.seedpolicies(11.1.1,11.1.1)	Active		Library	100
□	orai18n-adf(11,11.1.1.1.0)	Active		Library	100
□	UIX(11,11.1.1.1.0)	Active		Library	100
□	⊞ wsil-wls	Active	✓ OK	Enterprise Application	5

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:

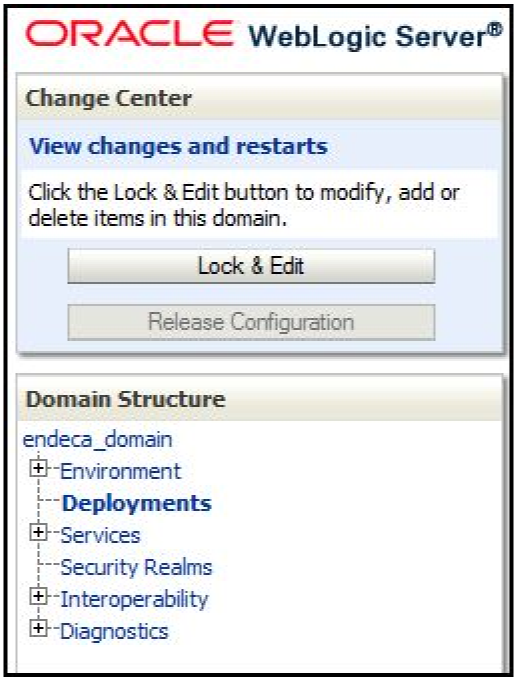
1. Make sure that the Administration Server for the Endeca domain is running.
2. From your browser, access the Administration Server console using this syntax:

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

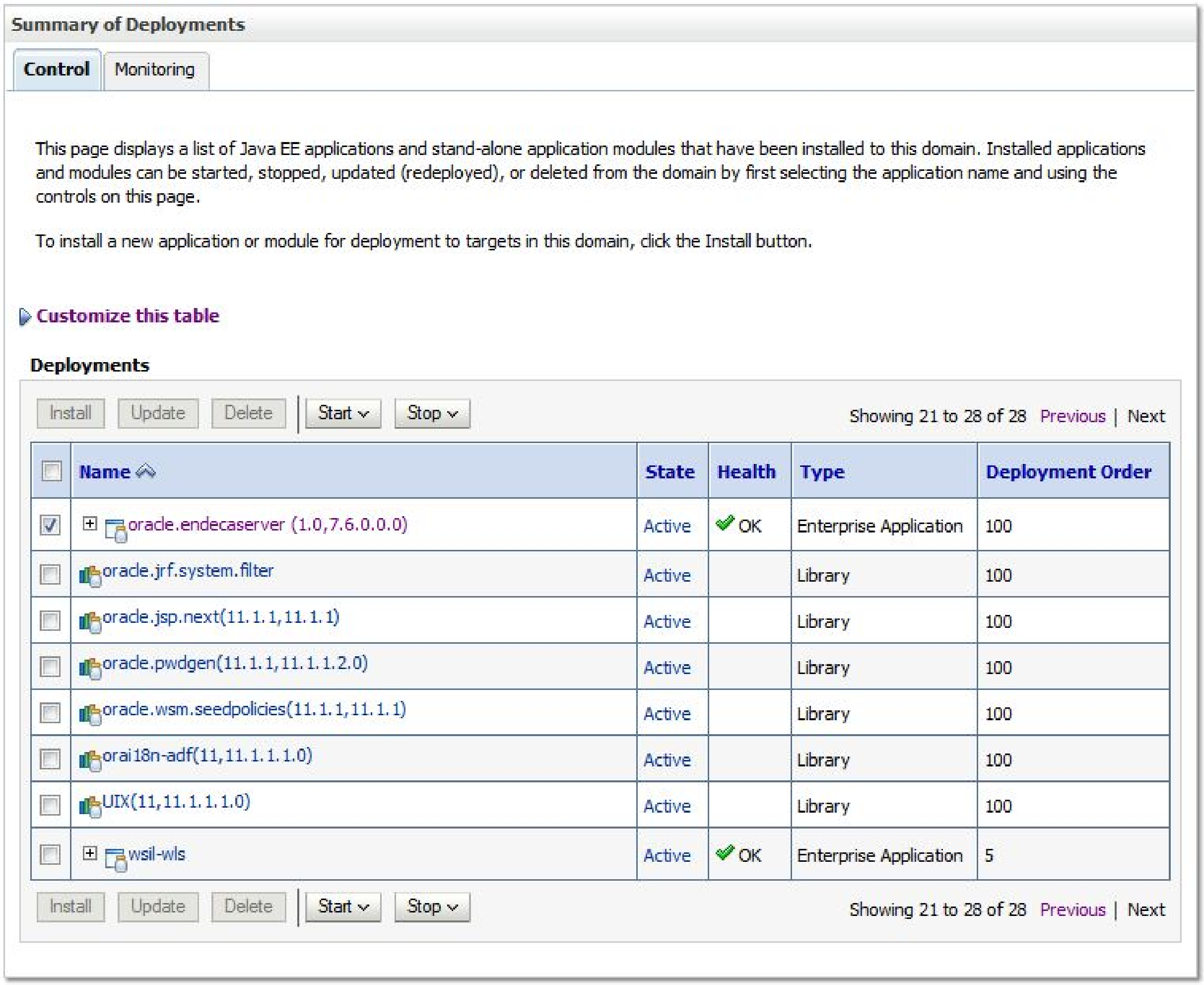
   For example:

   ```
   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:



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.

Deployments

□	Name	State	Health	Type	Deployment Order
☑	⊞ oracle.endecaserver (1.0,7.6.0.0.0)	Active	✓ OK	Enterprise Application	100
□	oracle.jrf.system.filter	Active		Library	100
□	oracle.jsp.next(11.1.1,11.1.1)	Active		Library	100
□	oracle.pwdgen(11.1.1,11.1.1.2.0)	Active		Library	100
□	oracle.wsm.seedpolicies(11.1.1,11.1.1)	Active		Library	100
□	orai18n-adf(11,11.1.1.1.0)	Active		Library	100
□	UIX(11,11.1.1.1.0)	Active		Library	100
□	⊞ wsil-wls	Active	✓ OK	Enterprise Application	5

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 in single-node mode, if you shut down WebLogic Server, Endeca Server starts without delay. If you installed in a cluster mode, 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. If you installed in a single-node mode, Endeca Server starts without delay.

When Endeca Server is deployed in a cluster mode, 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*.

Hostname resolution in the Endeca Server

This topic discusses how Endeca Server performs hostname resolution.

The Endeca Server Java application running in WebLogic relies on the hostname resolution by checking the network configuration on your machine. In the majority of network configurations, your machine is already set up properly and the Endeca Server can resolve its hostname correctly through DNS lookup, at startup. If this is the case, you can ignore this topic.

In some instances, however, additional troubleshooting might be required. This topic discusses the details.

The Endeca Server uses a Java System call to obtain the mapping between an externally-resolvable IP address and the machine's hostname. The Java System call is specific to the operating system and to the network setup on your machine. For example, if DNS is configured in an enterprise-wide domain by the IT department, the hostname resolution works fine (even if the `/etc/hosts` file on your machine is empty). However, if your `/etc/hosts` file includes any entries, Endeca Server uses this file (overriding what it finds from the DNS lookup), and this may prevent the Endeca Server to properly resolve its hostname. In such cases, you receive the "`Could not connect to Endeca Server`" error.

The summary is that Endeca Server relies on the hostname resolution by ultimately checking the network configuration on your machine which is done through `/etc/hosts`. That is, in the end, Endeca Server does not use any other OS-specific methods for determining the host name, but relies on `/etc/hosts`. In the `/etc/hosts` file, Endeca Server looks for the hostname that is mapped to the site-local IP address. It is important to know that Endeca Server cannot use a mapping to `localhost`. Therefore, to ensure host name resolution in the Endeca Server, check that your `/etc/hosts` file includes as its first entry the mapping from the machine-specific IP address to the full name of your machine. (This file can also include a mapping to `localhost`, as long as this entry is not the first, that is, it is listed after the mapping to the full name of your machine).

The following statements describe the Endeca Server’s hostname resolution logic in detail.

To resolve the hostname to the IP address that Endeca Server will use at startup, Endeca Server does the following:

1. It checks the DNS lookup first and compares it with `/etc/hosts`.
2. If it finds conflicting information, it uses `/etc/hosts` as the final source of truth. That is, Endeca Server relies on the `/etc/hosts` file, and does not rely on `/etc/hostname` and DNS lookup. Endeca Server relies on the fact that the `/etc/hosts` file contains an unambiguous mapping between either:

   - The site-local IP address and the Fully Qualified Domain Name (FQDN) of this machine, or
   - The site-local IP address and the first portion of the FQDN.

   If Endeca Server finds such a mapping, it uses the first part of the FQDN as the hostname, at startup time. For example, if the mapping in `/etc/hosts` is:

   ```
   192.168.0.1 myname.example.com
   ```

   Endeca Server will use `myname` as the hostname, at startup. Similarly, if the mapping is:

   ```
   192.168.0.1 myname
   ```

   Endeca Server will use `myname` as the hostname, at startup.
3. If the `/etc/hosts` file contains several entries for the same site-local IP address, Endeca Server uses the first hostname it finds in the file.

Important: The mapping from a site-local IP address to `localhost` is not accepted by the Endeca Server and does not allow it to resolve the host name. If `/etc/hosts` contains this mapping it should not be the first one listed. If it is the only mapping or the first mapping, Endeca Server issues an error indicating that it cannot resolve an IP address to the host name for its operations.

For example, this mapping in `/etc/hosts` is not resolvable by the Endeca Server:

```
192.168.0.1 localhost
```

Similarly, this mapping is also not resolvable by the Endeca Server:

```
127.0.0.1 localhost.localdomain localhost
::1 localhost6.localdomain localhost
```

These mappings are resolvable (`myhostname` is listed first, before `localhost`, and it is first mapped to the site-local externally-resolvable IP address):

```
192.168.0.45 myhostname localhost.localdomain localhost
```

```
127.0.0.1 myhostname localhost.localdomain localhost
::1 myhostname localhost6.localdomain localhost
```

4. Next, if Endeca Server cannot find the site-local IP address in `/etc/hosts`, it looks for any other address that is from the Ethernet network adapter configured on the system (this is another non-loopback address that the system might have), mapped to the hostname, and attempts to use it.
5. Finally, if it cannot find either of these addresses, Endeca Server falls back to using the first IP loopback address it can find on the system (this address is typically of the type 127.0.0.1).

To conclude, if you observe connection problems with the Endeca Sever, reconfigure networking configuration on the Endeca Server machines, so that `/etc/hosts` includes a mapping from the site-local IP address to FQDN, or the first portion of the correct, Fully Qualified Domain Name. If `/etc/hosts` includes a mapping to `localhost`, it should not be the only entry, and should not be the first entry in this file. Do not use any other OS-specific methods for determining the hostname — only use the configuration in `/etc/hosts`, because ultimately, this is the configuration that Endeca Server relies on.

Chapter 3

Endeca Server Command Reference

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

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\EndecaServer7.6.0\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 27*.

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

Option	Description
`--keystore`	Specifies the location of a keystore file needed for authentication to the Oracle Endeca Server. For example, the `endecaServerClientCert.ks` keystore can be used. If you use this option, it implies that SSL should be used for communication between `endeca-cmd` and the Oracle Endeca Server. This means that you don't need to use the `--ssl` option. **Note:** If you specify a keystore, this causes `endeca-cmd` to prompt for a password. Therefore, if you use the `--keystore` option, you cannot run `endeca-cmd` as part of a script.
`--truststore`	Specifies the location of a truststore file needed for verifying the authenticated connection to the Oracle Endeca Server. For example, the `endecaServerTrustStore.ks` truststore can be used. If you use this option, it implies that SSL should be used for communication between `endeca-cmd` and the Oracle Endeca Server. This means that you don't need to use the `--ssl` option. **Note:** If you specify a truststore, this causes `endeca-cmd` to prompt for a password. Therefore, if you use the `--truststore` option, you cannot run `endeca-cmd` as part of a script.
`--ssl`	Specifies whether to use an authenticated SSL connection to the Oracle Endeca Server. If you use either the `--keystore` or `--truststore` option, then you don't need to use the `--ssl` option, as the authenticated SSL connection is implied by specifying the keystore or the truststore file. If you use `--ssl` without either a `--keystore` or `--truststore` option, you can use an empty keystore and the `cacerts` file as your truststore. Alternatively, you can use the standard approaches for creating keystore and truststore files in the WebLogic Server.
`--password`	Specifies a password for accessing the keystore and its key, and the truststore. The keystore and truststore are both assumed to use this password. 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). This option is intended for use in scripts.
`--keystore-password`	Specifies a password for accessing the keystore (overrides the `--password` option). This option is intended for use in scripts.

Option	Description
`--truststore-password`	Specifies a password for accessing the truststore (overrides the `--password` option). This option is intended for use in scripts.
`--key-password`	Specifies a password for accessing the key within the keystore (overrides the `--password` 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 31*
- *get-node-profile on page 32*
- *get-node-health on page 32*
- *get-cluster-health on page 32*
- *delete-node on page 32*
- *list-nodes on page 33*

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
`--num-cpu-cores <int>`	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.
`--ram-size-mb <int>`	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.

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 34*
- *get-dd-profile on page 37*
- *list-dd-profiles on page 37*
- *delete-dd-profile on page 38*

put-dd-profile

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

The syntax for this command is:

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

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

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

Create Option	Description
`--description <text>`	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.
`--query-leader <bool>`	Whether the leader node can accept read-only queries. If not specified, defaults to `true`.
`--num-followers <int>`	How many follower nodes (Dgraph processes) should be configured in the data domain. If not specified, defaults to 0.
`--read-only <bool>`	Whether a data domain should be created as read-only and not accept updating requests, including data loading requests. If not specified, defaults to `false`.
`--oversubscribe <bool>`	Whether a data domain (or any of its nodes) can be placed on an oversubscribed Endeca Server instance. If not specified, defaults to `true`.
`--num-compute-threads <int>`	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.
`--compute-cache-size <int>`	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.
`--startup-timeout <int>`	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.
`--shutdown-timeout <int>`	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.

Create Option	Description
`--ancestor-counts <bool>`	Optional. If enabled, computes counts for root managed attribute values and any intermediate managed attribute value selections. The default is `false`. 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.
`--backlog-timeout <int>`	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.
`--refinement-sampling-min <int>`	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.
`--implicit-exact <bool>`	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.
`--implicit-sample <int>`	Optional. Sets the maximum number of records to sample when computing implicit refinements (which are a performance tuning parameter). The default value is 1024.
`--net-timeout <int>`	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.
`--search-max <int>`	Optional. Sets the maximum number of terms for text search. The default value is `10`.
`--search-char-limit <int>`	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.
`--snippet-cutoff <int>`	Optional. Limits the number of words in an attribute that the Dgraph evaluates to identify the snippet. If a match is not found within <*num*> 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 <*num*> is not specified, the default is 500.
`--snippet-disable <bool>`	Optional. Globally disables snippeting. The default is `false`, meaning snippeting is enabled.

Create Option	Description
`--dynamic-category-enable <bool>`	Optional. Enable all available dynamic attribute value characteristics. The default is `false`. Note that this option has performance implications and is not intended for production use.
`--contraction-disable <bool>`	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 `false`. 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.
`--wildcard-max <int>`	Optional. Specifies the maximum number of terms that can match a wildcard term in a wildcard query that contains punctuation, such as `ab*c.def*`. The default is `100`.
`--auto-idle <bool>`	Specifies whether the Endeca Server should turn this data domain idle, if the data domain receives no queries during the specified `--idle-timeout`. If `auto-idle` is set to `false`, the data domain never turns idle. If set to `true`, the data domain turns idle after `idle-timeout` expires and if no queries arrive during the timeout period. The data domain is considered idle if it is set to auto-idle 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.
`--idle-timeout <int>`	Optional. Specifies the timeout period, in minutes, after which the data domain that is configured to auto-idle is turned idle by the Endeca Server, if the data domain receives no queries during the timeout period. The default is 10 minutes, and it is used if the timeout is not specified. This setting is not used for data domains for which `auto-idle` is set to `false`.
`--session-id-type <type>`	The method to use for establishing session affinity. The available options are: `header` (for HTTP headers), `parameter` (for URL parameters), or `cookie`. The default method is `header`.
`--session-id-key <name>`	The name of the key to use for maintaining affinity. The default name is `X-Endeca-Session-ID`.
`--args <dgraph-flags>`	Specifies a list of the additional Dgraph flags that will be used for the data domain's Dgraph process. The `--args` flag must be the last flag on the command line, as all of its arguments are passed on to the Dgraph process.

Create Option	Description
`--args --usage`	Provides a list of the available Dgraph process flags. See *Dgraph flags on page 69*.

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 39*
- *clone-dd on page 40*
- *delete-dd on page 40*
- *disable-dd on page 40*
- *enable-dd on page 40*
- *export-dd on page 41*
- *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 44*
- *version 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`.

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
`--dd-profile-name` *`name`*	Specifies the name of the data domain profile name to be used. If not specified, defaults to using the default data domain profile.
`--is-enabled` *`boolean`*	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 create and register but not start the data domain, specify `--is-enabled false`.

Example 1:

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

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

Example 2:

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

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

clone-dd

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

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

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

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

This example:

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

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

delete-dd

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

The syntax for this command is:

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

disable-dd

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

The syntax for this command is:

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

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

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

enable-dd

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

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

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

export-dd

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

The 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
`--dd-profile-name`	The name of the data domain profile to use (if not specified, defaults to the default data domain profile).
`--is-enabled`	If set to `true`, 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 `--is-enabled false`.

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]
```

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

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

An Endeca ***data domain*** is a set of one or more Dgraph processes that together handle end-user query requests. One of the Dgraph processes in a data domain is responsible for handling all write operations (updates, configuration changes), while the remaining Dgraph processes serve as read-only. All Dgraph nodes in a given data domain typically utilize the same index residing on shared storage. Each data domain node is represented by the Dgraph process and is typically hosted on a separate Endeca Server instance application running in an Endeca Server cluster.

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

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

If you configure a data domain to run on a single Endeca Server instance, it typically also requires a single node that is started once you start the Endeca Server application. If you configure a data domain to run multiple Dgraph processes (such data domains are described as "data domains with multiple nodes"), then the Endeca Server cluster locates ports on which to start the Dgraph processes for the data domain and identifies which Dgraph configuration flags to use.

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

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

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

About the Dgraph process

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

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

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, the data domain would need to be recreated in the Endeca Server cluster that is 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 Sever 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.
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, 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.

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, 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
`deleteDataDomain` request	This request in the Manage Web Service deletes a data domain if you specify its name.

Option	Description
`endeca-cmd delete-dd` *name*	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-mame` 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.

6. 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.

Note: 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 one of the data domain profiles.

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 (for example, open a Command Prompt in Windows) and navigate to the `endeca-cmd` directory.
2. Use the options as shown in the following examples:

Option	Description
Exporting: `endeca-cmd export-dd my_dd --offline-name my_dd_offline`	This example of the command exports the index of the data domain `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 the name, 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.

Option	Description
Importing: `endeca-cmd import-dd new_dd --offline-name my_dd_offline --dd-profile-name TestDDProfile --is-enabled false`	This example of the command imports the index `my_dd_offline` into the new data domain `new_dd`, which is created with the `TestDDProfile` data domain profile and is not enabled after it is created. **Note:** Importing assumes that you have previously created a data domain whose index is currently exported; it also assumes that you kept track of the `--offline-name` value you used when exporting, or know the name that was automatically assigned.

If 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`, based on the default data domain profile. 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:ddProfileName>default</ns1:ddProfileName>
    <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
`enableDataDomain` of the Manage Web Service, or `endeca-cmd enable-dd` *`name`*	Use this option to enable a previously disabled data domain.
`createDataDomain` of the Manage Web Service, or `endeca-cmd create-dd --is-enabled true`	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.
`cloneDataDomain` of the Manage Web Service, or `endeca-cmd clone-dd --is-enabled true`	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-enable` 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
`disableDataDomain` or `endeca-cmd disable-dd` *`name`*	This command of the Manage Web Service disables a data domain.
`createDataDomain` or `endeca-cmd create-dd --is-enabled false`	When you create a new data domain, you can specify `false` for `is-enabled`. In this case, the created data domain is disabled (its processes are not running).
`cloneDataDomain` or `endeca-cmd clone-dd --is-enabled false`	When you clone an existing data domain, if you specify `false` for `is-enabled`, 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
`listDataDomains` in the Manage Web Service, or the equivalent `endeca-cmd list-dd --verbose` command.	Returns information about all data domains registered with the Endeca Server, along with their characteristics.
`getDataDomain` in the Manage Web Service, or the equivalent `endeca-cmd get-dd` *name* command.	Returns information about a single data domain with the specified `name`.

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 `allocateBulkLoadPort`*name* 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 53*.
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.

Endeca Server memory consumption

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

Endeca Server memory consumption

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 Endeca Server memory consumption is characterized by these statements:

- **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. This is observable if you run operating system diagnostic tools. However, measuring the Dgraph process size on a lightly-loaded machine is not an indication that the Dgraph process will actually utilize all of the physical memory it initially claims. The Dgraph process allocates considerable amounts of virtual memory while ingesting data or executing complex queries, however, the usage of physical memory depends on the demands of other processes on the system. The Dgraph process releases the physical memory when other processes in the operating system require it. When other memory-intensive processes are present, including other Dgraph processes, Dgraph releases a significant portion of its physical memory quickly. Without such pressure, it may retain the physical memory indefinitely.

  Hence, 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. To conclude, you should not rely on these estimates for predicting how much memory the Dgraph actually requires to run to support your data domains.

- **Endeca Server relies on internal heuristics to estimate whether it can host multiple data domains**. When you provision a new data domain, Endeca Server first determines whether it has sufficient capacity to host the data domain, and if yes, decides which nodes will host the Dgraph processes for the data domain. To calculate whether it has sufficient capacity to host (new or any additional) data domains Endeca Server uses the following parameters: oversubscribing, auto-idling, memory calculations, and threads calculations.

  Specifically, to allocate data domains, Endeca Server chooses nodes which have sufficient processing cores for the new data domains. It denies the creation of additional data domains when it does not have enough processing cores to host them (this behavior is better guaranteed on Linux 6 with cgroups enabled, if the Endeca Server is configured to utilize cgroups). To conserve resources, Endeca Server turns to idle those inactive data domains that are configured to auto-idle, and automatically activates such data domains when an end-user's request arrives. To help the cluster administrators of the Endeca Server tune the allocation of data domains to Endeca Server instances, Endeca Server allows specifying the number of threads per core, and the ratio of allocated virtual memory to the index size, for the nodes in the Endeca Server deployment.

  For detailed information about how the Endeca Server performs its own internal capacity calculations and which heuristics are being used, see the topic about data domain allocation in the *Oracle Endeca Server Cluster Guide*.

- **On Linux 6 cgroups can be utilized for OS-level data domain allocation guarantees**. In addition to the parameters you can specify (such as oversubscribing auto-idling of data domains, or the number of threads and CPU cores for nodes), if the Endeca Server is deployed on Oracle Linux 6 or RHEL 6, cgroups can be used for data domain allocation, providing additional guarantees.

  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.

For information on cgroups, see *About using control groups (cgroups) for data domains on page 61*.

- **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 Had 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.
6. 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 `root` 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.
7. 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 (`sonfig`), 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 `idle-timeout` 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 10 minutes is used. 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 the `putDataDomainProfile` operation (and its idling options) of the Cluster Web Service.

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 (and its oversubscribing option) 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
`endeca-cmd put-dd-profile --session-id-type` *`method`*	The *`method`* that will be used for establishing session affinity. The available options are: `HEADER` for HTTP headers, `PARAMETER` for URL parameters, or `COOKIE`. The default method is `HEADER`. The values are not case-sensitive.
`endeca-cmd put-dd-profile --session-id-key` *`name`*	The name of the object that will be checked by the specified method. The default name is `X-Endeca-Session-ID`.

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

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

Dgraph Flag that must be specified only with `endeca-cmd put-dd-profile --args --`*`<Dgraph_flag>`*	**Description**
`--sslcafile` *`<CA-certfile-path>`*	Specify the path of the `eneCA.pem` 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.
`--sslcertfile` *`<certfile-path>`*	Specify the path of the `eneCert.pem` 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.
`--syslog`	Direct all output to syslog.
`--validate_data`	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 33*, 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 `--args`**, as in:** `endeca-cmd put-dd-profile --`*`<Dgraph_flag>`*	**Syntax of the flag when you specify it with the** `--args`**, as in:** `endeca-cmd put-dd-profile --args --`*`<Dgraph_flag>`*
`--ancestor-counts`	`--ancestor_counts`
`--backlog-timeout`	`--backlog-timeout`
`--refinement-sampling-min`	`--esampmin`
`--implicit-exact`	`--implicit_exact`

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

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 utilize the following syntax.

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

Where:

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

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

For example:

```
http://localhost: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 Dgraph recognizes the following admin operations:

Admin operation	Description
`/admin/`*`datadomain`*`?op=help`	Returns the usage page for all of the admin operations.
`/admin/`*`datadomain`*`?op=flush`	Specifies when the Dgraph should flush its dynamic cache.
`/admin/`*`datadomain`*`?op=logroll`	Forces a query log roll.
`/admin/`*`datadomain`*`?op=merge`	Merges update generations in the index and sets the system's merge policy.
`/admin/`*`datadomain`*`?op=ping`	Checks the aliveness of a Dgraph and returns a lightweight message.
`/admin/`*`datadomain`*`?op=stats`	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.
`/admin/`*`datadomain`*`?op=statsreset`	Resets the Dgraph Statistics page for one of the Dgraphs in the data domain.

help

flush

logroll

merge

ping

stats

statsreset

help

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

flush

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

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

```
flushing cache...
```

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

logroll

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

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

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

merge

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

Managing the index merge policy for the data domain

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 Sever Statistics page. Note that this page is intended for use by Oracle Endeca Support only.

statsreset

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

The `statsreset` operation returns the following message:

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

Note that the Dgraph Server Statistics page is intended for use by Oracle Endeca 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

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 unlimit settings for merges

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

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

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

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

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

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 three categories of logging in the Endeca Server: the logging of the Endeca Server Java process in the WebLogic Server domain, the Endeca data domain logging, and the Dgraph process logs.

Endeca Server logs

Data domain logs

Dgraph request log and stdout/sterr log

Endeca Server logs

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

Each WebLogic Server instance maintains these log files:

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

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

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

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

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

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

Log entry format

Setting log levels

Creating log handlers for debugging

Customizing the HTTP access log

Log entry format

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

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

```
C:\Oracle\Middleware\EndecaServer7.6.0\endeca-cmd>endeca-cmd disable-dd text
SEVERE: Error while invoking endpoint "http://localhost:7001/endeca-server/ws/manage"
from client endeca-cmd encountered a problem.
caused by:
The Endeca Server returned an error: OES-000095: No such data domain: text
caused by:
OES-000095: No such data domain: text
  OES-000095: No such data domain: text
```

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

```
[2013-10-21T16:31:16.384-04:00] [AdminServer] [ERROR] [OES-000066]
[com.endeca.opmodel.ws.ManagePortImpl]
[tid: [ACTIVE].ExecuteThread: '34' for queue: 'weblogic.kernel.Default (self-tuning)'] [userId:
<anonymous>]
[ecid: 0000K7ShiBy9lZZ_TlDCic1IPOwH000001,0:1] [WEBSERVICE_PORT.name: managePort]
[APP: oracle.endecaserver#1.0@7.6.0.0.0] [J2EE_MODULE.name: endeca-server] [WEBSERVICE.name: manage]
[J2EE_APP.name: oracle.endecaserver_1.0@7.6.0.0.0] OES-000066: Service error:
com.endeca.cluster.ClusterException$NoSuchDataDomainException:
OES-000095: No such data domain: text com.endeca.cluster.ClusterException$NoSuchDataDomainException:
OES-000095: No such data domain: text[[
    at com.endeca.cluster.Cluster.setDataDomainEnabled(Cluster.java:534)
    at com.endeca.opmodel.ws.ManagePortImpl.disableDataDomain(ManagePortImpl.java:418)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
    at java.lang.reflect.Method.invoke(Method.java:597)
 ...
    at weblogic.work.ExecuteThread.run(ExecuteThread.java:221)
]]
```

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

Log entry field	Meaning
`[2013-10-21T16:31:16.384-04:00]`	The timestamp, which is the date and time when the message was generated.
`[AdminServer]`	The organization ID for the originating component, which is the WebLogic Admin Server.
`[ERROR]`	The ODL message type. See the table below for possible values.
`[OES-000095]`	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.

Log entry field	Meaning
`[com.endeca.opmodel.ws.ManagePortImpl]`	The component ID, which identifies the Endeca Server component that originated the message.
`[tid: [ACTIVE].ExecuteThread: '34' for queue: 'weblogic.kernel.Default (self-tuning)']`	The ID of the thread that generated the message.
`[userId: <anonymous>]`	The name of the user whose execution context generated the message.
`[ecid: 0000K7ShiBy9lZZ_TlDCic1IPOwH000001, 0:1]`	The Execution Context ID (ECID), which is a global unique identifier of the execution of a particular request in which the originating component participates.
`[WEBSERVICE_PORT.name: managePort]`	Supplemental Attribute that contains a component-specific attribute about the event.
`[APP: oracle.endecaserver_1.0@7.6.0.0.0]`	Application ID for the Endeca Server application.
`[J2EE_MODULE.name: endeca-server]`	Supplemental Attribute that contains a component-specific attribute about the event.
`[WEBSERVICE.name: manage]`	Supplemental Attribute that contains a component-specific attribute about the event.
`[J2EE_APP.name: oracle.endecaserver_1.0@7.6.0.0.0]`	Supplemental Attribute that contains a component-specific attribute about the event.
`[OES-000066: Service error: ...]`	Message Text. The text of the error message.

Message types and levels

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

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

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

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

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

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

Setting log levels

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

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

Note: This topic explains how to modify the default Endeca Server loggers. For information on creating a new log handler for Endeca Server (which uses a separate log file), see *Creating log handlers for debugging on page 85*.

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

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

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

The Endeca Server loggers are:

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

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

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

where:

- `logger` is a mandatory parameter that specifies the full name of the logger to be modified. Note that the logger name is case-sensitive (which also means that it cannot use a wildcard).
- `level` is a mandatory parameter that specifies the log level. You can specify either an ODL level (such as `TRACE:16`) or a Java level (such as `FINER`). For a list of the levels, see *Message types and levels on page 82*.
- `persist` is an optional parameter that specifies whether the level should be saved to the configuration file. The values are 0 (do not save the level) or 1 (save the level, which will persist across WebLogic re-starts). The default is 1.

To set a log level for an Endeca Server logger:

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

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

   For example:

   ```
   connect('weblogic', 'welcome1')
   ```
5. Use the `setLogLevel()` command to set a new log level for a specific Endeca Server logger.

   For example:

   ```
   setLogLevel(logger="com.endeca.opmodel.control.DGraph", level="TRACE:1")
   ```
6. Use the `listLoggers` command to verify that the log level was changed.

   For example:

   ```
   listLoggers(pattern="com.endeca.opmodel.control.DGraph")
   ```
7. When you have finished, use the `exit()` command to exit the WLST utility.

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

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

Creating log handlers for debugging

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

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

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

where:

- `name` specifies the name of the new log handler.
- `addToLogger` specifies the name of an existing logger to be added to this log handler.
- `path` specifies the absolute path and name of the log file.
- `handlerType` specifies the name of the Java class that provides the handler implementation.

  It must be an instance of `java.util.logging.Handler` or `oracle.core.ojdl.logging.HandlerFactory`.
- `addHandler` is a Boolean value (1=true, 0=false). If the value is true, then the named handler will be added.

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

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

To add a new log handler:

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

   For example:

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

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

   For example:

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

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

   See the example above for the syntax.

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

   For example:

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

   The command output should look similar to this:

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

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

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

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

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

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

Customizing the HTTP access log

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

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

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

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

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

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

To customize the HTTP access log:

1. Log into the Administration Server console.
2. In the Change Center of the Administration Console, click **Lock & Edit**.
3. In the left pane of the Console, expand **Environment** and select **Servers**.
4. In the Servers table, click the **AdminServer** name.
5. In the Settings for AdminServer page, select **Logging>HTTP**.
6. On the Logging > HTTP page, make sure that the **HTTP access log file enabled** checkbox is checked.
7. Click **Advanced**.
8. In the Advanced pane:
   (a) In the Format list box, select **Extended**.
   (b) In the Extended Logging Format Fields, enter this space-delimited string:
   ```
   date time time-taken cs-method cs-uri sc-status
   ```
9. Click **Save**.
10. In the Change Center of the Administration Console, click **Activate Changes**.
11. Stop and then re-start WebLogic Server.

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

```
#Version:    1.0
#Fields:    date time time-taken cs-method cs-uri sc-status
#Software:    WebLogic
#Start-Date:    2013-10-22    15:23:40
2013-10-22    15:27:07    0.967    POST    /endeca-server/ws/manage    200
2013-10-22    15:27:32    7.301    POST    /endeca-server/es/control    200
2013-10-22    15:27:32    7.567    POST    /endeca-server/ws/manage    200
2013-10-22    16:23:35    0.219    GET    /endeca-server/ws/conversation/sh?wsdl    200
2013-10-22    16:23:35    0.0    GET    /favicon.ico    404
```

```
2013-10-22    16:24:14    0.031    GET    /endeca-server/ws/conversation/sh?wsdl    200
2013-10-22    16:24:14    0.031    GET    /endeca-server/ws/conversation/sh?XSD
=lql_parser_types.xsd    200
2013-10-22    16:27:15    5.692    POST    /endeca-server/ws/conversation/sh    200
2013-10-22    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.

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 92*.

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:
endecaresultslistportlet_WAR_endecaresultslistportlet_INSTANCE_5RKp_LAYOUT_11601
```

As an example, if you set:

```
PORTLET=endecaresultslistportlet_WAR_endecaresultslistportlet_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
`/config/`*`ddomain`*`?op=help`	Returns the usage page for all of the config operations.
`/config/`*`ddomain`*`?op=log-enable`	Enables verbose logging for one or more specified variables.
`/config/`*`ddomain`*`?op=log-disable`	Disables verbose logging for one or more specified variables.
`/config/`*`ddomain`*`?op=log-status`	Returns verbose logging status.

The configuration operations utilize the following syntax:

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

Where:

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

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

For example:

```
http://localhost: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=textsearchrelrankverbose&name=textsearchspellverbose
```

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

However, this operation:

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

returns an unsupported logging setting message.

In addition, the following operations are supported:

- /config/*ddomain*?op=log-status returns a list of all logging variables with their values (true or false).
- The special name all can be used with /config/*ddomain*?op=log-enable or /config?/*dstore*op=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
verbose	Enables verbose mode.
requestverbose	Prints information about each request to stdout.
textsearchrelrankverbose	Enables verbose information about relevance ranking during search query processing.

Variable	Description
textsearchspellverbose	Enables verbose output for spelling correction features.
dgraphperfverbose	Enables verbose performance debugging messages during core Dgraph navigation computations.
dgraphrefinementgroupverbose	Enables refinement verbose/debugging messages.

log-enable

log-disable

log-status

help

log-enable

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

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

For example, this operation:

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

turns on verbose logging for queries, while this operation:

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

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

However, this operation:

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

returns an "Unsupported logging setting" message.

log-disable

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

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

log-status

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

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

```
Logging settings:

verbose - FALSE
requestverbose - TRUE
updateverbose - FALSE
recordfilterperfverbose - FALSE
```

```
textsearchrelrankverbose - TRUE
textsearchspellverbose - FALSE
dgraphperfverbose - FALSE
dgraphrefinementgroupverbose - FALSE
```

help

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

Index