

Oracle® Endeca Server

Cluster Guide

Version 7.7.0 • January 2016

Copyright and disclaimer

Copyright © 2003, 2016, Oracle and/or its affiliates. All rights reserved.

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

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

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

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

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

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

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

Table of Contents

Copyright and disclaimer	2
Preface	5
About this guide	5
Who should use this guide	5
Conventions used in this guide	5
Contacting Oracle Customer Support	6
Chapter 1: Introducing the Endeca Server Cluster	7
Overview	7
Endeca Server Cluster	7
Data domain cluster	8
Building a cluster	9
Data domain cluster concepts	10
Cluster diagrams	10
Diagram with a single data domain	11
Diagram with multiple data domains	12
Cluster behavior	13
Allocation of data domains	13
Load balancing and routing of requests	16
Data domain cluster startup	17
How updates are processed	17
Role of the Cluster Coordinator	18
How enhanced availability is achieved	19
Chapter 2: Managing the Endeca Server Cluster	22
Listing Endeca Server nodes	22
Obtaining version information	23
About the node profile	23
Configuring a custom node profile	24
Monitoring node status	25
Monitoring cluster status	25
Deleting an Endeca Server node	26
Chapter 3: Managing Data Domain Profiles	27
About the data domain profile	27
Deciding which data domain profile to use	28
Data Domain profile parameters	30
Default data domain profile	37
Defining a data domain profile	39
Listing data domain profiles	40
Removing a data domain profile	42

How idling affects data domain behavior	42
How oversubscribing affects hardware utilization	44
Example: data domain profile with low hardware usage	47
Example: data domain profile with high hardware usage	47
Chapter 4: Using the Cluster Web Service	49
About the Cluster Web Service	49
Accessing the Cluster Web Service	49
List of operations in the Cluster Web Service	50
Chapter 5: Using the Manage Web Service	56
About the Manage Web Service	56
Accessing the Manage Web Service	57
List of operations in the Manage Web Service	57

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 Oracle Endeca Server cluster, consisting of multiple Endeca Server instances that can host one or more Endeca data domains.

This guide does not discuss deploying a single Endeca Server with a single Endeca data domain. Even though you can configure single-node Endeca data domains hosted by single Endeca Server instances, single Endeca Server instances are not highly available and can only be used in development environments. For information on installing an Endeca Server on a single node and starting a data domain on it, see the *Oracle Endeca Server Installation Guide*.

Who should use this guide

This guide is intended for administrators who configure and deploy an Oracle Endeca Server cluster and make it available for use by the clients. It also addresses those administrators who use an already deployed Endeca Server cluster to create and manage the Endeca data domains.

This guide assumes that the Oracle Endeca Server software is already installed on multiple servers, and that the Endeca Server cluster is already deployed. For information on installation and deployment of an Endeca Server cluster, see the *Oracle Endeca Server Installation Guide*.

Conventions used in this guide

The following conventions are used in this document.

Typographic conventions

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

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

Typeface	Meaning
File Path	This formatting is used for file names and paths.

Symbol conventions

This table describes the symbol conventions used in this document.

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

Path variable conventions

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

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

Contacting Oracle Customer Support

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

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



Chapter 1

Introducing the Endeca Server Cluster

This section introduces the concepts and capabilities of the Endeca Server Cluster and includes diagrams of data domains hosted in it.

[Overview](#)

[Cluster diagrams](#)

[Cluster behavior](#)

Overview

This section discusses the Endeca Server Cluster and the data domain cluster, and describes the cluster concepts.

[Endeca Server Cluster](#)

[Data domain cluster](#)

[Building a cluster](#)

[Data domain cluster concepts](#)

Endeca Server Cluster

The **Endeca Server cluster** is a deployment of multiple Endeca Server instances that host and manage multiple clustered data domains.

A **data domain** is a logical collection of data and metadata managed by the Endeca Server. Through its interfaces, the Endeca Server allows for data loading, configuration, and querying of a data domain. A data domain may impose order on subsets of its data through semantic entities. A data domain is the largest unit of data over which the Endeca Server allows queries to be expressed. (Applications wishing to correlate, join, or display data from multiple data domains, must do so themselves.)

The Endeca Server cluster has the following characteristics:

- It serves as a building block for delivering other services on top of the Oracle Endeca Server software. It also supports on-premise deployments of Oracle Endeca Information Discovery Studio.
- It runs in WebLogic on a cluster of identical hardware nodes, all of which have access with write permissions to shared storage. Each such node in the Endeca Server cluster is known as the **Endeca Server node**. Each node in the cluster must have the same hardware. The hardware characteristics, such as CPU and RAM, are configured in the **node profile**. Only one node profile can be configured in the system, and it applies to all Endeca Server instances (nodes) in the Endeca Server cluster.

- It encapsulates the details of setting up **data domain clusters**, by keeping details about the location of the Endeca Server nodes and hosted data domain nodes hidden from the end users. (The end users are users of any front-end application configured on top of a particular data domain cluster.)
- The Endeca Server cluster makes decisions about how to allocate resources to data domains, performs routing of requests to the appropriate nodes, and maintains enhanced availability of query processing for data domains.
- It provides a unified interface for provisioning and managing data domains while automating the process of data domain cluster setup, allocation of processing resources, and configuration.
- It lets you create a reusable set of **data domain profiles** for data domains. Each of the data domain profiles maps to different allotments of the underlying hardware resources on the Endeca Server nodes. When you create a data domain cluster, a previously defined profile is used initially; a data domain cluster can be changed after it has been created.
- It provides routing and load balancing of requests for data domains.

Data domain cluster

One or more data domain clusters can be hosted in a clustered Endeca Server deployment. This topic introduces the data domain cluster and describes its capabilities.

A **data domain cluster** is a set of Dgraph processes that together handle requests for a given data domain. One of these Dgraph processes 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 cluster utilize the same index residing on shared storage. Each data domain cluster node is hosted on a separate Endeca Server node.

In this guide, the term **Dgraph node** is used interchangeably with the term **data domain node**. The Dgraph node is the node in the data domain cluster that runs the Dgraph process. The Dgraph process of the Oracle Endeca Server is the main computational module that provides the features of the Endeca Server, such as search, refinement computation, and Guided Navigation. For information about the capabilities of the Dgraph process, see the *Oracle Endeca Server Developer's Guide*.

Additionally, the term **data domain** is used interchangeably with the term **data domain cluster**.

For definitions of the data domain cluster and the data domain nodes see [Data domain cluster concepts on page 10](#).

Data domain cluster capabilities

A data domain cluster provides the following capabilities:

- **Enhanced availability of query processing by the Oracle Endeca Server.** In a cluster of Dgraphs (a data domain cluster), if one of the Dgraph nodes fails, queries continue to be processed by other Dgraph nodes.
- **Increased throughput by the Oracle Endeca Server.** In a data domain cluster, you change throughput capacity by adding Dgraph nodes. This allows you to spread the query load across them, without the need to increase storage requirements at the same rate.
- **Routing of requests to the data domains.** Requests to a given data domain can be submitted to any Endeca Server instance in the Endeca Server cluster, regardless of whether or not it hosts the target data domain. The Endeca Server instance receiving the request will route it appropriately.

- **Automatic allocation of the data domain leader node.** The leader and follower nodes differ in the types of queries they can process, however, this is transparent to the end users of applications running on top of Endeca data domains. The allocation of leader and follower node roles is performed by the Endeca Server cluster, when you add data domain clusters. Allocation of the leader node ensures high availability of data loading.

Building a cluster

This topic provides at-a-glance summary of high-level tasks required to deploy a data domain cluster hosted in the Endeca Server cluster.

The creation of an Endeca data domain hosted in the Endeca Server cluster assumes two administrative roles:

- An administrator who will be installing and deploying the Endeca Server software on a homogenous set of hardware nodes, and setting up the Endeca Server cluster.
- An administrator who will be creating an Endeca data domain on an already deployed Endeca Server cluster, utilizing the data domain profiles provided by the Endeca Server cluster administrator.

The following procedure describes the overall process end-to-end, with the assumption that various parts of this process are performed by different administrative roles: Steps 1 - 3 are performed by the administrator responsible for the Endeca Server cluster deployment and management; step 4 is performed by the administrator responsible for the creation of data domains hosted in an already provisioned Endeca Server cluster.

To build an Endeca Server cluster:

1. Install the Endeca Server software on a set of hardware nodes, and deploy an Endeca Server cluster. For information, see the *Endeca Server Installation Guide*.
2. Configure the node profile for the Endeca Server nodes. For information, see the section [Managing the Endeca Server Cluster on page 21](#) in this guide.
After you set up the node profile, its characteristics will be used by all nodes in the Endeca Server cluster.
3. Configure one or more data domain profiles. For information, see the section [Managing Data Domain Profiles on page 26](#) in this guide.
The data domain profiles you create will be available to the administrator responsible for the data domain cluster configuration — for each new data domain, one of these data domain profiles will be used.
4. Add a new Endeca data domain, and perform various other tasks in the data domain, including adding or deleting nodes, monitoring, or loading data into it. For information on all data domain tasks, see the *Oracle Endeca Server Administrator's Guide*.

Data domain cluster concepts

This topic introduces the terms used to describe data domain clusters — the leader and follower nodes, and the Cluster Coordinator.

Leader node

The **leader node** of a data domain cluster is a single Dgraph process in the data domain cluster responsible for receiving and processing updates to the index and configuration. This node can also be configured to process queries, like other nodes. This node is responsible for generating information about the latest versions of the index, and propagating this information to the follower nodes.

When you create a new data domain cluster, the Endeca Server software creates and starts the leader node and then the follower nodes. Each data domain cluster (that is not read-only) is created with one leader node and a specified number of follower nodes.

The leader node periodically receives full or incremental index updates. It also receives administration or configuration updates. It is the only node in the data domain cluster that makes updates to the index.

After processing updates, the leader publishes a new version of the data and notifies all follower nodes, alerting them to start using the updated version of the index. The follower nodes acquire read-only access to an updated version of the index.

Follower node

A **follower node** is a node in the data domain cluster responsible for processing queries. The follower node does not update the index. When you create a data domain cluster (that is not read-only), the Endeca Server software creates and starts the required number of the follower nodes after starting the leader node for the data domain. (In the read-only data domain cluster, all nodes are allocated follower node roles by the Endeca Server).

During the process of acquiring access to the recently updated index, follower nodes continue to serve queries. Each query is processed against a specific version of the index available to it at any given time. Query processing performance may slow down as follower nodes acquire read-only access to the updated index.

Cluster Coordinator

The Cluster Coordinator provides coordination services between cluster nodes. For detailed information about the tasks it performs, see [Role of the Cluster Coordinator on page 18](#).

Cluster diagrams

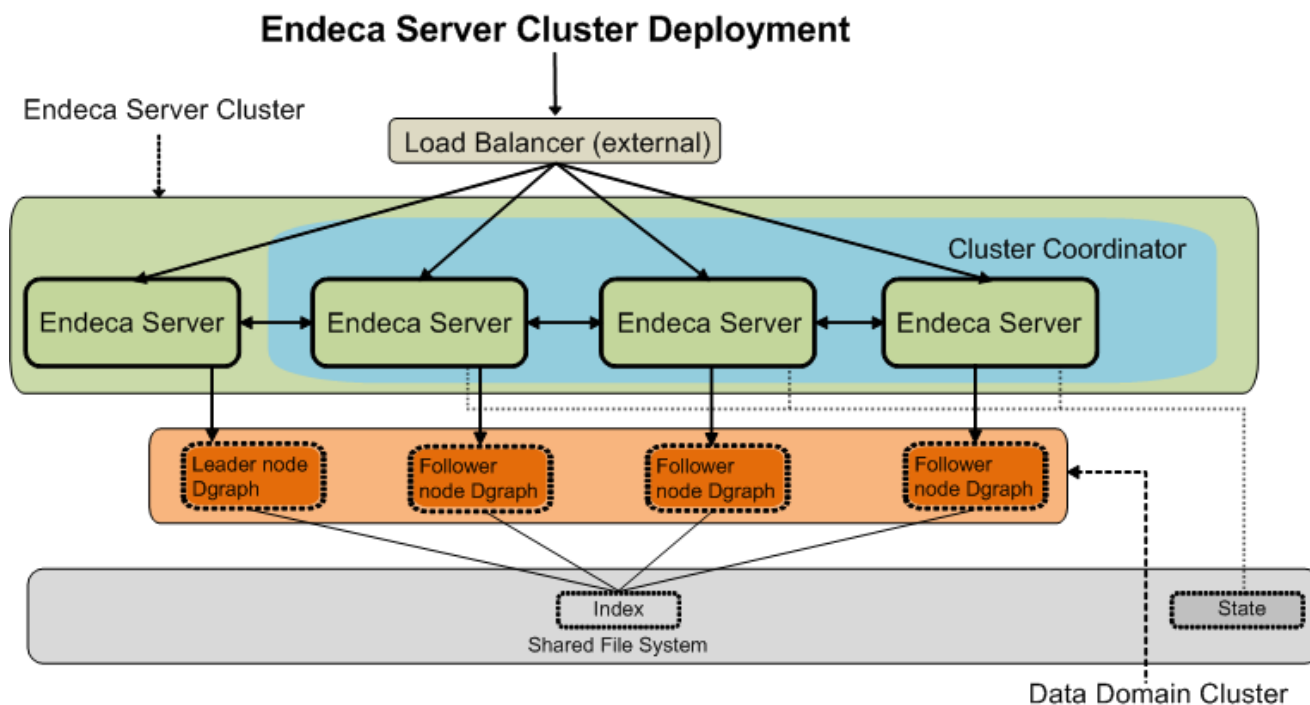
Many scenarios of data domain clusters are possible — a single Endeca Server cluster can host one or more data domains. This section includes diagrams that illustrate the cluster topology.

[Diagram with a single data domain](#)

[Diagram with multiple data domains](#)

Diagram with a single data domain

The diagram in this topic illustrates a clustered Endeca Server deployment serving a single data domain cluster.



In this diagram, starting from the top, the following components of the Endeca Server cluster deployment are included:

- The external load balancer serves as the single point of entry to the Endeca Server cluster. All requests to the single hosted data domain cluster go through this load balancer.



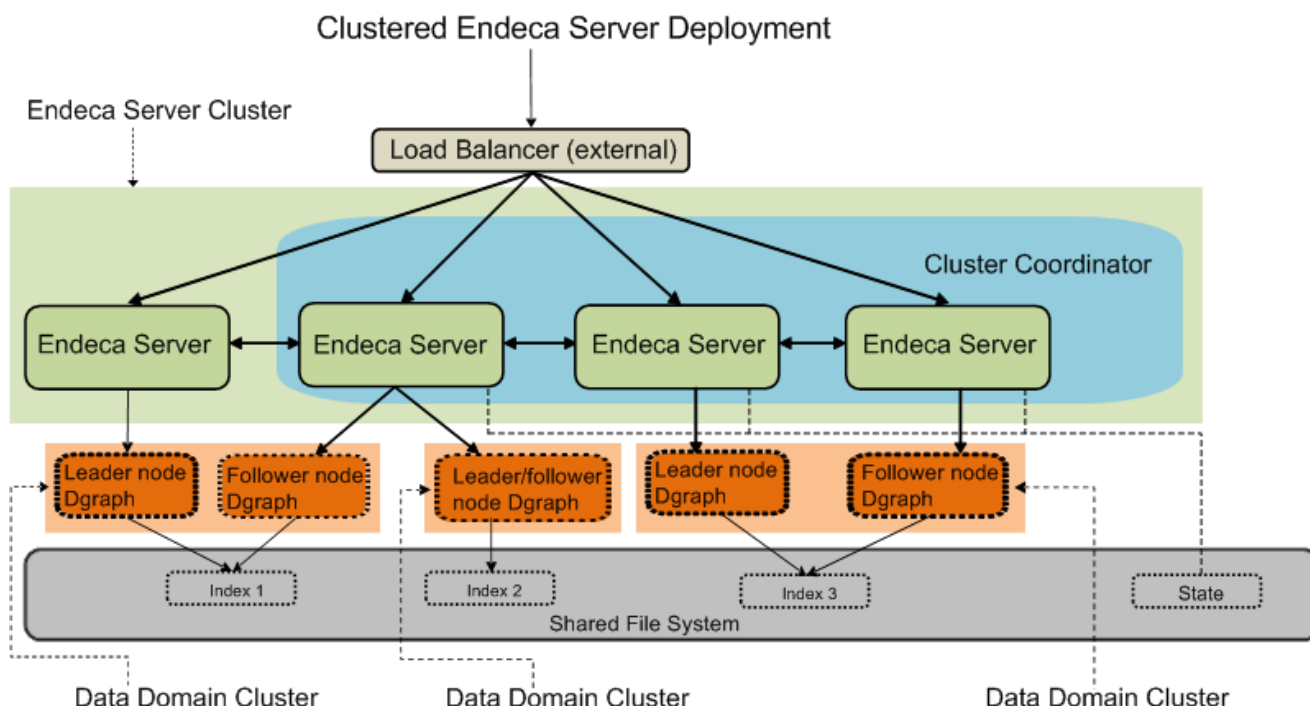
Note: Although it is recommended to use an external load balancer in your deployment, it is optional. For information, see [Load balancing and routing of requests on page 16](#).

- The Endeca Server cluster comprises a set of four Endeca Server nodes. All Endeca Server nodes in the cluster are aware of one another.
- Three out of four Endeca Server nodes are hosting the Cluster Coordinator instances. The Cluster Coordinator maintains a cluster state for all participating members of the Endeca Server cluster.
- A single data domain cluster consisting of four Dgraph nodes is hosted on the four Endeca Server nodes.
- Each node in this data domain cluster is represented by the Dgraph process running on each of the Endeca Server nodes. In this example, there is a one-to-one mapping between the data domain nodes and the Endeca Server nodes hosting them.
- In the data domain cluster that is not configured as read-only, exactly one node serves as the leader node, and the remaining nodes are follower nodes. For definitions of leader and follower nodes, see [Data domain cluster concepts on page 10](#).
- Finally, all nodes in the data domain cluster have write access to a shared file system on which the index for the data domain is stored. Only the leader Dgraph node in any data domain cluster is allowed to write

to the index located on the file system. Follower Dgraph nodes can only read from the index. On the diagram, the index resides on the shared file system. The cluster state is also maintained in the shared file system by the Cluster Coordinator.

Diagram with multiple data domains

The following diagram illustrates a clustered Endeca Server deployment serving multiple data domain clusters.



In this diagram, starting from the top, the following components of the Endeca Server cluster deployment are included:

- The external load balancer serves as the single point of entry to the Endeca Server cluster. All requests to all hosted data domain clusters go through this load balancer.



Note: Although it is recommended to use an external load balancer in your deployment, it is optional. For information, see [Load balancing and routing of requests on page 16](#).

- The Endeca Server cluster comprises a set of four Endeca Server nodes. All Endeca Server nodes in the cluster are aware of one another.
- Three out of four Endeca Server nodes are hosting the Cluster Coordinator instances. The Cluster Coordinator maintains a cluster state for all participating members of the Endeca Server cluster.
- Three data domain clusters are hosted on the Endeca Server nodes. (Depending on the size of the Endeca Server cluster, multiple data domain clusters can be hosted.)
- Starting from the left:
 - The first data domain cluster consists of two Dgraph nodes, one of which is the leader node. This data domain cluster uses "Index 1" stored on a shared file system.

- The second data domain cluster consists of one Dgraph node that serves both as the leader and follower node; this data domain cluster uses "Index 2" stored on a shared file system.
- Finally, the third data domain cluster has two Dgraph nodes, with one leader node and one follower node; this data domain cluster uses "Index 3", which is also stored on a shared file system.
- The following characteristics are shared by all data domain clusters in the Endeca Server cluster and are also reflected in this diagram:
 - Each Dgraph node is hosted on a different Endeca Server node — the allocation of Dgraph nodes to the Endeca Server nodes occurs automatically when you add data domains.
 - In each data domain cluster that is not read-only, exactly one node serves as the leader node, and the remaining nodes are follower nodes. (In a read-only data domain cluster, all Dgraph nodes serve as follower nodes.)
 - All Endeca Server nodes hosting all data domain nodes have access to a shared file system on which the indices for each data domain are stored. The cluster state is also maintained in the shared file system by the Cluster Coordinator.

Cluster behavior

This section discusses the Endeca Server cluster behavior, and the data domain cluster behavior in various scenarios, such as data domain cluster startup, updates to the data domain index, and response to a data domain node failure. It also describes how the Endeca Server ensures enhanced availability of its query processing for data domains.

[Allocation of data domains](#)

[Load balancing and routing of requests](#)

[Data domain cluster startup](#)

[How updates are processed](#)

[Role of the Cluster Coordinator](#)

[How enhanced availability is achieved](#)

Allocation of 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. This topic summarizes the configuration parameters that affect these calculations, so that you, as a system

administrator of the Endeca Server cluster can take them into account when configuring or administering your Endeca Server cluster.

Endeca Server capacity calculations

The Endeca Server uses the following parameters to calculate whether it has sufficient capacity to host (new or any additional) data domains:

- Requirements configured in the data domain profile:
 - **Oversubscribing.** You can allow Endeca Server to oversubscribe resources for a particular data domain while hosting other data domains. For detailed information on how this feature affects data domain allocation, see [How oversubscribing affects hardware utilization on page 44](#).
 - **Auto-idling.** You can allow Endeca Server to automatically idle a currently active data domain if it does not receive queries for the idling time period. If a data domain is set to automatically idle, this frees up resources on the Endeca Server nodes, and affects allocation of other data domains to Endeca Server nodes. For detailed information on how this feature affects data domain allocation, see [How idling affects data domain behavior on page 42](#).
- The available total capacity of the Endeca Server cluster at the time when a new data domain is being created and needs to be allocated. The term "total capacity" includes two aspects — the available total number of threads on each node, and the available amount of memory on each node that can be allocated to a particular data domain.

The available total capacity is calculated for these two aspects — memory and the number of threads, — as follows:

- **Memory calculations.** To determine whether the Endeca Server has sufficient amount of memory to host a data domain, the logic is as follows. When starting, the Endeca Server collects information about the total available RAM size and swap memory size. If a new data domain creation is requested, Endeca Server uses this information to decide which nodes should be assigned for the new data domain. Endeca Server only selects those nodes whose total virtual memory size is larger than the total memory footprint of all data domains already hosted on that node and the amount of memory needed for the new data domain. The memory footprint is estimated on each Endeca Server node based on this formula:

```
endeca-memory-to-index-size-ratio x indexes-size + computeCacheSizeMB
```

where:

- `endeca-memory-to-index-size-ratio` is the ratio of all virtual memory allocated for a data domain to the index size. The cluster administrator specifies this setting for each machine in the Endeca Server cluster, in the `EndecaServer.properties` file. The default ratio is 2.0; it is used if no other value is specified. For example, if the index size is 40MB, and the ratio is 2.0, the Endeca Server attempts to allocate 80MB of virtual memory to the data domain.
- `indexes-size` is the total calculated size of all indexes for all data domains currently hosted in the Endeca Server. This setting is calculated internally by the Endeca Server and is not reported to the administrator.
- `computeCacheSizeMB` is the amount of RAM, in MB, to allocate to the result cache for each Dgraph process of the data domain. The Endeca Server cluster administrator specifies this setting for the data domain profile, when creating it, by using `PutDataDomainProfile` of the Cluster Web Service, or `endeca-cmd put-dd-profile`. If you do not specify it, the default is used. The default of 0 is interpreted as follows. When an absolute value is 0, the default Dgraph cache size

is computed as 10% of the amount of RAM available on the Endeca Server node hosting the Dgraph node.

- **Threads calculations.** To determine whether the Endeca Server has sufficient number of processing threads to host a new (or additional) data domain, the logic is as follows. When starting, the Endeca Server collects information about the total number of used threads for its hosted data domains. If a new data domain creation is requested, Endeca Server creates it only if the total number of threads it has available is sufficient to accommodate both the new data domain and all already hosted data domains. The total number of threads is estimated on each Endeca Server node based on this formula:

```
numCpuCores x endeca-threads-allowed-per-core
```

where:

- `numCpuCores` is the setting specified in the Endeca Server node profile. (The node profile, once defined, is used on all nodes in the Endeca Server cluster.) This is the number of CPU cores allocated to each Endeca Server instance. When the Endeca Server application is deployed in the WebLogic Server, the number of CPU cores is determined automatically at startup. The default is the larger of 2, or the number of available CPU cores on the node. For example, if the number of all available CPU cores on the machine is 12, it is used by default as the `numCpuCores` in the node profile.
- `endeca-threads-allowed-per-core` is the setting specified in the `EndecaServer.properties` file. It is the number of threads allowed per core. The default is 1.0 and it is used if no other value is specified.

If the number obtained in this formula is larger than the "total number of threads used" — on all the Dgraphs and on all Endeca Server nodes, then the Endeca Server attempts to allocate threads to the new data domain.



Note: To calculate the "total number of threads used", the Endeca Server uses `numComputeThreads` from the data domain profiles. This is the setting specified in the data domain profile. It specifies the number of threads to allocate for processing requests on each Dgraph node serving a data domain using this profile. The number of threads should be equal to or greater than 4. The default is 4.

Here is another way to interpret this formula: If the user who wants to create a new data domain asks for more than the total number of threads required on the Endeca Server node so that it can host both the existing data domains and the new one, then allocation of that Endeca Server node to the data domain cannot occur, and the Endeca Server proceeds to look for other nodes in the cluster to allocate this data domain. If no nodes are found with enough threads to host the Dgraph processes for the new data domain, the data domain is not created.



Note: Data domain allocation strategy ignores disabled or idle data domains when calculating threads and memory footprint. Enabling a data domain and waking up an idle data domain operations succeed only if they meet all resources constraints.

- Additionally, if you have deployed Endeca Server on Linux 6, and enabled the configuration and use of cgroups in the Endeca Server, cgroups are used to guarantee that Endeca Server will not consume all of the machine's resources for its data domain Dgraph processes and will always remain reachable by the system administrator on that machine. If cgroup is enabled, Endeca Server uses the cgroup limit of total virtual memory instead of the total virtual memory of the machine to allocate data domains. For information on how cgroups are used and enabled, see the *Oracle Endeca Server Administrator's Guide*.

Endeca Server node allocation

Once the Endeca Server determines that it has sufficient capacity to host a data domain, it decides which Endeca Server nodes will host the particular Dgraph processes for the data domain.



Note: While this information is useful and visible to the Endeca Server cluster administrator, the allocation of data domain's nodes to Endeca Server nodes is not exposed to the data domain administrator. From the data domain administrator's point of view, the Endeca Server is hosting the data domain on its available nodes (making an effort to distribute the resource use as efficiently as possible), and the load balancing and routing of requests occurs automatically, to direct end-user queries to the Endeca Servers for processing, for each hosted data domain.

The Endeca Server uses the following principles for node allocation:

- The Dgraph nodes for a given data domain cluster are automatically hosted on different Endeca Server nodes. In other words, the Endeca Server software does not support hosting multiple Dgraph nodes for the same data domain on the same Endeca Server node.

For example, in a three-node Endeca Server cluster, you cannot create a data domain with five Dgraph nodes. For this data domain to be created, the number of Endeca Server nodes would need to also increase to five. The Manage Web Service of the Endeca Server returns a fault to any request that attempts to create or rescale a data domain beyond the number of Endeca Server nodes in the cluster.

- If more than one data domain cluster is hosted in the Endeca Server, then whenever possible, the leader nodes for each of the data domain clusters are hosted on different Endeca Server nodes.

Load balancing and routing of requests

This topic discusses the load balancing and routing of external requests targeting different data domains hosted by the Endeca Server cluster.

Load balancing of requests

Depending on your deployment strategy, to the external clients, the entry point of contact with the Endeca Server cluster could be either any Endeca Server node in the cluster, or an external load balancer.

The Endeca Server cluster relies on the following two levels of load balancing of requests:

1. Load balancing of requests across the nodes in the Endeca Server cluster. This task should be performed by an external load balancer, if you choose to use it in your deployment (it is not included in the Endeca Server package).

If an external load balancer is used, it receives all requests and distributes them across all of the nodes in the Endeca Server cluster. Once requests are received by any Endeca Server node, they are routed by the Endeca Server to the requested data domain.

If an external load balancer is not used, external requests can be sent to any node in the Endeca Server cluster. They will not be distributed across the nodes, but will be routed to the appropriate nodes serving the data domain, and then load-balanced between the data domain nodes.

2. Load balancing of requests across the nodes of a given data domain. This task is automatically handled by the Endeca Server cluster — it accepts requests on any node running an Endeca Server instance and provides load balancing of these requests across the nodes in the data domain.

Routing of requests

The Endeca Server cluster can host multiple data domain clusters. The allocation of data domains to specific Endeca Server nodes is not transparent to the clients issuing the requests. The Endeca Server automatically directs requests sent to any Endeca Server node to the subset of the cluster hosting the targeted data domain.

The following statements describe the behavior of the Endeca Server cluster for routing of requests:

- Requests can be submitted to any node in the Endeca Server cluster.
- The Endeca Server routes requests to the appropriate Endeca Server nodes hosting the data domain specified in the request.

For example, if the request is an updating request, such as a data loading request, or a configuration update, it will be routed to the leader node of the data domain cluster. If the request represents a non-updating (query processing) request, it will be routed to any of the follower nodes.

- Non-updating requests are load-balanced randomly across the data domain cluster. To support session affinity, you can optionally configure the data domain profile with a session ID parameter, URL, or cookie that will be used to re-route requests from the same session to the same node in the data domain.

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). For information on how to configure session affinity for an Endeca data domain, see the *Oracle Endeca Server Administrator's Guide*.

Data domain cluster startup

Once the Endeca Server cluster is started, it activates the Endeca data domain clusters it is hosting. This topic discusses the behavior of the data domain cluster at startup.

On startup, the following actions take place:

- Any data domain node is started in either a leader or follower mode and in any order. Any number of follower nodes and zero or one leader node are started in each data domain cluster.
- Once started, each data domain node registers with the Cluster Coordinator that manages the distributed state of the cluster. The leader node determines the current version of the index and informs the Cluster Coordinator.
- Follower nodes do not alter the index in any way; they continue answering queries based on the version of the index to which they have access at startup, even if the leader node is in the process of updating, merging, or deleting index versions on disk. Follower nodes do not receive updating requests. Follower nodes acquire access to the new index once the update completes. For information, see [How updates are processed on page 17](#).

How updates are processed

In a data domain cluster, updates to the records in the index and updates to the configuration are routed to the Oracle Endeca Server that is hosting the leader node.

The leader node processes the update and commits it to the on-disk index. Upon completion, the Cluster Coordinator informs all follower nodes that a new version of the index is available. The leader node and all

follower nodes can continue to use the previous version of the index to finish query processing that had started against that version.

As each node finishes processing queries on the previous version, it releases references to it. Once the follower nodes are notified of the new version, they acquire read-only access to it and start using it.

Interaction of outer transactions and updates

An outer transaction is a request to the Endeca Server that lets you include into it other requests, and process all of them as a single atomic operation. Using an outer transaction request is useful when running updates to the data domain index. To ensure that all updates either succeed or fail as a unit, it is recommended to wrap updates in the data domain cluster in an outer transaction request to the Endeca Server, although this is optional.

The following statements describe the behavior of an outer transaction request in the data domain cluster:

- A request to start an outer transaction can be sent to any of the nodes in the Endeca Server cluster. The Endeca Server routing service sends the request to the leader node in the data domain cluster, which processes it.
- If, while an outer transaction is in progress, the leader Dgraph node fails, the outer transaction request is not applied. This means that the data domain index is not changed in any way. In this case, the request logs for the data domain may contain messages about not finding an open transaction with the specified ID. This is because the transaction is no longer running in the data domain. It is recommended to restart the outer transaction once the Endeca Server cluster establishes a new Dgraph leader node for the data domain.

For more information about how outer transactions work, see the *Oracle Endeca Server Developer's Guide*.

Role of the Cluster Coordinator

The Cluster Coordinator provides configuration and state management and distributed coordination services to members of the Endeca Server cluster. It ensures high availability of the query processing provided by the nodes in the Endeca Server cluster and in the data domain clusters.

Any node in the Endeca Server cluster can optionally be configured to run the Cluster Coordinator. Since the availability of a clustered Endeca Server deployment depends on the running of the Cluster Coordinator services, it is highly recommended that an odd number (at least three) Endeca Server nodes are configured as Cluster Coordinators to avoid the Cluster Coordinator being a single point of failure.

The Cluster Coordinator service has the following characteristics:

- It is a shared information repository that provides a set of distributed coordination services. It ensures that all systems in the cluster coordinate their actions relative to all other systems running in your environment. If one of the nodes communicates any cluster information, all other nodes recognize it and react to it in a manner that ensures synchronization, event notification, and coordination between the nodes.

The communication and coordination mechanisms continue to work in the case when connections or data domain cluster nodes fail.

- It provides communication between the Endeca Server nodes, ensuring that if one of these nodes fails, requests are sent to other active nodes, until the node rejoins the Endeca Server cluster.
- It provides communication between Dgraphs in a data domain cluster. It prevents node isolation, ensures that updates from a cluster client are applied in the order that they were sent, guarantees that updates are either applied in full or not at all, and ensures that all data domain nodes have access to the most recent

version of the index after updates. It also controls the election of the Dgraph leader node in the data domain cluster, in case the current leader node fails. The newly-elected leader node identifies the most recent version of the index, and using the Cluster Coordinator, informs other nodes of the current version.

To summarize, in order to run, the Cluster Coordinator service requires a majority of its nodes to be active. Therefore, it is recommended that the Cluster Coordinator instance runs on an odd number (at least three) of the Endeca Server nodes in the deployed Endeca Server cluster. For information on deploying Cluster Coordinator services on a subset of Endeca Server nodes, see the *Oracle Endeca Server Installation Guide*.

How enhanced availability is achieved

This topic discusses how the Endeca Server cluster ensures enhanced availability of query-processing to the data domain clusters.



Important: The Endeca Server cluster provides enhanced availability, but does not provide high availability. This topic discusses the cluster behavior that enables enhanced availability, and notes instances where system administrators need to take action to restore services.

This topic discusses clusters of Endeca Server nodes that contain more than one running instance of the Endeca Server. Even though you can configure single-node data domains hosted by single Endeca Server instances, single Endeca Server instances can only be used in development environments, as they do not guarantee the availability of query processing to the data domain. Namely, in a single-node data domain hosted by a single Endeca Server instance, a failure of the Endeca Server node leads to the shutting down of the Dgraph process.

Availability of Endeca Server nodes

In an Endeca Server cluster with more than one Endeca Server instance, an ensemble of the Cluster Coordinator services running on a subset of nodes in the Endeca Server cluster ensures enhanced availability of the Endeca Server nodes in the Endeca Server cluster.

When an Endeca Server node in an Endeca Server cluster goes down, all Dgraph nodes hosted on it, and the Cluster Coordinator service (which may also be running on this node) also go down. As long as the Endeca Server cluster consists of more than one node, this does not disrupt the processing of non-updating user requests for the data domains. (It may negatively affect the Cluster Coordinator services. For information on this, see [Availability of Cluster Coordinator services on page 20](#).)

If an Endeca Server node fails, the Endeca Server cluster is notified and stops routing all requests to the data domain nodes hosted on that Endeca Server node, until you restart the Endeca Server node.

Let's consider an example that helps illustrate this case. Consider a three-node single data domain cluster hosted on the Endeca Server cluster consisting of three nodes, where each Endeca Server node hosts one Dgraph node for the data domain. In this case:

- If one Endeca Server node fails, incoming requests will be routed to the remaining nodes.
- If the Endeca Server node that fails happens to be the node that hosts the leader node for the data domain cluster, the Endeca Server cluster selects a new leader node for the data domain from the remaining Endeca Server nodes and routes subsequent requests accordingly. This ensures availability of the leader node for a data domain.
- If the Endeca Server node goes down, the data domain nodes (Dgraphs) it is hosting are not moved to another Endeca Server node. If your data domain has a total of two or three Dgraph nodes dedicated to processing queries, the data domain continues to function (because, if it had 3 Dgraphs, two Dgraphs

remain operational; if it had two Dgraphs, one Dgraph remains operational). If the data domain has a single Dgraph node, then if the Endeca Server node hosting it fails, query processing for this data domain stops until you restart the Endeca Server node.

When you restart the failed Endeca Server node, its processes are restarted by the Endeca Server cluster. Once the node rejoins the cluster, it will rejoin any data domain clusters for the data domains it hosts. Additionally, if the node hosts a Cluster Coordinator, it will also rejoin the ensemble of Cluster Coordinators.

Availability of data domain nodes

The ensemble of Cluster Coordinator services running on a subset of Endeca Server nodes in the cluster ensures the enhanced availability of the data domain cluster nodes and services:

- Failure of the leader node. When the leader node goes offline, the Endeca Server cluster elects a new leader node and starts sending updates to it. During this stage, follower nodes continue maintaining a consistent view of the data and answering queries. When the node that was the leader node is restarted and joins the cluster, it becomes one of the follower nodes. Note that it is also possible that the leader node is restarted and joins the cluster before the Endeca Server cluster needs to appoint a new leader node. In this case, the node continues to serve as the leader node.

If the leader node in the data domain changes, the Endeca Server continues routing those requests that require the leader node to the Endeca Server cluster node hosting the newly appointed leader node.



Note: If the leader node in the data domain cluster fails, and if an outer transaction has been in progress, the outer transaction is not applied and is automatically rolled back. In this case, a new outer transaction must be started. For information on outer transactions, see the section about the Transaction Web Service in the *Oracle Endeca Server Developer's Guide*.

- Failure of a follower node. When one of the follower nodes goes offline, the Endeca Server cluster starts routing requests to other available nodes, and attempts to restart the Dgraph process for this follower node. Once the follower node rejoins the cluster, the Endeca Server adjusts its routing information accordingly.

Availability of Cluster Coordinator services

The Cluster Coordinator services themselves must be highly available. The following statements describe the requirements in detail:

- Each Endeca Server node in the Endeca Server cluster can be optionally configured at deployment time to host a Cluster Coordinator instance. To ensure availability of the Cluster Coordinator service, it is recommended to deploy the Cluster Coordinator instances in a cluster of their own, known as an ensemble. At deployment time, it is recommended that a subset of the Endeca Server nodes is configured to host Cluster Coordinator services. As long as a majority of the ensemble is running, the Cluster Coordinator service is highly available and its services are used by the Endeca Server cluster and the data domain clusters hosted in it. Because the Cluster Coordinator requires a majority, it is best to start an odd number of its instances — this means that the Cluster Coordinator service must be started on at least three Endeca Server nodes in the Endeca Server cluster. An Endeca Server node that is configured to host a Cluster Coordinator assumes responsibility for ensuring the uptime of the Cluster Coordinator process it hosts — it will start the Cluster Coordinator service upon the start of the Endeca Server, and will restart it should it stop running.

To summarize, although the Cluster Coordinator can run on only one node, to ensure high availability of the Cluster Coordinator services, the Cluster Coordinator service must run on at least three nodes (or an

odd number of nodes that is greater than three) in any Endeca Server cluster. This prevents the Cluster Coordinator service itself from being a single point of failure. For information on deploying the Cluster Coordinator in the cluster, see the *Oracle Endeca Server Installation Guide*.

- If you do not configure at least three Endeca Server nodes to run the Cluster Coordinator service, the Cluster Coordinator service will be a single point of failure. Should the Cluster Coordinator service fail, access to the data domain clusters hosted in the Endeca Server cluster becomes read-only. This means that it is not possible to change the data domains in any way. You cannot create, resize, start, stop, or change data domains; you also cannot define data domain profiles. You can send read queries to the data domains and perform read operations with the Cluster and Manage Web Services, such as listing data domains or listing nodes. No updates, writes, or changes of any kind are possible while the Cluster Coordinator service in the Endeca Server cluster is down — this applies to both the Endeca Server cluster and data domain clusters. To recover from this situation, the Endeca Server instance that was running a failed Cluster Coordinator must be restarted or replaced (the action required depends on the nature of the failure).

Cluster Coordinator log

The Cluster Coordinator maintains a log named `coordinator.log`. By default, this log is stored in the `$DOMAIN_HOME/EndecaServer/logs` directory.

For more information (including how to change its log level), see the *Oracle Endeca Server Administrator's Guide*.



Chapter 2

Managing the Endeca Server Cluster

This section describes listing and deleting nodes in the Endeca Server cluster, obtaining the cluster status and version information, and listing and changing the hardware node profile.

[Listing Endeca Server nodes](#)

[Obtaining version information](#)

[About the node profile](#)

[Configuring a custom node profile](#)

[Monitoring node status](#)

[Monitoring cluster status](#)

[Deleting an Endeca Server node](#)

Listing Endeca Server nodes

To list the Endeca Server nodes, use the `endeca-cmd list-nodes` command. This command utilizes the `listNodes` request from the Cluster Web Service.

When you start any of the WebLogic servers on which the Endeca Server software is installed and deployed, the Endeca Server instances are started on these servers.



Note: You cannot add nodes to the Endeca Server cluster using any interfaces — the nodes are added automatically, once you start the WebLogic servers on which the Endeca Server instances are deployed.

To list Endeca Server cluster nodes:

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

For information on `endeca-cmd` and its location, see the *Endeca Server Administrator's Guide*.

Alternatively, access the Cluster Web Service of the Endeca Server cluster at the URL, similar to the following example: `https://my-endeca-server:7002/endeca-server/ws/cluster`.

In this example, `my-endeca-server` is the host name of the Endeca Server cluster, `7002` is the port, and `endeca-server` is the default root context in the WebLogic Server in which an SSL-enabled Endeca Server instance is running.

2. Issue the command as in this example:

```
endeca-cmd list-nodes
```

The command returns the protocol, host names, and ports of the Endeca Server instances running in your Endeca Server cluster.

Alternatively, you can issue a request with the `listNodes` operation of the Cluster Web Service.

Here is an abbreviated example of a response from `listNodes`, listing one Endeca Server node:

```
<ns3:nodeInfo>
  <ns2:protocol>https</ns2:protocol>
  <ns2:hostname>dev-node</ns2:hostname>
  <ns2:nodePort>7002</ns2:nodePort>
</ns3:nodeInfo>
```

Obtaining version information

You can use `endeca-cmd version`, or the `version` operation of the Manage Web Service, to list the version of the Oracle Endeca Server and the Dgraph processes for each of the enabled data domains.

The `version` command returns information about the Endeca Server software version, and the version of the Dgraph nodes in each of the enabled data domains hosted by this Endeca Server cluster.

To request a version:

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 at the URL, similar to the following example: `https://my-endeca-server:7002/endeca-server/ws/manage`.

2. Use one of these options:

- Issue this command:

```
endeca-cmd version
```

- Issue a request with the Manage Web Service, as in this abbreviated example:

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

About the node profile

The node profile for the Endeca Server defines the hardware characteristics that the Endeca Server cluster uses to create an Endeca Server node.

The **node profile** for the Endeca Server nodes defines:

- Number of CPU cores. This is the number of CPU cores allocated to each Endeca Server instance in the cluster. When the Endeca Server application is deployed in the WebLogic Server, the number of CPU cores is determined automatically at startup. The default is the larger of 2 or the number of available CPU cores on the node.
- The size of virtual memory in MB. The default is 1024 MB. This is the size of virtual memory allocated to each Endeca Server instance in the cluster.

Only one node profile can be configured in the system. It is used by all Endeca Server instances (nodes) in the Endeca Server cluster.

The hardware characteristics specified in the node profile have the following effect:

- When you initially deploy an Endeca Server instance in the WebLogic Server, the default node profile settings are used. In addition, depending on the choices you make during the installation process, the HTTPS protocol can be used for communication between the Endeca Server instances.
- Once the Endeca Server is deployed in a WebLogic Server, you can change the hardware node profile by running the command for changing it on any Endeca Server node. Once you change it, it overrides the default node profile, and all nodes in the Endeca Server cluster start using the new node profile.

Typically, administrators set the node profile to consume most if not all available hardware resources on the nodes. For example, if a machine has 12 CPUs, the node profile's `numCPUCores` will be set to 12 also. (The default is to use all available hardware resources on the node).

- Before changing the hardware node profile, it is useful to obtain the current node profile. Use the `endeca-cmd get-node-profile` command, or the `getNodeProfile` operation of the Cluster Web Service.



Important: It is not recommended to change the node profile after creating data domains, especially in cases when you reduce the number of CPU cores or the size of RAM available to the nodes in the Endeca Server cluster.

Configuring a custom node profile

When the Endeca Server instance is deployed in the WebLogic Server, it uses the hardware characteristics from the default node profile. You can override it with a custom node profile, by specifying different hardware characteristics in the node profile after the Endeca Server node is deployed, using `endeca-cmd set-node-profile`.

It is recommended to set the node profile for the Endeca Server nodes before any data domains are created. If the custom node profile is not set, the default node profile is used for data domains.



Important: All nodes use a single node profile, whether it is a default node profile, or the one you specify. In other words, once you change the node profile on any Endeca Server node, all nodes in the Endeca Server cluster start using this profile.

Before you configure a custom node profile, use `endeca-cmd get-node-profile` to request information about the default hardware node profile from the Endeca Server cluster.

To specify the hardware characteristics of the node profile:

1. On any Endeca Server node that is already deployed, access the `endeca-cmd` utility or the Cluster Web Service.

For information on `endeca-cmd` and its location, see the *Endeca Server Administrator's Guide*.

2. Use the command as in the following example:

```
endeca-cmd set-node-profile --num-cpu-cores 4 --ram-size 2048
```

Alternatively, you can use the `setNodeProfile` operation in the Cluster Web Service request:

```
<ns:setNodeProfile>
  <ns:nodeProfile>
    <ns1:numCpuCores>4</ns1:numCpuCores>
    <ns1:ramSizeMB>2048</ns1:ramSizeMB>
  </ns:nodeProfile>
```



```
</ns:setNodeProfile>
```

The hardware characteristics specified in this node profile will now be used for all of the Endeca Server instances running on managed nodes in a WebLogic domain.

After you have changed your Endeca Server node profile, the data domain profiles can be created. You can then use the data domain profiles to create data domain clusters hosted in the Endeca Server cluster.

Monitoring node status

Use the `endeca-cmd get-node-health node_hostname` command, or `getNodeHealth` operation in the Cluster Web Service request to obtain information about the Endeca Server node's status.

The `endeca-cmd get-node-health` operation lists the health information for the specified Endeca Server node in the cluster. It requires you to specify the node's hostname as the main argument.

To obtain information about the status of the Endeca Server node:

1. Issue the `endeca-cmd get-node-health` command as in the following example:

```
endeca-cmd get-node-health My_Node
```

Alternatively, you can issue the following request with the Cluster Web Service, as in the following abbreviated example:

```
<ns1:getNodeHealth xmlns:ns1="http://www.endeca.com/endeca-server/cluster/2/0">  
  <ns1:hostname>My_Node</ns1:hostname>  
</ns1:getNodeHealth>
```

The operation returns the Endeca Server node's host name, port, and status. Additionally, it lists the status of the Dgraph nodes hosted on this Endeca Server node (these Dgraph nodes belong to different data domains).

Monitoring cluster status

Use the `endeca-cmd get-cluster-health` command, or `getClusterHealth` operation in the Cluster Web Service request, to obtain information about the status of the Endeca Server cluster.

The `endeca-cmd get-cluster-health` operation lists the health information for the Endeca Server cluster. This operation does not require any arguments.

To obtain information about the status of the Endeca Server cluster:

1. Issue the `endeca-cmd get-cluster-health` command as in the following example:

```
endeca-cmd get-cluster-health
```

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

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

The command returns the Endeca Server nodes and their status, the host names and ports of the nodes running the Cluster Coordinator services and their status, and the names of data domains along with their status.

Deleting an Endeca Server node

To delete a node, use the `endeca-cmd delete-node node_hostname` command, or the `deleteNode` operation of the Cluster Web Service.

Deleting an Endeca Server node affects the Endeca Server cluster in the following ways:

- When you delete a node, you leave fewer nodes in the cluster for hosting data domain clusters. This may negatively affect the level of availability and the overall throughput of the Endeca Server cluster.
- If you delete one of those Endeca Server nodes on which the Cluster Coordinator service was running in the cluster, fewer active nodes will run the Cluster Coordinator services. This may negatively affect availability of the Cluster Coordinator services, which in turn will compromise availability of the Endeca Server cluster itself. For information, see [Role of the Cluster Coordinator on page 18](#).
- If you delete a node that is hosting any data domains, the Endeca Server cluster attempts to relocate data domains from this node to other nodes, before deleting the node. If it does not have sufficient resources to migrate data domains, the node is not deleted (it prevents you from deleting it).

Therefore, before removing an active Endeca Server node, ensure the following:

- A sufficient number of Endeca Server nodes continue hosting active data domain clusters.
To verify whether the Endeca Server node is hosting any Dgraph nodes, use the `endeca-cmd get-node-health` command, or the `getNodeHealth` request of the Cluster Web Service.
- A sufficient number of Endeca Server nodes continue running the Cluster Coordinator service. To guarantee high availability of Cluster Coordinator services, at least three Endeca Server nodes should be running the Cluster Coordinator service.

To verify the number of Cluster Coordinator services running in your cluster, use the `endeca-cmd get-cluster-health` command, or the `getClusterHealth` request of the Cluster Web Service.

To delete a node in the Endeca Server cluster:

1. Issue the command as in this example:

```
endeca-cmd delete-node my_node
```

where `my_node` is the host name of the node you want to delete.

The equivalent Web service request looks similar to the following example:

```
<ns1:deleteNode xmlns:ns1="http://www.endeca.com/endeca-server/cluster/2/0">
  <ns1:hostname>my_node</ns1:hostname>
</ns1:deleteNode>
```

If the node is not hosting any data domains, it is deleted from the Endeca Server cluster.

Additionally, if the node is hosting data domains and the Endeca Server cluster finds other nodes to which the data domains can be migrated, it moves the data domain nodes (Dgraph processes) to other Endeca Server nodes and deletes the data domain.



Note: This operation fails if the Endeca Server node does not have sufficient resources to re-allocate data domains hosted on the node you are attempting to delete.

2. In the WebLogic Server Administration Console, undeploy the Endeca Server application on that node.



Chapter 3

Managing Data Domain Profiles

This section describes how to create and manage profiles for data domains in the Endeca Server cluster. It also lists data domain profile parameters.

[*About the data domain profile*](#)

[*Deciding which data domain profile to use*](#)

[*Data Domain profile parameters*](#)

[*Default data domain profile*](#)

[*Defining a data domain profile*](#)

[*Listing data domain profiles*](#)

[*Removing a data domain profile*](#)

[*How idling affects data domain behavior*](#)

[*How oversubscribing affects hardware utilization*](#)

[*Example: data domain profile with low hardware usage*](#)

[*Example: data domain profile with high hardware usage*](#)

About the data domain profile

The **data domain profile** is a collection of characteristics that the Endeca Server uses to create a data domain. Before you can create a data domain, you must create a data domain profile (or use the default profile).

The profile serves as a template for the Endeca Server when it initially creates data domains and allocates resources to them, from the pool of provisioned Endeca Server instances.

A data domain profile defines the resources available to a data domain, such as the number of Dgraph nodes. It also defines the hardware utilization pattern the Endeca Server cluster will use when allocating its servers to the data domains using this profile.

A data domain profile is defined with the `putDataDomainProfile` operation of the Cluster Web Service, or using `endeca-cmd put-dd-profile`.

A data domain profile includes:

- A data domain's name and description
- 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 data domain should be auto-idled after a timeout
- Whether the Dgraph processes should be read-only
- Session affinity configuration
- Hardware characteristics of Dgraph processes — the number or required CPU threads and the cache size
- Optionally, the arguments passed to the Dgraph processes for this profile.

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

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

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

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

Deciding which data domain profile to use

You can define several data domain profiles, each serving a typical service use case. In addition, a default data domain profile exists.

Before you define a custom data domain profile, it is useful to know the following about the Endeca Server cluster and its data domains:

- **Determining query load distribution in the data domain.**

In the data domain profile, you can decide whether your data domain requires a dedicated leader node for handling updates, or the leader node that shares the regular query load with follower nodes.

For example, if the frequency and size of your updating requests are high, it is recommended to dedicate the leader node to process only the updating requests and not to process regular queries — this way the follower nodes process read query requests. On the contrary, if the index for the data domain is updated rarely, you can configure the leader node in the data domain to share the regular non-updating query load with other nodes, along with processing updates. The data domain profile allows you to specify this option.

The Dgraph nodes in the data domain cluster can process read queries and updating queries. Read queries can be processed by any Dgraph node (follower or leader node); they represent responses to Guided Navigation requests or search requests from the end users. Updating queries represent data updates to the index, changes in the records schema, or changes in the Dgraph configuration. Updating requests must be processed only by the leader node and cannot be processed by follower nodes.

- **Determining the desired number of query processing nodes in the data domain.** When creating data domain profiles, you can specify the number of follower nodes (these are the nodes that process only read queries, as opposed to the leader node that can process both queries and updates). The number of follower nodes you need depends on the usage patterns for the end users of the data domain. A data domain cluster may need more query processing Dgraph nodes if the number of end users is high, and they issue a high number of queries, often concurrently.

Keep in mind that you can only create a data domain with a certain number of Dgraph nodes if you have a sufficient number of Endeca Server instance nodes in the Endeca Server cluster. For each data domain, the Endeca Server creates only one Dgraph node on a specific Endeca Server instance. In other words, if you want to create a five-node data domain, the Endeca Server cluster hosting it should have at least five Endeca Server nodes.

- **Determining the allocation of processing hardware resources in the Endeca Server cluster.** When you create a new data domain, the Endeca Server cluster allocates the CPU resources from its servers to meet the needs of the data domain based on the configuration specified in the data domain profile for the number of threads required for each data domain node.

When defining a data domain profile, you can choose whether the Endeca Server cluster should use one of the following hardware utilization patterns:

- Dedicate 100% of its nodes capacity to one hosted data domain.
- Share its capacity with other data domains but remain within its total capacity.
- Is allowed to oversubscribe — it can start multiple Dgraph nodes (for different data domains) on its Endeca Server nodes, where the total number of CPU threads requested by data domains may exceed the total amount of CPU available to each Endeca Server node.

A data domain profile relies on the characteristics defined in the Endeca Server node profile. Such as, the node profile determines the potential limit on the number of dedicated data domains that could be hosted on the node (dedicated data domains are those for which the Endeca Server nodes dedicate 100% of their capacity).

- **Determining whether the data domain should be auto-idled.** When you create a data domain, you can specify whether the Endeca Server should automatically turn this data domain to idle after a specified timeout, if no queries are issued for it during the timeout period. This setting lets you limit the data domain proliferation, in self-service types of applications. For example, if many data domains are provisioned, but some of them are not used actively, they can be set to auto-idle, allowing the Endeca Server to stop allocating resources to them if these data domains are not used, and reactivate them once queries are again issued for them. This further conserves resources for the Endeca Server, allowing it to allocate resources more flexibly. Note that idling data domains is only supported for data domains with a single Dgraph process.
- **Determining whether the data domain should be read-only.** When defining a data domain profile, you can specify whether the data domain should be created as read-only. This is useful in the development environment or for demonstration purposes. For example, you can export an existing data domain 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.

- **Determining Dgraph process behavior.** In the data domain profile, you can optionally decide to specify configuration options for the Dgraph processes. If specified, these options will be used on all Dgraph processes started by the Endeca Server for this data domain.

You define these characteristics when configuring data domain profiles.

Data Domain profile parameters

The `endeca-cmd put-dd-profile` command, or the Cluster Web Service `putDataDomainProfile` operation, let you configure a data domain profile and its parameters. This topic lists these parameters and provides their descriptions.




Each data domain profile you create in the Oracle Endeca Server using the Cluster Web Service has the following parameters:



Note: The following table lists the parameters as they appear in the Cluster Web Service WSDL. Therefore, if you are using the Cluster Web Service requests directly, use this format. The `endeca-cmd` utility utilizes calls to the Cluster Web Service and therefore lets you specify the same characteristics with a slightly different format. For example, the `allowQueriesOnLeader` parameter from the Cluster Web Service is equivalent to the `endeca-cmd put-dd-profile --query-leader` command. Whenever possible, both formats are included in the table.

If you would like to use default values for any of the parameters, depending on the tool you use for sending web service requests, you may or may not need to specify the default values explicitly. For example, `soapUI` fills in the default if it is not specified, by looking it up in the web service's WSDL. However, other clients may not fill in the default if you do not specify the parameter and its value explicitly.

Parameter	Type	Description
<code>name</code>	string	Name of the data domain profile. The <code>endeca-cmd</code> command does not have an equivalent option; the name is specified to <code>endeca-cmd</code> on the command line.
<code>description</code>	string	Description of the data domain profile. The equivalent command using this parameter in <code>endeca-cmd</code> is: <pre>endeca-cmd put-dd-profile --description</pre> <p>If you use <code>endeca-cmd</code>, then if the description has spaces, it must be enclosed in double quotes.</p>

Parameter	Type	Description
allowQueriesOnLeader	boolean	<p>If set to <code>true</code>, indicates that the leader node should handle read (non-updating) queries in addition to handling updating queries. The default is <code>true</code>. If set to <code>false</code>, indicates that the leader node should be dedicated to handling updating queries only. The equivalent command using this parameter in <code>endeca-cmd</code> is:</p> <pre>endeca-cmd put-dd-profile --query-leader true</pre> <p> Note: Non-updating queries represent read requests to the index. Updating queries change the index or other configuration settings in the Dgraph process; they represent write requests to the index.</p>
numFollowers	integer	<p>Specifies the desired number of follower nodes to handle non-updating query load. The default is 0 (in this case, the data domain cluster consists of one leader node). If you set <code>allowQueriesOnLeader</code> to <code>false</code>, <code>numFollowers</code> must be equal to or greater than one.</p> <p> Note: Only one Dgraph node for each data domain could be hosted by each Endeca Server instance. Therefore, you must have a sufficient number of Endeca Server nodes before adding follower nodes to a data domain. For example, you cannot create a four-node data domain in an Endeca Server cluster with only three Endeca Server nodes.</p> <p>The equivalent command with this parameter is:</p> <pre>endeca-cmd put-dd-profile --num-followers 2</pre>
readOnly	boolean	<p>Indicates whether the Dgraph nodes in the data domain should be read-only. The default is <code>false</code>.</p> <p> Note: If you set <code>allowQueriesOnLeader</code> to <code>true</code>, <code>readOnly</code> must be <code>false</code>.</p> <p>The equivalent command with this parameter is:</p> <pre>endeca-cmd put-dd-profile --read-only false</pre>

Parameter	Type	Description
allowOversubscribe	boolean	<p>If set to <code>true</code>, the Endeca Server Cluster can exceed its capacity while sharing nodes hosting this data domain with other data domains. The default is <code>true</code>. The setting for this flag works together with <code>numComputeThreads</code>.</p> <p>When this setting is <code>true</code>, the data domain cluster nodes may still use less capacity than is available on the Endeca Server node. However, this flag allows the Dgraph nodes to compete for processing threads configured in the data domain profile, even if the Endeca Server node may not have these threads available. The thread allocation in this case is handled by the operating system. Allowing the node to oversubscribe can be useful if you are setting data domain profiles for the development environment and would like to conserve hardware resources.</p> <p>If a data domain cluster is allowed to oversubscribe, the Endeca Server cluster may host the nodes for this data domain on a node that does not have enough capacity. For example, an Endeca Server node with 12 CPUs can host 7 Dgraph nodes (for 7 different data domain clusters), each configured with <code>--num-compute-threads 2</code> and <code>--oversubscribe true</code>. The Endeca Server nodes in this data domain's configuration are considered "oversubscribed".</p> <p>If set to <code>false</code>, the nodes in the Endeca Server cluster hosting this data domain can share their resources only within their capacity. For example, if the Endeca Server node has 16 CPUs, and is hosting 2 Dgraph nodes each configured with 8 threads, it is sharing its capacity equally between two data domain nodes, utilizing 100% of its CPU, but not oversubscribing. Configuring the data domain profile that is not allowed to oversubscribe is useful when setting a data domain with high hardware-utilization requirements in a production environment. The equivalent command with this parameter is:</p> <pre>endeca-cmd put-dd-profile --oversubscribe true</pre>
numComputeThreads	integer	<p>The number of threads to allocate for processing requests on each Dgraph node serving a data domain using this profile. The number of threads should be equal to or greater than 4. The default is 4. The equivalent command with this parameter is:</p> <pre>endeca-cmd put-dd-profile ---num-compute-threads 4</pre>

Parameter	Type	Description
computeCacheSizeMB	integer	<p>The amount of RAM, in MB, to allocate to the result cache on each Dgraph node in the data domain. The default is 0, which is interpreted as follows:</p> <p>When an absolute value is 0, the default Dgraph cache size is computed as 10% of the amount of RAM available on the Endeca Server node hosting the Dgraph node. The equivalent command with this parameter is similar to this example that specifies 1MB as value of the Dgraph cache:</p> <pre>endeca-cmd put-dd-profile --compute-cache-size 1</pre>
startupTimeoutSeconds	integer	<p>The time for the Dgraph nodes to start up in the data domain. The default is 600 seconds. The equivalent command with this parameter is:</p> <pre>endeca-cmd put-dd-profile --startup-timeout <num_sec></pre>
shutdownTimeoutSeconds	integer	<p>The time for the Dgraph nodes to gracefully shut down. The default is 30 seconds. The equivalent command with this parameter is:</p> <pre>endeca-cmd put-dd-profile --shutdown-timeout <num_sec></pre>
enableAncestorCounts	boolean	<p>Optional. If enabled, computes counts for root managed attribute values and any intermediate managed attribute value selections. If the flag is not specified, or if it is specified, but the value is not specified, the default is <code>false</code>. In this case, 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. The equivalent command with this parameter is:</p> <pre>endeca-cmd put-dd-profile --ancestor-counts false</pre>
backlogTimeout	integer	<p>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. If the flag is specified, but the value is not specified, the default value is 0 seconds.</p>
minRefinementSamples	integer	<p>Optional. The minimum number of records to sample during refinement computation. If the flag is specified, but the value is not specified, 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. The equivalent command with this parameter is:</p> <pre>endeca-cmd put-dd-profile --refinement-sampling-min 0</pre>

Parameter	Type	Description
enableExactImplicits	boolean	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. The equivalent command with this parameter is: <pre>endeca-cmd put-dd-profile --implicit-exact false</pre>
numImplicitSamples	integer	Optional. Sets the maximum number of records to sample when computing implicit refinements (which are a performance tuning parameter). If the flag is specified, but the value is not specified, the default value is 1024. The equivalent command with this parameter is: <pre>endeca-cmd put-dd-profile --implicit-exact 1024</pre>
netTimeoutSeconds	integer	Optional. Specifies the maximum number of seconds the Dgraph waits for the client to download data from queries across the network. If the flag is specified, but the value is not specified, the default network timeout value is 30 seconds. The equivalent command with this parameter is: <pre>endeca-cmd put-dd-profile --net-timeout 30</pre>
maxSearchTerms	integer	Optional. Specifies the maximum number of terms for text search. If the flag is specified, but the value is not specified, the default value is 10. The equivalent command with this parameter is: <pre>endeca-cmd put-dd-profile --search-max 10</pre>
searchCharLimit	integer	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. The equivalent command with this parameter is: <pre>endeca-cmd put-dd-profile --search-char-limit 132</pre>
snippetCutoff	integer	Optional. Limits the number of words in an attribute that the Dgraph evaluates to identify the snippet. If a match is not found within <i><num></i> words, the Dgraph does not return a snippet, even if a match occurs later in the attribute value. If the flag is specified, but <i><num></i> is not specified, the default is 500. The equivalent command with this parameter is: <pre>endeca-cmd put-dd-profile --snippet-cutoff 500</pre>

Parameter	Type	Description
<code>disableSnippets</code>	boolean	Optional. Globally disables snippeting. The default is <code>false</code> , meaning snippeting is enabled. The equivalent command with this parameter is: <pre>endeca-cmd put-dd-profile --snippet-disable false</pre>
<code>enableAllDynamicsMAs</code>	boolean	Optional. Enable all available dynamic attribute value characteristics. If the flag is specified, but the value is not specified, the default value is <code>false</code> . Note that this option has performance implications and is not intended for production use. The equivalent command with this parameter is: <pre>endeca-cmd put-dd-profile --dynamic-category-enable false</pre>
<code>disableContraction</code>	boolean	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. If the flag is specified, but the value is not specified, the default is <code>false</code> . 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. The equivalent command with this parameter is: <pre>endeca-cmd put-dd-profile --contraction-disable false</pre>
<code>maxWildcards</code>	integer	Optional. Specifies the maximum number of terms that can match a wildcard term in a wildcard query that contains punctuation, such as <code>ab*c.def*</code> . If the flag is specified, but the value is not specified, the default value is <code>100</code> . The equivalent command with this parameter is: <pre>endeca-cmd put-dd-profile --wildcard-max 100</pre>

Parameter	Type	Description
<code>sessionIdType</code>	string	<p>Specifies the method you would like to use for handling session affinity, when routing requests to this data domain. The default method is <code>header</code> — the HTTP header is used for session affinity if you don't specify <code>sessionIdType</code> in the data domain profile.</p> <p>Available options are:</p> <ul style="list-style-type: none"> • <code>HEADER</code>. The HTTP headers will be used for session affinity. • <code>PARAMETER</code>. The URL parameters will be used. • <code>COOKIE</code>. Cookies will be used. <p>The values for these options are not case-sensitive. The equivalent <code>endeca-cmd</code> command with this parameter is:</p> <pre>endeca-cmd put-dd-profile --session-id-type header</pre> <p>This example specifies <code>header</code> as the session ID key.</p>
<code>sessionIdKey</code>	string	<p>Specifies the name of the object to be checked, for establishing session affinity through one of the methods specified with <code>sessionIdType</code>. The value of <code>sessionIdKey</code> can be any string that is allowed to be used as an HTTP header, a URL parameter, or a cookie name. The default is <code>X-Endeca-Session-ID</code>. If you don't specify the value for <code>sessionIdKey</code>, it is used either as a header, a URL parameter, or a cookie. The equivalent <code>endeca-cmd</code> command with this parameter is:</p> <pre>endeca-cmd put-dd-profile --session-id-key X-Endeca-Session-ID</pre>
<code>autoIdle</code>	boolean	<p>Indicates whether to idle a data domain after a timeout if no queries are issued for this data domain during the timeout period. The default is <code>false</code>. If set to <code>false</code>, the data domain is never made idle. If set to <code>true</code>, the data domain will be made idle after a timeout specified in the <code>idleTimeoutMinutes</code> setting of the data domain profile. When the data domain is made idle, the Endeca Server stops its Dgraph processes, and stops allocating resources to them. However, if end-users issue a query to such a data domain, the data domain is activated and its Dgraph processes are restarted by the Endeca Server. The equivalent command with this parameter is:</p> <pre>endeca-cmd put-dd-profile --auto-idle false</pre>

Parameter	Type	Description
idleTimeoutMinutes	string	<p>(Optional). Indicates the timeout to use for idling, in minutes. The default is 15 minutes, it is used if you do not specify the timeout. 15 minutes is the lowest idle timeout value you can set. This timeout allows for long-running queries to complete successfully before the data domain is turned idle. If you have any queries running longer than 15 minutes, increase the idle timeout.</p> <p>If a data domain is set to auto-idle and for the specified timeout period no queries arrive, it turns idle, and its Dgraph processes are stopped gracefully. If queries arrive, the data domain is activated. The timeout starts for a data domain that is set to auto-idle as soon as it is created and enabled. The timer is reset each time the data domain receives a query (and its Dgraph process is restarted) after being idle. The equivalent command with this parameter is:</p> <pre>endeca-cmd put-dd-profile --auto-idle true --idle-timeout 30</pre>
args	string	<p>List of flags to specify to the Dgraph processes serving a data domain using this profile. If not specified, no Dgraph flags are used. The equivalent command with this parameter is:</p> <pre>endeca-cmd put-dd-profile --args <list_of_args></pre> <p>To obtain a list of flags, use: <code>endeca-cmd put-dd-profile --args --usage</code>. For a list of Dgraph flags, see the <i>Oracle Endeca Server Administrator's Guide</i>.</p>

Default data domain profile

This topic lists parameters for the default data domain profile used for single-node data domain deployments hosted by a single Endeca Server instance.

The Endeca Server cluster uses the default data domain profile to support faster deployments of simplified data domains in a development environment. The default data domain profile lets you create a data domain in a single-node Endeca Server instance. If you start a data domain using the default profile, you can later expand the Endeca Server cluster by adding Endeca Server instances and adding a shared file system. You can then export this data domain (thus saving its index), and import it into an Endeca Server cluster, while adjusting its other characteristics, to ensure you can run it in the production environment. For information on installing and deploying an Endeca Server cluster, see the *Oracle Endeca Server Installation Guide*.

To list the details of the default data domain profile, use the `listDataDomainProfile` operation from the Cluster Web Service, or the equivalent `endeca-cmd get-dd-profile` command.



Note: This profile always exists in the system and cannot be deleted with the `deleteDataDomainProfile` operation, or with the equivalent `endeca-cmd delete-dd-profile` command.

The following table lists the default data domain profile parameters as they are returned by the Cluster Web Service. The `endeca-cmd get-dd-profile` command returns equivalent parameters. The command has no options, just the profile name as the argument (`default`).

Parameter	Type	Default
<code>name</code>	string	The name is <code>default</code> .
<code>description</code>	string	The default system data domain profile.
<code>allowQueriesOnLeader</code>	boolean	The default is <code>true</code> . The leader node should handle non-updating query load in addition to handling updating queries.
<code>numFollowers</code>	integer	<p>The default is 0. This profile creates a data domain with only one node that handles all types of requests (updating and non-updating requests).</p> <p>The number is 0, because the only node in this data domain is a single leader node. This node is created automatically, unlike follower nodes whose number may vary.</p>
<code>readOnly</code>	boolean	Indicates whether the Dgraph node is read-only (and thus can only process read-only requests to the data domain's index). The default is <code>false</code> .
<code>allowOversubscribe</code>	boolean	The default is <code>true</code> . This means that each Dgraph node in the data domain will request its configured number of processing threads from the Endeca Server nodes even if the hosting Endeca Server nodes may not have sufficient CPU.
<code>numComputeThreads</code>	integer	The default is 4. The number of threads allocated to each Dgraph process (known as the Dgraph node) serving the data domain.
<code>computeCacheSizeMB</code>	integer	<p>The default is 0 MB. The amount of RAM to allocate to the result cache on each Dgraph node in the data domain.</p> <p>The default of 0 is interpreted as follows. When an absolute value is 0, the default Dgraph cache size is computed as 10% of the amount of RAM available on the Endeca Server node hosting the Dgraph node.</p>
<code>startupTimeoutSeconds</code>	integer	The default is 600 seconds. The time, in seconds, it takes the Dgraph process to start up.
<code>shutdownTimeoutSeconds</code>	integer	The default is 30 seconds. The time, in seconds, it takes the Dgraph process to shut down.
<code>sessionIdType</code>	string	The default is <code>header</code> . The method used for establishing session affinity when routing requests to the appropriate data domain hosted by the Endeca Server cluster.

Parameter	Type	Default
sessionIdKey	string	The default is X-Endeca-Session-ID. The name of the object that is checked by the session affinity method.
autoIdle	boolean	The default is <code>false</code> . This profile creates a data domain that is not allowed to auto-idle.
idleTimeoutMinutes	string	The default is 15 minutes. This is the default timeout period, that is used if a data domain is allowed to auto-idle, and the custom timeout is not specified. This default value does not apply to the default data domain profile (because this profile does not allow a data domain to auto-idle).
args	string	No flags are specified for the Dgraph in the default data domain profile.

Example of the default data domain profile

If you issue the request `getDataDomainProfile` using the Cluster Web Service at the URL similar to the following: `http://host:7001/endeca-server/ws/cluster`, and specify `default` as the profile's name, the default data domain profile is returned in the response.

This example of the response omits web service namespaces. This example also assumes that the WebLogic Server uses the default port 7001, that `host` is the hostname of the Endeca Server, and `endeca-server` is the default root context of the Endeca Server Java application running in the WebLogic Server container:

```
<dataDomainProfile>
  <name>default</name>
  <description>The default system data domain profile</description>
  <allowQueriesOnLeader>true</allowQueriesOnLeader>
  <numFollowers>0</numFollowers>
  <readOnly>false</readOnly>
  <allowOversubscribe>true</allowOversubscribe>
  <numComputeThreads>4</numComputeThreads>
  <computeCacheSizeMB>0</computeCacheSizeMB>
  <startupTimeoutSeconds>600</startupTimeoutSeconds>
  <shutdownTimeoutSeconds>30</shutdownTimeoutSeconds>
  <sessionIdType>header</sessionIdType>
  <sessionIdKey>X-Endeca-Session-ID</sessionIdKey>
  <autoIdle>false</autoIdle>
  <idleTimeoutMinutes>15</idleTimeoutMinutes>
</dataDomainProfile>
```

Defining a data domain profile

To define a new data domain profile in the Endeca Server, use the `endeca-cmd put-dd-profile` command, or the `putDataDomainProfile` request of the Cluster Web Service. The domain profile must exist before you create data domains, otherwise, the default data domain profile is used.

A **data domain profile** defines the resources available to a data domain. It serves as a template for the Endeca Server when it creates data domains on provisioned Endeca Server nodes in the Endeca Server cluster.

A data domain profile is used by the Endeca Server only for the initial data domain creation. When you create a data domain in the Endeca Server cluster, it is configured based on either a default data domain profile or a custom data domain profile (if you specify it). After the data domain has been created, the profile is not associated with this data domain. That is, you can later change the characteristics of the data domain. You can also delete this data domain profile without affecting any data domains.



Note: Be careful when creating a data domain profile to ensure the Endeca Server has enough capacity to host a data domain created based on this profile. For example, the setting for `compute-cache-size` in the data domain profile is larger than the `ramSizeMB` that you specified in the node profile for the instances of Endeca Server nodes, then no data domain using this profile can be created.

To define a data domain profile:

1. Use the `endeca-cmd put-dd-profile` command, specifying the domain profile parameters. For information, see [Data Domain profile parameters on page 30](#).

For example, the following command specifies the data domain profile named `MyProfile`:

```
endeca-cmd put-dd-profile MyProfile
--description "To be used in development environment"
--query-leader true
--read-only false
--num-followers 2
--oversubscribe true
--num-compute-threads 4
--compute-cache-size 4
--startup-timeout 600
--shutdown-timeout 30
--session-id-type header
--session-id-key X-Endeca-Session-ID
--autoIdle true
--idleTimeoutMinutes 30
```

Alternatively, you can specify the data domain profile with a request to the Cluster Web Service, using the `putDataDomainProfile` operation, specifying the values in the request.

After creating this new profile, you can create data domains with it.

Listing data domain profiles

Custom data domain profiles are created by an administrator in charge of the Endeca Server cluster. Before you create a data domain, you need to know the names and characteristics of previously defined data domain profiles. This information is returned with the operations from the Manage Web Service and Cluster Web Service, or with `endeca-cmd`.



Note: Existing data domain profiles do not necessarily reflect the characteristics of existing data domains that used these profiles. The data domains could have been changed after they were created.

To obtain a list of one or more data domain profiles:

1. Use one of these options either from `endeca-cmd`, or the Cluster or Manage Web Services:

Option	Description
<code>endeca-cmd get-dd-profile name</code> , or <code>getDataDomainProfile</code> from the Manage Web Service	Lists the details of a single specified data domain profile.
<code>endeca-cmd list-dd-profiles --verbose</code> , or <code>listDataDomainProfiles</code> from the Manage Web Service	Lists all defined data domain profiles and their characteristics.



Note: The Cluster Web Service also has the same set of operations, `listDataDomainProfiles` and `getDataDomainProfiles`; both return the same information as the equivalent operations from the Manage Web Service. Each web service is intended for different users — administrators managing data domains use the Manage Web Service, while administrators managing the Endeca Server cluster use the Cluster Web Service. Having these operations in both interfaces allows each user to have a full list of features for managing either their data domains, or Endeca Server clusters.

Here is an abbreviated example of a `listDataDomainProfiles` request to the Manage Web Service:

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

The request returns the set of data domain profiles configured in the Endeca Server cluster. For each data domain profile, it returns its name, description, and other characteristics, such as the number of follower nodes, the number of query processing threads, and the list of arguments that are sent to the Dgraph processes for this data domain profile.

A `getDataDomainProfile` request is similar, but requires you to specify the name of the data domain profile.

Here is an example of a response from `endeca-cmd get-dd-profile MyTestProfile`. It lists the details of the profile named `MyTestProfile`:

```
MyTestProfile
Description: test_profile_with_0_followers_readonly_false
AllowQueriesOnLeader: true
AllowOverSubscribe: true
NumFollowers: 0
NumComputeThreads: 4
read-only: false
computeCacheSizeMB: 0
startupTimeoutSeconds: 60
shutdownTimeoutSeconds: 60
session-id-type: header
session-id-key: X-Endeca-Session-ID
autoIdle: false
IdleTimeoutMinutes: 30
args: []
```

Removing a data domain profile

You remove a data domain profile with `endeca-cmd delete-dd-profile`, or with the `deleteDataDomainProfile` operation of the Cluster Web Service.

Removing the data domain profile does not remove the data domains created using this profile. When the data domain profile is removed, its name can no longer be used to create new data domains. However, you can create a new data domain profile with that name, and use it for the creation of data domains.

To delete a data domain profile:

1. Use `endeca-cmd delete-dd-profile`, specifying a profile name.

Alternatively, you can delete a data domain profile with the `deleteDataDomainProfile` operation of the Cluster Web Service, as shown in this example:

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

How idling affects data domain behavior

You can create a data domain profile that automatically turns a data domain to idle if it receives no queries for the specified timeout period. Automatic idling of data domains is useful as it helps control the proliferation of data domains in the Endeca Server deployment, thus improving hardware utilization by the Endeca Server. This topic discusses how Endeca Server manages automatic idling of data domains.

A data domain is **idle** when the Dgraph process is stopped, on the Endeca Server node hosting this data domain. A data domain that is idle continues to be registered with the Endeca Server, however the Endeca Server stops allocating resources (memory and processing threads), to this data domain. An idle data domain is activated automatically if it receives a query. In other words, once an end user of a front-end application powered by the Endeca Server, (such as Studio), issues a request, an idle data domain is activated, and its Dgraph process is restarted automatically.

Limitations for the idling of data domains are as follows:

- Auto-idling is guaranteed on a data domain that has a single Dgraph process, and is ignored for data domains deployed in a cluster of more than one node. Specifically, if you create a data domain based on the data domain profile with `autoIdle` set to `true` in a cluster of more than one Endeca Server nodes, the data domain can be created successfully, but the `autoIdle` setting is ignored. Similarly, if you use a specific data domain profile with `autoIdle` set to `true` for a data domain that has one Dgraph node, and then rescale the data domain by adding nodes, the `autoIdle` setting is ignored by the Endeca Server.
- Enabling a data domain and activating an idle data domain operations succeed only if they meet all resources constraints. In other words, if a data domain is idle and a query is issued for it (which would normally activate it), but other data domains were configured and enabled in the Endeca Server in the meantime, it is possible that the Endeca Server may not have sufficient resources to activate this data domain. In this case, the Endeca Server issues an error and the data domain remains idle until you manually free up space in the Endeca Server by disabling other data domains.

A data domain profile contains these settings (in the Manage Web Service) for idling of a data domain:

- `autoIdle` indicates whether to turn a data domain to idle after a timeout, by stopping its Dgraph processes. If set to `false`, Endeca Server never makes this data domain idle. If set to `true`, the data

domain is made idle after the optional timeout specified in the `idleTimeoutMinutes`, if during this timeout the data domain does not receive queries. The default for `autoIdle` is `false`.

- `idleTimeoutMinutes` indicates the timeout to use for idling, in minutes. The timeout is counted from the time a data domain has been created and enabled. This parameter is optional. If not specified, the default of 15 minutes is used. 15 minutes is the lowest idle timeout value you can set. This timeout allows for long-running queries to complete successfully before the data domain is turned idle. If you have any queries running longer than 15 minutes, increase the idle timeout. To change the idling timeout for a data domain, disable a data domain, create a new data domain profile with new timeout, update the data domain to use the new data domain profile, and enable the data domain.

The following statements describe which operations activate (or do not activate) an idle data domain:

- Queries from the Conversation Web Service, Configuration Web Service, and Data Ingest Web Service activate an idle data domain and restart the timeout timer.
- Queries from Administration and Configuration URL commands also activate and restart the Dgraph processes for the idle data domain (and restart the timer), with the exception of `http://<host>:<port>/endeca-server/admin/<data_domain>?op=ping`, which does not activate an idle data domain.
- Bulk load interaction. The process of loading data through the Bulk Ingest interface starts with the `allocateBulkLoadPort` operation from the Manage Web Service, which activates an idle data domain. If a bulk ingest operation is issued and is in progress for the data domain that is set to `auto-idle`, it is possible that bulk load may take longer than the idling timeout. If the timeout expires while bulk data load is in progress, the Endeca Server refreshes the idling timeout and the data domain does not shut down, so that bulk load can complete successfully.
- The `endeca-cmd` commands do not activate an idle data domain with the exception of `endeca-cmd allocate-bulk-load-port`, which activates an idle data domain.



Note: It is important to distinguish an "idle data domain" from a "data domain that is configured to auto-idle": an idle data domain is the one that is already idle (its processes are stopped); while a data domain configured to auto-idle is the one that uses the data domain profile with `autoIdle` set to `true`. It can be active but set to auto-idle after a timeout, or, it can already be idle, if it received no queries during the timeout.

If a data domain is configured to auto-idle, stopping and restarting its Dgraph process is handled automatically by the Endeca Server, without requiring you to call these operations explicitly, as you would do when manually starting and stopping a data domain.

The following statements describe how an idle data domain behaves in various situations:

- If a data domain is set to automatically idle, but is currently active, and you disable and then enable it, it will be active.
- If a data domain is set to automatically idle, but is currently active and an outer transaction is open, it will not turn to idle until the transaction closes or is canceled (even if the auto-idling timeout expires and no queries are sent to the data domain during the timeout).
- If a data domain is idle and you disable it, it will be idle when you re-enable it, until a query arrives that activates the data domain by restarting its Dgraph process.
- If a data domain is idle, its idling timer is stopped. Once the data domain is activated, the clock for the timeout is restarted.

- Updating a data domain to use a new data domain profile does not interfere with idling, because updating is done on a disabled data domain.
- Cloning, exporting, and importing an idle data domain behaves in the same way as performing these operations for a disabled data domain.
- If you idle a data domain, and then restart the WebLogic Server hosting the Endeca Server, a data domain will continue to be idle after the Endeca Server restarts.

How oversubscribing affects hardware utilization

Data domain profiles define the hardware resources that the Endeca Server cluster makes available to the data domains it hosts. This topic explores the interaction between oversubscribing of data domains and hardware utilization.

Data domain profiles are defined by an administrator to control the resources allocated to data domains, and provide a mechanism to offer different levels of service for data domains hosted by the Endeca Server.

The following examples, along with their supporting diagrams, explore how data domain profiles relate to the utilization of CPU resources on the nodes of an Endeca Server cluster.



Note: The examples in this topic explore the data domain's profile settings for `oversubscribe` and `num-compute-threads`, but do not take into account the setting for `autoIdle`. If `autoIdle` is set to `true`, it also affects the fluctuating resources in the Endeca Server, and interferes with oversubscribing behavior. For more information, see [How idling affects data domain behavior on page 42](#).

CPU utilization by data domains without oversubscribing

The following diagram illustrates several possible ways to configure a data domain profile that is configured so that it does not oversubscribe CPU resources.

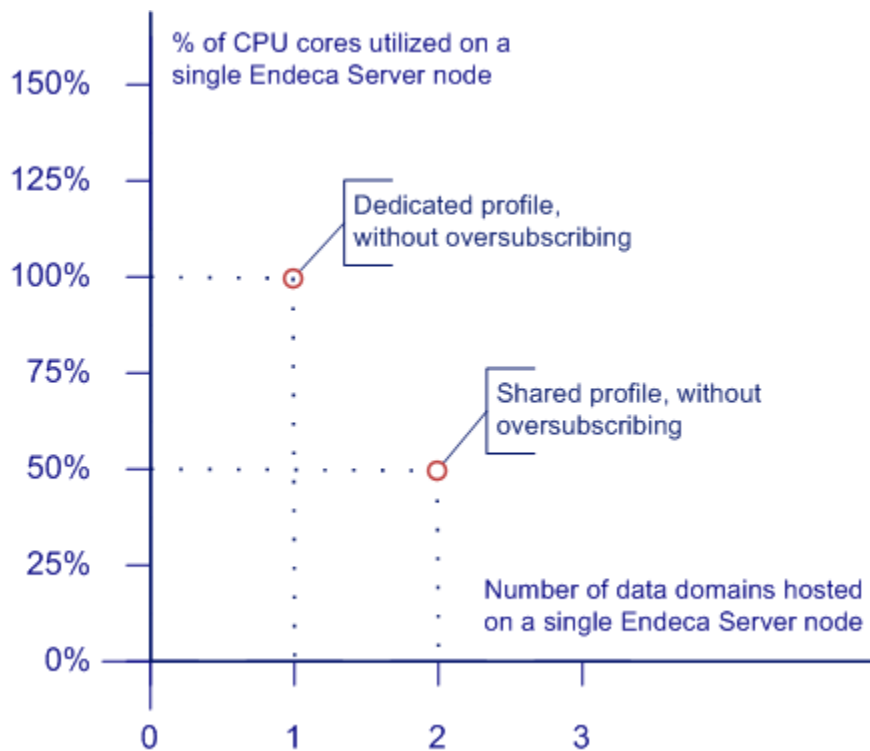
The fact that a data domain profile does not allow oversubscribing means that the Endeca Server cluster will allocate hardware resources to such a profile based on what is specified in the profile, but only if the Endeca Server nodes have these resources.

The following options of `endeca-cmd put-dd-profile` control the parameters for not allowing oversubscribing and for hardware utilization:

- `--oversubscribe false`.
- `--num-compute-threads <num_threads>`.

In addition to options from `endeca-cmd`, you can use equivalent parameters from the `putDataDomainProfile` operation of the Cluster Web Service — `allowOversubscribe` and `numComputeThreads`.

The diagram shows the CPU utilization of a single Endeca Server node as a function of the number of domains it hosts.



In this diagram, two use cases are described:

- **A dedicated profile, without oversubscribing.** This profile may be applicable for data domains that will support production applications. In the diagram, this type of usage is illustrated by a single data domain hosted on a single Endeca Server node that allocates 100% of its CPU capacity to this data domain.
- **A shared profile, without oversubscribing.** This profile divides an Endeca Server machine's CPUs between data domains. In the diagram, this type of usage is illustrated by two data domains each utilizing 25% of CPU capacity on each Endeca Server node (in this case, the total CPU usage is 50%).

CPU utilization by data domains with oversubscribing

The following examples demonstrate profiles that allow Endeca Server to host domains on nodes that may be oversubscribed.

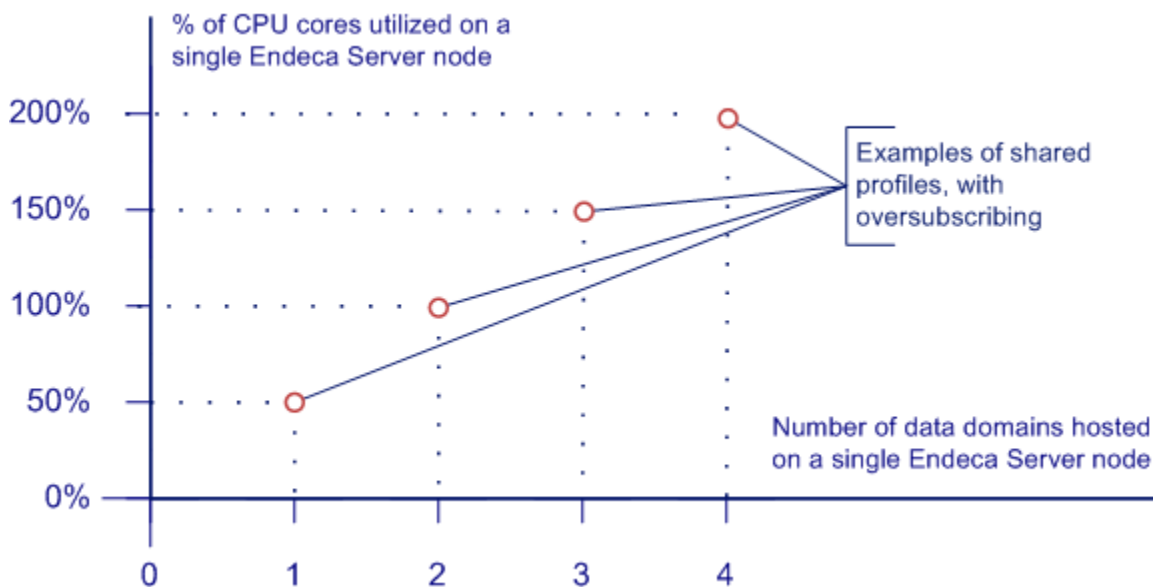
A profile that allows oversubscribing means that the Endeca Server will allocate the resources specified in the profile but may choose to host the domains on nodes where those resources are already allocated to other domains. In other words, an oversubscribed profile instructs the Endeca Server cluster to allow domains to be hosted on nodes where the resources allocated to domains exceed the actual available resources. For example, the Endeca Server may host ten 4-CPU Dgraphs on a 12 CPU machine.

When an Endeca Server node is oversubscribed, the domains it hosts will compete for the available resources. Managing the competing demands for resources is delegated to the underlying operating system.

The following options of `endeca-cmd put-dd-profile` control the parameters for allowing oversubscribing and for hardware utilization (equivalent parameters exist in the Cluster Web Service):

- `--oversubscribe true`.
- `--num-compute-threads <num_threads>`.

The following diagram illustrates four examples of shared oversubscribed data domain profiles:



In this diagram, from left to right, the following data domain profiles are described:

- The first scenario shows a single shared data domain, where a data domain hosted on the Endeca Server node takes 50% of its CPU. Even if the Endeca Server node may not have enough CPU, the data domain will be allocated the required CPU because its profile is configured to allow oversubscribing. In this example, since there is only a single domain hosted on the node, the node itself is not yet oversubscribed.
- The second scenario shows two shared data domains, where each Endeca Server node hosts two Dgraph nodes (each from a different data domain), configured so that they allow oversubscribing. In this example, the Endeca Server node is not oversubscribed but is fully subscribed, since the two domains together are using 100% of the CPU resources.
- The next scenario shows three shared data domains, where all data domains combined can take up to 150% of the CPU on the Endeca Server node. In this example, the node is oversubscribed, since 150% of its CPU resources are allocated to the three domains it hosts.
- The last scenario shows four shared data domains, where all data domains combined can take up to 200% of the CPU on the Endeca Server node. In this example, the Endeca Server node is also oversubscribed.

Oversubscribed profiles may be useful for domains where strict performance guarantees are not required, such as development or testing scenarios.

Example: data domain profile with low hardware usage

This example illustrates the characteristics of the data domain profile that defines data domain clusters with low hardware usage requirements.

The following example ensures the following:

- The Endeca Server machines hosting the data domain cluster are oversubscribed, meaning they are set to share their CPU capacity with more than one data domain cluster hosted on these machines, even if the total capacity required by these data domains exceeds the amount of CPU on each Endeca Server node. This behavior is defined by the `allowOversubscribe` element in the data domain profile definition that must be set to `true`.
- The data domain cluster's leader node shares the query load with the other nodes and is not dedicated to only processing updating requests. This behavior is defined by the `allowQueriesOnLeader` element in the data domain profile definition that must be set to `true`.

Example

This example illustrates the data domain profile for data domain clusters with low hardware usage requirements. Because it represents a snippet of a full web service request, the example does not list the namespace of the Cluster Web Service, or the operation used to define the profile.

The example specifies the number of follower nodes as 1. This number is arbitrary — you can modify it according to the needs of your data domain and the overall capacity of nodes in the Endeca Server cluster.

The important parts of this example are the settings for `allowQueriesOnLeader` and `allowOversubscribe`, which are both set to `true`:

```
<dataDomainProfile>
  <name>low hardware-usage profile</name>
  <description>Profile for a data domain with low hardware usage</description>
  <allowQueriesOnLeader>true</allowQueriesOnLeader>
  <numFollowers>1</numFollowers>
  <readOnly>false</readOnly>
  <allowOversubscribe>true</allowOversubscribe>
  <numComputeThreads>4</numComputeThreads>
  <computeCacheSizeMB>1</computeCacheSizeMB>
  <startupTimeoutSeconds>600</startupTimeoutSeconds>
  <shutdownTimeoutSeconds>30</shutdownTimeoutSeconds>
  <sessionIdType>header</sessionIdType>
  <sessionIdKey>X-Endeca-Session-ID</sessionIdKey>
</dataDomainProfile>
```

Example: data domain profile with high hardware usage

This example illustrates the characteristics of a data domain profile that defines a production-grade data domain cluster. In such a cluster, each Dgraph node runs on a dedicated Endeca Server node, and a dedicated leader node handles only updating requests.

The data domain profile is defined with the `dataDomainProfile` complex type of the Cluster Web Service. This complex type is used when you specify a data domain profile to the Endeca Server, using the `putDataDomainProfile` operation of the Cluster Web Service. You can also use `endeca-cmd put-dd-profile`.

The following example of a production-grade data domain configuration ensures the following:

- The Endeca Server machines host only that data domain, dedicate 100% of their CPU capacity to the data domain's nodes, and are not allowed to oversubscribe. This is accomplished by setting the value of `numComputeThreads` in the profile to the total number of CPU cores available on each Endeca Server node and setting the value of `allowOversubscribe` to `false`.
- The follower nodes in the data domain cluster are dedicated to processing only non-updating, end-user query requests. This ensures that the leader node is dedicated to processing updating requests to the data and to the configuration. This behavior is defined by setting the `allowQueriesOnLeader` element in the data domain profile definition to `false`.

Example

This example illustrates the production-grade data domain profile. Because it represents a snippet of a full web service request, the example does not list the namespace of the Cluster Web Service or the operation used to define the profile.

The example specifies the number of Dgraph nodes as 10 and the number of threads requested for each Dgraph node as 4. These numbers are arbitrary — you can modify them according to the needs of your data domain cluster and the overall capacity of nodes in the Endeca Server cluster. In this example, these numbers imply that the data domain will be hosted by ten Endeca Server nodes, each of which has at least 4 CPUs.

The important parts of this example are the settings for `allowQueriesOnLeader` and `allowOversubscribe`, which are both set to `false`, and the setting for `numComputeThreads`, which is equal to the number of CPUs on each Endeca Server node:

```
<dataDomainProfile>
  <name>high hardware-usage profile</name>
  <description>Production data domain profile</description>
  <allowQueriesOnLeader>false</allowQueriesOnLeader>
  <numFollowers>10</numFollowers>
  <readOnly>false</readOnly>
  <allowOversubscribe>false</allowOversubscribe>
  <numComputeThreads>4</numComputeThreads>
  <computeCacheSizeMB>1</computeCacheSizeMB>
  <startupTimeoutSeconds>600</startupTimeoutSeconds>
  <shutdownTimeoutSeconds>30</shutdownTimeoutSeconds>
  <sessionIdType>header</sessionIdType>
  <sessionIdKey>X-Endeca-Session-ID</sessionIdKey>
</dataDomainProfile>
```




Chapter 4

Using the Cluster Web Service

This section is a reference of all operations available in the Cluster Web Service of the Oracle Endeca Server.

[About the Cluster Web Service](#)

[Accessing the Cluster Web Service](#)

[List of operations in the Cluster Web Service](#)

About the Cluster Web Service

The Cluster Web Service lets system administrators define node profiles in the Endeca Server cluster, and list and delete Endeca Server nodes. It also contains operations for obtaining the cluster health and the data domain health, and for defining and managing data domain profiles.

In the Endeca Server cluster, individual nodes are represented by WebLogic Managed Servers, each of which is running an Endeca Server application. These Endeca Server nodes can host multiple data domains.

The Cluster Web Service is intended for those system administrators who will be installing and deploying the Endeca Server software on a homogenous set of hardware nodes, and setting up the Endeca Server cluster.

Using the Cluster Web Service, you can:

- Define the hardware node profile for the Endeca Server nodes, and then add Endeca Server nodes to an Endeca Server cluster that would use the predefined node profile.
- Define domain profiles — these profiles define the types of data domains that could be added to the Endeca Server cluster.

After you have defined Endeca Server nodes and the data domain profiles with the Cluster Web Service, you use the Manage Web Service to add the data domains based on the defined domain profiles.



Note: Both the Cluster and the Manage Web Services have options in `endeca-cmd` that utilize their operations. Instead of using either of these two Web services, you can use `endeca-cmd`.

Accessing the Cluster Web Service

The Cluster Web Service is declared in `cluster.wsdl`.

To access the WSDL of the Cluster Web Service at the following URL:

```
http://host:<port>/endeca-server/ws/cluster/<dataDomain>?wsdl
```

where the `host` and `port` are the host and port of any Endeca Server node in the Endeca Server cluster, and `dataDomain` is the name of the Endeca data domain hosted by the Endeca Server cluster.

To issue requests using the Cluster Web Service, use this URL:

```
http://host:port/endeca-server/ws/cluster
```

Requests can be issued in any tool that allows you to send web service requests, such as soapUI.

About the web service version

The namespace for the Cluster Web Service is similar to the following example and reflects the version of the service:

```
xmlns:v2_0="http://www.endeca.com/endeca-server/cluster/2/0"
```

This namespace is included in the WSDL document for the web service.

In this example, the string 2/0 indicates the version is 2.0, where 1 is the major version, and 0 is the minor version. The version in the service that you have installed may not match this example.



Important: After you upgrade the Oracle Endeca Server, verify the versions of the web services you have been using against the installed versions, to avoid version mismatch. It is recommended to use the web service versions that match the ones installed with the Oracle Endeca Server.

For more information on web service versions, see the *Oracle Endeca Server Developer's Guide*.

List of operations in the Cluster Web Service

The Cluster Web Service contains operations for creating the node profile and managing nodes in the Endeca Server cluster. It also contains operations for defining and managing data domain profiles, and for checking the status of the cluster and data domains.

Operation description

Use the Cluster Web Service to:

- Define a node profile. The node profile provides the underlying hardware characteristics for the Endeca Server nodes. The Endeca Server cluster uses the node profile at deployment time for all its nodes, to allocate hardware to nodes on which the data domains will reside.
- Define data domain profiles. Data domain profiles are templates for different kinds of data domains; before you add data domains to the Endeca Server cluster, data domain profiles must be created.

After you install and deploy Endeca Server nodes, you can configure Endeca Server node profiles and data domain profiles. Next, use the Manage Web Service to add and manage data domains that will be hosted by the Endeca Server nodes and will use defined data domain profiles.

Request

The input to the Cluster Web Service depends on the element in its request. For example:

- To configure a node profile, use the `setNodeProfile` element, specifying the number of CPUs and the amount of available RAM.
- To list Endeca Server nodes, use the `listNodes` element (no parameters are required for this operation).

Other operations are discussed in the list of operations in this topic.

Response

Depending on the success of the request, the Cluster Web Service returns:

- If there are no problems and the response was successful, an empty response element is returned (except for `listNodes` and `listDataDomainProfiles` operations, which return information).
- The request returns a SOAP fault, in the instances including the following: a data domain with the specified name already exists, the data domain profile is not defined in the Endeca Server cluster, or the Endeca Server cluster has insufficient resources to accommodate the request.

List of operations

The Cluster Web Service lets you perform the following operations, where operations are represented by elements in the request:

Request element	Description
<code>setNodeProfile</code>	<p>Define the custom node profile. The custom node profile lets you set the hardware characteristics for all Endeca Server cluster nodes, specifying the number of CPUs and the amount of available RAM. If the custom node profile does not exist, the default node profile is used by the Endeca Server cluster when it adds new nodes. You can use the <code>setNodeProfile</code> operation at any time when deploying new Endeca Server nodes in the cluster, for example, after the first Endeca Server node has been deployed based on the default node profile.</p> <p>Once you set a custom node profile, when you deploy new Endeca Server nodes in the cluster, this profile will be used for all Endeca Server nodes, including the current node.</p> <p>It is recommended to configure the custom node profile before creating data domain profiles and data domains (this is because they rely on the capacity of the Endeca Server cluster nodes determined by the node profile). Also, if you change the node profile after the data domains have been created, this may affect the data domains.</p> <p>To define a custom node profile, specify:</p> <ul style="list-style-type: none"> • <code>numCpuCores</code> this is the number of threads that will be configured on each Endeca Server machine. The default is the number of cores multiplied by the number of Endeca Server threads allowed per core, <code>endeca-threads-allowed-per-core</code>, from <code>EndecaServer.properties</code>. (The default for <code>endeca-threads-allowed-per-core</code> is 1.0. • <code>ramSizeMB</code> this is the amount of virtual memory you are requesting for the Endeca Server application on this machine. Based on this system, the system provides the larger of the amount you specify and the amount the system has available.
<code>getNodeProfile</code>	<p>Return the hardware profile of nodes in the Endeca Server cluster as set by <code>setNodeProfile</code>. The request returns the number of CPUs and the amount of available RAM. This operation does not have any arguments.</p>

Request element	Description
deleteNode	<p>Remove a node from the Endeca Server cluster, specifying its <code>hostname</code>.</p> <p>If the node you are attempting to delete is hosting data domains, the Endeca Server cluster attempts to migrate the data domains to other nodes, if the cluster has enough resources for this operation.</p> <p>The request returns a fault if the specified node does not exist in the cluster, or if the data domains hosted on the node cannot be migrated to other nodes due to insufficient resources in the Endeca Server cluster.</p>
listNodes	<p>Return a list containing the <code>hostname</code> and <code>port</code> of each node in the cluster. This operation does not require any arguments.</p>

Request element	Description
putDataDomainProfile	<p>Configure a new data domain profile, which defines resources available to a data domain when it is created. Specify these values:</p> <ul style="list-style-type: none"> • <code>name</code> and <code>description</code> are the name and description of the data domain profile. • <code>allowQueriesOnLeader</code> indicates whether the leader node should share query requests with follower nodes (if set to <code>true</code>), or should be dedicated exclusively to handling update queries (if set to <code>false</code>). The default is <code>true</code>. • <code>numFollowers</code> is the number of follower Dgraph nodes to handle query load. The default is 0. If queries on leader node are not allowed, the <code>numFollowers</code> must be equal to or greater than one. • <code>readOnly</code> indicates whether the data domain should process only read requests to its index. The default is <code>false</code>. • <code>allowOversubscribe</code> indicates whether the Endeca Server cluster can exceed its capacity while sharing nodes hosting this data domain with other data domains (if set to <code>true</code>), or should share its resources only within its capacity (if set to <code>false</code>). The default is <code>true</code>. • <code>numComputeThreads</code> is the number of threads to allocate for processing requests on each Dgraph node in the data domain using this profile. If no value is specified or 0 is specified, this defaults to the larger of 4 or the number of available processors. • <code>computeCacheSizeMB</code> is the amount of RAM, in MB, to allocate to the result cache on each Dgraph node. If no value is specified or 0 is specified, this defaults to the Dgraph default cache size computed as 10% of the amount of RAM available on the Endeca Server node hosting the Dgraph node. • <code>autoIdle</code> indicates whether to automatically turn this data domain idle if it does not receive queries during the idle timeout. If set to <code>false</code>, the data domain is never turned idle. If set to <code>true</code>, the data domain turns idle after the idle timeout expires. The default is <code>false</code>. • <code>idleTimeoutMinutes</code> (Optional). Is the time, in minutes, after which a data domain configured to auto-idle is turned idle if it does not receive queries. If not specified, the default of 15 minutes is used. This parameter does not affect data domains that are not allowed to auto-idle. If you specify a value less than 15 minutes, Endeca Server issues a warning and resets this timeout to 15 minutes, as this is the lowest timeout value you can set.

Request element	Description
putDataDomainProfile (continued)	<p>Additional values for putDataDomainProfile are:</p> <ul style="list-style-type: none"> • <code>startupTimeoutSeconds</code> is the time, in seconds, to allow for the Dgraph process to start up. The default is 600 seconds. • <code>shutdownTimeoutSeconds</code> is the time, in seconds, to allow for the Dgraph process to shut down. The default is 30 seconds. • <code>sessionIdType</code> is the method used for establishing session affinity when routing requests to the data domain. The default is <code>header</code>. The value is not case-sensitive. • <code>sessionIdKey</code> is the name of the object checked by the <code>sessionIdType</code> session affinity method. The default is <code>X-Endeca-Session-ID</code>. • The Dgraph flags: <code>enableAncestorCounts</code>, <code>backlogTimeout</code>, <code>minRefinementSamples</code>, <code>enableExactImplicits</code>, <code>numImplicitSamples</code>, <code>netTimeoutSeconds</code>, <code>maxSearchTerms</code>, <code>snippetCutoff</code>, <code>searchCharLimit</code>, <code>disableSnippets</code>, <code>enableAllDynamicsMAs</code>, <code>disableContraction</code>, <code>maxWildcards</code>. For descriptions, see Data Domain profile parameters on page 30. • <code>args</code> (Optional). The list of additional Dgraph flags. For a list, see the <i>Oracle Endeca Server Administrator's Guide</i>.
getDataDomainProfile	Return the data domain profile with the specified name. The request returns the same information as specified with <code>putDataDomainProfile</code> .
deleteDataDomainProfile	Remove a specified data domain profile from the Endeca Server.
listDataDomainProfiles	Return a list of all defined data domain profiles containing the same information as specified with <code>putDataDomainProfile</code> .
getClusterHealth	Return the Endeca Server nodes and their status, the host names and ports of the nodes running the Cluster Coordinator and their status, and the data domains and their status.
getDataDomainHealth	Return the health of the data domain with the specified name, and list the status of the Dgraph nodes in the data domain.
getNodeHealth	<p>Return the Endeca Server node's host name and port 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).</p> <p>Requires you to specify the host name of the node.</p>
usage	Return a list of flags for the Dgraph process. These flags could be used as values in <code>args</code> element of the <code>putDataDomainProfile</code> operation.

Example of a request to the Cluster Web Service

The following examples show the Cluster Web Service request and response for the `getDataDomainHealth` operation.

This abbreviated example shows the request for the health of the data domain `test2`:

```
<ns1:getDataDomainHealth xmlns:ns1="http://www.endeca.com/endeca-server/cluster/2/0">
  <ns1:name>test2</ns1:name>
</ns1:getDataDomainHealth>
```



Note: This documentation is complemented by the API reference. For reference information about the elements and types used in the Cluster Web Service, see the Cluster Web Service API Reference in the *Oracle Endeca Server API References*.



Chapter 5

Using the Manage Web Service

This section provides a reference of all operations available in the Manage Web Service of the Oracle Endeca Server.

[About the Manage Web Service](#)

[Accessing the Manage Web Service](#)

[List of operations in the Manage Web Service](#)

About the Manage Web Service

The Manage Web Service lets you create new data domains in the Endeca Server cluster and manage existing data domains.

The Manage Web Service is intended for those system administrators who will be creating an Endeca data domain on an already deployed Endeca Server cluster, utilizing the node profile and the data domain profiles provided by the Endeca Server cluster administrator.

Using the Manage Web Service, you can:

- Create a new data domain. In this case, the Oracle Endeca Server cluster starts a set of Dgraph processes, each serving this data domain.
- Clone, attach, detach, enable, export, import, rescale, update, or delete one or more existing data domains, and warm up the Dgraph cache for the specified data domain.
- List or check the status of one or more data domains.

The operations from the Manage Web Service can be performed directly by sending requests with this web service through a web services tool such as soapUI. In addition, you can run these operations through the Endeca Server command-line interface, `endeca-cmd`. This guide describes how to use both the Manage Web Service requests directly, and the commands from `endeca-cmd`. For a full list of `endeca-cmd` commands, see the *Oracle Endeca Server Administrator's Guide*.



Note: You can add data domains after the Oracle Endeca Server nodes are already running in the Endeca Server cluster. Additionally, each new data domain you create utilizes an Endeca Server node profile and a data domain profile that must already be created with the Cluster Web Service, or with the `endeca-cmd` utility.

Namespaces and schema

The Manage Web Service utilizes the following namespace specific to it (among other namespaces common between the Endeca Server web services):

```
xmlns:v1="http://www.endeca.com/endeca-server/manage/2"
```


The version indicated in this example may not correspond with the version of this web service that you are using. For information on web service versioning, see the *Oracle Endeca Server Developer's Guide*.

The Manage Web Services utilizes the schema named `types.xsd`.

For API-generated documentation on the Manage Web Service and its schema, see the *Oracle Endeca Server API References*.

Accessing the Manage Web Service

The Manage Web Service is declared in `manage.wsdl`.

You can access the WSDL of the Manage Web Service at the following URL:

```
http://<host>:<port>/endeca-server/ws/manage/?wsdl
```

where the `host` and `port` are the host and port of any Endeca Server node in the Endeca Server cluster, and `endeca-server` is the default root context of the Endeca Server Java application deployed in the WebLogic Server.

To issue requests using the Manage Web Service, use this URL:

```
http://host:port/endeca-server/ws/manage
```

Requests can be issued in any tool that allows you to send Web service requests, such as soapUI.

About the web service version

The namespace for the Manage Web Service is similar to the following example and reflects the version of the service:

```
xmlns:v2_0="http://www.endeca.com/endeca-server/manage/2/0"
```

This namespace is included in the WSDL document for the web service.

In this example, the string `2/0` indicates the version is 2.0, where 2 is the major version, and 0 is the minor version. The version in the service that you have installed may not match this example.



Important: After you upgrade the Oracle Endeca Server, verify the versions of the web services you have been using against the installed versions, to avoid version mismatch. It is recommended to use the web service versions that match the ones installed with the Oracle Endeca Server.

For more information on web service versions, see the *Oracle Endeca Server Developer's Guide*.

List of operations in the Manage Web Service

The Manage Web Service contains operations for creating new data domains and managing existing ones, in the context of the Endeca Server cluster hosting multiple data domain clusters.

Operation description

The Manage Web Service lets you issue requests to create new or manage existing data domains.

Before you use the Manage Web Service, deploy the Endeca Server nodes in the WebLogic Server container, and use the Cluster Web Service to create data domain profiles.

Each data domain you create with the Manage Web Service has these characteristics:

- It resides on a subset of Endeca Server nodes that themselves belong to an Endeca Server cluster.
- It is associated with an existing data domain profile. The domain profiles determine the resources available to each of the data domain clusters once you create them. (You can change the data domain's characteristics after it has been created.)

Request

The input to the Manage Web Service depends on the element in its request. For example:

- To create a new data domain, use the `createDataDomain` element, specifying its name and the data domain profile name.
- To create a new data domain using index files that already exist, use the `importDataDomain` element, specifying its name, the data domain profile name, and the label for the previously exported index files.

Other operations are discussed in the list of operations in this topic.

Response


Depending on the success of the request, the Manage Web Service returns:


- If there are no problems and the response was successful, an empty response element is returned (except for `listDataDomains` and `listDataDomainProfiles` operations, which return information).
- A SOAP fault indicating an Endeca Server error, in the instances including the following: a data domain with the specified name already exists, the data domain profile is not defined in the Endeca Server cluster, the Endeca Server cluster has insufficient resources to accommodate the request, or the data domain profile is configured with characteristics that exceed the current hardware resources of the Endeca Server cluster.


List of operations

The Manage Web Service lets you perform the following operations, where operations are represented by elements in the request:

Request element	Description
<code>allocateBulkLoadPort</code>	<p>Return a host name for the leader node and the port used for Bulk Load Interface, for a data domain with the specified <i>name</i>.</p> <p>This is a read-write operation — if the current leader 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.</p> <p>If a data domain is idle, this operation (and the equivalent <code>endeca-cmd allocate-bulk-load-port</code>) activates the data domain by restarting its Dgraph process.</p>

Request element	Description
<code>createDataDomain</code>	<p>Create a new data domain in the Endeca Server cluster, specifying its name, data domain profile name, and (optionally) whether it is enabled. The values are:</p> <ul style="list-style-type: none"> • <code>name</code> is the data domain name. • <code>ddProfileName</code> is the name of a data domain profile to use. • <code>enabled</code> (Optional). If set to <code>true</code> the data domain is enabled. This is the default. If set to <code>false</code>, the data domain is disabled.
<code>deleteDataDomain</code>	<p>Delete a data domain from the system in its entirety removing all associated resources, which include source records, configuration, logs, and index files for this data domain. No backup of any sort is performed. When a data domain is deleted, it cannot be restored.</p> <p>To delete a data domain, specify its <code>name</code> in this element.</p>
<code>cloneDataDomain</code>	<p>Create a new data domain by making a copy of a specified existing data domain.</p> <p>To clone a data domain, specify:</p> <ul style="list-style-type: none"> • <code>name</code> is the name of the new data domain you will be creating. • <code>sourceName</code> is the name of the source data domain from which you will be copying. • <code>enabled</code> (Optional). Whether the new data domain should be enabled. If you don't specify the <code>enabled</code> value, the new data domain is enabled by default.
<code>rescaleDataDomain</code>	<p>Resize a specified data domain by adding the number of follower nodes to the data domain cluster. To rescale a data domain, specify its <code>name</code>, and <code>numFollowers</code>, which is the desired number of follower Dgraph nodes to handle queries. Ensure that the Endeca Server cluster contains a sufficient number of nodes to accommodate the changed size of the data domain. With this operation, you can only add nodes to the data domain, but not remove them.</p> <p> Note: When the data domain's profile disallows queries on the leader node (<code>allowQueriesOnLeader</code> is set to <code>false</code>), <code>numFollowers</code> must be greater than one to ensure that at least one follower node is available to handle queries.</p>

Request element	Description
exportDataDomain	<p>Export the index of the specified data domain by taking a snapshot of the index and copying it into an internally stored file. A snapshot represents a copy of the index files only, and does not capture any other characteristics of the data domain, such as the Dgraph configuration, or the data domain profile used when it was created.</p> <p>You can export a data domain that is either enabled or disabled.</p> <p>When an enabled data domain is exported, it continues to run, but a copy of its index is stored in an offline directory on a shared file system.</p> <p>To export a data domain, specify:</p> <ul style="list-style-type: none">• <code>name</code> is the name of the data domain whose index you want to export.• <code>nameOfOfflineDataDomain</code> (Optional). The name that will be given to this data domain's index once it is exported. This name must be unique. If not specified, it is assigned automatically, by appending the date to the original data domain name, as in this format: <code>name_MMMMMM-dd-yyyy-hh-mm</code>. The request returns the resulting name used for the exported index. <p> Important: Keep track of the value of <code>nameOfOfflineDataDomain</code>, because you will need it later for importing this index.</p>

Request element	Description
importDataDomain	<p>Create a new data domain using previously exported index files. 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 <code>nameOfOfflineDataDomain</code> you used in the <code>exportDataDomain</code> operation, or know the name that was automatically assigned.</p> <p>To import a data domain, specify:</p> <ul style="list-style-type: none"> • <code>name</code> is the name of the new data domain into which you are importing the index files from a previously exported data domain. • <code>nameOfOfflineDataDomain</code> is the name of the previously exported index files. • <code>ddProfileName</code> (Optional). The name of the data domain profile that will be used for the new data domain created with imported index files. <p> Note: Starting with Endeca Server version 7.6.1.13, each previously exported data domain contains a <code>.profile</code> internal file, as part of its index. This file is used automatically as a data domain profile when you import the index and create a new data domain based on this index. You do not need to specify it. However, if you are importing a data domain from a version before Endeca Server 7.6.1.13, then its index does not have the <code>.profile</code> file. (You are required to know the data domain profile name.) When you import this index, you can then either specify the data domain profile name (if you have it and if this data domain profile exists in your cluster), or if you don't have the data domain profile for the domain you are importing, then Endeca Server imports the index and creates the data domain based on the default data domain profile.</p> <ul style="list-style-type: none"> • <code>enabled</code> (Optional). Specifies whether the newly imported data domain should be enabled (if set to <code>true</code>). If the value is not specified, the imported data domain is enabled by default.
enableDataDomain	<p>Enable a data domain by specifying its <code>name</code>. This starts the Dgraph nodes (processes) for the data domain.</p>
disableDataDomain	<p>Disable a data domain by specifying its <code>name</code>. A disabled data domain exists and is still allocated resources, but its Dgraph nodes are shut down and must be enabled with <code>enableDataDomain</code> before the data domain can handle requests.</p>
getDataDomain	<p>Obtain information about a data domain with a specified <code>name</code>. The response returns the data domain name and its status (enabled or disabled).</p>

Request element	Description
<code>listDataDomains</code>	List all existing data domains and includes additional status information, such as the name, description, whether the data domain is enabled, the number of nodes, the number of query processing threads, and the list of arguments specified for the Dgraph processes (if any are specified).
<code>getDataDomainStats</code>	Return runtime statistics about the data domain, if you specify the data domain name. The response includes the following information: <ul style="list-style-type: none"> • The size of the index records on shared storage (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. The statistics pages are intended for Oracle Endeca Support.
<code>getDataDomainProfile</code>	Obtain information about the data domain profile with the specified name. Returns the name and description of the profile and other data domain profile characteristics, such as the number of nodes, the number of query processing threads, and the list of arguments that are sent to the Dgraph processes.
<code>listDataDomainProfiles</code>	Return the set of data domain profiles configured in the Endeca Server cluster. For each data domain profile, returns its name, description, and other characteristics, such as the number of nodes, the number of query processing threads, and the list of arguments sent to the Dgraph processes.
<code>updateDataDomain</code>	Update the data domain to use another data domain profile. To update a data domain, specify: <ul style="list-style-type: none"> • Its name • The new data domain profile name, <code>ddProfileName</code> Before running this operation, ensure that the data domain is disabled, the new data domain profile exists, and the Endeca Server has sufficient resources to allocate the data domain using the new profile. After you run this operation, the data domain continues to be disabled and must be enabled.

Request element	Description
warmCache	<p>Request the Dgraph process to warm up its cache without the need for you to create a custom warm-up script. The command allows the Dgraph to reuse its computation results across queries, and thus helps reduce the user-observable latencies in query processing and improves performance. If the data domain runs in a cluster, the cache warming job runs on all Dgraph processes for the data domain. If you issue queries for the data domain, the Endeca Server processes them before any cache warming queries.</p> <p>To warm up the Dgraph process cache, specify:</p> <ul style="list-style-type: none"> • The data domain's name • <code>timeLimitSec</code> (optional). The default is 1800 seconds (30 minutes). It is used if you do not specify the time limit.
version	<p>List the version of the Oracle Endeca Server and the versions of the Dgraph processes powering each of the data domains (if the Dgraph processes for these data domains are currently running).</p> <p>You do not need to specify the data domain names when using this command.</p>

Example

The following examples show the Manage Web Service request and response for creating a data domain:

This abbreviated example shows a Manage Web Service request that creates a data domain `myDD` based on the data domain profile `my_dd_profile`. Once created, this data domain will be enabled:

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

This example shows an empty `createDataDomainResponse` from the Manage Web Service. This indicates that the data domain was created successfully:

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

To verify that the data domain was added, issue a `listDataDomains` operation.

It is possible that, based on the existing data domain profiles, the Endeca Server cluster cannot create a data domain. This situation can occur if, for example, the data domain profile specifies a greater number of follower nodes in the data domain than the number of deployed Endeca Server nodes. For example, if you have deployed a single Endeca Server node and created a data domain profile with four follower nodes and a dedicated leader node (`allowQueriesOnLeader` is set to `false`), you may receive an error similar to the following:

```
Request for 5 nodes would exceed configured maximum allowable number of oversubscribed nodes
```

To summarize, if you receive an error when creating a data domain, verify the number and characteristics of your Endeca Server nodes and the configuration of the data domain profile you used. Check that the Endeca Server cluster is configured with a sufficient number of Endeca Server nodes, has enough hardware capacity, and that the data domain profile matches your data domain requirements. For information, see [How oversubscribing affects hardware utilization on page 44](#) and [Deciding which data domain profile to use on page 28](#).



Note: For reference information about the elements and types used in the Manage Web Service, see the Manage Web Service API Reference, in the *Oracle Endeca Server API References*.

Index

A

- allow queries on leader 28
- auto-idling 29

C

- cluster
 - building, high-level overview 9
 - data domain 8
 - Endeca Server 7
 - monitoring status 25
- Cluster Coordinator
 - about 18
 - requirements 20
- Cluster Web Service 49
 - accessing 49
 - list of operations 50
 - using 50

D

- data domain
 - auto-idling 29
 - behavior 14
 - bringing it online 17
 - cluster definition 8
 - defining profile 39
 - definition 7
 - deleting profile 42
 - enhanced availability 20
 - follower node 10
 - hardware utilization 44
 - idling 42
 - leader node 10
 - listing profiles 41
 - node allocation 14
 - outer transaction 18
 - oversubscribing, explained 29
 - profile 27
 - profile example, high hardware usage 47
 - profile example, low hardware usage 47
 - profile parameters 30
 - read-only 29
 - updates 17
- data domain cluster
 - hosting a single data domain, diagram 11
 - hosting multiple data domains, diagram 12
- deleting
 - node, in the Endeca Server cluster 26
- Dgraph node
 - See node

E

- Endeca Server
 - cluster 7
 - diagram of cluster 12
 - listing cluster nodes 22
 - node profile 23
 - node profile, configuring 24
 - version 23
- enhanced availability 19

F

- failure
 - Cluster Coordinator 20
 - data domain node 20
 - Endeca Server node 19
- follower node 10
- follower nodes, deciding number 29

I

- idle data domains 29
- idling 42

L

- leader node 10
- listing nodes for Endeca Server cluster 22

M

- Manage Web Service
 - about 56
 - accessing 57
 - list of operations 57
 - using 57

N

- node
 - deleting, in Endeca Server cluster 26
 - Endeca Server, definition 7
 - Endeca Server, failure 19
 - profile, for Endeca Server nodes 23
- node profile
 - configuring, for Endeca Server nodes 24
 - Endeca Server nodes 23

O

- oversubscribing 29

P

- profile

- choosing, for a data domain 28
- CPU utilization, by data domains 45
- data domain 27
 - default, for data domain 37
 - defining, for data domain 39
 - deleting, for data domain 42
 - for Endeca Server nodes 23
 - for the data domain 28
 - listing, for data domains 41
 - oversubscribing, for a data domain 44
 - parameters, for data domain 30
- .profile file 28

R

- read-only
 - data domain 29

S

- status
 - Endeca Server cluster 25
 - Endeca Server nodes 25

V

- version, Endeca Server 23

W

- Who should use this guide 5