Oracle® Big Data Discovery

Administrator's Guide

Version 1.1.3 • May 2016

Copyright and disclaimer

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

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

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

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

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

Table of Contents

Part I: Overview of Big Data Discovery Administration

Part II: Administering Big Data Discovery

Part III: Administering the Dgraph and Dgraph Gateway

Part IV: Administering Studio

Part V: Controlling User Access to Studio

Part VI: Administering Big Data Discovery Using Enterprise Manager Cloud Control

Part VII: Logging for Studio, Dgraph, and Dgraph Gateway

Preface

Oracle Big Data Discovery is a set of end-to-end visual analytic capabilities that leverage the power of Hadoop to transform raw data into business insight in minutes, without the need to learn complex products or rely only on highly skilled resources.

About this guide

This guide describes administration tasks associated with Oracle Big Data Discovery.

Audience

This guide is intended for administrators who configure, monitor, and control access to Oracle Big Data Discovery.

Conventions

The following conventions are used in this document.

Typographic conventions

The following table describes the typographic conventions used 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 segments within a paragraph.
Variable	This formatting is used for variable values. For variables within a code sample, the formatting is ***`Variable`***.
`File Path`	This formatting is used for file names and paths.

Symbol conventions

The following table describes 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
`$ORACLE_HOME`	Indicates the absolute path to your Oracle Middleware home directory, where BDD and WebLogic Server are installed.
`$BDD_HOME`	Indicates the absolute path to your Oracle Big Data Discovery home directory, `$ORACLE_HOME/BDD-<version>`.
`$DOMAIN_HOME`	Indicates the absolute path to your WebLogic domain home directory. For example, if your domain is named `bdd-<version>_domain`, then `$DOMAIN_HOME` is `$ORACLE_HOME/user_projects/domains/bdd-<version>_domain`.
`$DGRAPH_HOME`	Indicates the absolute path to your Dgraph home directory, `$BDD_HOME/dgraph`.

Contacting Oracle Customer Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. This includes important information regarding Oracle software, implementation questions, product and solution help, as well as overall news and updates from Oracle.

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

Part I

Overview of Big Data Discovery Administration

Chapter 1

Introduction

This section lists administrative tasks and tools that you can use to do these tasks. It also lists all Big Data Discovery logs and describes the backup strategy.

List of administrative tasks

Administrative tools

List of administrative tasks

This topic lists top-level administrator tasks for Studio, the Dgraph, the Dgraph HDFS Agent, and the Dgraph Gateway.

Section	Tasks
Overview of Big Data Discovery Administration	Learning about available administrative tools and logs used in Big Data Discovery, as well as learning about which files need to be backed up. Also, viewing the diagram of the Big Data Discovery cluster, learning about the cluster behavior, such as routing or requests, handling of data updates, and maintaining high availability.
Administering Big Data Discovery	Using the `bdd-admin` script for administering the product — starting, stopping and restarting the components, and checking the status of Big Data Discovery services. Also, performing administrative tasks for the BDD cluster deployment.
Administering the Dgraph and Dgraph Gateway	• Learning about the Dgraph, its memory consumption, the Dgraph internal cache, and a way to limit the Dgraph memory consumption for expensive queries. • Starting and stopping the Dgraph Gateway in the WebLogic Server Administration Console. • Running the Dgraph administrative operations with the `bdd-admin` script. • Using flags for the Dgraph and for the Dgraph HDFS Agent. • Administering the Dgraph with the Dgraph Gateway Command Utility.

Section	Tasks
Administering Studio	- Configuring framework settings. - Configuring Hadoop settings for file upload. - Managing data sources and viewing summary reports of project usage. - Configuring the locale and email notifications. - Managing projects in the Control Panel.
Controlling User Access to Studio	- Configuring user-related settings in Studio. - Creating and managing users in Studio. - Integrating with an LDAP system to manage users. - Setting up Single Sign-On (SSO).
Administering Big Data Discovery using Enterprise Manager	- Tasks for the Big Data Discovery targets (Cluster, Studio, Dgraph). - Running various Dgraph administrative operations, such as viewing the Dgraph statistics, or saving the Dgraph Tracing Utility data.
Logging	- Logging options in the `bdd-admin` script. - Logging options for Big Data Discovery targets in Enterprise Manager. - Studio logs, their format and types, and customization options. - Dgraph Gateway logs, their format, log levels, and customization options. - Dgraph request log and stdout/sterr log.

Administrative tools

Two tools for administering Big Data Discovery exist: Enterprise Manager and the `bdd-admin` script. This topic introduces these administrative tools and discusses when to use each.

Enterprise Manager for Big Data Discovery

The Enterprise Manager plug-in lets you monitor, diagnose, and manage Big Data Discovery components. The plug-in includes three targets: a target for the entire Big Data Discovery cluster, as well as targets for Studio and the Dgraph.

For information on performing administrative tasks through the plug-in, see *Using Enterprise Manager for Big Data Discovery on page 137*.

The `bdd-admin` script

The `bdd-admin` script lets you perform a number of administrative tasks for the Dgraph, the HDFS Agent, Studio, and the Dgraph Gateway from the command line.

For information on performing administrative tasks through the script, see *The bdd-admin Script on page 24*.

Administrative tool comparison

This table illustrates which administrative tasks you can perform using the script or the Enterprise Manager plug-in. Use these guidelines:

- The `bdd-admin` script is available to you regardless of whether you are using the Enterprise Manager plug-in. Use it for all administrative tasks or for those that you cannot do in any other way. For example, you can use it to perform administrative tasks for the Dgraph and HDFS Agent, including starting and stopping, logging, updating the configuration, and running administrative operations.
- The Enterprise Manager plug-in is optional. The plug-in is desirable especially for monitoring and log access, thus it is recommended to be used if you have the license for it and have installed it.

This table compares these tools.

Administrative task	Enterprise Manager	bdd-admin	Notes
Starting and stopping Big Data Discovery (all components)	No	Yes	Enterprise Manager: You can only use it after Big Data Discovery is already up and running. Script: After you install Big Data Discovery, you use the script to start and stop the entire stack (Dgraph, Dgraph HDFS Agent, Studio, Dgraph Gateway).
Starting and stopping the Dgraph	Yes	Yes	Both tools can start and stop the Dgraph target. Note that the Dgraph HDFS Agent is also started and stopped at the same time.
Starting and stopping Dgraph Gateway and Studio	Yes	Yes	Both tools start and stop both Dgraph Gateway and Studio. Note that both must be stopped and started at the same time.
Starting and stopping Hadoop nodes (used for Data Processing)	No	No	Neither tool can start and stop CDH or HDP nodes.
Adding and removing nodes in the cluster (Dgraph nodes, Studio nodes, CDH or HDP nodes)	No	No	Neither tool can add or remove nodes in the BDD deployment.
Exploring Dgraph logs	Yes	Yes	You can use `bdd.conf` to specify the location of the Dgraph and HDFS Agent output files and the Dgraph Gateway log level. You can use the `bdd-admin` script to enable, disable, and check the status of extended logging features and perform a log roll for the Dgraph.

Administrative task	**Enterprise Manager**	**bdd-admin**	**Notes**
Monitoring node status	Yes	Yes	Enterprise Manager: Node status indicator lets you see if the node is up or down. Script: You can use the `bdd-admin` script to check the current status of the Dgraph, the HDFS Agent, Studio, and the Dgraph Gateway.
Creating and deleting services	No	No	You cannot use the `bdd-admin` script or Enterprise Manager to create or delete services.
Enabling and disabling auto-start	No	Yes	You can use `bdd.conf` to enable the entire cluster to automatically restart after a reboot.
Refreshing configuration	No	Yes	Enterprise Manager: It does not have an option to refresh the configuration. Script: You can use the `bdd-admin` script to copy an updated version of `bdd.conf` to all servers in the BDD cluster deployment.
Running administrative operations	Yes	Yes	You can use administration operations for the Dgraph through Enterprise Manager and through the `bdd-admin` script.

Chapter 2

Cluster Architecture

This section describes the architecture of a Big Data Discovery cluster.

Cluster components

A Big Data Discovery cluster is a deployment of Big Data Discovery on multiple machines. Such a deployment can be made up of any number of nodes: you determine the number of nodes at deployment time.

About a BDD cluster, nodes, and deployments

This topic provides an overview of the components in a Big Data Discovery cluster.

What is a BDD cluster?

A BDD cluster:

- Supports on-premise deployments of Big Data Discovery, both on commodity hardware and on engineered systems, such as Oracle Big Data Appliance (BDA).
- Has anywhere from three to more nodes (a minimum number of three nodes are required for the production environment to ensure enhanced availability of query processing). For example, a production deployment can include six nodes. Each node in the cluster is known as a ***BDD node***.
- Performs routing and load balancing of query requests arriving from Studio to the nodes that run the Dgraph. This assumes that there is at least one Dgraph node available to process queries arriving from Studio.

Nodes

Nodes in the BDD cluster deployment have different roles:

- They can serve as Hadoop cluster nodes. This is because you deploy Big Data Discovery on a set of nodes running Hadoop.

- They can serve as WebLogic Server nodes on which Java-based components of BDD (Studio and Dgraph Gateway) are running in the WebLogic Server.
- They can serve as Dgraph-only nodes. Together, these nodes constitute a Dgraph cluster, within the overall BDD cluster deployment. These Dgraph nodes communicate with Hadoop nodes and utilize ZooKeeper from the Hadoop installation to maintain high availability of the Dgraph processes.

For more information on nodes and their roles shown on a diagram, see *Diagram of a Big Data Discovery Cluster on page 17*.

Note: These roles are not mutually-exclusive. For example, in demo or learning deployments, you can co-locate Dgraph instances on the same nodes that run WebLogic Server or experiment with other configurations that have nodes serving dual roles. See the *Installation and Deployment Guide* for information on deployment scenarios and co-location.

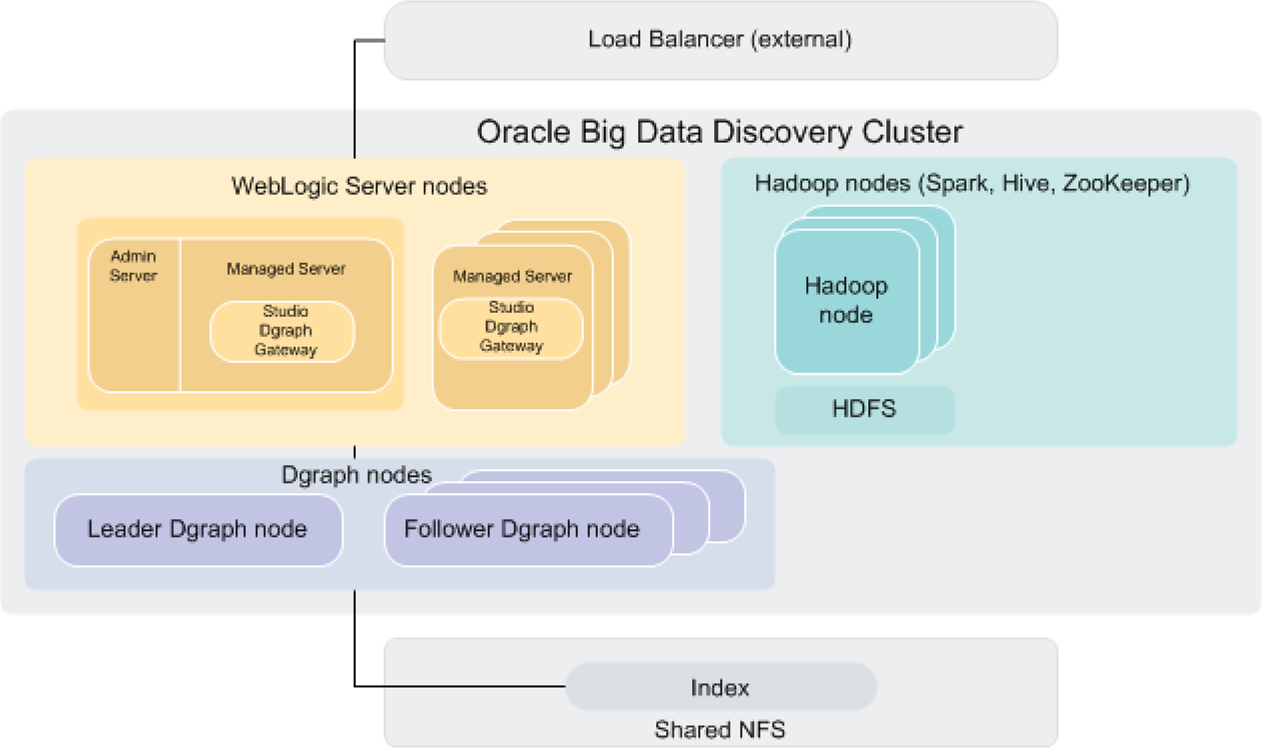
Types of BDD cluster deployments

You can choose between many ways in which to deploy BDD to utilize hardware efficiently. In the *Installation and Deployment Guide*, there are several recommended deployment scenarios to help you efficiently deploy BDD:

- A learning or demo deployment on one or two machines (this deployment is not intended to be turned into a production deployment).
- A production deployment on a set of six machines (three of which are running a Hadoop cluster only, two run WebLogic Servers with Studio and Dgraph Gateway, and one node is dedicated to running the Dgraph). The number of nodes in the production deployment can be less than six (with some software components co-located), and there can be more, depending on your needs.

Diagram of a Big Data Discovery Cluster

This diagram illustrates a cluster of Big Data Discovery nodes deployed on top of an existing Hadoop cluster.



This diagram depicts a suggested deployment topology for production, although many configurations are possible. For information on staging and learning, demo and production-level deployment topology, see the *Installation and Deployment Guide*.

In this diagram, starting from the top, the following components of the Big Data Discovery cluster deployment are included:

- An optional external load balancer serves as the single point of entry to the Big Data Discovery cluster. All browser requests are routed through this load balancer to Studio nodes.

  Note: Although it is recommended to use an external load balancer in your deployment, it is optional. For information, see *Load balancing and routing requests on page 20*.

- The Big Data Discovery cluster comprises three categories of nodes:
  1. Nodes that host WebLogic Server with Studio and Dgraph Gateway.
  2. Hadoop only nodes. These nodes do not host WebLogic Server or Dgraph instances. They run Data Processing jobs, within a Big Data Discovery deployment.
  3. Dgraph nodes. These nodes are solely dedicated to hosting Dgraph instances.
- **WebLogic Server nodes**. These nodes represent machines that the WebLogic Server is deployed on. These nodes are hosting two Java applications: Studio and Dgraph Gateway.
- **Hadoop nodes**. Big Data Discovery is deployed on top of an Hadoop cluster. This diagram shows only those Hadoop nodes on which BDD is deployed. These nodes represent a subset of the entire pre-

existing Hadoop cluster, onto which BDD is deployed. These nodes have both Hadoop and BDD installation on them and share access to HDFS. Optimally, three Hadoop nodes are required for hosting ZooKeeper instances. ZooKeeper maintains a cluster state for all participating members of the Big Data Discovery cluster, in particular, it ensures automatic Dgraph leader node election, in case the leader Dgraph node fails.

- **Dgraph nodes**. These nodes form a Dgraph cluster that is part of the larger BDD cluster deployment. One node serves as the leader Dgraph node, and the remaining nodes are follower Dgraph nodes. All nodes in the Dgraph cluster have write access to a shared file system (NFS) on which the index is stored. Only the leader Dgraph node writes to the index located on the file system. Follower Dgraph nodes can only read from the index. The index includes internal indexes for each of the data sets in BDD.
- Enterprise Manager for Big Data Discovery is not shown on this diagram. It can be optionally used with any Big Data Discovery deployment. When used, Enterprise Manager is installed on a separate WebLogic Server. For more information, see *Using Enterprise Manager for Big Data Discovery on page 137*.

Cluster of Dgraph nodes

A typical BDD cluster deployment includes a set of machines that are solely dedicated to running the Dgraph. This set of machines is known as the Dgraph cluster.

A ***Dgraph cluster*** is a set of Dgraphs that together handle requests for data sets in Big Data Discovery. Requests arriving from Studio are routed and load-balanced between the Dgraph nodes. One of these Dgraph nodes is responsible for handling all write operations (updates, configuration changes), while the remaining Dgraphs serve as read-only. All Dgraph nodes in the cluster utilize an index residing on shared storage.

The leader and follower Dgraph nodes differ in the types of queries they can process; however, this is transparent to the end users of Big Data Discovery. The allocation of leader and follower Dgraph node roles is performed by the BDD cluster automatically.

A ***Dgraph node*** is a node in BDD cluster deployment that runs the Dgraph. The Dgraph is the main computational module that provides search, refinement computation, Guided Navigation, and many other features, all of which you can observe and use in Studio.

In a BDD cluster deployment, you can have only one cluster of Dgraph nodes. All nodes in BDD that run Studio and Dgraph Gateway in WebLogic Server talk to the same single cluster of Dgraph nodes. The Dgraph cluster can have any number of nodes, even though a certain number of Dgraph nodes is recommended for production environment. For more information, see the *Installation and Deployment Guide*.

Dgraph Cluster role

A Dgraph cluster is responsible for:

- **Enhanced availability of query processing by the Oracle Big Data Discovery**. In a cluster of Dgraph nodes, if one of the Dgraph nodes fails, queries continue to be processed by other Dgraph nodes.
- **Increased throughput**. At deployment time, you can add one or more Dgraph nodes to the same Dgraph cluster. This lets you spread the query load across them, without the need to increase storage requirements at the same rate.

Leader and follower Dgraph nodes

This topic introduces the terms used to describe Dgraphs: the leader and follower nodes.

Leader Dgraph node

The ***leader node*** is a single Dgraph node responsible for receiving and processing updates to the index and configuration. This node also does query processing, like other nodes. This node is responsible for generating information about the latest versions of the data set index and propagating this information to the follower Dgraph nodes.

The Dgraph Gateway automatically determines which Dgraph node is the leader Dgraph node. The other Dgraph nodes are follower Dgraph nodes. Thus, each BDD cluster deployment with multiple Dgraph nodes is started with one leader Dgraph node and a number of follower Dgraph nodes.

The leader Dgraph node periodically receives full or incremental index updates. It also receives administration or configuration updates. It is the only node in the Dgraph 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 Dgraph node

A ***follower node*** is a node in the Dgraph cluster responsible for processing queries arriving from Studio. Typically, in any Big Data Discovery cluster deployment, there is a subset of nodes serving as the Dgraph follower nodes. The follower nodes do not update the index. When the Dgraph nodes are started, the Dgraph Gateway elects a leader Dgraph node, with the other Dgraph nodes being follower nodes.

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.

Cluster behavior

There are many possible scenarios of Big Data Discovery deployment clusters. This section describes how the BDD cluster behaves and maintains enhanced availability in various scenarios, such as during node startup, updates to the indexes, or individual node failures.

Load balancing and routing requests

This topic discusses the load balancing and routing requests from Studio nodes to the Dgraph nodes in Oracle Big Data Discovery.

Load balancing requests

Depending on your deployment strategy, to the external clients, the entry point of contact with the on-premise deployment of the Big Data Discovery cluster could be either any Studio-hosting node in the cluster, or an external load balancer configured in front of Studio instances.

The Big Data Discovery cluster relies on the following two levels of requests load balancing:

1. Load balancing requests across the nodes hosting multiple instances of Studio. This task should be performed by an external load balancer, if you choose to use it in your deployment (an external load balancer is not included in the Big Data Discovery package).

   If you use an external load balancer, it receives all requests and distributes them across all of the nodes in the Big Data Discovery cluster deployment that host the Studio application. Once a request is received from a Studio node, it is routed by BDD to the appropriate Dgraph node.

   If you don't use an external load balancer, external requests can be sent to any Studio node. They are then load-balanced between the nodes hosting the Dgraph.

2. Load balancing requests across the Dgraph nodes. This task is automatically handled by the BDD cluster. The Big Data Discovery software accepts requests from its Studio and Data Processing components on any node hosting the Dgraph and provides their internal load balancing across the other Dgraph-hosting nodes.

Routing requests

The Big Data Discovery cluster automatically directs requests to the subset of the cluster nodes hosting the Dgraph instances.

The following statements describe the behavior of the BDD cluster for routing requests to Dgraph nodes:

- Requests can be submitted from Studio or Data Processing components to any Dgraph Gateway in the BDD cluster, which in turn will route the request to an appropriate Dgraph node.

  For example, if the request is an updating request, such as a data loading request or a configuration update, it is routed to the leader Dgraph node in the cluster. If the request represents a non-updating (query processing) request, it is routed to the leader Dgraph node or to any of the follower Dgraph nodes. If a BDD cluster has only one node hosting the Dgraph, this node serves as the leader (with no followers).

- Non-updating requests are load-balanced using round-robin algorithm across the Dgraph nodes for processing.
- The Big Data Discovery cluster utilizes session affinity for all requests arriving from Studio to the Dgraph, by relying on the session ID in the header of each Studio request. Requests from the same session ID are always routed to the same Dgraph node in the cluster. This improves query processing performance by efficiently utilizing the Dgraph cache, and improves performance of caching entities (known in Studio as views).

How session affinity is used

When a WebLogic Server node hosting Studio and Dgraph Gateway receives a client request, it routes the request to a Dgraph node using session affinity, based on the session ID specified in the header of the request.

When end users issue queries, Studio sets the session ID for the requests in the HTTP headers. Requests with the same session ID are routed to the same Dgraph node. If the BDD software cannot locate the session ID, it relies on a round-robin strategy for deciding which Dgraph node the request should be routed to.

Session affinity is enabled by default, via the `endeca-session-id-key` and `endeca-session-id-type` properties in the `EndecaServer.properties` file of the Dgraph Gateway (do not change these values):

Property	Description
`endeca-session-id-key=`*`name`*	This is the name of the object checked by the specified method. Its default value is `session ID`.
`endeca-session-id-type=`*`method`*	This is the method used for establishing session affinity. Its default value is `HEADER`.

Startup of Dgraph nodes

Once the Big Data Discovery cluster is started, it activates the Dgraph processes on a subset of the nodes that are hosting the Dgraph instances. This topic discusses the behavior of the Dgraph nodes at startup.

On startup, the following actions take place:

- Any Dgraph node is started in either a leader or follower mode and in any order. Any number of follower nodes and one leader node are started in each Big Data Discovery cluster deployment.

  Note: If the BDD cluster deployment has only one node that runs the Dgraph, then this node serves as the leader. This configuration is possible but is not recommended for production environments.

- Once started, each Dgraph node registers with ZooKeeper that manages the distributed state of the Dgraph nodes. The leader node determines the current version of the index and informs ZooKeeper.
- 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 Dgraph node is in the process of updating, merging, or deleting indexed versions on disk. Follower Dgraph nodes do not receive updating requests; they acquire access to the new index once the updates complete.

How updates are processed

In a Dgraph cluster (it is part of the BDD cluster deployment), updates to the records in the indexes and updates to the configuration are routed to the leader Dgraph node.

The leader Dgraph node processes the update and commits it to the on-disk index. Upon completion, all follower nodes are informed that a new version of the index is available. The leader Dgraph node and all follower Dgraph nodes can continue to use the previous version of the index to finish query processing that had started against that version.

As each Dgraph 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.

Role of ZooKeeper

The ZooKeeper utility provides configuration and state management and distributed coordination services to Dgraph nodes of the Big Data Discovery cluster. It ensures high availability of the query processing by the Dgraph nodes in the cluster.

ZooKeeper is part of the Hadoop package. The Hadoop package is assumed to be installed on all Hadoop nodes in the BDD cluster deployment. Even though ZooKeeper is installed on all Hadoop nodes in the BDD cluster, it may not be running on all of these nodes. To ensure availability of a clustered Dgraph deployment, configure an odd number (at least three) of Hadoop nodes to run ZooKeeper instances. This will prevent ZooKeeper from being a single point of failure.

ZooKeeper has the following characteristics:

- It is a shared information repository that provides a set of distributed coordination services. It ensures synchronization, event notification, and coordination between the nodes. The communication and coordination mechanisms continue to work in the case when connections or Dgraph-hosting nodes fail.
- It provides communication between the Dgraph-hosting nodes, ensuring that if one of these nodes fails, requests are sent to other active Dgraph nodes, until the node rejoins the Dgraph cluster (this is true if more than two instances of ZooKeeper are running in the deployment).
- It provides communication between Dgraph nodes. It controls the election of the Dgraph leader node in the Dgraph cluster, in case the current leader Dgraph node fails. The newly-elected leader Dgraph node identifies the most recent version of the index, and, using ZooKeeper, informs other nodes of the current version of the index.

To summarize, in order to run, ZooKeeper requires a majority of its hosting nodes to be active. Therefore, it is recommended that ZooKeeper run on an odd number (at least three) of the Hadoop nodes in the deployed Big Data Discovery cluster. You can ensure this during the installation, when running the deployment script.

How enhanced availability is achieved

This topic discusses how the BDD cluster deployment ensures enhanced availability of query-processing.

Important: The BDD cluster deployment 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.

The following three sections discuss the BDD cluster behavior for providing enhanced availability.

Note: This topic discusses BDD deployments with more than one running instance of the Dgraph. Even though you can deploy BDD on a single node, such deployments can only serve development environments, as they do not guarantee the availability of query processing in BDD. Namely, in a BDD deployment where only one node is hosting a single Dgraph instance, a failure of the Dgraph node shuts down the Dgraph process.

Availability of WebLogic Server nodes hosting Studio

When a WebLogic Server node goes down, Studio also goes down. As long as the BDD cluster utilizes an external load balancer and consists of more than one WebLogic Server node on which Studio is started, this does not disrupt Big Data Discovery operations.

If a WebLogic Studio node hosting Studio fails, the BDD cluster (that uses an external load balancer) stops using it and relies on other Studio nodes, until you restart it.

Availability of Dgraph nodes

The ZooKeeper ensemble running on a subset of Hadoop (CDH or HDP) nodes ensures the enhanced availability of the Dgraph cluster nodes and services:

- Failure of the leader Dgraph node. When the leader Dgraph node goes offline, the BDD 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. You should manually restart this node with the `bdd-admin` script. When the node that was the leader node is restarted and joins the cluster, it becomes one of the follower nodes. It is also possible that the leader node is restarted and joins the cluster before the cluster needs to appoint a new leader node. In this case, the node continues to serve as the leader node.

  If the leader node changes, the BDD cluster starts routing updating requests to the newly-elected leader Dgraph node.

- Failure of a follower Dgraph node. When one of the follower nodes goes offline, the BDD cluster starts routing requests to other available nodes. You should manually restart this node using the `bdd-admin` script. Alternatively, you can start the Dgraph through the Enterprise Manager plug-in for Big Data Discovery (if you are using it). Once the node is restarted, it rejoins the cluster, and the cluster adjusts its routing information accordingly.

Availability of ZooKeeper instances

The ZooKeeper instances themselves must be highly available. The following statements describe the requirements in detail:

- Each Hadoop node in the BDD cluster deployment can be optionally configured at deployment time to host a ZooKeeper instance. To ensure availability of ZooKeeper instances, it is recommended to deploy them in a cluster of their own, known as an ensemble. At deployment time, it is recommended that a subset of the Hadoop nodes is configured to host ZooKeeper instances. As long as a majority of the ensemble is running, ZooKeeper services are used by the BDD cluster. Because ZooKeeper requires a majority, it is best to start an odd number of its instances. This means that ZooKeeper must be started on at least three Hadoop nodes in the BDD cluster. A Hadoop node hosting a ZooKeeper instance assumes responsibility for ensuring the ZooKeeper process uptime. It will start ZooKeeper when BDD is deployed and will restart it should it stop running.

  To summarize, although ZooKeeper can run on only one Hadoop node, to ensure high availability of ZooKeeper instances, ZooKeeper must run on at least three Hadoop nodes (or an odd number of nodes that is greater than three) in any BDD cluster. This prevents ZooKeeper itself from being a single point of failure.

- If you do not configure at least three Hadoop nodes to run ZooKeeper, it will be a single point of failure. Should ZooKeeper fail, access to the data sets served by BDD becomes read-only. No updates, writes, or changes of any kind are possible while ZooKeeper in the BDD cluster is down. To recover from this situation, the Hadoop node that was running a failed ZooKeeper must be restarted or replaced (the action required depends on the nature of the failure).

Part II

Administering Big Data Discovery

Chapter 3

The bdd-admin Script

You can use the bdd-admin script to perform administrative operations for your BDD cluster from the command line. This section describes the script and its commands.

About the bdd-admin script

The bdd-admin script includes a number of commands that perform different administrative tasks for your cluster, like starting components and updating BDD's configuration. The script is located in the $BDD_HOME/BDD_manager/bin directory.

Important: bdd-admin can only be run from the Admin Server as the Linux user that installed BDD. This user requires passwordless sudo enabled on all nodes in the cluster.

bdd-admin has the following syntax:

```
./bdd-admin.sh <command> [command args]
```

When you run the script, you must specify a command. This determines the operation it will perform. Note that you can't specify multiple commands at once, and you must wait for a command to complete before running it a second time. Additionally, you can't run the following commands at the same time:

- start
- stop
- restart
- backup
- restore

For example, if you run stop, you can't run start until all components have been stopped.

You can also include any of the specified command's supported arguments to further control the script's behavior. For example, you can run most commands on all nodes or one or more specific ones. The arguments each command supports are described later in this chapter.

The following sections describe the commands bdd-admin supports.

Lifecycle management commands

`bdd-admin` supports the following lifecycle management commands.

Command	Description
`start`	Starts components.
`stop`	Stops components.
`restart`	Restarts components.

System management commands

`bdd-admin` supports the following system management commands.

Command	Description
`autostart`	Enables/disables autostart for components. Components that have autostart enabled will automatically restart after their hosts are rebooted.
`backup`	Backs up your cluster's data and metadata to a single tar file.
`restore`	Restores your cluster's data and metadata from a backup tar file.
`publish-config`	Publishes updated BDD, Hadoop, and Kerberos configuration to all BDD nodes.
`update-model`	Either updates the model files for Data Enrichment modules, or restores them to their original states.
`flush`	Flushes component caches.

Diagnostics commands

`bdd-admin` supports the following diagnostics commands.

Command	Description
`get-blackbox`	Generates the Dgraph's on-demand tracing blackbox file and returns its name and location. This command is intended for use by Oracle Support only.
`status`	Returns either component statuses or the overall health of the cluster.
`get-stats`	Returns component statistics. This command is intended for use by Oracle Support only.
`reset-stats`	Resets component statistics. This command is intended for use by Oracle Support only.

Command	Description
`get-log-levels`	Outputs the current levels of component logs.
`set-log-levels`	Sets the log levels for components and subsystems.
`get-logs`	Generates a zip file of component logs. This command is intended for use by Oracle Support only.
`rotate-logs`	Rotates component logs. This command is intended for use by Oracle Support only.

Global options

`bdd-admin` supports the following global options. You can include these with any command, or without a command.

Command	Description
`--help`	Prints the usage information for the `bdd-admin` script and its commands.
`--version`	Prints version information for your BDD installation.

For example, to view the usage for the entire `bdd-admin` script, run:

```
./bdd-admin.sh --help
```

To view the usage for a specific command, run the command with the `--help` flag:

```
./bdd-admin.sh <command> --help
```

For the version number of your BDD installation, run:

```
./bdd-admin.sh --version
```

Lifecycle management commands

You can use the `bdd-admin` script's lifecycle management commands to perform such operations as starting and stopping BDD components.

start

stop

restart

start

You can start components by running the `bdd-admin` script with the `start` command.

Note: You can't run `start` if `stop`, `restart`, `backup`, or `restore` are currently running.

To start components, run the following from the Admin Server:

```
./bdd-admin.sh start [option <arg>]
```

You can specify the following options.

Option	Description
`-c, --component <component(s)>`	A comma-separated list of the components the script will start: • `agent`: Dgraph HDFS Agent • `dgraph`: Dgraph • `dp`: Data Processing • `bddServer`: Studio and Dgraph Gateway Note the following: • The script will prompt you for the WebLogic Server username and password when starting the `bddServer` component (or all components) if the `BDD_WLS_USERNAME` and `BDD_WLS_PASSWORD` environment variables aren't set. • You can't start the `dp` component unless the `dgraph` and `agent` components are already running. • If you start only the `dgraph` component, the script will automatically start the `agent`, as well, and vice versa.
`-n, --node <hostname(s)>`	A comma-separated list of the nodes the script will run on. Each must be defined in `bdd.conf`.

If you don't specify any options, the script will start all supported components.

Examples

The following command starts all supported components:

```
./bdd-admin.sh start
```

The following command starts the Dgraph and the HDFS Agent on the `web009.us.example.com` node:

```
./bdd-admin.sh start -c dgraph,agent -n web009.us.example.com
```

stop

You can stop components by running the `bdd-admin` script with the `stop` command.

Note: You can't run `stop` if `start`, `restart`, `backup`, or `restore` is currently running.

To stop components, run the following from the Admin Server:

```
./bdd-admin.sh stop [option <arg>]
```

You can specify the following options.

Option	Description
`-t, --timeout` <*minutes*>	The amount of time the script will wait (in minutes) before terminating the component(s). If this value is 0, the script will force the component(s) to shut down immediately. If it's greater than 0, the script will wait the specified amount of time for the component(s) to shut down gracefully, then terminate them if they don't. If this option isn't specified, the script will shut the component(s) down gracefully, which may take a long time.
`-c, --component` <*component(s)*>	A comma-separated list of the components the script will stop: • `agent`: Dgraph HDFS Agent • `dgraph`: Dgraph • `dp`: Data Processing • `bddServer`: Studio and Dgraph Gateway Note the following: • The script will prompt you for the WebLogic Server username and password when starting the `bddServer` component (or all components) if the `BDD_WLS_USERNAME` and `BDD_WLS_PASSWORD` environment variables aren't set. • If you stop only the `dgraph` component, the script will automatically stop the `agent`, as well, and vice versa.
`-n, --node` <*hostname(s)*>	A comma-separated list of the nodes the script will run on. Each must be defined in `bdd.conf`.

If you don't specify any options, the script will stop all supported components gracefully.

Examples

The following command gracefully shuts down all supported components:

```
./bdd-admin.sh stop
```

The following command waits 10 minutes for the Dgraph HDFS Agent, Dgraph, and Data Processing to shut down gracefully, then terminates any that are still running:

```
./bdd-admin.sh stop -t 10 -c agent,dgraph,dp
```

restart

You can restart components by running the `bdd-admin` script with the `restart` command. This will restart components regardless of whether they're currently running or stopped.

Note: You can't run `restart` if `start`, `stop`, `backup`, or `restore` is currently running.

To restart components, run the following from the Admin Server:

```
./bdd-admin.sh restart [option <arg>]
```

You can specify the following options.

Option	Description
`-t, --timeout` *`<minutes>`*	The amount of time the script will wait (in minutes) before terminating the component(s). If this value is 0, the script will force the component(s) to shut down immediately. If it's greater than 0, the script will wait the specified amount of time for the component(s) to shut down gracefully, then terminate them if they don't. If this option isn't specified, the script will shut the component(s) down gracefully, which may take a long time.
`-c, --component` *`<component(s)>`*	A comma-separated list of the components the script will restart: • `agent`: Dgraph HDFS Agent • `dgraph`: Dgraph • `dp`: Data Processing • `bddServer`: Studio and Dgraph Gateway Note the following: • The script will prompt you for the WebLogic Server username and password when starting the `bddServer` component (or all components) if the `BDD_WLS_USERNAME` and `BDD_WLS_PASSWORD` environment variables aren't set. • You can't restart the `dp` component unless the `dgraph` and `agent` components are either running or being restarted, as well. • If you restart only the `dgraph` component, the script will automatically stop the `agent`, as well, and vice versa.
`-n, --node` *`<hostname(s)>`*	A comma-separated list of the nodes the script will run on. Each must be defined in `bdd.conf`.

If you don't specify any options, the script will restart all supported components gracefully.

Examples

The following command gracefully shuts down and then restarts all supported components:

```
./bdd-admin.sh restart
```

The following command waits 5 minutes for the Dgraph and the HDFS Agent on the `web009.us.example.com` node to shut down gracefully, terminates it if it's still running, then restarts it:

```
./bdd-admin.sh restart -t 5 -c dgraph -n web009.us.example.com
```

System management commands

You can use the `bdd-admin` script's system management commands to perform such operations as backing up your cluster and updating BDD's configuration.

autostart

backup

restore

publish-config

update-model

flush

autostart

You can enable and disable autostart for components by running the `bdd-admin` script with the `autostart` command. Components that have autostart enabled will restart automatically after their hosts are rebooted.

Note: `autostart` doesn't restart components that crashed or were stopped by `bdd-admin` before a reboot.

To enable or disable autostart, run the following from the Admin Server:

```
./bdd-admin.sh autostart <operation> [option <arg>]
```

You must specify one of the following operations.

Operation	Description
`on`	Enables autostart for the specified component(s).
`off`	Disables autostart for the specified component(s).
`status`	Returns the status of autostart for the specified component(s).

You can also specify the following options.

Option	Description
`-c, --component <component(s)>`	A comma-separated list of the components the script will enable or disable autostart for: • `agent`: Dgraph HDFS Agent • `dgraph`: Dgraph • `bddServer`: Studio and Dgraph Gateway Note that if the `dgraph` component is automatically restarted, the `agent` component will be, as well, and vice versa.
`-n, --node <hostname(s)>`	A comma-separated list of the nodes the script will run on. Each must be defined in `bdd.conf`.

If you don't specify any options, the script will run on all supported components.

Examples

The following command enables autostart for all supported components:

```
./bdd-admin.sh autostart on
```

The following command returns whether the HDFS Agent running on the `web009.us.example.com` node has autostart enabled or disabled:

```
./bdd-admin.sh autostart status -c agent -n web009.us.example.com
```

backup

You can back up your cluster's data and metadata by running the `bdd-admin` script with the `backup` command.

Note: You can't run `backup` if `start`, `stop`, `restart`, or `restore` is currently running.

This backs up the following data to a single tar file, which you can later use to restore your cluster:

- Studio database
- Schema and data for Hive tables created in Studio
- Dgraph indexes
- Sample files in HDFS
- Configuration files

Note that the backup doesn't include transient data, like state in Studio. This information will be lost if you restore your cluster.

To back up your cluster, run the following from the Admin Server:

```
./bdd-admin.sh backup [option <arg>] <file>
```

Before running `backup`, verify the following:

- The `BDD_STUDIO_JDBC_USERNAME` and `BDD_STUDIO_JDBC_PASSWORD` environment variables are set. Otherwise, the script will prompt you for this information at runtime.
- The database client is installed on the Admin Server. For MySQL databases, this should be MySQL client. For Oracle databases, this should be Oracle Database Client, installed with a type of Administrator. Note that the Instant Client isn't supported.
- If you have an Oracle database, the `ORACLE_HOME` environment variable is set to the directory one level above the `/bin` directory that the `sqlplus` executable is located in. For example, if the `sqlplus` executable is located in `/u01/app/oracle/product/11/2/0/dbhome/bin`, `ORACLE_HOME` should be set to `/u01/app/oracle/product/11/2/0/dbhome/bin`.

You can specify the following options.

Option	Description
`-o, --offline`	Performs a cold backup. Use this option if your cluster is down. If this option isn't specified, the script will perform a hot backup. More information on hot and cold backups is available below.
`-r, --repeat` *<num>*	The number of times the script will repeat the backup process if verification fails. This is only used for hot backups. If this option isn't specified, the script will only make one attempt to back up your cluster. If it fails, you must rerun it. More information on verification is available below.
`-v, --verbose`	Enables debugging messages.

If you don't specify any options, the script will make one attempt to perform a hot backup, and won't output debugging messages.

You must provide the *<file>* argument, which defines the absolute path to the tar file the script will back up your cluster to. This file must not exist and its parent directory must be writable.

For more information on backing up your cluster, see *Backing up Big Data Discovery on page 52*.

Hot vs. cold backups

`backup` can perform both hot and cold backups:

- Hot backups are performed while your cluster is running. Specifically, they're performed on the first Managed Server (defined by `MANAGED_SERVERS` in `bdd.conf`), and require that the components on that node are running. This is `backup`'s default behavior.
- Cold backups are performed while your cluster is down. You must include `backup`'s `-o` option to perform a cold backup.

Verification

Because hot backups are performed while your cluster is running, it's possible for the data in the backups of the database, index, and sample files to become inconsistent. For example, something could be added to the index after the database was backed up, which would mean the two locations would contain different data.

To prevent this, `backup` verifies that the data in all three backups is consistent. If it isn't, the operation fails.

By default, `backup` only backs up and verifies the data once. However, you can configure it to repeat this process by including the `-r` *<num>* option, where *<num>* is the number of times to repeat the backup and verification steps. This increases the likelihood that the backup operation will succeed.

Note: It's unlikely that verification will fail the first time, so it's not necessary to repeat the process more than once or twice.

Examples

The following command performs a hot backup with debugging messages:

```
./bdd-admin.sh backup -v /tmp/bdd_backup1.tar
```

The following command performs a cold backup:

```
./bdd-admin.sh backup -o /tmp/bdd_backup2.tar
```

restore

You can restore your cluster from a backup tar file by running the `bdd-admin` script with the `restore` command.

Note: You can't run `restore` if `start`, `stop`, `restart`, or `backup` is currently running.

This it completely restores the following from backup:

- Studio database
- Schema and data for Hive tables created in Studio
- Dgraph indexes
- Sample files in HDFS

Note: The script makes a copy of the old index named */<index_path>/<index_name>*`.old.copy`. You should delete this if you decide to keep the restored version.

It also restores some of the configuration settings, but not all of them. See below for more information.

Before running `restore`, verify the following:

- The `BDD_STUDIO_JDBC_USERNAME` and `BDD_STUDIO_JDBC_PASSWORD` environment variables are set. Otherwise, the script will prompt you for this information at runtime.
- The database client is installed on the Admin Server. For MySQL databases, this should be MySQL client. For Oracle databases, this should be Oracle Database Client, installed with a type of Administrator. Note that the Instant Client isn't supported.

- If you have an Oracle database, the ORACLE_HOME environment variable is set to the directory one level above the /bin directory that the sqlplus executable is located in. For example, if the sqlplus executable is located in /u01/app/oracle/product/11/2/0/dbhome/bin, ORACLE_HOME should be set to /u01/app/oracle/product/11/2/0/dbhome/bin.

To restore your cluster, run the following from the Admin Server:

```
./bdd-admin.sh restore [option] <file>
```

You can specify the following options.

Option	Description
-v, --verbose	Enables debugging messages.

You must provide the *<file>* argument, which defines the absolute path to the tar file you want to restore from. This must be a tar file created by the backup command.

Additionally, the cluster that was backed up to this file and your current cluster must have the same major version of BDD. They must also have the same type of database, Oracle or MySQL (restore doesn't support Hypersonic databases.). Both clusters can have different topologies.

For more information on restoring your cluster, see *Restoring Big Data Discovery on page 53*.

Configuration restoration

restore can't completely restore the configuration files because the current cluster may have a different topology than the backup cluster. Instead, it merges some of them with the ones from the current cluster and leaves others unchanged.

The following table describes the changes the script makes to each configuration file.

File	Changes
bdd.conf	The script doesn't merge bdd.conf at all. The restored cluster will have the same version as the current cluster.
portal-ext.properties, esconfig.properties	The script adds any properties from the backup versions of these files that aren't in the current ones. It doesn't modify any other settings.
edp.properties	The script restores all settings that *don't* affect cluster topology. Note that all other Data Processing configuration files will be fully restored.

Examples

The following command restores your cluster from the /tmp/bdd_backup1.tar file with no debugging messages:

```
./bdd-admin.sh restore /tmp/bdd_backup1.tar
```

publish-config

You can publish configuration changes to your BDD cluster by running the `bdd-admin` script with the `publish-config` command.

To update your cluster's configuration, run the following from the Admin Server:

```
./bdd-admin.sh publish-config <config type> [option <arg>]
```

Note: After running the `publish-config` command, you must restart your BDD cluster for the changes to take effect.

You must specify one of the following configuration types.

Configuration type	Description
`bdd` *`<path>`*	Publishes the updated version of `bdd.conf` specified by *`<path>`* to all BDD nodes. See *bdd on page 36* for more information.
`hadoop` *`[option <arg>]`*	Publishes Hadoop configuration changes to all BDD nodes and performs any other operations defined by the specified options. See *hadoop on page 37* for more information.
`kerberos` *`<option <arg>>`*	Publishes the specified Kerberos principal, `krb5.conf` file, or keytab file to all BDD nodes. See *kerberos on page 38* for more information.

System management commands

bdd

hadoop

kerberos

bdd

You can use the `publish-config` command with the `bdd` configuration type to publish an updated version of `bdd.conf` to all BDD nodes. This updates the configuration of your entire cluster.

To update your BDD cluster's configuration, edit a copy of `bdd.conf` on the Admin Server, then run:

```
./bdd-admin.sh publish-config bdd <path>
```

Where *`<path>`* is the absolute path to the modified copy of `bdd.conf`.

When the script runs, it makes a backup of the original `bdd.conf` in `$BDD_HOME/BDD_manager/conf` on the Admin Server. The backup is named `bdd.conf.bak`*`<num>`*, where *`<num>`* is the number of the backup; for example, `bdd.conf.bak2`. You can use this file to revert your configuration changes, if necessary.

The script then copies the modified version of `bdd.conf` to all BDD nodes in the cluster.

When the script completes, you must restart the cluster for your changes to take affect.

Note: When you update `bdd.conf`, any component log levels you've set on specific nodes using the `set-log-levels` command will be overwritten by the `DGRAPH_LOG_LEVELS` property in the updated file.

For more information on updating your cluster configuration, see *Updating the cluster configuration on page 50*.

hadoop

You can use the `publish-config` command with the `hadoop` configuration type to make changes to BDD's Hadoop configuration.

Depending on the options you specify, you can use `hadoop` to:

- Publish new or updated Hadoop client configuration files to your BDD cluster.
- Reset the `HUE_URI` property in `bdd.conf` (this is only available for HDP clusters).
- Switch to a different version of your Hadoop distribution without having to reinstall BDD.

Note: You can't use this configuration type to switch to a different Hadoop distribution.

To update BDD's Hadoop configuration, run the following from the Admin Server:

```
./bdd-admin.sh publish-config hadoop [option <arg>]
```

You can specify the following options.

Option	Description
`-u, --hueuri` *`<host>:<port>`*	Sets the `HUE_URI` property in `bdd.conf` to the specified URI. This option is only available for HDP clusters.
`-l, --clientlibs` *`<path[,path]>`*	Regenerates the Hadoop fat jar from the specified the client libraries. *`<path[,path]>`* must be a comma-separated list of the new libraries. You must run this option with `--sparkjar`.
`-j, --sparkjar` *`<file>`*	Sets the location of the Spark on YARN jar in all BDD configuration files to the specified path. *`<file>`* must be the absolute path to the Spark on YARN jar on your Hadoop nodes. You must run this option with `--clientlibs`.

If you don't include any options, the script publishes the Hadoop client configuration files to all BDD nodes and updates the Hadoop-related properties in all BDD configuration files.

For more information on the actions performed by this configuration type, see:

- *Updating the Hadoop client configuration files on page 54*
- *Setting the Hue URI on page 55*
- *Switching Hadoop versions on page 56*

kerberos

You can use the `publish-config` command with the `kerberos` configuration type to make changes to BDD's Kerberos configuration.

Depending on the options you specify, you can use `kerberos` to do the following:

- Enable Kerberos
- Update the location of the `krb5.conf` file in BDD's configuration files
- Update the BDD principal
- Publish a new keytab file to all BDD nodes

To update BDD's Kerberos configuration, run the following from the Admin Server:

```
./bdd-admin.sh publish-config kerberos [operation] <option>
```

You can include one of the following operations.

Operation	Description
`on`	Enables Kerberos. When you run this operation, you must specify the `-k`, `-t`, and `-p` options.
`config`	Updates BDD's Kerberos configuration. When you run this operation, you must specify at least one option. This is the command's default behavior, so this operation is optional. You can only use this if Kerberos is already enabled.

You must include at least one of the following options.

Option	Description
`-k, --krb5 <file>`	Updates the location of the `krb5.conf` file in all BDD configuration files. `<file>` must be the absolute path to the file. Note that you must manually move the file to its new location on all BDD nodes before running this option.
`-t, --keytab <file>`	Publishes the specified keytab file to all BDD nodes. `<path>` must be the absolute path to the new keytab file. The script will rename this file `bdd.keytab` and copy it to `$BDD_HOME/common/kerberos`.
`-p, --principal <principal>`	Publishes the specified principal to all BDD nodes. Note that you can't use this option to change the primary component of the principal.

For more information on updating your Kerberos configuration, see *Updating BDD's Kerberos configuration on page 57*.

update-model

You can update or reset the models used by some of the Data Enrichment modules by running the bdd-admin script with the update-model command.

To update or reset the models used by the Data Enrichment modules, run the following command from the Admin Server:

```
./bdd-admin.sh update-model <model_type> [path]
```

You must specify a model type.

Model type	Description
geonames	The model for the GeoTagger Data Enrichment modules.
tfidf	The model for the TF.IDF Data Enrichment module.
sentiment	The model for the Sentiment Analysis Data Enrichment modules.

You can optionally specify *[path]*, which is the absolute path to the location of the files you want to update the model with. You must move these files to a single directory on the Admin Server before running the script.

If you include *[path]*, the script will create a jar from the files in the specified directory. It will then replace the model's current jar on the YARN worker nodes with the new one. If you don't include *[path]*, the script will reset the specified model to its original state.

For details on configuring the input directories and files for the models, see the *Data Processing Guide*.

Reverting model changes

You can revert the changes made to the models by running the script without the *[path]* argument.

For example, the following command resets the tfidf model:

```
./bdd-admin.sh update-model tfidf
```

flush

You can flush component caches by running the bdd-admin script with the flush command.

To flush component caches, run the following from the Admin Server:

```
./bdd-admin.sh flush [option <arg>]
```

You can specify the following options.

Option	Description
`-c, --component` *`<component(s)>`*	A comma-separated list of the component caches the script will flush: • `dgraph`: Dgraph • `gateway`: Dgraph Gateway If you are debugging query issues, you can approximate cold-start or post-update performance by cleaning the Dgraph cache before running a request.
`-n, --node` *`<hostname(s)>`*	A comma-separated list of the nodes the script will run on. Each must be defined in `bdd.conf`.

If you don't specify any options, the script will flush the caches of all supported components.

Examples

The following command flushes all Dgraph and Dgraph Gateway caches in the cluster:

```
./bdd-admin.sh flush
```

The following command flushes the Dgraph cache on the `web009.us.example.com` node:

```
./bdd-admin.sh flush -c dgraph -n web009.us.example.com
```

Diagnostics commands

You can use the `bdd-admin` script's diagnostics commands to perform such operations as checking the status of your cluster and retrieving component log files.

get-blackbox

You can generate the Dgraph's on-demand tracing blackbox file by running the `bdd-admin` script with the `get-blackbox` command. This returns the name and location of the file.

Note: This command is intended for use by Oracle Support.

To generate the Dgraph blackbox file, run the following from the Admin Server:

```
./bdd-admin.sh get-blackbox [option <arg>]
```

You can specify the following options.

Option	Description
`-n, --node` *`<hostname(s)>`*	A comma-separated list of the nodes the script will run on. Each must be defined in `bdd.conf`.

If you don't specify any options, the script will generate blackbox files for all Dgraph nodes in the cluster.

Examples

The following command generates blackbox files for all Dgraph nodes:

```
./bdd-admin.sh get-blackbox
```

The following generates a blackbox file for the Dgraph running on the `web009.us.example.com` node:

```
./bdd-admin.sh get-blackbox -n web009.us.example.com
```

status

You can check either component statuses or the overall health of your BDD cluster by running the `bdd-admin` script with the `status` command.

`status` can perform two types of checks:

- Ping, which returns the status (up or down) of the specified components. This is the command's default behavior.
- Health check, which returns the overall health of the cluster and the Hive Table Detector.

To check component statuses or cluster health, run the following from the Admin Server:

```
./bdd-admin.sh status [option <arg>]
```

You can specify the following options.

Option	Description
`-c, --component` *`<component(s)>`*	A comma-separated list of the components the script will run on: • `agent`: Dgraph HDFS Agent • `dgraph`: Dgraph • `dp`: Data Processing • `gateway`: Dgraph Gateway • `studio`: Studio
`-n, --node` *`<hostname(s)>`*	A comma-separated list of the nodes the script will run on. Each must be defined in `bdd.conf`.
`--health-check`	Returns the health of the cluster and the Hive Table Detector. When you specify this option, you can't include the `-c` or `-n` options. If the healthcheck fails, you can find more information on what went wrong in the Studio and Data Processing logs.

If you don't specify any options, the script will return the statuses of all supported components.

Examples

The following command returns the statuses of all supported components:

```
./bdd-admin.sh status
```

The following command returns the health of the cluster and the Hive Table Detector:

```
./bdd-admin.sh status --health-check
```

The output from the above command will be similar to the following:

```
[2015/08/04 11:38:54 -0400] [Admin Server] Checking the health of BDD cluster...
[2015/08/04 11:40:06 -0400] [web009.us.example.com] Check BDD functionality......Pass!
[2015/08
/04 11:40:08 -0400] [web009.us.example.com] Check Hive Data Detector health......Hive Data Detector
has previously run.
[2015/08/04 11:40:10 -0400] [Admin Server] Successfully checked statuses.
```

get-stats

You can obtain Dgraph statistics by running the `bdd-admin` script with the `get-stats` command.

Note: Statistics are intended for use by Oracle Support only.

To obtain the Dgraph statistics, run the following from the Admin Server:

```
./bdd-admin.sh get-stats [option <arg>] <dest>
```

You can specify the following options.

Option	Description
`-c, --component` *`<component(s)>`*	A comma-separated list of the components the script will run on: • `dgraph`: Dgraph
`-n, --node` *`<hostname(s)>`*	A comma-separated list of the nodes the script will run on. Each must be defined in `bdd.conf`.

If you don't specify any options, the script will obtain the statistics for all supported components.

You must provide the <*dest*> argument. This defines the absolute path to the directory the script will output the requested statistics to. When the script completes, this directory will contain a file named `<`*`hostname`*`>-<`*`timestamp`*`>-dgraph-stats.xml`.

For more information on Dgraph statistics, see *About Dgraph statistics on page 67*.

Examples

The following command outputs the statistics of all Dgraph instances in the cluster to the `/tmp` directory:

```
./bdd-admin.sh get-stats /tmp
```

The following command outputs the statistics of the Dgraph running on the `web009.us.example.com` node to the `/tmp` directory:

```
./bdd-admin.sh get-stats -n web009.us.example.com /tmp
```

reset-stats

You can reset Dgraph statistics by running the `bdd-admin` script with the `reset-stats` command.

Note: Statistics are intended for use by Oracle Support only.

To reset Dgraph statistics, run the following from the Admin Server:

```
./bdd-admin.sh reset-stats [option <arg>]
```

You can specify the following options.

Option	Description
`-c, --component` *`<component(s)>`*	A comma-separated list of the components the script will run on: • `dgraph`: Dgraph
`-n, --node` *`<hostname(s)>`*	A comma-separated list of the nodes the script will run on. Each must be defined in `bdd.conf`.

If you don't specify any options, the script will reset the statistics for all Dgraph instances in the cluster.

For more information on Dgraph statistics, see *About Dgraph statistics on page 67*.

Examples

The following command resets the statistics for all Dgraph instances in the cluster:

```
./bdd-admin.sh reset-stats
```

The following command resets the statistics for the Dgraph running on the `web009.us.example.com` node:

```
./bdd-admin.sh reset-stats -n web009.us.example.com
```

get-log-levels

You can obtain a list of component logs and their current levels by running the `bdd-admin` script with the `get-log-levels` command.

To obtain component log levels, run the following from the Admin Server:

```
./bdd-admin.sh get-log-levels [option <arg>]
```

You can specify the following options.

Option	Description
`-c, --component <component(s)>`	A comma-separated list of the components the script will run on: • `dgraph`: Dgraph • `dp`: Data Processing • `gateway`: Dgraph Gateway The `dgraph` option returns the current levels of all Dgraph out log subsystems. For more information, see *Dgraph out log on page 173*.
`-n, --node <`*`hostname(s)`*`>`	A comma-separated list of the nodes the script will run on. Each must be defined in `bdd.conf`.

If you don't specify any options, the script will return the current log levels for all supported components.

If the script completes successfully, its output will be similar to the following:

```
[2015/06/01 22:36:24 -0400] [Admin Server] Retrieving log levels...
[2015/06
/01 22:36:30 -0400] [web009.us.example.com] Retrieving Dgraph Gateway log level.......Success!
   Gateway                      : WARNING
[2015/06/01 22:36:33 -0400] [web009.us.example.com] Retrieving DP log level.......Success!
   DP                           : INCIDENT_ERROR
[2015/06/01 22:36:45 -0400] [web009.us.example.com] Retrieving Dgraph log levels.......Success!
All Dgraph log subsystems:
   background_merging           : ERROR
   bulk_ingest                  : ERROR
   cluster                      : WARNING
   datalayer                    : ERROR
   dgraph                       : ERROR
   eql                          : ERROR
   eve                          : WARNING
   http                         : ERROR
   lexer                        : ERROR
   splitting                    : ERROR
   ssl                          : ERROR
   task_scheduler               : ERROR
   text_search_rel_rank         : ERROR
   text_search_spelling         : ERROR
   update                       : ERROR
   workload_manager             : ERROR
   ws_request                   : ERROR
   xq_web_service               : ERROR

[2015/06/01 22:36:49 -0400] [Admin Server] Successfully retrieved all log levels.
```

Examples

The following command prints the current log levels of all supported components:

```
./bdd-admin.sh get-log-levels
```

The following command prints the current log level of the Dgraph Gateway running on the `web009.us.example.com` node:

```
./bdd-admin.sh get-log-levels -c gateway -n web009.us.example.com
```

set-log-levels

You can set component log levels by running the `bdd-admin` script with the `set-log-levels` command. This updates the log levels in the components' configuration files so that the levels persist if the components are restarted.

To set component log levels, run the following from the Admin Server:

```
./bdd-admin.sh set-log-levels [option <arg>]
```

You can specify the following options.

Option	Description
`-c, --component` *`<component(s)>`*	A comma-separated list of the components the script will run on: • `dgraph`: Dgraph • `dp`: Data Processing • `gateway`: Dgraph Gateway

Option	Description
`-s, --subsystem` *`<subsystem(s)>`*	A comma-separated list of the Dgraph out log subsystems the script will run on: • `background_merging` • `bulk_ingest` • `cluster` • `datalayer` • `dgraph` (Note that this is different from the `dgraph` component.) • `eql` • `eve` • `http` • `lexer` • `splitting` • `ssl` • `task_scheduler` • `text_search_rel_rank` • `text_search_spelling` • `update` • `workload_manager` • `ws_request` • `xq_web_service` You can only include this option when you run the script on the `dgraph` component. If you specify the `dgraph` component but don't provide any subsystems, the script will run on all supported subsystems. **Note:** If you set the levels of Dgraph log subsystems, the script will automatically update the `DGRAPH_LOG_LEVELS` property in `bdd.conf` accordingly. If you set them on specific nodes, `bdd.conf` will only be updated on those nodes. These settings will be overwritten if you later update the cluster configuration with the `update-config` command. For more information on the Dgraph out log and its subsystems, see *Dgraph out log on page 173*.

Option	Description
`-l, --level <level>`	The log level the script will set for the components: • `INCIDENT_ERROR` • `ERROR` • `WARNING` • `NOTIFICATION` • `TRACE` You can only specify one log level. If you don't provide this option, the script will set the specified component logs to `NOTIFICATION`.
`--non-persistent`	Indicates that the log levels will be reset when the components are restarted. If you provide this option, the script won't set the log levels in the component configuration files. This option is only available for the `dgraph` and `gateway` components. Data Processing log levels are always persistent.
`-n, --node <hostname(s)>`	A comma-separated list of the nodes the script will run on. Each must be defined in `bdd.conf`.

If you don't specify any options, the script will set the log levels of all supported components and Dgraph log subsystems to `NOTIFICATION`. These settings will persist if the components are restarted.

Examples

The following command sets the log levels of Data Processing and the Dgraph log subsystems `cluster` and `datalayer` to `WARNING`:

```
./bdd-admin.sh set-log-levels -c dgraph,dp -s cluster,datalayer -l WARNING
```

The following command sets the log levels of the Dgraph Gateway and all Dgraph subsystems to `ERROR`, which will not be persistent:

```
./bdd-admin.sh set-log-levels -c dgraph,gateway -l ERROR --non-peristent
```

get-logs

You can obtain component log files by running the `bdd-admin` script with the `get-logs` command. When the script runs, it collects the requested logs and compresses them to a single zip file.

To obtain components logs, run the following from the Admin Server:

```
./bdd-admin.sh get-logs [option <arg>] <file>
```

You can specify the following options.

Option	Description
`-t, --time` *`<hours>`*	If you specify this option, the script will return the logs that were modified within the last *`<hours>`* hours. If you don't specify this option, the script will only return the most recently updated log file for each component.
`-c, --component` *`<component(s)>`*	A comma-separated list of the component logs the script will collect: • `agent`: Dgraph HDFS Agent logs • `all`: All component logs • `dgraph`: Dgraph logs • `dg-on-crash`: Dgraph on-crash tracing logs • `dg-on-demand`: Dgraph on-demand tracing logs • `dp`: Data Processing logs • `gateway`: Dgraph Gateway logs • `spark`: Spark logs • `studio`: Studio logs • `weblogic`: WebLogic Server logs • `zk-log`: ZooKeeper logs • `zk-transaction`: ZooKeeper transaction logs Note the following: • If you run this command on the `spark`, `zk-log`, and `zk-transaction` components, you will be prompted for the username and password for Cloudera Manager/Ambari if the `BDD_HADOOP_UI_USERNAME` and `BDD_HADOOP_UI_PASSWORD` environment variables aren't set. • The `dg-on-demand` log is only generated when you run the `get-blackbox` command. This means that if you run `get-logs` with the `-t` option, it will only return the `dg-on-demand` log if `get-blackbox` was run during the specified time frame. And if you run it without the `-t` option, it won't return the `dg-on-demand` log if `get-blackbox` has never been run.
`-n, --node` *`<hostname(s)>`*	A comma-separated list of the nodes the script will run on. Each must be defined in `bdd.conf`.

If you don't provide any options, the script will obtain the most recently updated logs for all components except `dg-on-crash`, `dg-on-demand`, and `zk-transaction`.

When you run `get-logs`, you must provide the *`<file>`* argument, which defines the absolute path of the zip file the script will output the logs to. This file must not exist and must include the `.zip` file extension.

Examples

The following command obtains the most recently modified logs for all supported components and outputs them to `/localdisk/logs/all_logs.zip`:

```
./bdd-admin.sh get-logs -c all /localdisk/logs/all_logs.zip
```

The following command obtains all `zk-log` and `zk-transaction` logs modified within the last 24 hours and outputs them to `/localdisk/logs/zk_logs.zip`:

```
./bdd-admin.sh get-logs -t 24 -c zk-log,zk-transaction /localdisk/logs/zk_logs.zip
```

rotate-logs

You can rotate component logs by running the `bdd-admin` script with the `rotate-logs` command.

Note: This command is intended for use by Oracle Support only.

To rotate component logs, run the following from the Admin Server:

```
./bdd-admin.sh rotate-logs [option <arg>]
```

You can specify the following options.

Option	Description
`-c, --component <component(s)>`	A comma-separated list of the component logs the script will rotate: • `agent`: Dgraph HDFS Agent logs • `dgraph`: Dgraph logs • `gateway`: Dgraph Gateway logs • `studio`: Studio logs • `weblogic`: WebLogic Server logs
`-n, --node <hostname(s)>`	A comma-separated list of the nodes the script will run on. Each must be defined in `bdd.conf`.

If you don't specify any options, the script will rotate all supported component logs.

Examples

The following command rotates all supported component logs:

```
./bdd-admin.sh rotate-logs
```

The following command rotates the logs of the Dgraph and Dgraph HDFS Agent running on the `web009.us.example.com` node:

```
./bdd-admin.sh rotate-logs -c dgraph,agent -n web009.us.example.com
```

Chapter 4

Administering a Big Data Discovery Cluster

This section describes how to perform different administrative tasks for your BDD cluster deployment as a whole, such as backing it up and updating its configuration.

Updating the cluster configuration

You can make changes to BDD's configuration by editing a copy of `bdd.conf` and then running the `bdd-admin` script to distribute the changes to the rest of the cluster.

You can edit `bdd.conf` in any text editor. Note that you can't modify all of the properties in the file; for example, you can't change the lists of Dgraph and Managed Server nodes. For the full list of properties you can change, see *Configuration properties that can be modified on page 51*.

Note: When you update `bdd.conf`, any component log levels you've set on specific nodes using the `set-log-levels` command will be overwritten by the `DGRAPH_LOG_LEVELS` property in the updated file.

When the script runs, it backs up the original version of `bdd.conf` to `bdd.conf.bak<`*`num`*`>` so you can revert your changes, if necessary. It then copies the updated file to all BDD nodes.

To update your cluster configuration:

1. On the Admin Server, copy `bdd.conf` in `$BDD_HOME/BDD_manager/conf` to a different directory.
2. Open the copy in a text editor and make your desired changes.

   Be sure to save the file before closing.
3. Go to `$BDD_HOME/BDD_manager/bin` and run:

   ```
   ./bdd-admin.sh publish-config bdd <path>
   ```

   Where `<`*`path`*`>` is the absolute path to the modified copy of `bdd.conf`.
4. Restart your cluster so the changes take effect:

   ```
   ./bdd-admin.sh restart
   ```

   The above command will shut the cluster down gracefully, which may take a long time. You can optionally specify `-t <`*`minutes`*`>` to force a shutdown sooner.

Configuration properties that can be modified

The table below describes the properties in `bdd.conf` that you can modify. Be sure to read this information carefully before making changes to `bdd.conf`. Don't update any other properties in this file, as this could have negative effects on your cluster.

Property	Description
`DGRAPH_INDEX_DIR`	The path to the Dgraph index on the NFS. This location contains the directory defined by `DGRAPH_INDEX_NAME`. You must prepare the index files on the NFS before changing the value of this property.
`JAVA_HOME`	The JDK used when starting the BDD components. If you change this value, you must also update the location used by the CLI and Studio. Note that this must be in the same location on all nodes in the cluster.
`DGRAPH_THREADS`	The number of threads the Dgraph starts with. Oracle recommends the following: • For machines running only the Dgraph, the number of threads should be equal to the number of CPU cores on the machine. • For machines running the Dgraph and other BDD components, the number of threads should be the number of CPU cores minus 2. For example, a machine with 4 cores should have 2 threads. Be sure that the number you use is in compliance with the licensing agreement.
`DGRAPH_CACHE`	The Dgraph cache size, in MB. There is no default value for this property, so you must provide one. For enhanced performance, Oracle recommends allocating at least 50% of the node's available RAM to the Dgraph cache. If you later find that queries are getting cancelled because there is not enough available memory to process them, you should increase this amount.
`DGRAPH_INDEX_NAME`	The name of the Dgraph index, which is located in the directory defined by `DGRAPH_INDEX_DIR`. You must prepare the index files on the NFS before changing the value of this property.
`DGRAPH_OUT_FILE`	The path to the Dgraph's stdout/stderr file.
`DGRAPH_LOG_LEVEL`	Optional. Defines the log levels for the Dgraph's out log subsystems. This must be in the format `"subsystem1 level1|subsystem2 level2|subsystemN levelN"` (including quotes). You can include as many subsystems as you want. Any you don't include will be set to `NOTIFICATION`. For more information on the Dgraph's out log subsystems and their supported levels, see *Dgraph out log on page 173*.

Property	Description
DGRAPH_ADDITIONAL_ARG	**Note:** This property is only intended for use by Oracle Support. Defines one or more flags to start the Dgraph with. Each flag must be quoted. Note that you cannot include flags that map to properties in bdd.conf. For more information on Dgraph flags, see *Dgraph flags on page 67*.
AGENT_OUT_FILE	The path to the HDFS Agent's stdout/stderr file.

Backing up Big Data Discovery

Because Big Data Discovery doesn't perform automatic backups, you must back up your system manually. Oracle recommends that, at a minimum, you back up your cluster immediately after deployment.

You back up your cluster by running the bdd-admin script with the backup command. This backs up the following data to a single tar file, which you can later use to restore your cluster:

- Studio database
- Schema and data for Hive tables created in Studio
- Dgraph indexes
- Sample files in HDFS
- Cluster configuration files

Note: The script doesn't back up transient data, like state in Studio. This information not be available in a restored cluster.

Before you back up your cluster, verify the following:

- The BDD_STUDIO_JDBC_USERNAME and BDD_STUDIO_JDBC_PASSWORD environment variables are set. Otherwise, the script will prompt you for this information at runtime.
- The database client is installed on the Admin Server. For MySQL databases, this should be MySQL client. For Oracle databases, this should be Oracle Database Client, installed with a type of Administrator. Note that the Instant Client isn't supported.
- If you have an Oracle database, the ORACLE_HOME environment variable is set to the directory one level above the /bin directory that the sqlplus executable is located in. For example, if the sqlplus executable is located in /u01/app/oracle/product/11/2/0/dbhome/bin, ORACLE_HOME should be set to /u01/app/oracle/product/11/2/0/dbhome/bin.

For more information on backup and its supported options, see *backup on page 32*. For instructions on restoring your cluster, see *Restoring Big Data Discovery on page 53*.

To back up BDD:

1. On the Admin Server, go to $BDD_HOME/BDD_manager/bin.

2. Run one of the following commands:
   - If your cluster is running:

     ```
     ./bdd-admin.sh backup -v <file>
     ```

   - If your cluster is down:

     ```
     ./bdd-admin.sh backup -o -v <file>
     ```

   Where `<file>` is the absolute path to the tar file the script will back up your cluster to. This file must not exist and its parent directory must be writable.

   The `-v` flag enables debugging messages. This is optional but recommended because the script might take a long time to finish and the output will keep you informed about its current status.

3. If you haven't set the `STUDIO_JDBC_USERNAME` and `STUDIO_JDBC_PASSWORD` environment variables, enter the database username and password when prompted.

Restoring Big Data Discovery

You can restore your cluster from a backup tar file by running the `bdd-admin` script with the `restore` command.

Before restoring your cluster, you should verify that:

- You have access to a backup tar file created by the `backup` command.
- Your current cluster and the backup cluster both have the same major version of BDD.
- Both clusters have the same type of database, Oracle or MySQL. Note that `restore` doesn't support Hypersonic databases.
- The `BDD_STUDIO_JDBC_USERNAME` and `BDD_STUDIO_JDBC_PASSWORD` environment variables are set. Otherwise, the script will prompt you for this information at runtime.
- The database client is installed on the Admin Server. For MySQL databases, this should be MySQL client. For Oracle databases, this should be Oracle Database Client, installed with a type of Administrator. Note that the Instant Client isn't supported.
- If you have an Oracle database, the `ORACLE_HOME` environment variable is set to the directory one level above the `/bin` directory that the `sqlplus` executable is located in. For example, if the `sqlplus` executable is located in `/u01/app/oracle/product/11/2/0/dbhome/bin`, `ORACLE_HOME` should be set to `/u01/app/oracle/product/11/2/0/dbhome/bin`.

Your current cluster can have a different topology than the backup cluster. For example, node IP addresses, the total number of nodes, and the locations of the BDD components can be different between the two.

When the script runs, it restores the database, Hive tables created in Studio, Dgraph index, and sample files from backup.

Note that the script doesn't completely restore the configuration files from backup—it merges them with the current cluster's configuration files. The restored cluster will contain some of the backup cluster's configuration, but most of it will be from the current cluster.

For more information on the `restore` command, see *restore on page 34*.

Important: The script will overwrite the data on your current cluster with the backed up data and won't roll the restoration back if it fails. Because of this, if your current cluster contains any important data, you should back it up before restoring.

To restore your cluster:

1. On the Admin Server, go to `$BDD_HOME/BDD_manager/bin`.
2. Stop your cluster if it's running:

   ```
   ./bdd-admin.sh stop
   ```

   The above command will shut the cluster down gracefully, which may take a long time. You can optionally specify `-t <minutes>` to force a shut down sooner.
3. Run the `restore` command:

   ```
   ./bdd-admin.sh restore <file>
   ```

   Where `<file>` is the absolute path to the backup tar file you want to restore from.
4. If you haven't set the `STUDIO_JDBC_USERNAME` and `STUDIO_JDBC_PASSWORD` environment variables, enter the database username and password when prompted.
5. When the script finishes running, restart your cluster so the changes take effect:

   ```
   ./bdd-admin.sh restart
   ```

   The above command will shut the cluster down gracefully, which may take a long time. You can optionally specify `-t <minutes>` to force a shut down sooner.

When the script runs, it makes a copy of the current Dgraph index named `/<index_path>/<index_name>.old.copy`. You should delete this if you decide to keep the restored version of the index.

Updating BDD's Hadoop configuration

You can update your BDD cluster's Hadoop configuration with the `bdd-admin` script.

Updating the Hadoop client configuration files

Setting the Hue URI

Switching Hadoop versions

Updating the Hadoop client configuration files

If you update your Hadoop client configuration files, you can publish your changes to BDD with the `bdd-admin` script. This distributes the Hadoop client configuration files to all BDD nodes and updates the relevant properties in BDD's configuration files.

When the script runs, it obtains the Hadoop client configuration files from Cloudera Manager/Ambari, then updates the following:

- All Hadoop properties in `bdd.conf`
- The `zookeeper-servers` property in `EndecaServer.properties`
- The following properties in Studio's `portal-ext.properties` file:
  - `dp.settings.hadoop.cluster.host`

  - dp.settings.hive.metastore.port
  - dp.settings.namenode.port
  - dp.settings.hive.jdbc.port
  - dp.settings.hue.http.port
- The following properties in Data Processing's edp.properties:
  - hiveServerHost
  - hiveServerPort

When the script finishes running, you must restart your cluster for the changes to take effect.

To update your cluster's Hadoop client configuration files:

1. On the Admin Server, go to $BDD_HOME/BDD_manager/bin and run:

   ```
   ./bdd-admin.sh publish-config hadoop
   ```

2. Restart your cluster so the changes take effect:

   ```
   ./bdd-admin.sh restart
   ```

   The above command will shut the cluster down gracefully, which may take a long time. You can optionally specify -t <*minutes*> to force a shutdown sooner.

Setting the Hue URI

If you have an HDP cluster, you can use the bdd-admin script to update the URI of the node running Hue in bdd.conf.

When the script runs, it sets the HUE_URI property in bdd.conf to the hostname and port you specify. It also updates your cluster's Hadoop configuration files and performs the steps described in *Updating the Hadoop client configuration files on page 54*.

After the script finishes, you must restart your cluster for the changes to take effect.

To update the Hue URI:

1. On the Admin Server, go to $BDD_HOME/BDD_manager/bin and run:

   ```
   ./bdd-admin.sh publish-config hadoop --hueuri <hostname>:<port>
   ```

   Where <*hostname*> and <*port*> are the fully qualified domain name and port number of the node running Hue.

2. Restart your cluster so the changes take effect:

   ```
   ./bdd-admin.sh restart
   ```

   The above command will shut the cluster down gracefully, which may take a long time. You can optionally specify -t <*minutes*> to force a shutdown sooner.

Switching Hadoop versions

If you upgrade to a new version of your Hadoop distribution, you need to update your BDD cluster to integrate with it. You can do this using the `bdd-admin` script.

Before you run the script, you must obtain the new Hadoop client libraries for your distribution and move them to the Admin Server. When the script runs, it uses these libraries to generate a new fat jar, which it then distributes to all BDD nodes.

The script also obtains and distributes the new Hadoop client configuration files as described in *Updating the Hadoop client configuration files on page 54*.

Note: You can't use `bdd-admin` to switch to a different Hadoop distribution. For example, you can upgrade from CDH 5.3.2 to CDH 5.4, but not to HDP 2.3.

To switch to a different Hadoop version:

1. Stop your BDD cluster by running the following from `$BDD_HOME/BDD_manager/bin` on the Admin Server:

   ```
   ./bdd-admin.sh stop [-t <minutes>]
   ```

2. Upgrade your Hadoop cluster according to the instructions in your distribution's documentation.
3. Verify that any configuration changes you made prior to installing BDD (for example, to your YARN settings) weren't reset during the upgrade.

   Additionally, if you have HDP:

   (a) In `mapred-site.xml`, replace all instances of `${hdp.version}` with your HDP version number.

   (b) In `hive-site.xml`, remove s from the values of the following properties:

   - `hive.metastore.client.connect.retry.dealay`
   - `hive.metastore.client.cocket.timeout`

4. Obtain the client libraries for the new version of your Hadoop distribution and put them on the Admin Server.

   The location you put them in is arbitrary, as you will provide the `bdd-admin` script with their paths at runtime.

   - If you have a CDH cluster, download the following packages from http://archive-primary.cloudera.com/cdh5/cdh/5/ and unzip them:
     - `spark-<spark_version>.cdh.<cdh_version>.tar.gz`
     - `hive-<hive_version>.cdh.<cdh_version>.tar.gz`
     - `hadoop-<hadoop_version>.cdh.<cdh_version>.tar.gz`
     - `avro-<avro_version>.cdh.<cdh_version>.tar.gz`
   - If you have an HDP cluster, copy the following directories from your Hadoop nodes to the Admin Server:
     - `/usr/hdp/<version>/pig/lib/h2/`
     - `/usr/hdp/<version>/hive/lib/`

- `/usr/hdp/<version>/spark/lib/`
- `/usr/hdp/<version>/spark/external/spark-native-yarn/lib/`
- `/usr/hdp/<version>/hadoop/`
- `/usr/hdp/<version>/hadoop/lib/`
- `/usr/hdp/<version>/hadoop-hdfs/`
- `/usr/hdp/<version>/hadoop-hdfs/lib/`
- `/usr/hdp/<version>/hadoop-yarn/`
- `/usr/hdp/<version>/hadoop-yarn/lib/`
- `/usr/hdp/<version>/hadoop-mapreduce/`
- `/usr/hdp/<version>/hadoop-mapreduce/lib/`

5. Start your BDD cluster:

   ```
   ./bdd-admin.sh start
   ```

6. Run the following up update BDD's Hadoop configuration:

   ```
   ./bdd-admin.sh publish-config hadoop -l <path[,path]> -j <file>
   ```

   `<path[,path]>` is a comma-separated list of the absolute paths to each of the client libraries on the Admin Server. For HDP clusters, the libraries *must* be specified in the order they are listed in above.

   `<file>` is the absolute path to the Spark on YARN jar on your Hadoop nodes.

7. Restart your cluster so the changes take effect:

   ```
   ./bdd-admin.sh restart
   ```

   The above command will shut the cluster down gracefully, which may take a long time. You can optionally specify `-t <minutes>` to force a shutdown sooner.

Updating BDD's Kerberos configuration

You can update your BDD cluster's Kerberos configuration with the `bdd-admin` script.

Enabling Kerberos

Changing the location of the Kerberos krb5.conf file

Updating the Kerberos keytab file

Updating the Kerberos principal

Enabling Kerberos

BDD supports Kerberos 5+ to authenticate its communications with Hadoop. You can enable this for BDD to improve the security of your cluster and data.

Before you can configure Kerberos for BDD, you must install it on your Hadoop cluster. If your Hadoop cluster already uses Kerberos, you must enable it for BDD so it can access the Hive tables it requires.

To enable Kerberos:

1. Install the `kinit` and `kdestroy` utilities on all BDD nodes.
2. Create the following directories in HDFS:
   - `/user/<BDD user>`, where `<BDD user>` is the name of the Linux user that runs all BDD processes.
   - `/user/<HDFS_DP_USER_DIR>`, where `<HDFS_DP_USER_DIR>` is the value of `HDFS_DP_USER_DIR` defined in `bdd.conf`.

   The owner of both directories must be the Linux user that runs all BDD processes, and their group must be `supergroup`.
3. Add the Linux user that runs all BDD processes to the `hdfs` and `hive` groups on all BDD nodes.
4. If you use HDP, add the group that the BDD user belongs to to the `hadoop.proxyuser.hive.groups` property in `core-site.xml`.

   You can do this in Ambari.
5. Create a principal for BDD.

   The primary component must be the name of the Linux user that runs all BDD processes and the realm must be your default realm.
6. Generate a keytab file for the BDD principal and move it to the Admin Server.

   The name and location of this file are arbitrary as you will pass this information to the `bdd-admin` script at runtime.
7. Copy your `krb5.conf` file to the same location on all BDD nodes.

   The location is arbitrary, but the default is `/etc`.
8. On the Admin Server, go to `$BDD_HOME/BDD_manager/bin` and run:

   ```
   ./bdd-admin.sh publish-config kerberos on -k <krb5> -t <keytab> -p <principal>
   ```

   Where:

   - `<krb5>` is the absolute path to `krb5.conf` on all BDD nodes
   - `<keytab>` is the absolute path to the BDD keytab file on the Admin Server
   - `<principal>` is the BDD principal

   The script updates BDD's configuration files with the name of the principal and the location of the `krb5.conf` file. It also renames the keytab file to `bdd.keytab` and distributes it to `$BDD_HOME/common/kerberos` on all BDD nodes.
9. If you use HDP, publish the change you made to `core-site.xml`:

   ```
   ./bdd-admin.sh publish-config hadoop
   ```
10. Restart your cluster for the changes to take effect:

    ```
    ./bdd-admin.sh restart
    ```

    The above command will shut the cluster down gracefully, which may take a long time. You can optionally specify `-t <minutes>` to force a shutdown sooner.

Once Kerberos is enabled, you can use the `bdd-admin` script to update its configuration as needed. For more information, see *kerberos on page 38*.

Changing the location of the Kerberos krb5.conf file

If you want to change the location of the `krb5.conf` file, you can use the `bdd-admin` script to update BDD's configuration accordingly.

You must provide the script with the absolute path to the `krb5.conf` file on all BDD nodes. When it runs, it updates the location of `krb5.conf` in BDD's configuration files.

For more information on updating your Kerberos configuration with `bdd-admin`, see *kerberos on page 38*.

To change the location of the `krb5.conf` file:

1. On all BDD nodes, move the `krb5.conf` file to the new location.

   The location is arbitrary, but must be the same on all nodes.

2. On the Admin Server, go to `$BDD_HOME/BDD_manager/bin` and run:

   ```
   ./bdd-admin.sh kerberos -k <file>
   ```

   Where `<file>` is the new absolute path to `krb5.conf`.

3. Restart your cluster so the changes take effect:

   ```
   ./bdd-admin.sh restart
   ```

   The above command will shut the cluster down gracefully, which may take a long time. You can optionally specify `-t <minutes>` to force a shut down sooner.

Updating the Kerberos keytab file

If you update BDD's current keytab file or create a new one, you can use the `bdd-admin` script to publish the new or updated file to the rest of the cluster.

When you run the script, you must provide it with the absolute path to the new or modified file. The script renames the specified file to `bdd.keytab` (if necessary) and copies it to `$BDD_HOME/common/kerberos` on all nodes.

For more information on updating your Kerberos configuration with the `bdd-admin` script, see *kerberos on page 38*.

To update the keytab file:

1. On the Admin Server, edit the current BDD keytab file or create a new one.

   The current file is named `bdd.keytab` and located in `$BDD_HOME/common/kerberos`.

2. Go to `$BDD_HOME/BDD_manager/bin` and run:

   ```
   ./bdd-admin.sh publish-config kerberos -t <file>
   ```

   Where `<path>` is the absolute path to the new or modified keytab file.

3. Restart your cluster so the changes take effect:

   ```
   ./bdd-admin.sh restart
   ```

   The above command will shut the cluster down gracefully, which may take a long time. You can optionally specify `-t <minutes>` to force a shutdown sooner.

Updating the Kerberos principal

If you edit the BDD principal or create a new one, you can use the `bdd-admin` script to publish your changes to the rest of the cluster.

When the script runs, it updates the following properties with the new or modified principal:

- `KERBEROS_PRINCIPAL` in `bdd.conf`
- `krb5.principal` in Studio's `portal-ext.properties` file
- `localKerberosPrincipal` and `clusterKererosPrincipal` in the `data_processing_CLI` file

Note: You can't change the primary component of the principal.

For more information on updating your Kerberos configuration with the `bdd-admin` script, see *kerberos on page 38*.

To update the Kerberos principal:

1. On the Admin Server, edit the current BDD principal or create a new one.

   Be sure to keep the primary component of the principal the same as the original.

2. Go to `$BDD_HOME/BDD_manager/bin` and run:

   ```
   ./bdd-admin.sh publish-config kerberos -p <principal>
   ```

   Where `<principal>` is the name of the new or modified principal.

3. Restart your cluster so the changes take effect:

   ```
   ./bdd-admin.sh restart
   ```

   The above command will shut the cluster down gracefully, which may take a long time. You can optionally specify `-t <minutes>` to force a shutdown sooner.

Part III

Administering the Dgraph and Dgraph Gateway

Chapter 5

The Dgraph

This section describes the Dgraph, its administrative operations, and flags. It also describes various Dgraph characteristics and behavior, such as memory consumption, Dgraph cache, and managing the Dgraph core dump files.

About the Dgraph

The Dgraph is a component of Big Data Discovery that runs search analytical processing of the data sets. It handles requests users make to data sets.

The Dgraph uses data structures and algorithms to provide real-time responses to client requests for analytic processing and data summarization. The Dgraph stores the indexes created after source data is loaded into Big Data Discovery. After the index is stored, the Dgraph receives client requests through Studio, queries the indexes, and returns the results.

An Oracle Big Data Discovery cluster has one or more Dgraph processes that handle end-user query requests accessing the index on shared storage. One of the Dgraphs in a Big Data Discovery cluster is the leader and therefore responsible for handling all write operations (updates, configuration changes), while the remaining Dgraphs serve as read-only followers.

Dgraph Tracing Utility

The Dgraph Tracing Utility is a Dgraph diagnostic program used by Oracle Support. It stores the Dgraph trace data, which are useful in troubleshooting the Dgraph. It starts when the Dgraph starts, and keeps track of all Dgraph operations. It stops when the Dgraph shuts down. You can save and download trace data to share it with Oracle Support.

The Tracing Utility stores the Dgraph target trace data it collects in `*.ebb` files, which are useful in analyzing Dgraph crashes. The files are intended for use by Oracle Support. The files are saved in the `$DGRAPH_HOME/bin` directory. You can also manually save the trace data, as described in *Saving trace data for the Dgraph on page 155*.

Additionally, you can download the `*.ebb` files, as described in *Downloading Dgraph trace files on page 156*.

Memory consumption by the Dgraph

This topic discusses the logic used by the Dgraph to control its memory consumption.

The Dgraph query performance depends on characteristics of your specific deployment: query workload and complexity, the characteristics of the loaded records, and the size of the index.

These statements describe how the Dgraph utilizes memory:

- After the installation, when the Dgraph is started it allocates considerable amounts of virtual memory on the system. This is needed for ingesting data and executing queries, including those that are complex. This is an expected behavior and is observable if you use system diagnostic tools.
- If the Dgraph is installed on a machine that is hosting other processes, other memory-intensive processes are present in the operating system and require memory. In this case, the Dgraph releases a significant portion of its physical memory quickly. Without such pressure, that is in cases when the Dgraph is the sole process on the hosting machine, the Dgraph may retain the physical memory indefinitely. This is an expected behavior.

  Because of this, depending on your deployment requirements, such as the size of your deployment, it may be highly desirable to deploy the Dgraph instances on servers dedicated solely to each of the Dgraph processes (this means that these machines are not hosting any other processes, for BDD or other applications).
- By default, the memory limit that the Dgraph is allowed to use on the machine is set to 80% of the machine's available RAM. This behavior ensures that the Dgraph does not run out of memory on the machine hosting the Dgraph. In other words, with this limit in place, the Dgraph is protected from running into out-of-memory performance issues.
- In addition to the default memory consumption limit of 80% of RAM, after the installation you can set a custom limit on the amount of memory the Dgraph can consume, using the Dgraph `--memory-limit` flag. If this limit is set, then, upon the Dgraph restart, the amount of memory required by the Dgraph to process all current queries cannot exceed this custom limit.

  Note: The Dgraph `--memory-limit` flag is intended for Oracle Support. For information on how to set it, see *Setting Dgraph memory consumption limit on page 64*. Also, a value of `0` for the flag means there is no limit set on the amount of memory the Dgraph can use. In this case, you should be aware that the Dgraph will use all the memory on the machine that it can allocate for its processing without any limit, and will not attempt to cancel any queries that may require the most amount of memory. This, in turn, may lead to out-of-memory page thrashing and require manually restarting the Dgraph.
- Once the Dgraph reaches a memory consumption limit (it could be the default limit of 80% of RAM, or a custom memory limit set with `--memory-limit`), it starts to automatically cancel queries, beginning with the query that is currently consuming the most amount of memory. When the Dgraph cancels a query, it logs the amount of memory the query was using and the time it was cancelled for diagnostic purposes.

- In addition to the memory consumption limit, before you install Big Data Discovery, you can specify the Dgraph cache size, using the `DGRAPH_CACHE` property in the `bdd.conf` file located in your installation directory. The orchestration script uses this value at installation time. You can adjust the size of `DGRAPH_CACHE` later, at any point after the installation. For information, see *Setting the Dgraph internal cache size on page 65*.
- There is one additional consideration about the Dgraph cache that is useful to keep in mind, before you decide to adjust the cache size:

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

Setting Dgraph memory consumption limit

It is possible to specify the custom memory limit the Dgraph is allowed to use for processing. If the memory limit is changed, this overrides the default memory consumption setting in the Dgraph that is set to 80% of the machine's available RAM.

Note: It is recommended that Oracle Support change the limit on Dgraph memory consumption.

By default, the memory limit that the Dgraph is allowed to use is 80% of the machine's available RAM. This behavior ensures that the Dgraph never runs out of memory during the course of its query processing or data ingest activity.

You can override the default limit and set a custom limit on the amount of memory the Dgraph can consume in MB, using the `--memory-limit` flag. If this value is set, then the amount of memory required by the Dgraph to process all current queries can't exceed this limit.

Once the Dgraph reaches a memory consumption limit set with this flag, then, similar to how it behaves with the default memory limit of 80%, the Dgraph starts to cancel queries, beginning with the query that is consuming the most amount of memory. When the Dgraph cancels a query, it logs the amount of memory the query was using and the time it was cancelled for diagnostic purposes.

The Dgraph `--memory-limit` can be set after the installation through the `DGRAPH_ADDITIONAL_ARG` parameter in the `bdd.conf` file in the `$BDD_HOME/BDD_manager/conf` directory.

Using the `--memory-limit` flag with a value of `0` means there is no limit set on the amount of memory the Dgraph can use.

For information on all Dgraph flags, see *Dgraph flags on page 67*.

To change the memory limit:

1. Go to `$BDD_HOME/BDD_manager/conf` directory and locate the `bdd.conf` file.
2. In the setting for `DGRAPH_ADDITIONAL_ARG`, specify the `--memory-limit` flag.
3. Save the `bdd.conf` file.

4. Run the `bdd-admin.sh publish-config bdd` command.

   This refreshes the configuration on all the Dgraph hosting machines with the modified settings from the `bdd.conf` file. For information on how to do this, see *Updating the cluster configuration on page 50*.

5. Restart the Dgraph with the `bdd-admin.sh` script.

Setting the Dgraph internal cache size

The Dgraph cache size should be configured to be large enough to allow the Dgraph to operate smoothly under normal query load.

For enhanced performance, Oracle recommends allocating at least 50% of the node's available RAM to the Dgraph cache. This is a significant amount of memory that you can adjust if needed. For example, if you later find that queries are getting cancelled because there is not enough available memory to process them, you should decrease this amount.

You configure the Dgraph cache size initially by setting the `DGRAPH_CACHE` value in the `bdd.conf` file in the installation directory. The orchestration script uses this value during the BDD installation process.

After the installation, you can adjust the size of the Dgraph cache by gradually changing the `DGRAPH_CACHE` value in the `bdd.conf` file in the `$BDD_HOME/BDD_manager/conf` directory and use the `bdd-admin publish-config` command to update the configuration for the entire cluster. For more information, see *publish-config on page 36*.

Before you adjust the Dgraph cache, keep the following consideration in mind:

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

This means that an occasional spike in Dgraph cache usage should not be the cause of alarm and that you should only consider adjusting the Dgraph cache size after observing Dgraph performance over longer periods of time.

Linux ulimit settings for merges

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

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

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

```
ERROR 04/03/13 05:24:35.668 UTC (1364966675668) DGRAPH {dgraph} BackgroundMergeTask:
exception thrown: Can't parse generation file, caused by I/O Exception: While mapping file,
caused by mmap failure: Cannot allocate memory
```

In this case, the problem is caused because the Dgraph cannot allocate enough virtual memory for its merging task.

Managing Dgraph core dump files

In the rare case of a Dgraph crash, the Dgraph writes its core dump files on disk. It is recommended to use the `ulimit -c unlimited` setting for the Dgraph core dump files. Non-limited core files contain all Dgraph data that is resident in memory (RSS of the Dgraph process).

When the Dgraph runs on a very large data set, the size of its index files stored in-memory may exceed the size of the physical RAM. If such a Dgraph fails, it may need to write out potentially very large core dump files on disk. The core files are written to the directory from which the Dgraph was started.

To troubleshoot the Dgraph, it is often useful to preserve the entire set of core files written out as a result of such failures. When there is not enough disk space, only a portion of the files is written to disk until this process stops. Since the most valuable troubleshooting information is contained in the last portion of core files, to make these files meaningful for troubleshooting purposes, it is important to provision enough disk space to capture the files in their entirety.

Two situations are possible, depending on your goal:

- You can afford to provision enough disk space.

  Large applications may take up the entire amount of available RAM. Because of this, the Dgraph core dump files can also grow large and take up the space equal to the size of the physical RAM on disk plus the size of the server data files in memory. To troubleshoot a Dgraph crash, provision enough disk space to capture the entire set of core files. In this case, the files are saved at the expense of potentially filling up the disk.

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

- You would like to limit the amount of disk space allotted for saving core files.

  To prevent filling up the disk, you can limit the size of these files on the operating system level, with the `ulimit -c <size>` command, although this is not recommended. If you set the limit size in this way, the core files cannot be used for debugging, although their presence will confirm that the Dgraph had crashed. In this case, with large Dgraph applications, only a portion of core files is saved on disk. This may limit their usefulness for debugging purposes. To troubleshoot the crash in this case, change this setting to `ulimit -c unlimited`, and reproduce the crash while capturing the entire core file. Similarly, to enable support to troubleshoot the crash, you will need to reproduce the crash while capturing the full core file.

Appointing a new Dgraph leader node

You can use the `appointNewDgraphLeader.sh` script to appoint a new Dgraph leader.

The use case for this script is when there is a long-running ingest in progress in the Dgraph HDFS Agent, and the Dgraph goes down for some reason. Instead of waiting until a new write request comes in, the administrator can just run this script to restart the ingest on another machine. (A file is maintained in HDFS that logs the exact progress of the ingest. The newly-appointed Dgraph HDFS Agent leader reads the file and knows at what point to pick up the ingest).

For example, the Dgraph HDFS Agent on Dgraph_A is performing an ingest when the Dgraph crashes (which results in the ingest being suspended). When the script is run, the new leader can be Dgraph_B, in which case the ingest is picked up at the point when it was stopped (except that Dgraph_B is now performing the ingest instead of Dgraph_A). Because there is only one index shared among the Dgraphs, the ingest can be resumed by the new leader.

Note that if the script is run but a new leader has been appointed in the interim, then the script basically reappoints the same leader.

The syntax for running the script is:

```
./appointNewDgraphLeader.sh <dg_address>
```

where *dg_address* is the FQDN (fully-qualified domain name) and port of the Dgraph Gateway server. For example:

```
./appointNewDgraphLeader.sh web009.us.example.com:7003
```

To appoint a new Dgraph leader:

1. Navigate to the `$DGRAPH_HOME/dgraph-hdfs-agent/bin` directory.
2. Run the `appointNewDgraphLeader.sh` script with the FQDN and port of the Dgraph Gateway, as in the example above.

If a new Dgraph leader is successfully appointed, the script returns this message:

```
New Dgraph Leader appointed
```

An unsuccessful operation could return either of these messages:

```
Unable to appoint new Dgraph leader

Could not reach Dgraph gateway
```

Note that an unsuccessful attempt could be caused by an incorrect address.

About Dgraph statistics

The Dgraph statistics page provides information such as startup time, host, port, and process information, data and log paths, and so on. This information is useful to help to tune your Dgraph and useful for Oracle Support.

The statistics page information is valid as long as the Dgraph is running; it is reset upon a Dgraph restart or by resetting the statistics page.

You can view and reset the Dgraph statistics page using one of these utilities:

- Using the `bdd-admin` script: *get-stats on page 42* and *reset-stats on page 43*
- Using Enterprise Manager: *Viewing Dgraph statistics on page 153* and *Resetting Dgraph statistics on page 154*.

Dgraph flags

Dgraph flags modify the Dgraph's configuration and behavior.

Important: Dgraph flags are intended for use by Oracle Support only. They are included in this document for completeness.

You can set Dgraph flags by adding them to the `DGRAPH_ADDITIONAL_ARG` property in `bdd.conf` in `$BDD_HOME/BDD_manager/conf` directory, then using the `bdd-admin publish-config` script to update the cluster configuration. Any flag included in this list will be set each time the Dgraph starts. For more information, see *publish-config on page 36*.

Note: Some of the Dgraph flags have the same names as HDFS Agent flags. These must have the same settings as their HDFS Agent counterparts.

Flag	Description
?	Prints the help message and exits. The help message includes usage information for each Dgraph flag.
`-v`	Enables verbose mode. The Dgraph will print information about each request it receives to either its stdout/stderr file (`dgraph.out`) or the file set by the `--out` flag.
`--backlog-timeout`	Specifies the maximum number of seconds that a query is allowed to spend waiting in the processing queue before the Dgraph responds with a timeout message. The default is `0` seconds.
`--bulk_load_port`	Sets the port on which the Dgraph listens for bulk load ingest requests. This must be the same as the port specified for the HDFS Agent `--bulk_load_port` flag. This flag maps to the `DGRAPH_BULKLOAD_PORT` property in `bdd.conf`.
`--cluster_identity`	Specifies the cluster identity of the Dgraph running on this node. The syntax is: `protocol:hostname:dgraph_port:dgraph_bulk_load_port:agent_port` This must be the same as the cluster identity specified for the HDFS Agent `--custer_identity` flag.
`--cmem`	Specify the maximum memory usage (in MB) for the Dgraph cache. For more information, see *Setting the Dgraph internal cache size on page 65*. This flag maps to the `DGRAPH_CACHE` property in `bdd.conf`.
`--coordinator`	Specifies the host and port that ZooKeeper is running on. The syntax is: `<hostname>:<port>` This must be the same as the value specified for the HDFS Agent `--coordinator` flag.
`--coordinator_auth`	Obtains the ZooKeeper authentication password from stdin.

Flag	Description
`--coordinator_index`	Specifies the index of the Dgraph cluster in the ZooKeeper ensemble. ZooKeeper uses this value to identify the Dgraph cluster. This must be the same as the value specified for the HDFS Agent `--coordinator_index` flag. This flag maps to the `COORDINATOR_INDEX` property in `bdd.conf`.
`--coordinator_session_cache`	Specifies the name and (optionally) location of the session cache file used by the leader Dgraph. The leader uses this file to resume its last session with ZooKeeper if it exits abnormally. This file is created when a Dgraph is promoted to leader and deleted when the leader exists normally. If the leader exits abnormally, the file remains on disk so that the leader can resume its last session. Follower Dgraphs don't produce session cache files, and only leaders resume sessions. The default file is `$BDD_HOME/dgraph/clustercache.token`. The file location should always be the same to ensure the Dgraph will be able to find it. Additionally, you should avoid modifying the contents of this file.
`--export_port`	Specifies the port on which the Dgraph listens for requests from the HDFS Agent. This should be the same as the number specified for the HDFS Agent `--export_port` flag. It should be different from the numbers specified for both the `--port` and `--bulk_load_port` flags. This flag maps to the `AGENT_EXPORT_PORT` property in `bdd.conf`.
`--help`	Prints the help message and exits. The help message includes usage information for each Dgraph flag.
`--log`	Specifies the path to the Dgraph request log file. The default file used is `dgraph.reqlog`.
`--log-level`	Specifies the log level for the Dgraph log subsystems. For information on setting this flag, see *Setting the Dgraph log levels on page 176*. This flag maps to the `DGRAPH_LOG_LEVEL` property in `bdd.conf`.

Flag	Description
--memory-limit	Specifies the maximum amount of memory (in MB) the Dgraph is allowed to use for processing. If you do not use this flag, the memory limit is by default set to 80% of the machine's available RAM. If you specify a limit in MB for this flag, this number is used as the memory consumption limit, for the Dgraph, instead of 80% of the machine's available RAM. If you specify 0 for this flag, this overrides the default of 80% and means there is no limit on the amount of memory the Dgraph can use for processing. For a summary of how Dgraph allocates and utilizes memory, see *Memory consumption by the Dgraph on page 63*.
--net-timeout	Specifies the maximum amount of time (in seconds) the Dgraph waits for the client to download data from queries across the network. The default value is 30 seconds.
--out	Specifies a file to which the Dgraph's stdout/stderr will be remapped. If this flag is omitted, the Dgraph uses its default stdout/stderr file, dgraph.out. This file must be different from the one specified by the HDFS Agent's --out flag. This flag maps to the DGRAPH_OUT_FILE property in bdd.conf.
--pidfile	Specifies the file the Dgraph's process ID (PID) will be written to. The default filename is dgraph.pid.
--host	Specifies the name of the Dgraph's host server. This flag maps to the DGRAPH_SERVERS property in bdd.conf.
--port	Specifies the port used by the Dgraph's host server. This flag maps to the DGRAPH_WS_PORT property in bdd.conf.
--leader	Creates a read/write Dgraph leader for the index. This flag is used internally.
--read-only	Sets the index files to read-only. This flag is for internal use only by Oracle Support and should not be used by BDD system administrators. When this flag is set, the Dgraph can only perform read-only operations. Any operations that attempt to write to the index files are rejected and return an HTTP status code 403.

Flag	Description
`--search_char_limit`	Specifies the maximum number of characters that a text search term can contain. The default value is `132`.
`--search_max`	Specifies the maximum number of terms that a text search query can contain. The default value is `10`.
`--snip_cutoff`	Specifies the maximum number of words in an attribute that the Dgraph will evaluate to identify a snippet. If a match is not found within the specified number of words, the Dgraph won't return a snippet, even if a match occurs later in the attribute value. The default value is `500`.
`--snip_disable`	Globally disables snippeting.
`--sslcafile`	**Note:** This flag is not used in Oracle Big Data Discovery. Specifies the path to the SSL Certificate Authority file that the Dgraph will use to authenticate SSL communications with other components.
`--sslcertfile`	**Note:** This flag is not used in Oracle Big Data Discovery. Specifies the path of the SSL certificate file that the Dgraph will present to clients for SSL communications.
`--stat-brel`	**Note:** This flag is deprecated and not used in Oracle Big Data Discovery. Creates dynamic record attributes that indicate the relevance rank assigned to full-text search result records.
`--syslog`	Directs all output to syslog.
`--threads`	Specifies the number of threads the Dgraph will use to process queries and execute internal maintenance tasks. The value you provide must be a positive integer (2 or greater). The default is 2 threads. The recommended number of threads for machines running only the Dgraph is the number of CPU cores the machine has. For machines co-hosting the Dgraph with other Big Data Discovery components, the recommended number of threads is the number of CPU cores the machine has minus two. This flag maps to the `DGRAPH_THREADS` property in `bdd.conf`.

Flag	Description
`--validate_data`	Validates that all indexed data loads and then exits.
`--version`	Prints version information and then exits. The version information includes the Oracle Big Data Discovery version number and the internal Dgraph identifier.
`--wildcard_max`	Specifies the maximum number of terms that can match a wildcard term in a wildcard query that contains punctuation, such as `ab*c.def*`. The default is `100`.

Dgraph HDFS Agent flags

This topic describes the flags used by the Dgraph HDFS Agent.

The Dgraph HDFS Agent requires several flags, which are described in the following table. Note that some flags have the same name as their Dgraph flag counterpart, and (except for `--out`) must have the same settings.

The `startDgraphHDFSAgent.sh` script can use the following flags:

Dgraph HDFS Agent flag	Description
`--agent_port`	Sets the port on which the Dgraph HDFS Agent is listening for HTTP requests. Note that there is no Dgraph version of this flag.
`--export_port`	Sets the port on which the Dgraph HDFS Agent is listening for requests from the Dgraph. This port number must be the same as specified for the Dgraph `--export_port` flag.
`--port`	Specifies the port on which the Dgraph is listening for HTTP requests. This port number must be the same as specified for the Dgraph `--port` flag.
`--bulk_load_port`	Sets the port on which the Dgraph HDFS Agent is listening for bulk load ingest requests. This port number must be the same as specified for the Dgraph `--bulk_load_port` flag.
`--cluster_identity`	Specifies the cluster identity of the Dgraph running on this node. The syntax is: `protocol:hostname:dgraph_port:dgraph_bulk_load_port:agent_port` This cluster identity must be the same as specified for the Dgraph `--cluster_identity` flag.

Dgraph HDFS Agent flag	Description
`--coordinator`	Specifies the host and port on which Zookeeper is running. The syntax is: `host:port` (with a semicolon separating the host name and port). This host:port must be the same as specified for the Dgraph `--coordinator` flag.
`--coordinator_index`	Specifies the index of the cluster in the Zookeeper ensemble. This index must be the same as specified for the Dgraph `--coordinator_index` flag.
`--out`	Specifies the file name and path of the Dgraph HDFS Agent's stdout/stderr log file. The log name must be different from that specified with the Dgraph `--out` flag.

Hadoop configuration files

The `core-site.xml` and `hdfs-site.xml` files are used to configure a Hadoop cluster, especially the one machine in the cluster that is designated as the NameNode. The NameNode contains the HDFS file system from which the Dgraph HDFS Agent will read ingest files and write export files.

At start-up, the Dgraph HDFS Agent reads in the `core-site.xml` and `hdfs-site.xml` files so it can determine the location of the NameNode.

Startup example

The following is an example of using the `startDgraphHDFSAgent.sh` to start the Dgraph HDFS Agent:

```
./startDgraphHDFSAgent.sh --agent_port 7102 --export_port 7101 --port 5555
    --bulk_load_port 5556 --coordinator web04.example.com:2181 --coordinator_index cluster1
    --cluster_identity http:web04.example.com:5555:5556:7102 --out /tmp/agent.log
```

Chapter 6

The Dgraph Gateway

This section describes the Dgraph Gateway role in the Big Data Discovery cluster deployment. It also discusses its configuration file and tells you how to start and stop the Dgraph Gateway through the Administration Console of the WebLogic Server.

About the Dgraph Gateway

Together with Studio, the Dgraph Gateway is a Java-based application that is co-hosted in the same WebLogic Server instance.

The Dgraph Gateway provides:

- Routing of requests to the Dgraph nodes in the BDD cluster
- Caching, business logic, and handling of cluster services for the Dgraph nodes.

Within the Big Data Discovery cluster deployment, you can have one or more WebLogic Server Managed nodes each of which run Studio and Dgraph Gateway. Once the Dgraph Gateway is deployed, you use the WebLogic Server's Administration Console to manage it.

Dgraph Gateway configuration file

A configuration file sets global parameters for the Dgraph Gateway, such as the default locations of files and directories.

The name of the configuration file is `EndecaServer.properties`, and it is located in the `$DOMAIN_HOME/config` directory. The default values in the file are set when the domain is created at installation time.

Most of these parameters are used by the Dgraph Gateway application and should not be modified. If you do need to modify some of them, stop the Dgraph Gateway on the machine on which you are modifying the parameter and restart it. If you have multiple nodes in the BDD cluster deployment that run WebLogic Server hosting Studio and the Dgraph Gateway, make the changes to the `EndecaServer.properties` on all WebLogic Server machines and then restart the Dgraph Gateway instances. BDD relies on this file being the same on all WebLogic Server Managed nodes in the BDD cluster deployment.

Dgraph Gateway settings

The following configuration settings are specific to Dgraph Gateway operations:

Dgraph Gateway parameter	Description
`endeca-session-id-key`	Specifies name of the key used to maintain session affinity. `X-Endeca-Session-ID` is the default value.
`endeca-session-id-type`	Specifies the method used to establish session affinity. The `header` value is the default.

ZooKeeper settings

For some of its functions, the Dgraph Gateway relies on the ZooKeeper package found in the CDH or HDP installation.

The following configuration settings are specific to ZooKeeper:

ZooKeeper parameter	Description
`endeca-cluster-identifier`	Specifies a string identifier for a BDD cluster. This property identifies the cluster. The default is `cluster1`.
`zookeeper-servers`	Specifies a comma-separated list of `host:port` pairs to describe each ZooKeeper server in a ZooKeeper ensemble. The `host` represents a server running ZooKeeper. The corresponding `port` is the port on which ZooKeeper clients connect to that server. ZooKeeper servers are those CDH or HDP nodes in the Big Data Discovery cluster deployment that run ZooKeeper instances. The default `host` value is the name of the Managed Server. The default `port` value is `2181`. If a single server runs ZooKeeper, specify the server name, such as `zookeeper-servers=web009.us.example.com:2181`. If multiple servers run ZooKeeper, specify comma-separated `host:port` names of all servers that are part of the Zookeeper ensemble.

Starting Dgraph Gateway

When you start the WebLogic Server in which the Dgraph Gateway application is deployed, it automatically starts the Dgraph Gateway.

If the application was running when WebLogic Server was shut down, the Dgraph Gateway automatically re-starts as part of the WebLogic Server start-up procedure. Additionally, you can manually start the Dgraph Gateway from the WebLogic Server Administration Console.

To start a stopped Dgraph Gateway:

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

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

   For example:

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

3. At the Administration Console login screen, log in with the administrator user name and password.
4. In the **Domain Structure** pane, click **Deployments**.
5. In the **Deployments** table, check the **oracle.endecaserver** Web application. Its State should be "Prepared" and its Health should be "OK".
6. In the Deployments table, click **Start>Servicing all requests** (which makes the application immediately available to all WebLogic Server clients).

   You can also choose the **Servicing only administration requests** option, which makes the application available in Administration Mode only.

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

As a result, the Dgraph Gateway is started and its State now changes to "Active".

Stopping Dgraph Gateway

You can manually stop the Dgraph Gateway from the WebLogic Server Administration Console.

Note that it is not necessary to stop Dgraph Gateway in order to shut down WebLogic Server; in this case, WebLogic Server will stop Dgraph Gateway as part of its shut-down procedure.

To stop the Dgraph Gateway:

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

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

   For example:

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

3. At the Administration Console login screen, log in with the administrator user name and password.
4. In the Domain Structure pane, click **Deployments**.
5. In the Deployments table, check the **oracle.endecaserver** Java application. Its State should be "Active" and its Health should be "OK", as in this abbreviated example:
6. In the **Deployments** table, click **Stop**, and select one of the stop options:
   - **When work completes:** Specifies that WebLogic Server waits for the Dgraph Gateway to finish its work and for all currently connected users to disconnect.
   - **Force Stop Now:** Specifies that WebLogic Server stops the Dgraph Gateway immediately, regardless of the work that is being performed and the users that are connected.

- **Stop, but continue servicing administration requests:** Specifies that WebLogic Server stops the Dgraph Gateway once all its work has finished, but then puts the application in Administration Mode so it can be accessed for administrative purposes.

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

As a result, the Dgraph Gateway is stopped and its State now changes to "Prepared".

Note: If the Dgraph Gateway is in a "Prepared", (that is, stopped), state when you shut down WebLogic Server, then the application is not automatically restarted when you start WebLogic Server. In this case, you must manually start Dgraph Gateway.

Chapter 7

Using the Dgraph Gateway Command Utility

This section describes the Dgraph Gateway commands (`endeca-cmd`) used for Dgraph nodes.

About the Dgraph Gateway Command Utility

The Dgraph Gateway has a command-line interface that lists Dgraph nodes, allocates bulk load port, provides version information, and performs cache warming operations for the Dgraphs.

The Dgraph Gateway Command Utility resides by default in the `$BDD_HOME/server/endeca-cmd` directory. The directory contains a script named `endeca-cmd` with which you can run the commands.

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

Commands

The `endeca-cmd` script allows you to run the following Dgraph Gateway commands.

Option	Description
`allocate-bulk-load-port`	Returns a host name for the leader node and the port used for Bulk Load Interface.
`dump-session`	Returns session information from a Dgraph for a specified session Id.
`list-compute-nodes`	Returns a list of running Dgraph nodes in a cluster.
`version`	Lists the version of the Dgraph Gateway and the version of the Dgraph (if the Dgraphs are currently running).

Option	Description
`warm-cache`	Warms the Dgraph cache without requiring a custom warm-up script.

Syntax

The syntax for running the `endeca-cmd` script is:

```
endeca-cmd <operation> [operation options] [global options]
```

Getting online help

The `--help` option provides usage help for the Dgraph Gateway commands. The syntax for obtaining general help is:

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

The syntax for obtaining help on a specific commands is:

```
endeca-cmd <operation> --help
```

This example displays usage help for the `list-compute-nodes` commands:

```
endeca-cmd list-compute-nodes --help
```

Global options for host, port, and context root

The command utility has several global options that allow you to specify the host, port, and context root of the Dgraph Gateway.

The global options are:

- `--host`
- `--port`
- `--root`
- `--help`

Do not forget to specify global options with `endeca-cmd`. If you do not specify them, `endeca-cmd` assumes that the defaults are used for the Dgraph Gateway (such as the default port and host). For example, assume you have configured the Dgraph Gateway application in WebLogic domain to use a port that is different from the default port. In this case, in order for the `endeca-cmd` utility to find the correct port, you should list it explicitly, as one of the global options. For example, this operation returns a list of the Dgraphs available to the Dgraph Gateway running on port 9001:

```
endeca-cmd list-compute-nodes --port 9001
```

--host option

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

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

```
endeca-cmd list-compute-nodes --host web7.example.com
```

The command tells the Dgraph Gateway running on **web7.example.com** (and listening on its default port) to return a list of the Dgraphs compute nodes.

--port option

7001 is the default HTTP port in the WebLogic Server on which the Dgraph Gateway application is listening.

Important: HTTP is used for communication of the Dgraph Gateway with other components within Big Data Discovery. Therefore, the node hosting WebLogic Server for the Dgraph Gateway (it is the same node that hosts Studio within WebLogic Server) must be deployed behind the site's firewall.

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

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

```
endeca-cmd list-compute-nodes --host web7.example.com --port 7003
```

The command tells the Dgraph Gateway running on the **web7.example.com** remote machine (and listening on a non-default port 9090) to return the list of Dgraph compute nodes.

--root option

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

Allocating a bulk load port

The `allocate-bulk-load-port` operation returns a host name for the leader node and the port used for the internally-used Bulk Load Interface.

The syntax for this command is:

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

This is a read-write operation. If the current leader node is available, the operation 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.

To allocate a bulk load port:

1. From the command line, navigate to the `endeca-cmd` directory.
2. Run the `allocate-bulk-load-port` command.

Example

```
endeca-cmd allocate-bulk-load-port --port 7003
Bulk load host: web009.us.example.com
Bulk load port: 7019
```

Returning version information

The `version` command lists the version of the Dgraph Gateway and the version of the Dgraph nodes (if the Dgraph nodes are currently running).

The syntax for this command is:

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

To return version information:

1. From the command line, navigate to the `endeca-cmd` directory.
2. Run the `version` command.

Example

```
endeca-cmd version --port 7003
Oracle Endeca Server 1.1.1.970778
```

Listing Dgraph nodes

The `list-compute-nodes` operation returns a list of running Dgraph nodes in a cluster. This includes both leader and follower nodes. The operation does not list Dgraphs that are stopped.

The syntax for this command is:

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

To list Dgraph nodes:

1. From the command line, change to the directory where `endeca-cmd` is installed.
2. Run the `list-compute-nodes` operation.

Example

```
endeca-cmd list-compute-nodes --port 7003
HTTP:web009.us.example.com:7010       Leader
```

Returning Dgraph session information

The `dump-session` operation returns session information from a Dgraph for a specified session ID. (Dgraph Gateway tracks which Dgraph instance is processing a request for a particular session.)

The syntax for this command is:

```
endeca-cmd dump-session [global-options] operation options
```

where operation options are the following:

- `--latest <argument>` is the number of most recent sessions to return.
- `--session-id <argument>` is the session ID to return.

For this command to work, you must specify one of the options and an argument for it. There is no default behavior for this command.

To return Dgraph session information:

1. From the command line, change to the directory where `endeca-cmd` is installed.
2. Run the `dump-session` operation with additional options as desired.

Examples

Example with `--latest <argument>`:

```
endeca-cmd dump-session --latest 30 --port 7003
Dump the 30 most recent sessions.
Session id: faked-session-2. DGraph node: web009:7010. Time of first query: 2014-27-21 01:27:56.
Time of the last query: 2014-27-21 01:27:56. Request count: 1
Session id: faked-session-1. DGraph node: web009:7010. Time of first query: 2014-27-21 01:27:44.
Time of the last query: 2014-27-21 01:27:44. Request count: 1
Session id: faked-session-0. DGraph node: web009:7010. Time of first query: 2014-27-21 01:27:04.
Time of the last query: 2014-27-21 01:27:04. Request count: 1
```

Example with `--session-id <argument>`:

```
endeca-cmd dump-session --session-id 1234 --port 7003
Dump session information with the given session ID: 1234
Session id: 1234. DGraph node: web009:7010. Time of first query: 2014-27-21 01:27:04.
Time of the last query: 2014-27-21 01:27:04. Request count: 1
```

Warming the Dgraph cache

The `warm-cache` command warms each Dgraph cache for all Dgraph instances in a cluster.

The command takes into account usage patterns of the Dgraphs and replays a set of previous queries for a specified period of time against each Dgraph. That replay warms the cache, allows a Dgraph to reuse cached results across subsequent user queries, and helps reduce the user-observable latencies in query processing.

You must explicitly issue the cache warming request. It does not run automatically. The only parameter for the command is the time limit for which the cache warming runs.

A successful invocation of `warm-cache` returns immediately with an empty response and starts the cache warming job in the background. Once the time limit is reached, the cache warming stops. If during this time you issue any other requests to the Dgraph, they take priority over cache warming.

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

The syntax for this command is:

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

The `--time-limit-sec` parameter is optional. It specifies a time limit to replay previous queries. If you do not specify the timeout, the default value of 1800 seconds (30 minutes) is used.

To warm the Dgraph cache:

1. From the command line, navigate to the `endeca-cmd` directory.

2. Run the `warm-cache` command.

Example

```
endeca-cmd warm-cache --port 7003
Warmed the cache on DGraph node: we009.us.example.com:7010.
```

Part IV

Administering Studio

Chapter 8

Managing Data Sources

You can add, configure, and delete database connections and JDBC data sources on the **Control Panel>Big Data Discovery>Data Source Library** page of Studio.

About database connections and JDBC data sources

Studio users can import data from an external JDBC database and access it from Studio as a data set in the Catalog.

A default installation of Big Data Discovery includes JDBC drivers to support the following relational database management systems:

- Oracle 11g and 12c
- MySQL

To set up this feature, there are both Studio administrator tasks and Studio user tasks.

A Studio administrator goes to the **Data Source Library** page and creates a connection to a database and creates any number of data sources, each with unique log in information, that share that database connection. The administrator configures each new data source with log in information to restrict who is able to create data sets from it. Data sources are not available to Studio users until an administrator sets them up.

Next, a Studio user clicks **Create a data set from a database** to import and filter the JDBC data source. After upload, the data source is available as a data set in the Catalog.

Creating data connections

To create a data connection:

1. Log in to Studio as an administrator.
2. Click **Configuration Options>Control Panel** and navigate to **Big Data Discovery>Data Source Library**.
3. Click **+ Connection**.

4. On the **New data connection** dialog, provide the name, URL, and authentication information for the data connection.
5. Click **Save**.

Deleting data connections

If you delete a data connection, the associated data sources also are deleted. Any data sets created from those data sources can no longer be refreshed once the connection has been deleted.

To delete a data connection:

1. Log in to Studio as an administrator.
2. Click **Configuration Options>Control Panel** and navigate to **Big Data Discovery>Data Source Library**.
3. Locate the data source connection and click the delete icon.
4. In the confirmation dialog, click **Delete**.

Creating a data source

When you create a data source, you specify a SQL query to select the data to include.

To create a data source:

1. Log in to Studio as an administrator.
2. Click **Configuration Options>Control Panel** and navigate to **Big Data Discovery>Data Source Library**.
3. Click **+ data source** for a data connection you created previously.
4. Provide the required authentication information for the data connection, then click **Continue**.
5. Provide a name and description for the data source.
6. In **Maximum number of records**, specify the maximum number of records to include in the data set.

   Studio does not control the order of the records. The SQL statement can indicate the order of records to import using an ORDER BY clause.
7. In the text area, enter the SQL query to retrieve the records for the data source, then click **Next**.

   The next page shows the available columns, with a sample list of records for each.
8. Click **Save**.

Once you have completed this task, the data source displays on the Studio Catalog as a new data set available to users.

Editing a data source

Once a data source is created, you can change the data or edit it.

Displaying details for a data source

To display detailed information for a data source, click the data source name. On the details panel:

- The **Data Source Info** tab provides a summary of information about the data source, including tags, the types of attributes, and the current access settings.
- The **Associated Data Sets** tab lists data sets that have been created from the data source.

Editing a data source

To edit a data source, click the **Edit** link on the data source details panel, or click the name itself.

Deleting a data source

To delete a data source:

1. Log in to Studio as an administrator.
2. Click **Configuration Options>Control Panel** and navigate to **Big Data Discovery>Data Source Library**.
3. In the Data Connections part of the page, expand the data connection on which your data source is based.
4. Click the information icon for the data source you want to delete.
5. Click the **Delete** link
6. In the confirmation dialog, click **Delete**.

Chapter 9

Configuring Studio Settings

The **Studio Settings** page on the **Control Panel** configures many general settings for the Studio application.

Studio settings list

Changing the Studio setting values

Studio settings list

Studio settings include configuration options for timeouts, default values, and the connection to Oracle MapViewer, for the **Map** and **Thematic Map** components.

The Studio settings are:

Setting	Description
Default access to derived data sets	Controls whether new data sets created by exporting or forking existing data sets are set to **Private** (restricted to the creator and all Studio Administrators) or made publically available. Defaults to **Public**.
`df.bddSecurityManager`	The fully-qualified class name to use for the BDD Security Manager. If empty, the Security Manager is disabled.
`df.clientLogging`	Sets the logging level for messages logged on the Studio client side. Valid values are ALL, TRACE, DEBUG, INFO, WARN, ERROR, FATAL and OFF. Messages are logged at the set level or above.
`df.countApproxEnabled`	Specifies a Boolean value to indicate that components perform approximate record counts rather than precise record counts. A value of `true` indicates that Studio display approximate record counts using the `COUNT_APPROX` aggregation in an EQL query. A value of `false` indicates precise record counts using the `COUNT` aggregation. Setting this to `true` increases the performance of refinement queries in Studio. The default value is `false`.
`df.dataSourceDirectory`	The directory used to store keystore and certificate files for secured data.
`df.defaultCurrencyList`	A comma-separated list of currency symbols to add to the ones currently available.

Setting	Description
`df.exportBatchSize`	When exporting a large number of records, Studio splits the records into batches. This setting determines the number of records in each batch. The default value is 2000.
`df.helpLink`	Used to configure the path to the documentation for this release. Used for links to specific information in the documentation.
`df.mapLocation`	The URL for the Oracle MapViewer eLocation service. The eLocation service is used for the text location search on the **Map** component, to convert the location name entered by the user to latitude and longitude. By default, this is the URL of the global eLocation service. If you are using your own internal instance, and do not have Internet access, then set this setting to "None", to indicate that the eLocation service is not available. If the setting is "None", Big Data Discovery disables the text location search. If this setting is not "None", and Big Data Discovery is unable to connect to the specified URL, then Big Data Discovery disables the text location search. Big Data Discovery then continues to check the connection each time the page is refreshed. When the service becomes available, Big Data Discovery enables the text location search.
`df.mapTileLayer`	The name of the MapViewer Tile Layer. By default, this is the name of the public instance. If you are using your own internal instance, then you must update this setting to use the name you assigned to the Tile Layer.
`df.mapViewer`	The URL of the MapViewer instance. By default, this is the URL of the public instance of MapViewer. If you are using your own internal instance of MapViewer, then you must update this setting to connect to your MapViewer instance.
`df.maxExportRecords`	The maximum allowable number of records that can be exported from a component. The default value is 1000000.
`df.mdexCacheManager`	Internal use only.

Setting	Description
`df.stringTruncationLimit`	The maximum number of characters to display for a string value. This value may be overridden when configuring the display of a string value in an individual component. The default value is 10000.
`df.performanceLogging`	This property can only be modified from the `portal-ext.properties` file.

Changing the Studio setting values

To set the values of Studio settings, you can either use the fields on the **Studio Settings** page, or add the values to `portal-ext.properties`. If you configure a setting in `portal-ext.properties`, then the field on the **Studio Settings** page is locked.

Configuring settings in `portal-ext.properties` makes it easier to migrate settings across different environments. For example, after testing the settings in a development system, you can simply copy the properties file to the production system, instead of having to reset the production settings manually from the **Control Panel**.

To change the Studio setting values:

1. To configure Studio settings:

   (a) From the Control Panel, select **Big Data Discovery>Studio Settings**.

   (b) For each setting you want to update, provide a new value in the setting configuration field.

   Note: Take care when modifying these settings, as incorrect values can cause problems with your Studio instance.

   If the setting is configured in `portal-ext.properties`, then you cannot change the setting from this page. You must set it in the file.

   (c) Click **Update Settings**.

   (d) To apply the changes, restart Big Data Discovery.

2. To add a setting to `portal-ext.properties`:

   (a) Stop the server.

   (b) Open a command prompt and change to `$ORACLE_HOME/user_projects/domains/bdd_<version>_domain/config/studio`

   (c) Open `portal-ext.properties` in a text editor and add the setting.

   In the file, the format for adding a setting is:

   ```
   <settingname>=<value>
   ```

   Where:

   - *<settingname>* is the name of the setting from the **Studio Settings** page.

- *<value>* is the value of the setting.

For example, to set the maximum number of records to export, the entry would be:

```
df.maxExportRecords=50000
```

(d) Save and close the file.

(e) Restart Studio.

On the **Framework Settings** page, the setting is now read only.

Chapter 10

Configuring Data Processing Settings

In order to upload files (Excel and CSV) and perform other data processing tasks, you must configure the **Data Processing Settings** on Studio's Control Panel.

List of Data Processing Settings

The settings listed in the table below must be set correctly in order to perform data processing tasks.

Many of the default values for these setting are populated based the values specified in `bdd.conf` during the installation process.

In general, the settings below should match the Data Processing CLI configuration properties which are contained in the script itself. Parameters that must be the same are noted as such in the table below. For information about the Data Processing CLI configuration properties, see the *Data Processing Guide*.

Hadoop Setting	Description
`bdd.enableEnrichments`	Specifies whether to run data enrichments during the sampling phase of data processing. This setting controls the Language Detection, Term Extraction, Geocoding Address, Geocoding IP, and Reverse Geotagger modules. A value of `true` runs all the data enrichment modules and `false` does not run them. You cannot enable an individual enrichment. The default value is `true`.
`bdd.kryoBufferSize`	Specifies the amount of buffer space allocated to Kryo. If you encounter Kryo-related exceptions, you may need to increase this value. The default value is 1024 MB.
`bdd.kryoMode`	Specifies a Boolean value to enable or disable Kryo mode. Kryo mode provides an alternative way to serialize and move data among Spark worker nodes. A value of `true` enables Kryo mode and `false` uses Java serialization. Kryo mode is generally faster for data processing but may cause exceptions in situations that are hard to anticipate. The default value is `false`.

Hadoop Setting	Description
`bdd.maxRecordsToProcess`	Specifies the maximum number of records that are processed to become the sample size of a data set in the **Catalog**. This is a global setting controls the sample size for all Excel and CSV files uploaded using Studio. For example, you if upload a file that has 5,000,000 rows, you could restrict the total number of sampled records to 1,000,000. The default value is 1,000,000. (This value is approximate. After data processing, the actual sample size may be slightly more or slightly less than this value.)
`bdd.maxSplitSize`	The maximum partition size for Spark jobs measured in MB. This controls the size of the blocks of data handled by Data Processing jobs. Partition size directly affects Data Processing performance — when partitions are smaller, more jobs run in parallel and cluster resources are used more efficiently. This improves both speed and stability. The default is set by the `MAX_INPUT_SPLIT_SIZE` property in the `bdd.conf` file (which is 32, unless changed by the user). The 32MB is amount should be sufficient for most clusters, with a few exceptions: • If your Hadoop cluster has a very large processing capacity and most of your data sets are small (around 1GB), you can decrease this value. • In rare cases, when data enrichments are enabled the enriched data set in a partition can become too large for its YARN container to handle. If this occurs, you can decrease this value to reduce the amount of memory each partition requires. Note that this property overrides the HDFS block size used in Hadoop.

Data Processing Topology

In addition to the configurable settings above, you can review the data processing topology by navigating to the **Big Data Discovery>About Big Data Discovery** page and expanding the **Data Processing Topology** drop-down. This exposes the following information:

Hadoop Setting	Description
Hadoop Host	The hostname of the machine that acts as the Master for your Hadoop cluster.

Hadoop Setting	Description
Name Node Port	The NameNode internal Web server port.
Hive metastore Server port	The Hive metastore listener port.
Hive Server port	The Hive server listener port.
Hue HTTP port	The Hue Web interface port.
Cluster OLT Home	The OLT home directory in the BDD cluster. The BDD installer detects this value and populates the setting.
Database Name	The name of the Hive database that stores the source data for Studio data sets.
EDP Data Directory	The directory that contains the contents of the edp_cluster_*.zip file on each worker node.
Sandbox	The HDFS directory in which to store the avro files created when users export data from Big Data Discovery. The default value is `/user/bdd`.

Changing the data processing settings

You configure the settings on the **Data Processing Settings** page on the **Control Panel**.

To change the Hadoop setting values:

1. Log in to Studio as an administrator.
2. From the **Control Panel**, select **Big Data Discovery>Data Processing Settings**.
3. For each setting, update the value as necessary.
4. Click **Update Settings**.

The changes are applied immediately.

Chapter 11

Viewing Project Usage Summary Reports

Big Data Discovery provides basic reports to allow you to track project usage.

About the project usage logs

Big Data Discovery stores project creation and usage information in its database.

When entries are added to the usage logs

Entries are added when users:

- Log in to Big Data Discovery
- Navigate to a project
- Navigate to a different page in a project
- Create a data set from the **Data Source Library**
- Create a project

When entries are deleted from the usage logs

By default, whenever you start Big Data Discovery, all entries 90 days old or older are deleted from the usage logs.

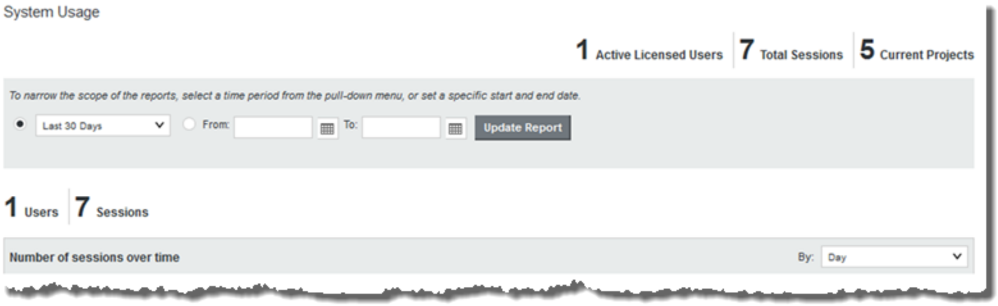
To change the age of the entries to delete, add the following setting to `portal-ext.properties`:

```
studio.startup.log.cleanup.age=entryAgeInDays
```

In addition to the age-based deletions, Big Data Discovery also deletes entries associated with data sets and projects that have been deleted.

About the System Usage page

The **System Usage** page of the **Control Panel** provides access to summary information on project usage logs.



To narrow the scope of the reports, select a time period from the pull-down menu, or set a specific start and end date.

The page is divided into the following sections:

Section	Description
Summary totals	At the top right of the page are the total number of: • Users in the system • Sessions that have occurred • Projects
Date range fields	Contains fields to set the range of dates for which to display report data.
Current number of users and sessions	Lists the number of users that were logged in and the number of sessions for the date range that you specify.
Number of sessions over time	Report showing the number of sessions that have been active for the date range that you specify Includes a list to set the date unit to use for the chart.
User Activity	Report that initially shows the top 10 number of sessions per user for the selected date range across all projects. You can click on any bars in this chart to drill down into the reporting data. At the top of the report are lists to select: • A specific user, or all users • A specific project, or all projects • Whether to display the top or bottom values (most or least sessions) • The number of values to display

Section	Description
Project Usage	Report that initially shows the top 10 number of sessions per project for the selected date range across all projects. You can click on any bars in this chart to drill down into the reporting data. At the top of the report are lists to select: • A specific project, or all projects • Whether to display the top or bottom values (most or least sessions) • The number of values to display
System	Contains a pie chart that shows the relative number of sessions by browser type and version for the selected date range.

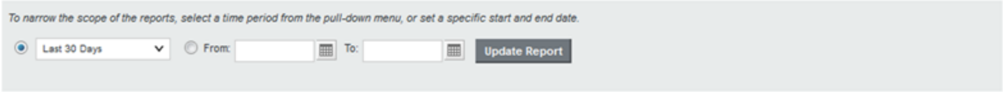
Using the System Usage page

On the **System Usage** page, you use the fields at the top to set the date range for the report data. You can also change the displayed data on individual reports.

To use the **System Usage** page:

1. To set the date range for the displayed data on all of the reports, you can either set a time frame from the current day, or a specific range of dates.

   By default, the page is set to display data from the last 30 days.

   To narrow the scope of the reports, select a time period from the pull-down menu, or set a specific start and end date.

   ● Last 30 Days ○ From: To:

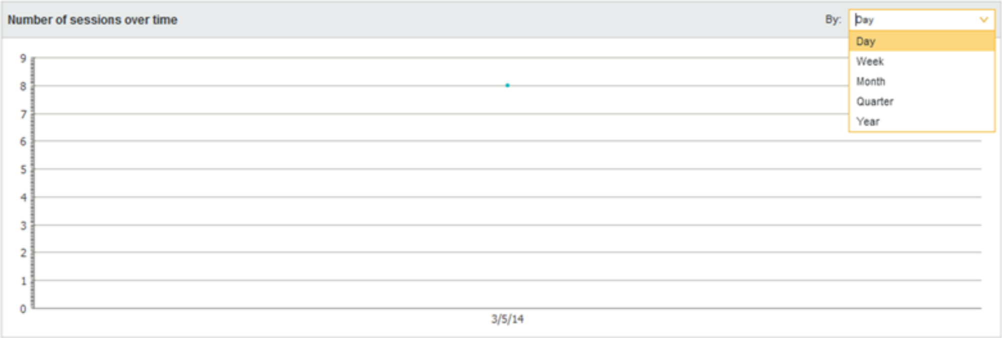
   (a) To select a different time frame, from the list, select the time frame to use.

   (b) To select a specific range of dates, click the other radio button, then in the **From** and **To** date fields, provide the start and end dates.

   (c) After selecting a time frame or range of dates, to update the reports to reflect the new selection, click **Update Report**.

2. For the **Number of sessions over time** report, you can control the date/time unit used to display the results.

   To change the date/time unit, select the new unit from the list.

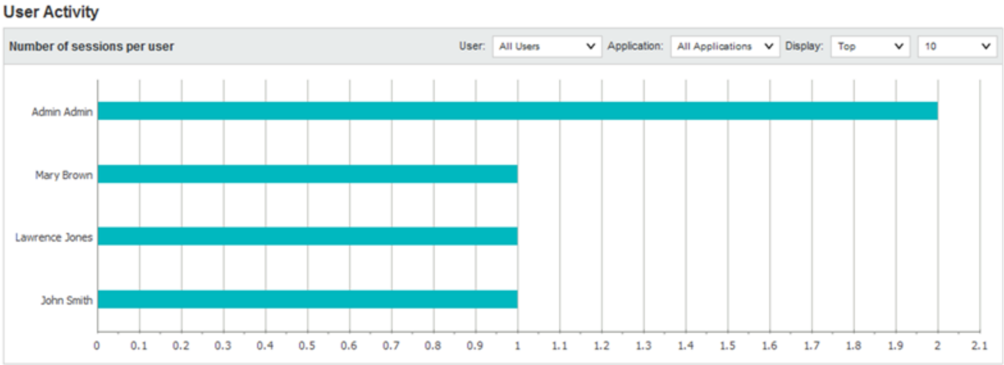


Number of sessions over time

By: Day

   The report is updated automatically to use the new value.

3. By default, the **User Activity** report shows the top 10 number of sessions per user for all projects during the selected time period.



User Activity

Number of sessions per user

User: All Users Application: All Applications Display: Top 10

   You can narrow the report to show values for a specific user or project, and change the number of values displayed.

   (a) To narrow the report to a specific user, from the **User** list, select the user.

   The report is updated to display the top or bottom number of sessions for projects the user has used.

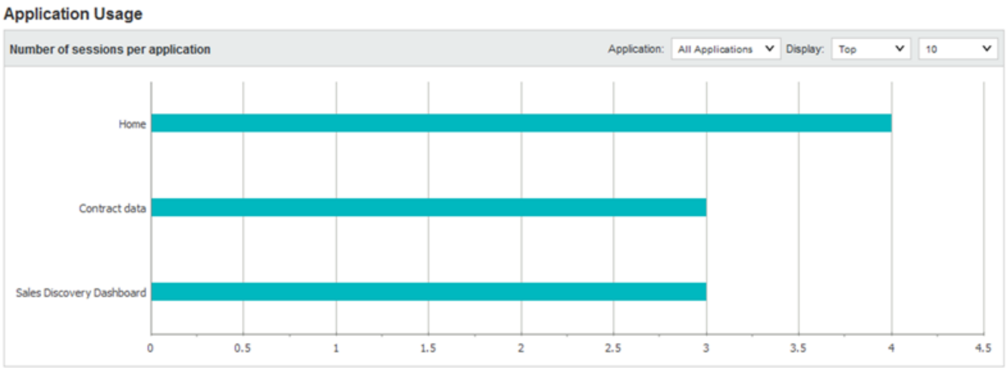
   (b) To narrow the report to a specific project, from the **Project** list, select the project.

   The report is updated to show the users with the top or bottom number of sessions for users.

   If you select both a specific project and a specific user, the report displays a single bar showing the number of sessions for that user and project.

   (c) Use the **Display** settings to control the number of values to display and whether to display the top or bottom values.

4. By default, the **Project Usage** report shows the 10 projects with the most sessions for the selected time range.

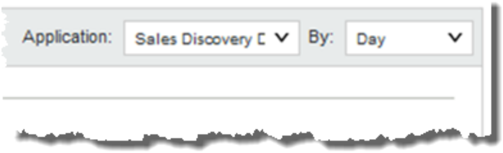

Application Usage
Number of sessions per application
Application: All Applications
Display: Top
10
Home
Contract data
Sales Discovery Dashboard
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5

You can narrow the report to show values for a specific project, and change the number of values displayed.

(a) To narrow the report to a specific project, from the **Project** list, select the project.

The report is changed to a line chart showing the number of sessions per day for the selected project.

A date unit list is added to allow you to select the unit to use.



For example, you can display the number of sessions per day, per week, or per month.

(b) If you are displaying the number of sessions for all projects, use the **Display** settings to control the number of values to display and whether to display the top or bottom values.

Chapter 12

Determining and Configuring the Locale to Use

The Big Data Discovery user interface and project data can be displayed in different locales.

Locales and their effect on the user interface

The locale determines the language in which to display the user interface. It can also affect the format of displayed data values.

Big Data Discovery is configured with a default locale as well as a list of available locales.

Each user account also is configured with a preferred locale, and the user menu includes an option for users to select the locale to use.

In Big Data Discovery, when a locale is selected:

- User interface labels display using the locale.
- Display names of attributes display in the locale.

  If there is not a version for that locale, then the default locale is used.
- Data values are formatted based on the locale.

Supported locales

Big Data Discovery supports the following languages:

- Chinese - Simplified
- English - US
- Japanese
- Korean
- Portuguese - Brazilian
- Spanish

Note that this is a subset of the languages supported by the Dgraph.

How Studio determines the locale to use

When users log in, Studio needs to determine the locale to use to display the user interface and data.

Locations where the locale may be set

Scenarios for selecting the locale

Locations where the locale may be set

The locale is set in different locations.

The locale can come from:

- Cookie
- Browser locale
- Default locale
- User preferred locale, stored as part of the user account
- Locale selected using the **Change locale** option in the user menu, which is also available to users who have not yet logged in.

Scenarios for selecting the locale

The locale used depends upon the type of user, the Big Data Discovery configuration, and how the user entered Big Data Discovery.

For the scenarios listed below, Big Data Discovery determines the locale as follows:

Scenario	How the locale is determined
A new user is created	The locale for a new user is initially set to **Use Browser Locale**, which indicates to use the current browser locale. This value can be changed to a specific locale. If the user is configured with a specific locale, then that locale is used for the user unless they explicitly select a different locale or enter with a URL that includes a supported locale.
A non-logged-in user navigates to Big Data Discovery	For a non-logged-in user, Big Data Discovery first tries to use the locale from the cookie. If there is no cookie, or the cookie is invalid, then Big Data Discovery tries to use the browser locale. If the current browser locale is not one of the supported locales, then the default locale is used.

Scenario	How the locale is determined
A registered user logs in	When a user logs in, Big Data Discovery first checks the locale configured for their user account. • If the user's locale is set to **Use Browser Locale**, then Big Data Discovery tries to use the locale from the cookie. If there is no cookie, or if the cookie is invalid, then Big Data Discovery tries to use the browser locale. If the current browser locale is not a supported locale, then the default locale is used. • If the user account is configured with a locale value other than **Use Browser Locale**, then Big Data Discovery uses that locale, and also updates the cookie with that locale.
A non-logged-in user uses the user menu option to select a different locale	When a non-logged-in user selects a locale, Big Data Discovery updates the cookie with the new locale. Note that this locale change is only applied locally. It is not applied to all non-logged-in users.
A logged-in user uses the user menu option to select a different locale	When a logged-in user selects a locale, Big Data Discovery updates both the user's account and the cookie with the selected locale.

Setting the available locales

Big Data Discovery is configured with a list of available locales. This list is used to populate the list for configuring the default locale, user default locale, and the available locales displayed for the **Change locale** option.

You can add a the setting to `portal-ext.properties` to constrain the list.

There is an implicit list of the following options for Studio:

```
locales=en_US, es_ES, ja_JP, ko_KR, pt_BR, zh_CN
```

To constrain this list:

1. Log in to the machine running Studio, locate `portal-ext.properties` and open it in a text editor.
2. Copy the following `locales` parameter to a new line in the file:

   ```
   locales=en_US, es_ES, ja_JP, ko_KR, pt_BR, zh_CN
   ```

3. Update the list to remove the locales that you do not want to be available in Studio.

   For example, to only support English, French, and Japanese, you would update it to:

   ```
   locales=en_US, fr_FR, ja_JP
   ```

4. Save and close the file.
5. Restart Studio for the changes to take effect.

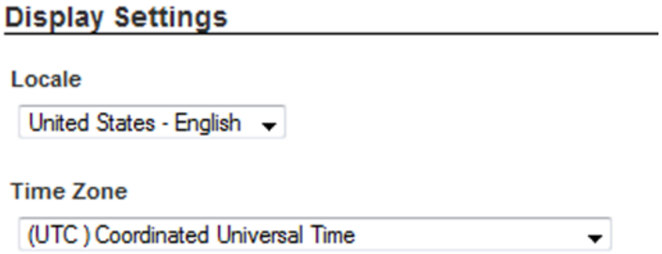
Selecting the default locale

Big Data Discovery is configured with a default locale, which you can update from the **Control Panel**.

Note that if you have a clustered implementation, make sure to configure the same locale for all of the instances in the cluster.

To select the default locale:

1. From the **Control Panel**, select **Platform Settings>Display Settings**.
2. On the **Display Settings** page, from the **Locale** list, select the default locale.



Display Settings

Locale

United States - English

Time Zone

(UTC) Coordinated Universal Time

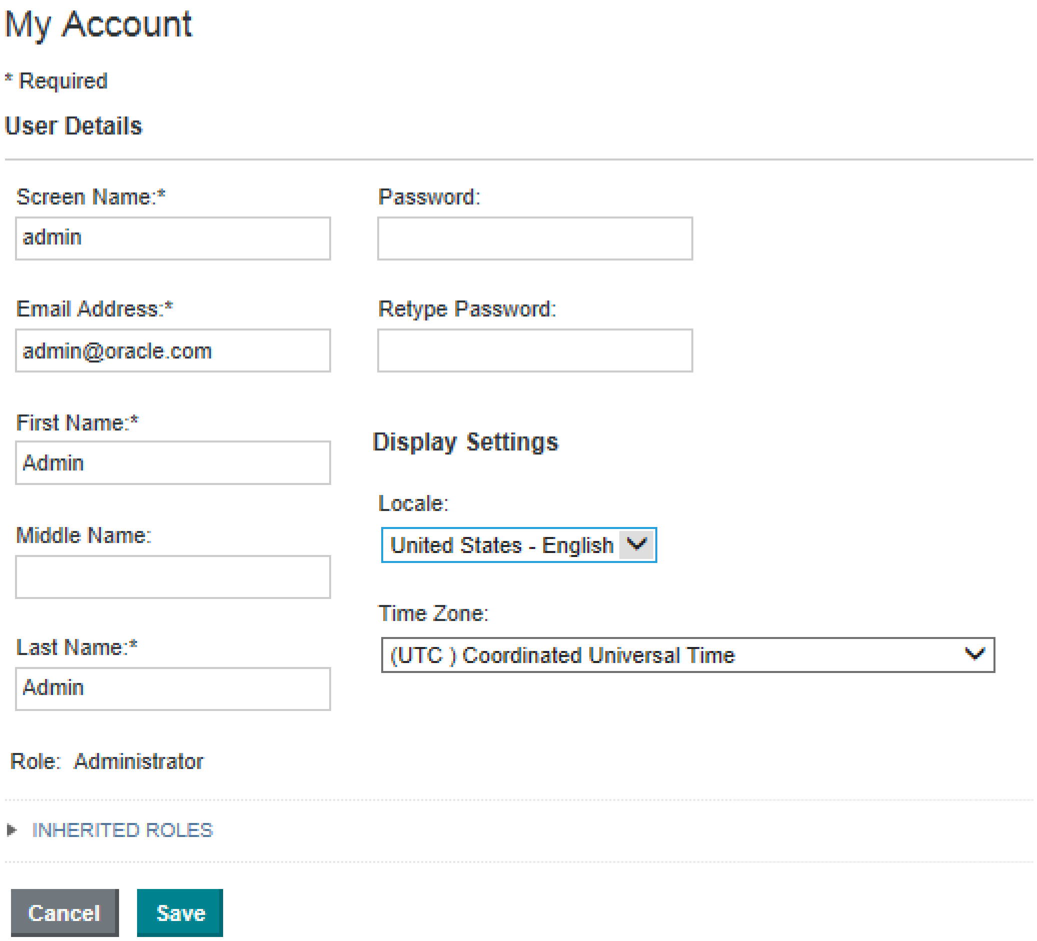
3. Click **Save**.

Configuring a user's preferred locale

Each user account is configured with a preferred locale. The default value for new users is **Use Browser Locale**, which indicates to use the current browser locale.

To configure the preferred locale for a user:

1. To display the setting for your own account, sign in to Studio, and in the header, select **User Options>My Account**.



My Account

* Required

User Details

Screen Name:* admin

Password:

Email Address:* admin@oracle.com

Retype Password:

First Name:* Admin

Middle Name:

Last Name:* Admin

Display Settings

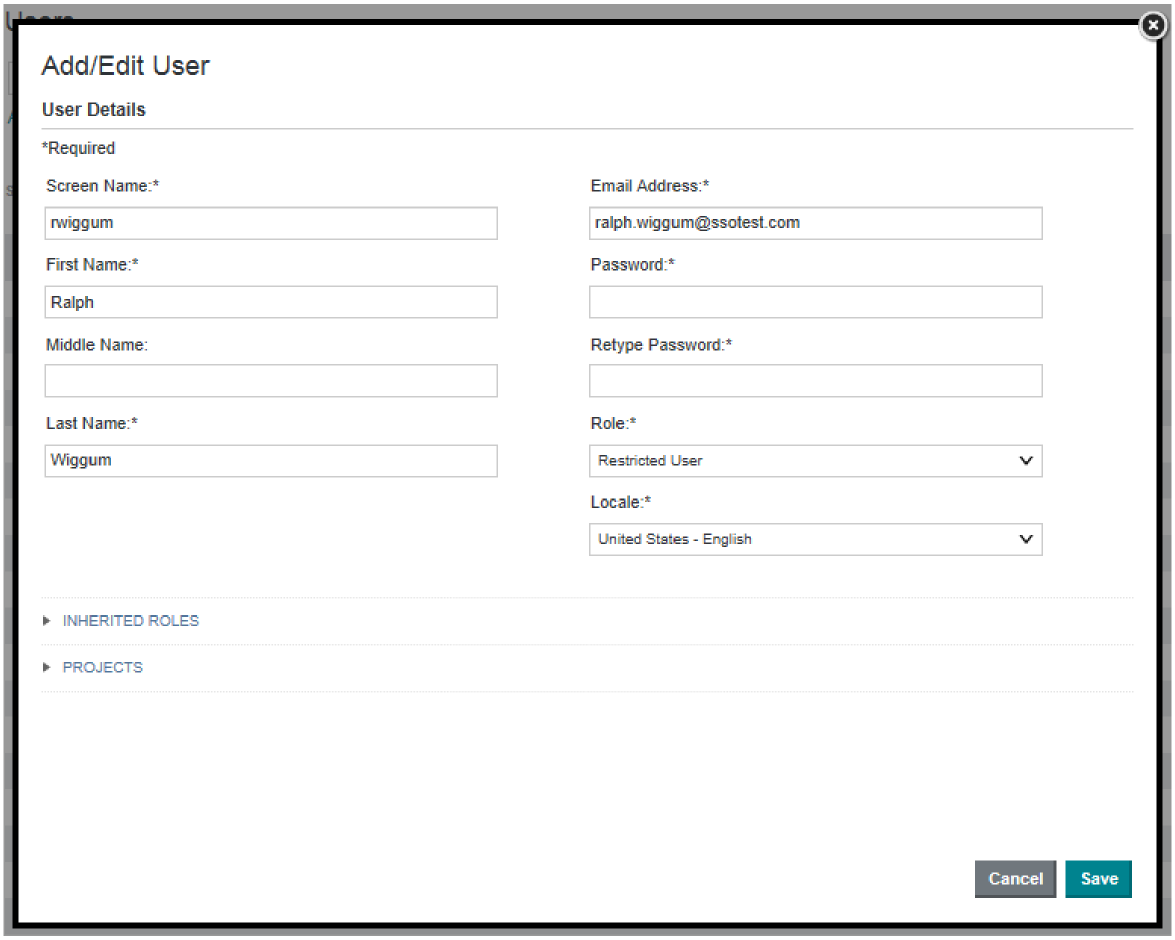
Locale: United States - English

Time Zone: (UTC) Coordinated Universal Time

Role: Administrator

2. To display the setting for another user:
   (a) In the Big Data Discovery header, click the **Configuration Settings** icon and select **Control Panel**.
   (b) Select **User Settings>Users**.

(c) Locate the user and click **Actions>Edit**.



Add/Edit User

User Details

*Required

Screen Name:* rwiggum

First Name:* Ralph

Middle Name:

Last Name:* Wiggum

Email Address:* ralph.wiggum@ssotest.com

Password:*

Retype Password:*

Role:* Restricted User

Locale:* United States - English

3. From the **Locale** list, select the preferred locale for the user.

4. Click **Save**.

Chapter 13

Configuring Settings for Outbound Email Notifications

Big Data Discovery includes settings to enable sending email notifications. Email notifications can include account notices, bookmarks, and snapshots.

Configuring the email server settings

In order for users to be able to email bookmarks, you must configure the email server settings. The email address associated with the outbound server is used as the From address on the bookmark email message.

To configure the email server settings:

1. In the Big Data Discovery header, click the **Configuration Settings** icon and select **Control Panel**.
2. Select **Server>Server Administration**
3. Click the **Mail** tab.
4. Fill out the fields for the incoming mail server:
   (a) In the **Incoming POP Server** field, enter the name of the POP server to use to receive email.

   (b) In the **Incoming Port** field, enter the port number for the POP server.

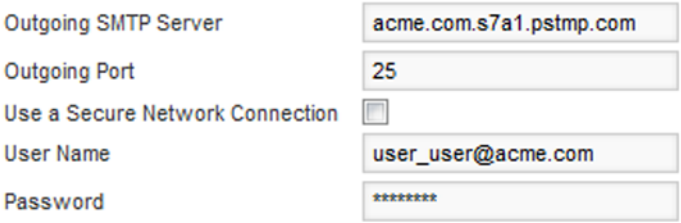
   (c) If you are not using the SMTPS mail protocol to send the email, then you must deselect the **Use a Secure Network Connection**.

   (d) In the **User Name** field, type the email address to associate with the mail server.

   This is the email address used as the **From:** address when end users email bookmarks.

   (e) In the **Password** field, type the email password associated with the email address.

5. Fill out the fields for the outbound mail server:



Outgoing SMTP Server	acme.com.s7a1.pstmp.com
Outgoing Port	25
Use a Secure Network Connection	☐
User Name	user_user@acme.com
Password	********

   (a) In the **Outgoing SMTP Server** field, enter the name of the SMTP server to use to send the email.

   (b) In the **Outgoing Port** field, enter the port number for the SMTP server.

   (c) If you are not using the SMTPS mail protocol to send the email, then the **Use a Secure Network Connection** check box must be deselected.

   (d) In the **User Name** field, type the name to display for the notification sender.

   This is the email address used as the From address when end users email bookmarks.

   (e) In the **Password** field, type the email password associated with the email address.

6. Click **Save**.

Configuring the sender name and email address for notifications

From the **Email Settings** page of the **Control Panel**, you can configure the sender name an email address to display on outbound notifications.

To configure the sender name and email address:

1. From the **Control Panel**, select **Platform Settings>Email Settings**.
2. On the **Settings** tab, in the **Name** field, type the name to display for the notification sender.
3. In the **Address** field, type the email address to display for the notification sender. The sender address is used as the reply-to address for most notifications. For bookmarks and snapshots, the reply-to address is the email address of the user who creates the request.
4. Click **Save**.

Setting up the Account Created and Password Changed notifications

From the **Email Settings** page of the **Control Panel**, you can configure the notifications sent when an account is created and when a user's password is changed.

These notifications only apply to users created and managed within Big Data Discovery.

The configuration includes:

- Whether to send the notification

- The subject line of the email message
- The content of the email message

To set up the Account Created and Password Changed notifications:

1. From the **Control Panel**, select **Platform Settings>Email Settings**.
2. To configure the Account Created notification:
   (a) Click the **Account Created Notification** tab.

   (b) By default, the notification is enabled, meaning that when new users are created in Big Data Discovery, they receive the notification. To disable the notification, deselect the **Enabled** check box.

   (c) In the **Subject line** field, type the text of the email subject line.

   The subject line can include any of the dynamic values listed at the bottom of the tab. For example, to include the user's Big Data Discovery screen name in the subject line, include `[$USER_SCREENNAME$]` in the subject line.

   (d) In the **Body** text area, type the text of the email message.

   The message text can include any of the dynamic values listed at the bottom of the tab. For example, to include the user's Big Data Discovery screen name in the message text, include `[$USER_SCREENNAME$]` in the message text.

   (e) To save the message configuration, click **Save**.
3. To configure the Password Changed notification:
   (a) Click the **Password Changed Notification** tab.

   (b) By default, the notification is enabled, meaning that when new users are created in Big Data Discovery, they receive the notification. To disable the notification, deselect the **Enabled** check box.

   (c) In the **Subject line** field, type the text of the email subject line.

   The subject line can include any of the dynamic values listed at the bottom of the tab. For example, to include the user's Big Data Discovery screen name in the subject line, include `[$USER_SCREENNAME$]` in the subject line.

   (d) In the Body text area, type the text of the email message.

   The message text can include any of the dynamic values listed at the bottom of the tab. For example, to include the user's Big Data Discovery screen name in the message text, include `[$USER_SCREENNAME$]` in the message text.

   (e) To save the message configuration, click **Save**.

Chapter 14

Managing Projects from the Control Panel

The **Control Panel** provides options for Big Data Discovery administrators to configure and remove projects.

Configuring the project type

The project type determines whether the project is visible to users on the **Catalog**.

The project types are:

Project Type	Description
Private	• The project Creator and Studio Administrators are the only users with access • The **All Big Data Discovery users** group is set to **No Access** Projects are Private by default. Access must be granted by the Creator or by a Studio Administrator.
Public	• The **All Big Data Discovery users** group is set to **Project Restricted Users** Public projects grant view access to Studio users.
Shared	The project has been modified in any of the following ways: • Users other than the Creator are added to the project • User Groups other than **All Big Data Discovery admins** and **All Big Data Discovery users** are added to the project • The **All Big Data Discovery users** group is set to **Project Authors** Projects are set to Shared to indicate changes from the default Public or Private permissions.

If you change the project type, then the page visibility type for all of the project pages changes to match the project type.

To change the project type for a project:

1. In the Studio header, click the **Configuration Options** icon and select **Control Panel**.
2. Select **User Settings>Projects**
3. Click the **Actions** link for the project, then select **Edit**
4. From the **Type** drop-down list, select the appropriate project type.

   You cannot explicitly select **Shared** as a project type. Instead, it is assigned if the default permissions have been modified.
5. Click **Save**.

Assigning users and user groups to projects

You can manage access to projects from the **Project Settings>Sharing** page or from the project details panel in the Catalog. For details, see "Assigning project roles" in the *Data Exploration and Analysis Guide*.

Certifying a project

Big Data Discovery administrators can certify a project.

Certifying a project can be used to indicate that the project content and functionality has been reviewed and the project is approved for use by all users who have access to it.

Note that only Big Data Discovery administrators can certify a project. Project Authors cannot change the certification status.

To certify a project:

1. From the **Control Panel**, select **User Settings>Projects**.
2. Click the **Actions** link for the project, then click **Edit**.
3. On the project configuration page, to certify the project, select the **Certified** check box.
4. Click **Save**.

Making a project active or inactive

By default, a new project is marked as active. From the **Control Panel**, Big Data Discovery administrators can control whether a project is active or inactive. Inactive projects are not displayed on the **Catalog**.

Note that this option only available to Big Data Discovery administrators.

To make a project active or inactive:

1. In the Studio header, click the **Configuration Options** icon and select **Control Panel**.
2. Select **User Settings>Projects**
3. Click the **Actions** link for the project, then click **Edit**.

4. To make the project inactive, deselect the **Active** check box. If the project is inactive, then to make the project active, check the **Active** check box.
5. Click **Save**.

Deleting projects

From the **Control Panel**, Big Data Discovery administrators can delete projects.

To delete a project:

1. From the **Control Panel**, select **User Settings>Projects**.
2. Click the **Actions** link for the project you want to remove.
3. Click **Delete**.

Part V

Controlling User Access to Studio

Chapter 15

Configuring User-Related Settings

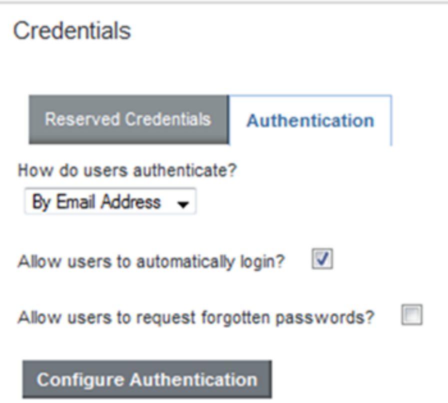
You configure settings for passwords and user authentication in the Studio **Control Panel**.

Configuring authentication settings for users

Each user has both an email address and a screen name. By default, users log in to Studio using their email addresses.

To configure the authentication settings for users:

1. In the Studio header, click the **Configuration Options** icon and select **Control Panel**.
2. Select **Platform Settings>Credentials** .
3. On the **Credentials** page, click the **Authentication** tab.



How do users authenticate?
By Email Address

Allow users to automatically login? ☑

Allow users to request forgotten passwords? □

4. From the **How do users authenticate?** list, select the name used to log in.

   To enables users log in using their email address, select **By Email Address**. This is the default.

   To enable users log in using their screen name, select **By Screen Name**.
5. To enable the **Remember me** option on the login page, so that login information is saved when users log in, select the **Allow users to automatically login?** check box.
6. To enable the **Forgot Your Password?** link on the login page, so that users can request a new password if they forget it, select the **Allow users to request forgotten passwords?** check box.
7. Click **Save**.

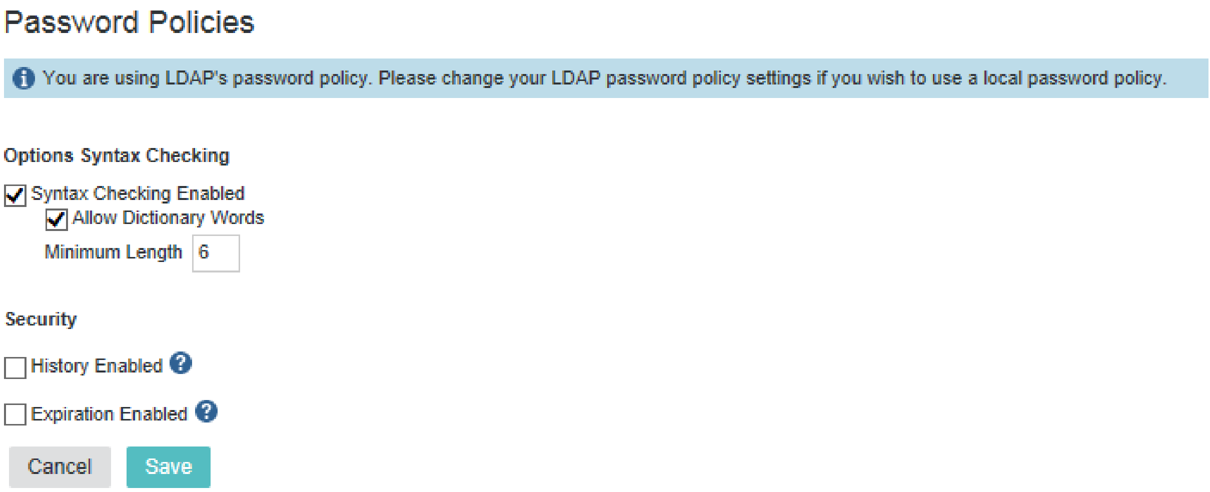
Configuring the password policy

The password policy sets the requirements for creating and setting Studio passwords. These options do not apply to Studio passwords managed by an LDAP system.

To configure the password policy:

1. Select **Configuration Options>User Settings>Password Policies**.

   The **Password Policies** page displays.

   Password Policies

   You are using LDAP's password policy. Please change your LDAP password policy settings if you wish to use a local password policy.

   Options Syntax Checking

   ☑ Syntax Checking Enabled

   ☑ Allow Dictionary Words

   Minimum Length 6

   Security

   □ History Enabled

   □ Expiration Enabled

2. Under **Options Syntax Checking** to enable syntax checking (enforcing password requirements), select **Syntax Checking Enabled**.

   If the box is not selected, then there are no restrictions on the password format.

3. If syntax checking is enabled, then:

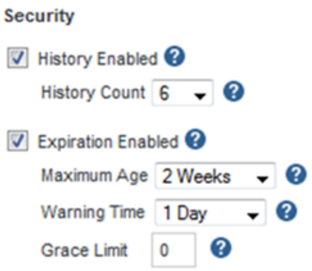
   (a) To allow passwords to include words from the dictionary, select the **Allow Dictionary Words** check box.

   If the box is not selected, then passwords cannot include words.

   (b) In the **Minimum Length** field, type the minimum length of a password.

4. To prevent users from using a recent previous password:

   (a) Under **Security**, select the **History Enabled** check box.

   Security

   ☑ History Enabled

   History Count 6

   ☑ Expiration Enabled

   Maximum Age 2 Weeks

   Warning Time 1 Day

   Grace Limit 0

   (b) From the **History Count** list, select the number of previous passwords to save and prevent the user from using.

      For example, if you select 6, then users cannot use their last 6 passwords.

5. To enable password expiration:

   (a) Select the **Expiration Enabled** check box.

      You should not enable expiration if users cannot change their passwords in Big Data Discovery.

   (b) From the **Maximum Age** list, select the amount of time before a password expires.

   (c) From the **Warning Time** list, select the amount of time before the expiration to begin displaying warnings to the user.

   (d) In the **Grace Limit** field, type the number of times a user can log in using an expired password.

6. Click **Save**.

Restricting the use of specific screen names and email addresses

If needed, you can configure lists of screen names and email addresses that should not be used for Studio users.

To restrict the user of specific screen names and email addresses:

1. In the Studio header, click the **Configuration Options** icon and select **Control Panel**.
2. Select **Platform Settings>Credentials** .
3. On the **Reserved Credentials** tab, in the **Screen Names** text area, type the list of screen names that cannot be used.

   Put each screen name on a separate line.

4. In the **Email Addresses** text area, type the list of email addresses that cannot be used.

   Put each email address on a separate line.

Chapter 16

Creating and Editing Studio Users

In Studio, roles are used to control access to general features as well as to access specific projects and data. The **Users** page on the **Control Panel** provides options for creating and editing Studio users.

About role privileges

Each Studio user is assigned a user role. The user role determines a user's access to features within Studio.

User roles and project roles

Studio roles are divided into Studio-wide user roles and project-specific roles. The user roles are Administrator, Power User, Restricted User, and User. These roles control access to Studio features in data sets, projects, and Studio administrative configuration. The project-specific roles are Project Author and Project Restricted User. These roles control access to project-specific configuration and project data. All Studio users have a user role, and they may also have project-specific roles that have been assigned to them individually or to any of their user groups.

Administrators can assign user roles. They also have Project Author access to all projects, which allows them to assign project roles as well.

Inherited roles

A Studio user might have a number of assigned roles. In addition to a user role, they may have a project-specific role and belong to a user group that grants additional roles. In these cases, the highest privileges apply to each area of Studio, regardless of if these privileges have been assigned directly or inherited from a user group.

User Roles

The user roles are as follows:

Role	Description
Administrator	Administrators have full access to all features in Studio. Administrators can: • Access the **Control Panel** • Create and delete data sets and projects • Transform data within a project • View, configure, and manage all projects
Power User	Power users can: • Create and delete data sets and projects • Transform data within a project • Export data to HDFS and create new data sets • View, configure, and manage projects for which they have a project role • Edit their account information Power users cannot: • Access the **Control Panel**
User	Users can: • Create and delete data sets and projects • Transform data within a project • View, configure, and manage projects for which they have a project role • Edit their account information Users cannot: • Access the **Control Panel** • Export data to HDFS

Role	Description
Restricted User	This is the default user role for new users. It has the most restricted privileges and is essentially a read-only role. This is the default user role for new users. Restricted users can: • Create new projects • View data sets in the Catalog • View, configure, and manage projects for which they have a project role Restricted users cannot: • Edit their account information • Access the **Control Panel** • Create new data sets • Transform data within a project • Export data to HDFS

Note: Power Users, Users, and Restricted Users have no project roles by default, but they can access any projects that grant roles to the **All Big Data Discovery users** group. They can also access projects for which they have a project role, outlined below.

Project Roles

Project roles grant access privileges to project content and configuration. You can assign project roles to individual users or to user groups, and they define access to a given project regardless of a user's user role in Big Data Discovery Studio. The roles are:

Role	Description
Project Author	Project authors can: • Configure and manage a project • Add or remove users and user groups • Assign user and user group roles • Transform project data • Export project data Project authors cannot: • Create new data sets • Access the Big Data Discovery Control Panel

Role	Description
Project Restricted User	Project Restricted Users can: • View a project and navigate through the configured pages • Add and configure project pages and components Project restricted users cannot: • Access **Project Settings** • Create new data sets • Transform data • Export project data

Creating a new Studio user

If you are not using LDAP, you may want to create Studio users manually.

For example, for a small development instance, you may just need a few users to develop and test projects. Or if your LDAP users for a production site are all end users, you may need a separate user account for administering the site.

To create a new Studio user:

1. In the Studio header, click the **Configuration Options** icon and select **Control Panel**.
2. Select **User Settings>Users** .
3. Click **Add**.

   The **Details** page for the new user displays.
4. In the **Screen Name** field, type the screen name for the user.

   The screen name must be unique, and cannot match the screen name of any current active or inactive user.
5. In the **Email Address** field, type the user's email address.
6. For the user's name, enter values for at least the **First Name** and **Last Name** fields.

   The **Middle Name** field is optional.
7. To create the initial password for the user:

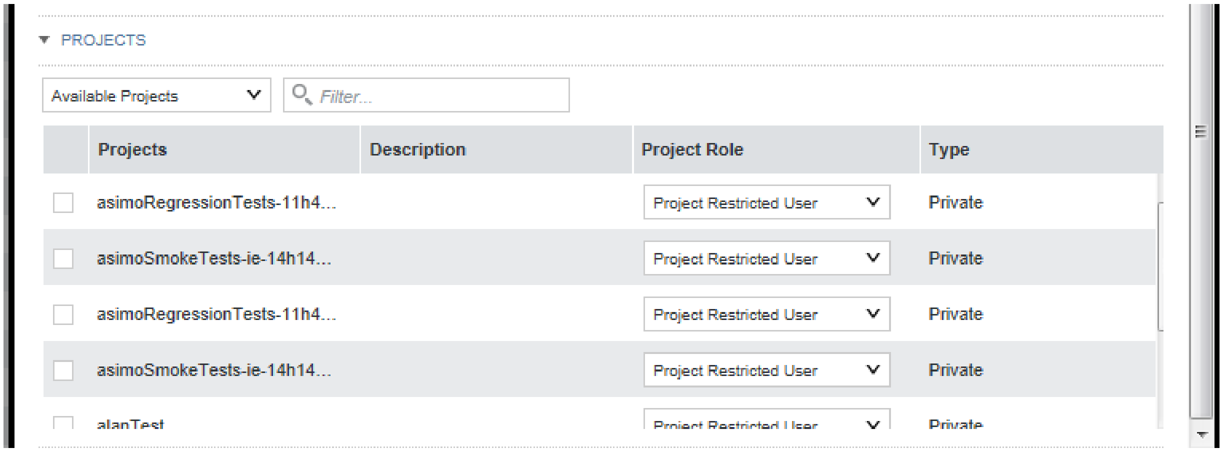
   (a) In the **Password** field, enter the password to assign to the new user.

   (b) In the **Retype Password** field, type the password again.

   By default, the Studio password policy requires users to change their password the first time they log in.
8. From the **Locale** list, select the preferred locale for the user.
9. From the **Role** list, select the user role to assign to the user.

   For details, see *About role privileges on page 116*.

10. From the **Projects** section at the bottom of the dialog, to assign the user to projects:



	Projects	Description	Project Role	Type
□	asimoRegressionTests-11h4...		Project Restricted User	Private
□	asimoSmokeTests-ie-14h14...		Project Restricted User	Private
□	asimoRegressionTests-11h4...		Project Restricted User	Private
□	asimoSmokeTests-ie-14h14...		Project Restricted User	Private
□	alanTest		Project Restricted User	Private

(a) Select the check box next to each project you want the new user to be a member of.

(b) For each project, from the **Role** list, select the project role to assign to the user.

11. Click **Save**.

The user is added to the list of users.

Editing a Studio user

The **Users** page also allows you to edit a user's account.

From the **Users** page, to edit a user:

1. In the Studio header, click the **Configuration Options** icon and select **Control Panel**.
2. Select **User Settings>Users**
3. Click the **Actions** button next to the user.
4. Click **Edit**.
5. To change the user's password:

   (a) In the **Password** field, type the new password.

   (b) In the **Retype Password** field, re-type the new password.
6. To change the user role, from the **Role** list, select the new role.
7. Under **Projects**, to add a user as an project member:

   (a) Make sure the list is set to **Available Projects**. These are projects the user is not yet a member of.

   (b) Select the check box next to each project you want to add the user to.

   (c) For each project, from the **Role** list, select the project role to assign to the user.

8. Under **Projects**, to change the project role for or remove the user from a project:

   (a) From the list, select **Assigned Projects**.

   The list shows the projects the user is currently a member of.

   (b) To change the user's project role, from the **Role** drop-down list, select the new project role.

   (c) To remove the user from a project, deselect the check box.

9. Click **Save**.

Deactivating, reactivating, and deleting Studio users

From the **Users** page of the **Control Panel**, you can make an active user inactive. You can also reactivate or delete inactive users.

Note that you cannot make your own user account inactive, and you cannot delete an active user.

From the **Users** page, to change the status of a user account:

1. To make an existing user inactive:

   (a) In the users list, select the check box for the user you want to deactivate.

   (b) Click **Deactivate**.

   Big Data Discovery prompts you to confirm that you want to deactivate the user.

   The user is then removed from the list of active users.

   Note that inactive users are not removed from Big Data Discovery.

2. To reactivate or delete an inactive user:

   (a) Click the **Advanced** link below the user search field.

   Big Data Discovery displays additional user search fields.

   (b) From the **Active** list, select **No**.

   Note that if you change the **Match type** to **Any**, you must also provide search criteria in at least one of the other fields.

   (c) Click **Search**.

   The users list displays only the inactive users.

   (d) Select the check box for the user you want to reactivate or delete.

   (e) To reactivate the user, click **Restore**.

   (f) To delete the user, click **Delete**.

Chapter 17

Integrating with an LDAP System to Manage Users

If you have an LDAP system, you can allow users to use those credentials to log in to Big Data Discovery.

About using LDAP

Integrating Studio with Lightweight Directory Access Protocol (LDAP) allows users to sign in to Studio using their existing LDAP user accounts, rather than creating separate user accounts from within Studio. LDAP is also used when integrating with a single sign-on (SSO) system.

You can integrate Studio with one LDAP directory but not multiple LDAP directories.

You can set up mixed authentication systems with both LDAP and manually created Studio users. In such a scenario, Studio pulls users and groups from an LDAP directory, and you can supplement those LDAP users with additional Studio users that you create.

If Studio uses LDAP for user management, you are notified in a blue banner across the **Password Policies** page. In this scenario, Studio relies entirely on the LDAP system for user names, passwords, syntax checking, minimum length settings, and so on. The settings on the **Password Policies** page do not apply to your LDAP users. However, if you create users directly in Studio, you can modify some basic settings about the password configuration on the **Password Policies** page.

Configuring the LDAP settings and server

The LDAP settings on the **Control Panel>Credentials** page include whether LDAP is enabled and required for authentication, the connection to the LDAP server, and whether to support batch import or export to or from the LDAP directory. The method for processing batch imports is set in `portal-ext.properties`.

In `portal-ext.properties`, the setting `ldap.import.method` determines how to perform batch imports from LDAP. This setting is only applied if batch import is enabled. The available values for `ldap.import.method` are:

Value	Description
`user`	Specifies a user-based import. This is the default value. User-based batch import uses the import search filter configured in the **User Mapping** section of the **LDAP** tab. For user-first import, Big Data Discovery: 1. Uses the user import search filter to run an LDAP search query. 2. Imports the resulting list of users, including all of the LDAP groups the user belongs to. The group import search filter is ignored.
`group`	Specifies a group-based import. Group-based import uses the import search filter configured in the **Group Mapping** section of the **LDAP** tab. For group-based import, Big Data Discovery: 1. Uses the group import search filter to run an LDAP search query. 2. Imports the resulting list of groups, including all of the users in those groups. The user import search filter is ignored.

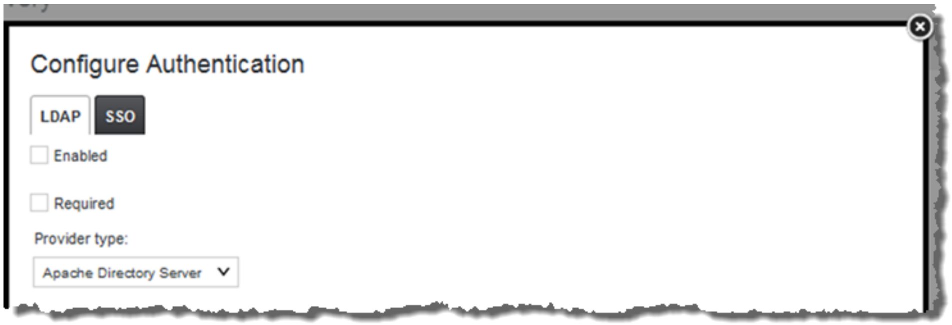
The value you should use depends partly on how your LDAP system works. If your LDAP directory only provides user information, without any groups, then you have to use user-based import. If your LDAP directory only provides group information, then you have to use group-based import.

To configure the LDAP settings:

1. In the Big Data Discovery header, click the **Configuration Options** icon and select **Control Panel**.
2. Click **Credentials>Authentication**

3. Click the **Configure Authentication** button.

   The **Configure Authentication** dialog displays, with the **LDAP** tab selected.



Configure Authentication

□ Enabled

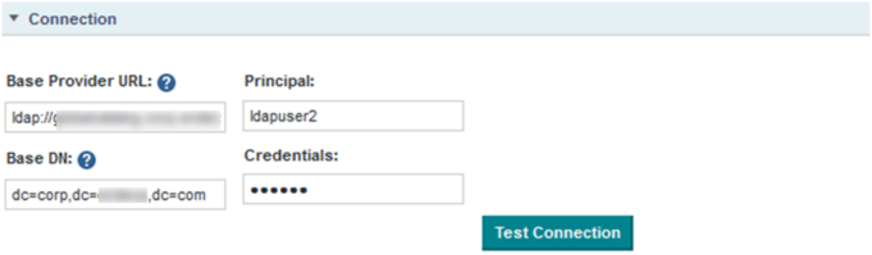
□ Required

Provider type: Apache Directory Server

4. To enable LDAP authentication, select the **Enabled** check box.
5. To only allow users to log in using an LDAP account, select the **Required** check box.

   If this box is selected, then any users that you create manually in Big Data Discovery cannot log in. To make sure that users you create manually can log in, make sure that this box is deselected.
6. To populate the LDAP server configuration fields with default values based on a specific type of provider, select the type of server you are using from the **Provider type** list.

   If you select the **Custom** option, then the fields are cleared.
7. The **Connection** settings cover the basic connection to LDAP:



Connection

Base Provider URL: ldap://[redacted]

Principal: ldapuser2

Base DN: dc=corp,dc=[redacted],dc=com

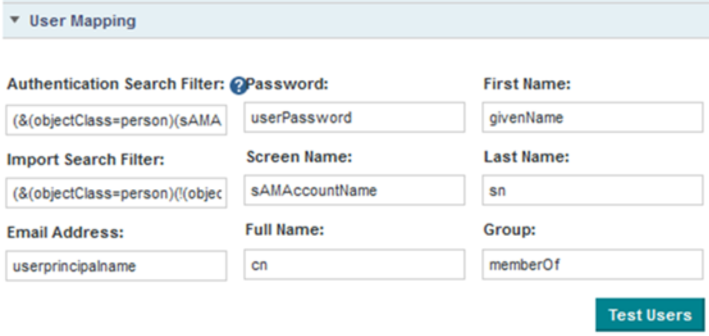
Credentials: ••••••

Field	Description
Base Provider URL	The location of your LDAP server. Make sure that the machine on which Big Data Discovery is installed can communicate with the LDAP server. If there is a firewall between the two systems, make sure that the appropriate ports are opened.
Base DN	The Base Distinguished Name for your LDAP directory. For a commercial organization, it may look something like: `dc=companynamehere,dc=com`

Field	Description
Principal	The user name of the administrator account for your LDAP system. This ID is used to synchronize user accounts to and from LDAP.
Credentials	The password for the administrative user.

After providing the connection information, to test the connection to the LDAP server, click **Test Connection**.

8. Under **User Mapping**:



Authentication Search Filter: (&(objectClass=person)(sAMA

Password: userPassword

First Name: givenName

Import Search Filter: (&(objectClass=person)(!(objec

Screen Name: sAMAccountName

Last Name: sn

Email Address: userprincipalname

Full Name: cn

Group: memberOf

(a) Use the search filter fields to configure the filters for finding and identifying users in your LDAP directory.

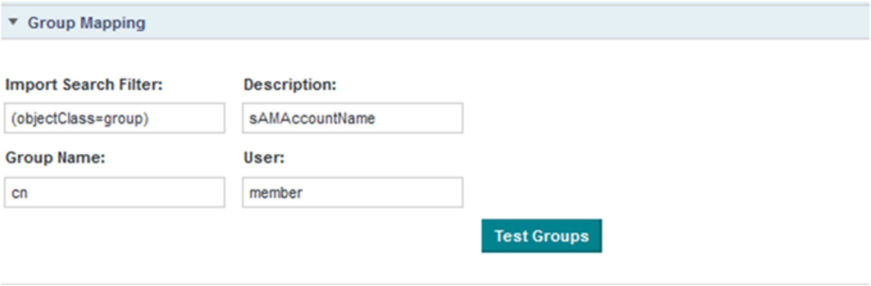
Field	Description
Authentication Search Filter	The search criteria for user logins. If you do not enable batch import of LDAP users, then the first time a user tries to log in, Big Data Discovery uses this authentication search filter to search for the user in the LDAP directory. By default, users log in using their email address. If you have changed this setting, you must modify the search filter here. For example, if you changed the authentication method to use the screen name, you would modify the search filter so that it can match the entered login name: `(cn=@screen_name@)`

Field	Description
Import Search Filter	The search filter to use for batch import of users. This filter is used if: • You enable batch import of LDAP users • In `portal-ext.properties`, `ldap.import.method` is set to `user` Depending on the LDAP server, there are different ways to identify the user. The default setting (`objectClass=inetOrgPerson`) usually is fine, but to search for only a subset of users or for users that have different object classes, you can change this.

(b) Use the remaining fields to map your LDAP attributes to the Big Data Discovery user fields.

(c) After setting up the attribute mappings, to test the mappings, click **Test Users**.

9. Under **Group Mapping**, map your LDAP groups.



Import Search Filter: (objectClass=group)

Description: sAMAccountName

Group Name: cn

User: member

(a) In the **Import Search Filter** field, type the filter for finding LDAP groups.

This filter is used if:

- You enable batch import of LDAP users
- In `portal-ext.properties`, `ldap.import.method` is set to `group`

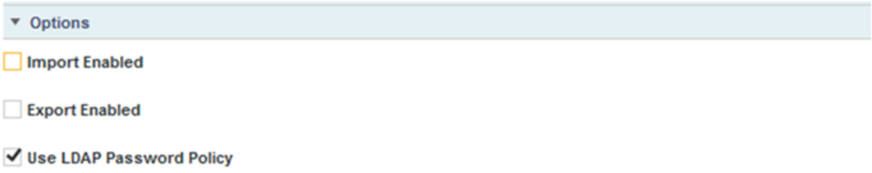
(b) Map the following group fields:

- Group Name
- Description
- User

(c) To test the group mappings, click **Test Groups**.

The system displays a list of the groups returned by your search filter.

10. The **Options** section is used to configure importing and exporting of LDAP user data and to select the password policy:



▾ Options

□ Import Enabled

□ Export Enabled

☑ Use LDAP Password Policy

   (a) If you selected the **Import Enabled** check box, then batch import of LDAP users is enabled.

   If you did not select this box, then Big Data Discovery synchronizes each user as they log in. It is recommended that you leave this box deselected.

   If you do enable batch import, then the import process is based on the value of `ldap.import.method`.

   Note also that when using batch import, you cannot filter both the imported users and imported groups at the same time. For user-based batch import mode, you cannot filter the LDAP groups to import. For group-based batch import mode, you cannot filter the LDAP users to import.

   (b) If the **Export Enabled** check box is selected, then any changes to the user in Big Data Discovery are exported to the LDAP system.

   It is recommended that you leave this box deselected.

   (c) To use the password policy from your LDAP system, instead of the Big Data Discovery password policy, select the **Use LDAP Password Policy** check box.

Preventing encrypted LDAP passwords from being stored in BDD

By default, when you use LDAP for user authentication, each time a user logs in, Big Data Discovery stores a securely encrypted version of their LDAP password. For subsequent logins, Big Data Discovery can then authenticate the user even when it cannot connect to the LDAP system. For even stricter security, you can configure Big Data Discovery to prevent the passwords from being stored.

To prevent Big Data Discovery from storing the encrypted LDAP passwords:

1. Stop Big Data Discovery.
2. Add the following settings to `portal-ext.properties`:

```
ldap.password.cache.hashed=false
ldap.auth.required=true
auth.pipeline.enable.liferay.check=false
```

3. Restart Big Data Discovery.

Big Data Discovery no longer stores the encrypted LDAP passwords for authenticated users. If the LDAP system is unavailable, Big Data Discovery cannot authenticate previously authenticated users.

Assigning roles based on LDAP user groups

For LDAP integration, it is recommended that you assign roles based on your LDAP groups.

To ensure that users have the correct roles as soon as they log in, you create groups in Big Data Discovery that have the same name as your LDAP groups, but in lowercase, and assign the correct roles to each group.

To create a user group, and assign roles to that group:

1. In the Big Data Discovery header, click the **Configuration Options** icon and select **Control Panel**.
2. Select **User Settings>User Groups**.
3. On the **User Groups** page, to add a new group, click **Add**.

   The **Add Group** dialog displays.
4. In the **Name** field, type the name of the group.

   Make sure the name is the lowercase version of the name of a group from your LDAP system. For example, if the LDAP group is called `SystemUsers`, then the user group name would be `systemusers`.
5. In the **Description** field, type a description of the group.
6. To assign roles to the group, from the **Role** list, select the user role to assign to the group.

   The selected roles are assigned to all of the users in the group. For details on the available user roles, see *About role privileges on page 116*.
7. Click **Save**.

   The group is added to the **User Groups** list.

Chapter 18

Setting up Single Sign-On (SSO)

You can provide user access by integrating with an SSO system.

About using single sign-on

Integrating with single sign-on (SSO) allows Studio users to be logged in to Big Data Discovery automatically once they are logged in to your SSO system.

Note that once Big Data Discovery is integrated with SSO, you cannot create and edit users from within Big Data Discovery. All users get access to Big Data Discovery using their SSO credentials. This means that you can no longer use the default administrative user provided with Big Data Discovery. You will need to make sure that there is at least one SSO user with an Administrator user role for Big Data discovery.

The officially supported method for integrating with SSO is to use Oracle Access Manager, with an Oracle HTTP Server in front of the Big Data Discovery application server. While you may be able to use another SSO tool that supports passing the user name in an HTTP header, you would have to use the documentation and support materials for that tool in order to set up the integration.

The information in this guide focuses on the details and configuration that are specific to the Big Data Discovery integration. For general information on installing Oracle Access Manager and Oracle HTTP Server, see the associated documentation for those products.

Overview of the process for configuring SSO with Oracle Access Manager

Here is an overview of the steps for using Oracle Access Manager to implement SSO in Big Data Discovery.

1. Install Oracle Access Manager 11g, if you haven't already. See the Oracle Access Manager documentation for details.
2. Install Oracle HTTP Server (OHS) 11g. See the Oracle HTTP Server documentation for details.
3. Install OHS Webgate 11g. See the Webgate documentation for details.

4. Create an instance of OHS and confirm that it is up and running. See the OHS documentation for details.
5. Configure the reverse proxy module for the Big Data Discovery application server in Oracle HTTP Server. See *Configuring the reverse proxy module in OHS on page 130*.
6. Install the Webgate module into the Oracle HTTP Server. See *Registering the Webgate with the Oracle Access Manager server on page 131*.
7. In Big Data Discovery, configure the LDAP connection for your SSO implementation. See *Configuring the LDAP connection for SSO on page 133*.
8. In Big Data Discovery, configure the Oracle Access Manager SSO settings. See *Configuring the Oracle Access Manager SSO settings on page 134*.
9. Configure Big Data Discovery's web server settings to use the OHS server. See *Completing and testing the SSO integration on page 135*.
10. Disable direct access to the Big Data Discovery application server, to ensure that all traffic to Big Data Discovery is routed through OHS.

Configuring the reverse proxy module in OHS

For WebLogic Server, you need to update the file `mod_wl_ohs.conf` to add the logout configuration for SSO.

Here is an example of the file with the `/bdd/oam_logout_success` section added:

```
LoadModule weblogic_module   "${ORACLE_HOME}/ohs/modules/mod_wl_ohs.so"
<IfModule weblogic_module>
      WebLogicHost hostName
      WebLogicPort portNumber
</IfModule>

<Location /bdd/oam_logout_success>
      PathTrim /bdd/oam_logout_success
      PathPrepend /bdd/c/portal
      DefaultFileName logout
      SetHandler weblogic-handler
</Location>

<Location />
      SetHandler weblogic-handler
</Location>
```

The `/bdd/oam_logout_success Location` configuration is special for Big Data Discovery. It redirects the default Webgate Logout Callback URL (`/bdd/oam_logout_success`) to an application tier logout within Big Data Discovery. With this configuration, when users sign out of SSO from another application, it is reflected in Big Data Discovery.

Registering the Webgate with the Oracle Access Manager server

After you have installed the OHS Webgate, you use the remote registration (RREG) tool to register the OHS Webgate with the OAM server.

To complete the registration:

1. Obtain the RREG tarball (`rreg.tar.gz`) from the Oracle Access Manager server.
2. Extract the file to the OHS server.
3. Modify the script `oamreg.sh`.

   Correct the `OAM_REG_HOME` and `JAVA_HOME` environment variables.

   `OAM_REG_HOME` should point to the extracted `rreg` directory created in the previous step.

   You may not need to change `JAVA_HOME` if it's already set in your environment.
4. In the `input` directory, create an input file for the RREG tool. The file can include the list of resources secured by this Webgate.

   You can omit this list if the application domain already exists.

   Here is an example of an input file where the resources have not been set up for the application domain and host in Oracle Access Manager:

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

   <OAM11GRegRequest>

   <serverAddress>http://oamserver.us.mycompany.com:7001</serverAddress>
   <hostIdentifier>myserver-1234</hostIdentifier>
   <agentName>myserver-1234-webgate</agentName>
   <applicationDomain>Big Data Discovery</applicationDomain>
   <protectedResourcesList>
     <resource>/bdd</resource>
     <resource>/bdd/.../*</resource>
   </protectedResourcesList>
   <publicResourcesList>
     <resource>/public/index.html</resource>
   </publicResourcesList>
   <excludedResourcesList>
     <resource>/excluded/index.html</resource>
   </excludedResourcesList>

   </OAM11GRegRequest>
   ```

   In this example, the resources have already been set up in Oracle Access Manager:

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

   <OAM11GRegRequest>

   <serverAddress>http://oamserver.us.mycompany.com:7001</serverAddress>
   <hostIdentifier>myserver-1234</hostIdentifier>
   <agentName>myserver-1234-webgate</agentName>
   <applicationDomain>Big Data Discovery</applicationDomain>

   </OAM11GRegRequest>
   ```

In the input file, the parameter values are:

Parameter Name	Description
`serverAddress`	The full address (`http://`*`host`*`:`*`port`*) of the Oracle Access Manager administrative server. The port is usually 7001.
`hostIdentifier`	The host identifier string for your host. If you already created a host identifier in the Oracle Access Manager console, use its name here.
`agentName`	A unique name for the new Webgate agent. Make sure it doesn't conflict with any existing agents in the application domain.
`applicationDomain`	A new or existing application domain to add this agent into. Each application domain may have multiple agents. An application domain associates multiple agents with the same authentication and authorization policies.

5. Run the tool:

   ```
   ./bin/oamreg.sh inband input/inputFileName
   ```

   For example:

   ```
   ./bin/oamreg.sh inband input/my-webgate-input.xml
   ```

   When the process is complete, you'll see the following message:

   ```
   Inband registration process completed successfully! Output artifacts are created in the
   output folder.
   ```

6. Copy the generated output files from the `output` directory to the OHS instance `config` directory (under `webgate/config/`).
7. Restart the OHS instance.
8. Test your application URL via OHS.

   It should forward you to the SSO login form.

   Check the OAM console to confirm that the Webgate is installed and has the correct settings.

Testing the OHS URL

Before continuing to the Big Data Discovery configuration, you need to test that the OHS URL redirects correctly to Big Data Discovery.

To test the OHS URL, use it to browse to Big Data Discovery.

You should be prompted to authenticate using your SSO credentials.

Because you have not yet configured the Oracle Access Manager SSO integration in Big Data Discovery, after you complete the authentication, the Big Data Discovery login page displays.

Log in to Big Data Discovery using an administrator account.

Configuring Big Data Discovery to integrate with SSO via Oracle Access Manager

In Big Data Discovery, you configure the LDAP connection and Oracle Access Manager connection settings.

Configuring the LDAP connection for SSO

Configuring the Oracle Access Manager SSO settings

Configuring the LDAP connection for SSO

The SSO implementation uses LDAP to retrieve and maintain the user information. For the Oracle Access Manager SSO, you configure Big Data Discovery to use Oracle Internet Directory for LDAP.

In Big Data Discovery, to configure the LDAP connection for SSO:

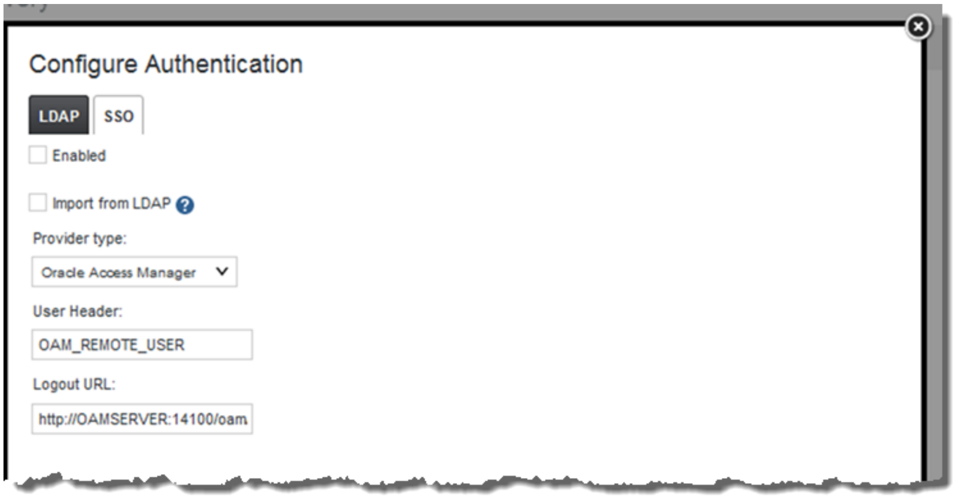
1. From the **Control Panel**, select **Platform Settings>Credentials**.
2. On the **Credentials** page, click **Authentication**.
3. On the **Authentication** tab, click the **Configure Authentication** button.

   The **Configure Authentication** dialog is displayed, with the **LDAP** tab selected.
4. On the **LDAP** tab, check the **Enabled** check box. Do not check the **Required** check box.
5. From the **Default values** drop-down list, select **Oracle Internet Directory**.
6. Configure the LDAP connection, users, and groups as described in *Configuring the LDAP settings and server on page 123*.
7. To save the LDAP connection information, click **Save**.
8. Configure the user roles for your user groups as described in *Assigning roles based on LDAP user groups on page 128*.

Configuring the Oracle Access Manager SSO settings

After you configure the LDAP connection for your SSO integration, you configure the Oracle Access Manager SSO settings.

The settings are on the **SSO** tab on the **Configure Authentication** dialog.



Configure Authentication

□ Enabled

□ Import from LDAP

Provider type: Oracle Access Manager

User Header: OAM_REMOTE_USER

Logout URL: http://OAMSERVER:14100/oam

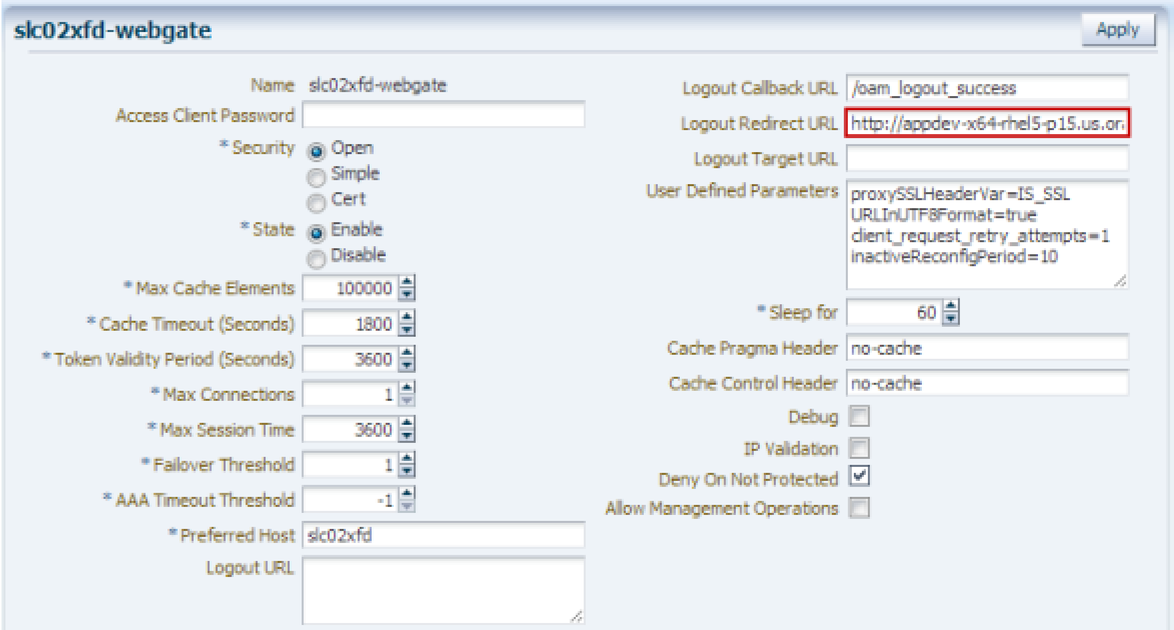
To configure the SSO settings:

1. From the **Control Panel**, select **Platform Settings>Credentials**.
2. In the **Credentials** page, click **Authentication**.
3. On the **Authentication** tab, click **Configure Authentication**.
4. On the **Configure Authentication** dialog, click **SSO**.
5. Select the **Enabled** check box.
6. Select the **Import from LDAP** check box.
7. From the **Provider Type** list, select **Oracle Access Manager**.

   Note that the only other option is **Custom**, which clears the fields. You would use the **Custom** option if you are using some other tool that passes the user name in an HTTP header. For information on setting up an SSO tool other than Oracle Access Manager, see the documentation and support materials for that tool.
8. Leave the default user header `OAM_REMOTE_USER`.

9. In the **Logout URL** field, provide the URL to navigate to when users log out.

   Make sure it is the same logout redirect URL you have configured for the Webgate:



slc02xfd-webgate

Name	slc02xfd-webgate
Access Client Password	
* Security	● Open ○ Simple ○ Cert
* State	● Enable ○ Disable
* Max Cache Elements	100000
* Cache Timeout (Seconds)	1800
* Token Validity Period (Seconds)	3600
* Max Connections	1
* Max Session Time	3600
* Failover Threshold	1
* AAA Timeout Threshold	-1
* Preferred Host	slc02xfd
Logout URL	
Logout Callback URL	/oam_logout_success
Logout Redirect URL	http://appdev-x64-rhel5-p15.us.or
Logout Target URL	

User Defined Parameters	proxySSLHeaderVar=IS_SSL
	URLInUTF8Format=true
	client_request_retry_attempts=1
	inactiveReconfigPeriod=10

* Sleep for	60
Cache Pragma Header	no-cache
Cache Control Header	no-cache
Debug	□
IP Validation	□
Deny On Not Protected	☑
Allow Management Operations	□

   For the logout URL, you can add an optional `end_url` parameter to redirect the browser to a final location after users sign out. To redirect back to Big Data Discovery, configure `end_url` to point to the OHS host and port.

   For example:

```
http://oamserver.us.mycompany.com:14100/oam/server/logout?end_url=http:/
/bddhost.us.company.com:7777/
```

10. To save the configuration, click **Save**.

Completing and testing the SSO integration

The final step in setting up the SSO integration is to add the OHS server host name and port to `portal-ext.properties`.

To complete and test the SSO configuration:

1. In `portal-ext.properties`:

   If OHS is not using SSL, then add the following lines:

```
web.server.host=ohsHostName
web.server.http.port=ohsPortNumber
```

   If OHS is using SSL, then add the following lines:

```
web.server.protocol=https
web.server.host=ohsHostName
web.server.https.port=ohsPortNumber
```

Where:

- *`ohsHostName`* is the fully qualified domain name (FQDN) of the server where OHS is installed. The name must be resolvable by Big Data Discovery users.

  For example, you would use `webserver01.company.com`, and not `webserver01`.

  You need to specify this even if OHS is on the same server as Big Data Discovery.

- *`ohsPortNumber`* is the port number used by OHS.

2. Restart Big Data Discovery.

   Make sure to completely restart the browser to remove any cookies or sessions associated with the Big Data Discovery user login you used earlier.

3. Navigate to the Big Data Discovery URL. The Oracle Access Manager SSO form displays.

4. Enter your SSO authentication credentials.

   You are logged in to Big Data Discovery.

   As you navigate around Big Data Discovery, make sure that the browser URL continues to point to the OHS server and port.

Part VI

Administering Big Data Discovery Using Enterprise Manager Cloud Control

Chapter 19

Using Enterprise Manager for Big Data Discovery

This section describes how to use the Enterprise Manager plug-in for Big Data Discovery to administer Big Data Discovery components with Enterprise Manager Cloud Control.

Before using Enterprise Manager

About Enterprise Manager

Starting and stopping the Dgraph and Studio using Enterprise Manager

Logging for BDD targets

Configuring log levels for Dgraph targets

Dgraph Administration Operations in Enterprise Manager

Before using Enterprise Manager

Before you can use Enterprise Manager to manage Big Data Discovery targets, you must already have installed Enterprise Manager Cloud Control and deployed the Enterprise Manager plug-in.

For details about these tasks, see *Enterprise Manager Plug-in for Big Data Discovery Installation Guide*.

About Enterprise Manager

The Enterprise Manager Plug-in for Big Data Discovery extends Oracle Enterprise Manager Cloud Control to add support for monitoring, diagnosing, and managing Big Data Discovery components.

The Enterprise Manager plug-in supports three targets for Oracle Big Data Discovery components:

- Cluster target
- Studio target
- Dgraph target

The Dgraph target also includes information about the Dgraph HDFS Agent.

In addition to providing support for targets, the plug-in has several customized features, such as support for starting and stopping the Dgraph, and support for the Dgraph administrative operations.

The plug-in provides a convenient way to view and monitor logs and also search Studio and Dgraph queries.

The Cluster target

The Dgraph target

The Studio target

Roles and privileges for BDD targets

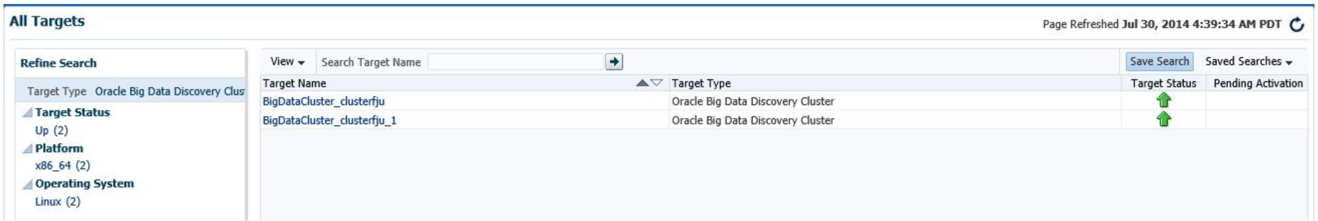
Security credentials for BDD targets

Connecting to Studio over a secure port

The Cluster target

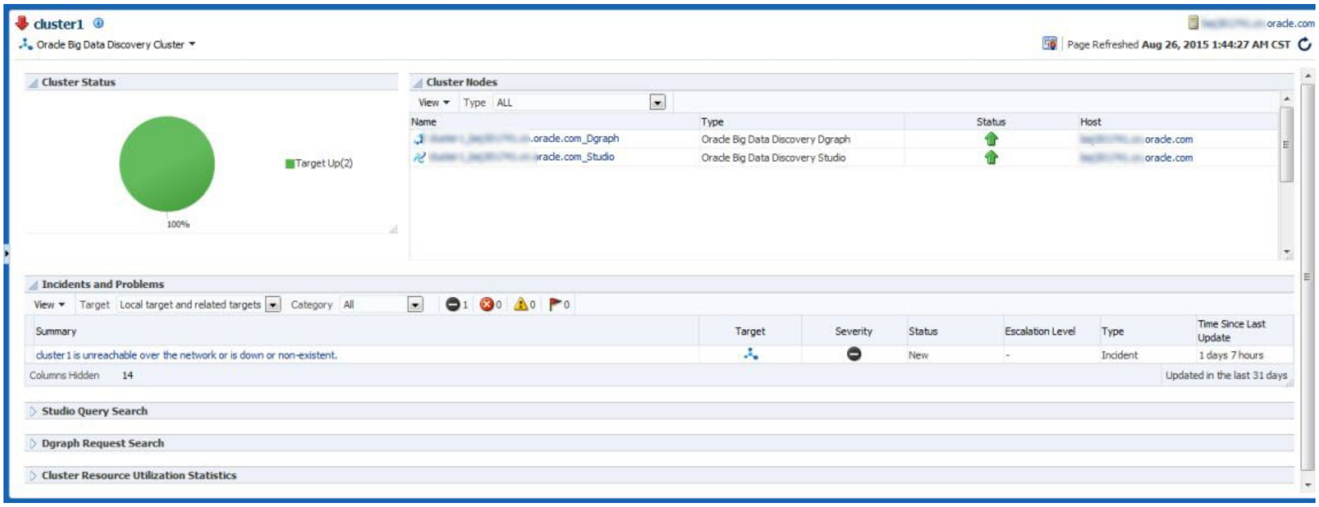
A Cluster target represents an entire Big Data Discovery cluster deployment, including the Studio and Dgraph instances.

The **All Targets** page provides a table that lists the Big Data Discovery clusters and corresponding status (up or down). For example:



Target Name	Target Type	Target Status	Pending Activation
BigDataCluster_clusterfju	Oracle Big Data Discovery Cluster	↑	
BigDataCluster_clusterfju_1	Oracle Big Data Discovery Cluster	↑	

Clicking a **Cluster** target in the table displays that cluster's home page, as in this example:



Name	Type	Status	Host
[redacted].oracle.com_Dgraph	Oracle Big Data Discovery Dgraph	↑	[redacted]oracle.com
[redacted]racle.com_Studio	Oracle Big Data Discovery Studio	↑	[redacted]oracle.com

Summary	Target	Severity	Status	Escalation Level	Type	Time Since Last Update
cluster 1 is unreachable over the network or is down or non-existent.		●	New	-	Incident	1 days 7 hours

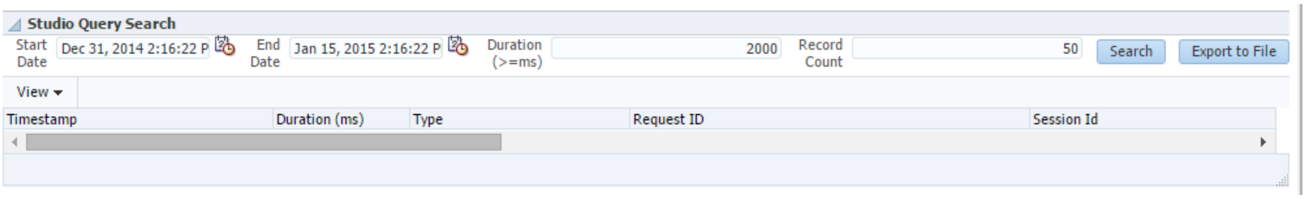
Columns Hidden 14

Updated in the last 31 days

On this page, you can see the status and name of each node in the cluster. In particular, on the Cluster page you can see which Dgraph is the leader node (this is useful because some of the Dgraph administrative operations can be run only from the leader node). Note that a Cluster target is attached to a WebLogic Admin Server, which means that the Cluster target is also an Admin Server target.

Some of the regions on the Cluster home page are standard to all Enterprise Manager plug-ins, such as the **Incidents and Problems** region. Other regions, such as **Cluster Resource Utilization Statistics**, are unique to the plug-in for Oracle Big Data Discovery.

Clicking the **Studio Query Search** region displays search features that allow you to search all Studio queries for a range of dates and identify any queries that took longer than the number of milliseconds you specify. For example:



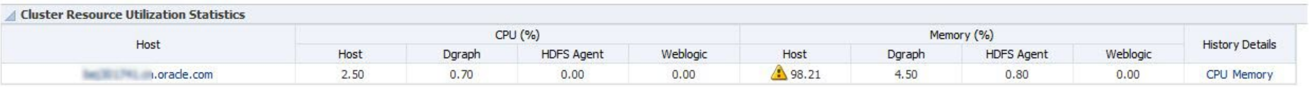
Studio Query Search

Start Date: Dec 31, 2014 2:16:22 P
End Date: Jan 15, 2015 2:16:22 P
Duration (>=ms): 2000
Record Count: 50

Timestamp	Duration (ms)	Type	Request ID	Session Id

Note that the **Studio Query Search** region also appears in the Studio target home page, while the **Dgraph Request Search** region is also available on the Dgraph target home page.

The **Cluster Resource Utilization Statistics** region collects and shows resource utilization data (CPU and memory usage) for different components in the BDD cluster (Dgraph process, Dgraph HDFS Agent process, WebLogic process, and the host). For example:



Cluster Resource Utilization Statistics

Host	CPU (%)				Memory (%)				History Details
	Host	Dgraph	HDFS Agent	Weblogic	Host	Dgraph	HDFS Agent	Weblogic	
[redacted].oracle.com	2.50	0.70	0.00	0.00	⚠ 98.21	4.50	0.80	0.00	CPU Memory

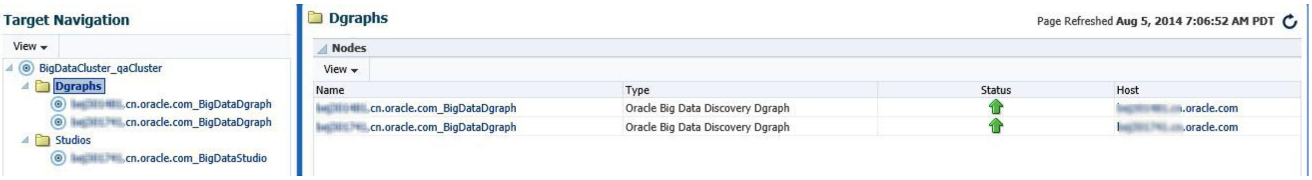
You can also export these results to a file in Excel.

The Dgraph target

A Dgraph target lists information about a single Dgraph instance running on a host and includes information about the Dgraph HDFS Agent associated with that Dgraph. You can view resource utilization, and search Dgraph queries and export your search results to a file in Excel. You can also search and export the HDFS Agent information.

Before accessing this page, set the default preferred credentials for the Dgraph target.

The **Dgraphs** folder page provides a table that lists all Dgraph nodes and each node's status (up or down). For example:



Target Navigation
View ▾
BigDataCluster_qaCluster
Dgraphs
[redacted].cn.oracle.com_BigDataDgraph
[redacted].cn.oracle.com_BigDataDgraph
Studios
[redacted].cn.oracle.com_BigDataStudio

Dgraphs

Page Refreshed **Aug 5, 2014 7:06:52 AM PDT**

Nodes

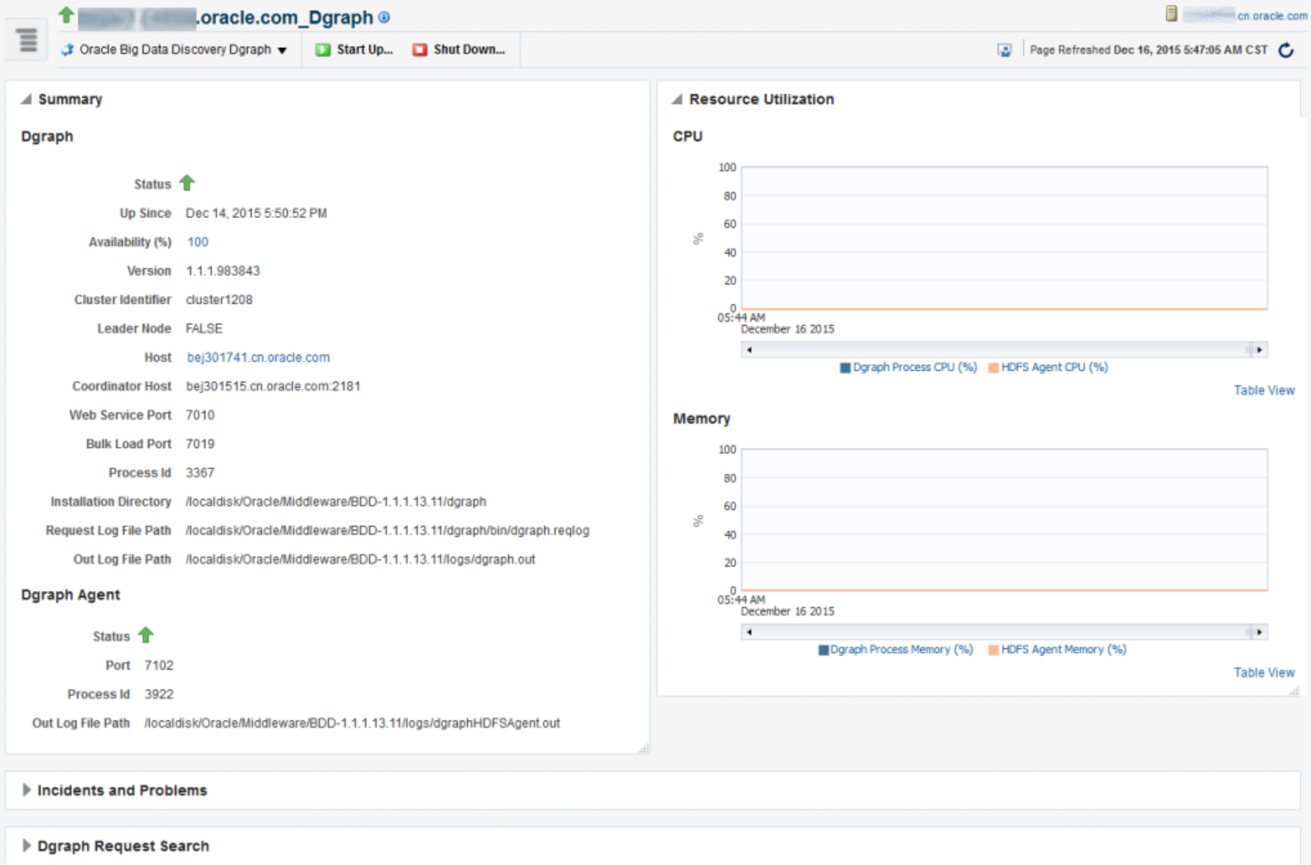
Name	Type	Status	Host
[redacted].cn.oracle.com_BigDataDgraph	Oracle Big Data Discovery Dgraph	⬆	[redacted].oracle.com
[redacted].cn.oracle.com_BigDataDgraph	Oracle Big Data Discovery Dgraph	⬆	[redacted].oracle.com

Clicking a Dgraph target in the table displays that Dgraph's home page. This page provides a way to start and stop the Dgraph. The Dgraph manages the Dgraph HDFS Agent, so starting or stopping the Dgraph also starts or stops the Dgraph HDFS Agent. The page also provides the following regions that describe the Dgraph node:

- A **Summary** region lists the basics about installation paths, general configuration, leader/follower information, and so on. This **Summary** region also includes an area for the Dgraph HDFS Agent.
- A **Resource Utilization** region is about CPU and RAM usage for the Dgraph and Dgraph HDFS Agent on that host.
- An **Incidents and Problems** region is standard to all Enterprise Manager plug-ins. This region allows you to search, view, manage, and resolve incidents and problems impacting your environment.

- A **Dgraph Request Search** region allows you to search all Dgraph queries for a range of dates and identify any queries that took longer than the number of milliseconds you specify. You can also export your search results to a file in Excel.

For example:



[redacted].oracle.com_Dgraph

[redacted]cn.oracle.com

Oracle Big Data Discovery Dgraph ▾ | Start Up... | Shut Down...

Page Refreshed Dec 16, 2015 5:47:05 AM CST

Summary

Dgraph

Status	↑
Up Since	Dec 14, 2015 5:50:52 PM
Availability (%)	100
Version	1.1.1.983843
Cluster Identifier	cluster1208
Leader Node	FALSE
Host	bej301741.cn.oracle.com
Coordinator Host	bej301515.cn.oracle.com:2181
Web Service Port	7010
Bulk Load Port	7019
Process Id	3367
Installation Directory	/localdisk/Oracle/Middleware/BDD-1.1.1.13.11/dgraph
Request Log File Path	/localdisk/Oracle/Middleware/BDD-1.1.1.13.11/dgraph/bin/dgraph.reqlog
Out Log File Path	/localdisk/Oracle/Middleware/BDD-1.1.1.13.11/logs/dgraph.out

Dgraph Agent

Status	↑
Port	7102
Process Id	3922
Out Log File Path	/localdisk/Oracle/Middleware/BDD-1.1.1.13.11/logs/dgraphHDFSAgent.out

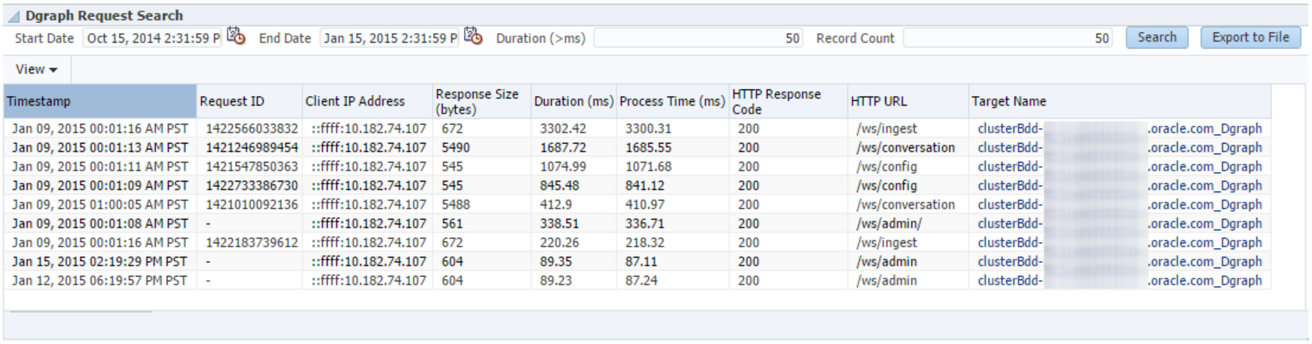
Resource Utilization

▸ **Incidents and Problems**

▸ **Dgraph Request Search**

Searching the Dgraph queries

Clicking the **Dgraph Request Search** region lets you search all Dgraph queries for a range of dates and identify any queries that took longer than the number of seconds you specify. For example:



Dgraph Request Search

Start Date Oct 15, 2014 2:31:59 P End Date Jan 15, 2015 2:31:59 P Duration (>ms) 50 Record Count 50 [Search] [Export to File]

View ▾

Timestamp	Request ID	Client IP Address	Response Size (bytes)	Duration (ms)	Process Time (ms)	HTTP Response Code	HTTP URL	Target Name
Jan 09, 2015 00:01:16 AM PST	1422566033832	::ffff:10.182.74.107	672	3302.42	3300.31	200	/ws/ingest	clusterBdd-[redacted].oracle.com_Dgraph
Jan 09, 2015 00:01:13 AM PST	1421246989454	::ffff:10.182.74.107	5490	1687.72	1685.55	200	/ws/conversation	clusterBdd-[redacted].oracle.com_Dgraph
Jan 09, 2015 00:01:11 AM PST	1421547850363	::ffff:10.182.74.107	545	1074.99	1071.68	200	/ws/config	clusterBdd-[redacted].oracle.com_Dgraph
Jan 09, 2015 00:01:09 AM PST	1422733386730	::ffff:10.182.74.107	545	845.48	841.12	200	/ws/config	clusterBdd-[redacted].oracle.com_Dgraph
Jan 09, 2015 01:00:05 AM PST	1421010092136	::ffff:10.182.74.107	5488	412.9	410.97	200	/ws/conversation	clusterBdd-[redacted].oracle.com_Dgraph
Jan 09, 2015 00:01:08 AM PST	-	::ffff:10.182.74.107	561	338.51	336.71	200	/ws/admin/	clusterBdd-[redacted].oracle.com_Dgraph
Jan 09, 2015 00:01:16 AM PST	1422183739612	::ffff:10.182.74.107	672	220.26	218.32	200	/ws/ingest	clusterBdd-[redacted].oracle.com_Dgraph
Jan 15, 2015 02:19:29 PM PST	-	::ffff:10.182.74.107	604	89.35	87.11	200	/ws/admin	clusterBdd-[redacted].oracle.com_Dgraph
Jan 12, 2015 06:19:57 PM PST	-	::ffff:10.182.74.107	604	89.23	87.24	200	/ws/admin	clusterBdd-[redacted].oracle.com_Dgraph

Exporting Dgraph target page search results to a file

You can export the results of your search to a file in Microsoft Office Excel.

To export to file, in the Dgraph target home page, enter the values for search and click **Search**, then click **Export to File**.

About Enterprise Manager

Dgraph HDFS Agent statistics

Dgraph HDFS Agent statistics

The Dgraph target contains a region for information about the Dgraph HDFS Agent. You can search export activities, search ingest activities, and export searched results.

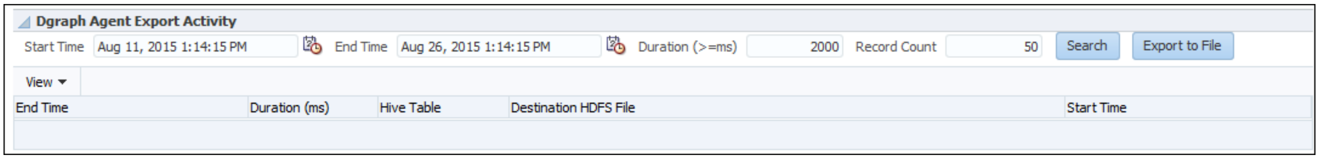
In addition to the basic set of metrics for Dgraph Agent on the Dgraph target home, the Dgraph target also provides a command to display the activities of the agent

Clicking **Oracle Big Data Discovery Dgraph>Dgraph Agent Activity** displays two tables:

- **Dgraph Agent Export Activity** table
- **Dgraph Agent Ingest Activity** table

Export activities

The **Dgraph Agent Export Activity** table shows information about the start and end time of each operation for exporting to HDFS, as well as the destination, duration and record count of the export. For example:



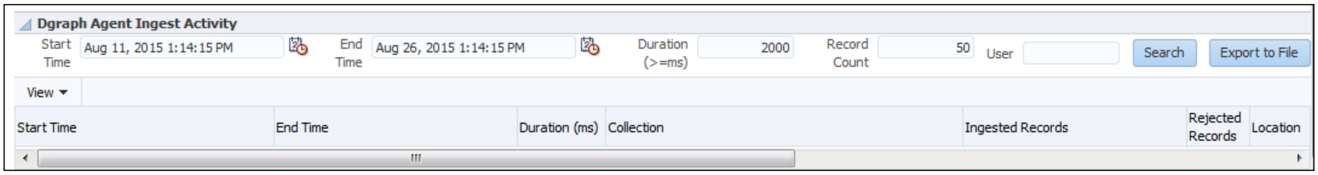
Dgraph Agent Export Activity

Start Time: Aug 11, 2015 1:14:15 PM
End Time: Aug 26, 2015 1:14:15 PM
Duration (>=ms): 2000
Record Count: 50

End Time	Duration (ms)	Hive Table	Destination HDFS File	Start Time

Ingest activities

The **Dgraph Agent Ingest Activity** table shows information about the start and end time of each ingest operation. For example:



Dgraph Agent Ingest Activity

Start Time: Aug 11, 2015 1:14:15 PM
End Time: Aug 26, 2015 1:14:15 PM
Duration (>=ms): 2000
Record Count: 50
User:

Start Time	End Time	Duration (ms)	Collection	Ingested Records	Rejected Records	Location

You can use a **Search** feature to search recent Ingest activities that have been run by the Dgraph HDFS Agent. For example, you can search by date range, to list all the ingest activities during that time period.

Aggregated statistics for the cluster

At the cluster home page, you can view aggregated HDFS activities statistics.

Clicking **Oracle Big Data Discovery Cluster>Dgraph Agent Activity** displays a two-tabbed table that provides the same type of export and import information pertaining to the Dgraph

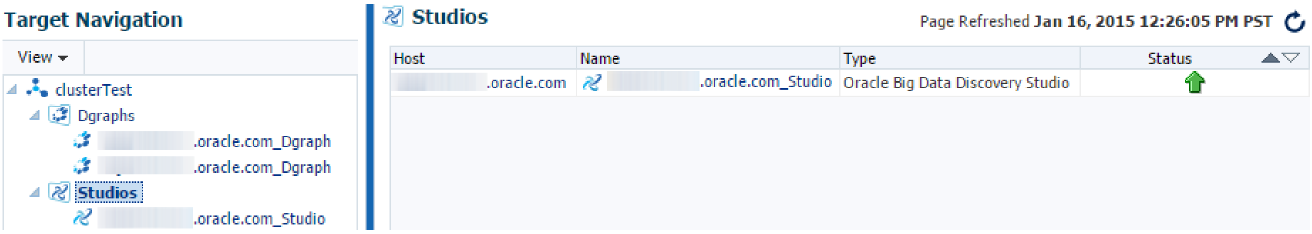
Exporting searched results

In all the above tables. you can export the results of your search to a file in Microsoft Office Excel, by using **Export to File**.

The Studio target

A Studio target represents a single Studio node running on a host. The **Studios** folder page provides a table that lists all Studio nodes and each node's status.

For example:



Target Navigation

View ▾

- clusterTest
  - Dgraphs
    - [redacted].oracle.com_Dgraph
    - [redacted].oracle.com_Dgraph
  - Studios
    - [redacted].oracle.com_Studio

Studios

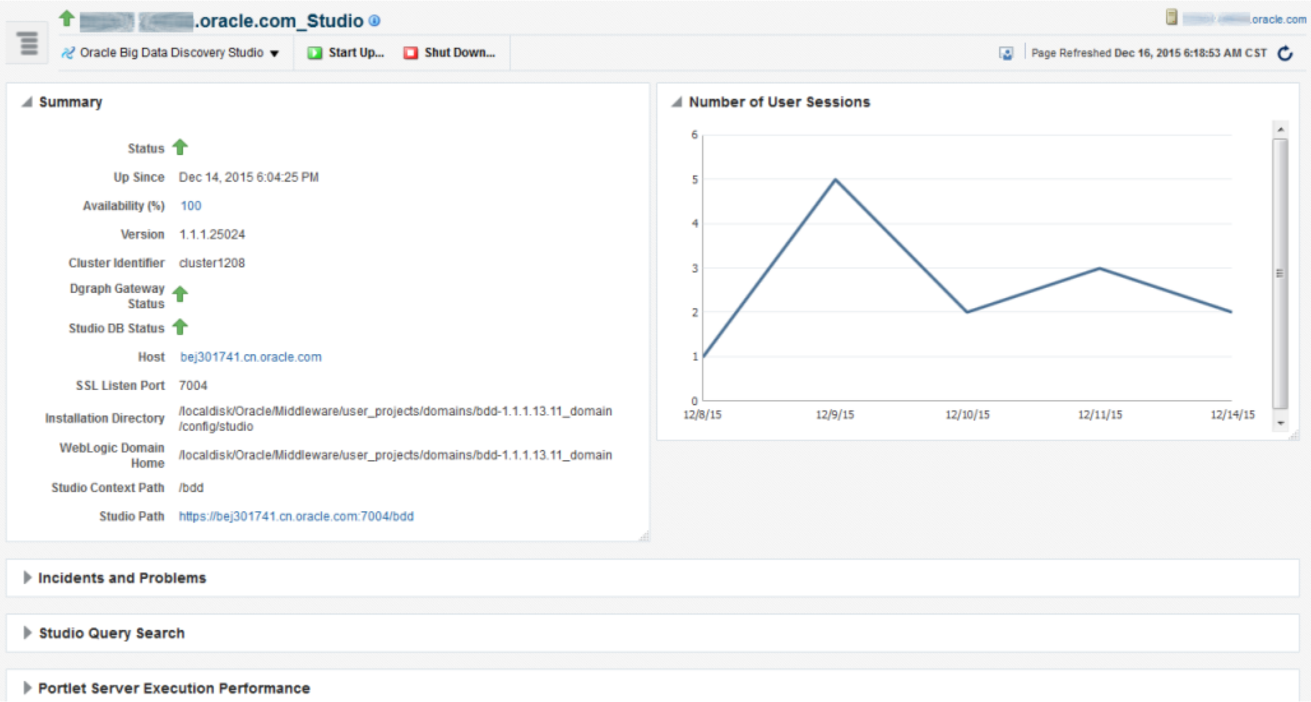
Page Refreshed **Jan 16, 2015 12:26:05 PM PST**

Host	Name	Type	Status
[redacted].oracle.com	[redacted].oracle.com_Studio	Oracle Big Data Discovery Studio	⇧

Clicking a Studio target in the table displays that Studio's home page and provides the following regions that describe the Studio node:

- A **Summary** region lists the basics about target status, availability, installation paths, server status, and so on.
- A **Number of User Sessions** region is about the total number of unique user sessions per day for the last 30 days.
- An **Incidents and Problems** region is standard to all Enterprise Manager plug-ins. This region allows you to search, view, manage, and resolve incidents and problems impacting your environment.
- A **Studio Query Search** region lets you search all Studio queries for a range of dates and identify any queries that took longer than the number of specified milliseconds. You can also export your search results to a file in Excel.
- A **Portlet Server Execution Performance** region displays running time for each portlet in Studio. (A portlet is another name for a component on a page in Studio).

For example:



Status	↑
Up Since	Dec 14, 2015 6:04:25 PM
Availability (%)	100
Version	1.1.1.25024
Cluster Identifier	cluster1208
Dgraph Gateway Status	↑
Studio DB Status	↑
Host	bej301741.cn.oracle.com
SSL Listen Port	7004
Installation Directory	/localdisk/Oracle/Middleware/user_projects/domains/bdd-1.1.1.13.11_domain /config/studio
WebLogic Domain Home	/localdisk/Oracle/Middleware/user_projects/domains/bdd-1.1.1.13.11_domain
Studio Context Path	/bdd
Studio Path	https://bej301741.cn.oracle.com:7004/bdd

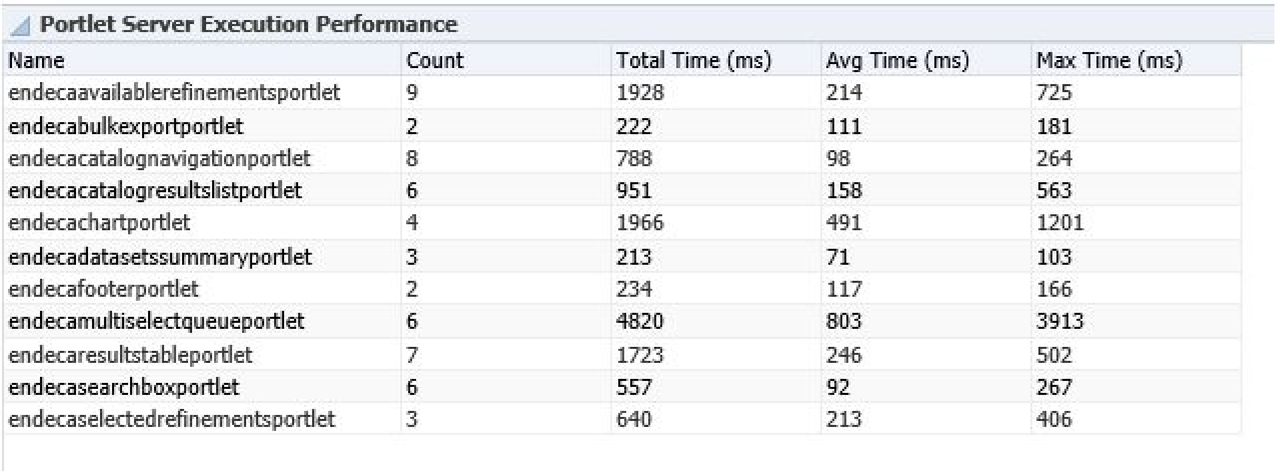
Tracking performance of Studio components

Clicking the **Portlet Server Execution Performance** region displays running times for each portlet (component) in Studio.

For each portlet in Studio, the table tracks:

- Total number of queries or runs
- Total running time
- Average running time
- Maximum running time

For example:

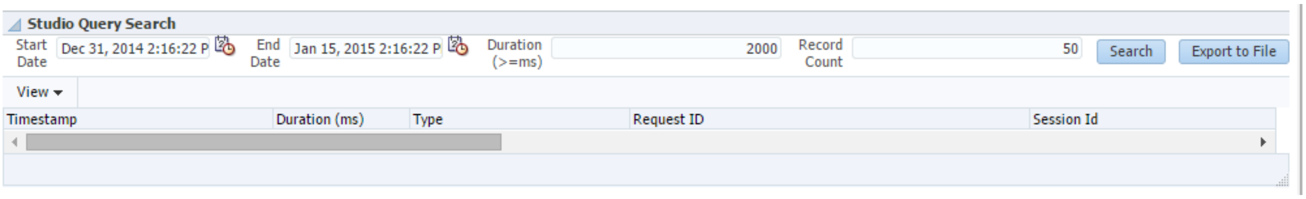


Portlet Server Execution Performance

Name	Count	Total Time (ms)	Avg Time (ms)	Max Time (ms)
endecaavailablerefinementsportlet	9	1928	214	725
endecabulkexportportlet	2	222	111	181
endecacatalognavigationportlet	8	788	98	264
endecacatalogresultslistportlet	6	951	158	563
endecachartportlet	4	1966	491	1201
endecadatasetssummaryportlet	3	213	71	103
endecafooterportlet	2	234	117	166
endecamultiselectqueueportlet	6	4820	803	3913
endecaresultstableportlet	7	1723	246	502
endecasearchboxportlet	6	557	92	267
endecaselectedrefinementsportlet	3	640	213	406

Searching Studio queries

Clicking the **Studio Query Search** region lets you search all Studio queries for a range of dates and identify any queries that took longer than the number of specified milliseconds. For example:



Studio Query Search

Start Date: Dec 31, 2014 2:16:22 P

End Date: Jan 15, 2015 2:16:22 P

Duration (>=ms): 2000

Record Count: 50

Timestamp	Duration (ms)	Type	Request ID	Session Id

Exporting search results from the Studio target page to a file

You can export the results of your search to a file in Microsoft Office Excel.

To export to file, in the Studio target home page, enter the values for search and click **Search**, then click **Export to File**.

Roles and privileges for BDD targets

Only users with sufficient roles and permissions can perform operations, such as Start and Stop, on the targets for Big Data Discovery.

Enterprise Manager Cloud Control lets you use various roles and privileges, when working with targets that you are managing. For information on how to create, grant and use roles and privileges, see the *Oracle Enterprise Manager Cloud Control Security Guide*.

To make sure the BDD plug-in lets you perform the tasks you need for the Big Data Discovery, you, as the administrator of Enterprise Manager, must:

- Have a super user privilege, or

- Have both of these privileges:
  - "Operator" privilege for the host target on nodes where the Big Data Discovery product is installed. (This is because host commands are used by Enterprise Manager to operate on BDD targets.)
  - "Create job" privilege for the host targets on nodes where the Dgraph is installed. (This is because Enterprise Manager jobs are used to start and stop the Dgraph.)

Security credentials for BDD targets

The plug-in is configured in a way that lets it discover and work with targets that are deployed in either secure mode (SSL), or non-secure mode. The plug-in also relies on preferred credentials for two of the Big Data Discovery targets — Dgraph and Studio, and sets them automatically at discovery time to the host credential. Many features of the plug-in depend on these credentials being set.

About security credentials in the plug-in

By default, the preferred credentials in the plug-in are set for two targets — Dgraph and Studio, and they are set with a host credential. It is important to have these credentials set so that the plug-in behaves correctly.

You can always set your own credentials. To do so, go to **Setup>Security>Preferred Credentials**, and on the **Security** page, select the target, and click **Manage Preferred Credentials**. For complete details about providing preferred credentials during the target discovery process, see the topic "Setting preferred credentials for a target", *http://docs.oracle.com/cd/E24628_01/timesten.121/e28645/install.htm#TTEMP604*, in the *Enterprise Manager System Monitoring Plug-in for Oracle TimesTen In-Memory Database User's Guide*.

If the preferred credentials are not set

The following table list instances where the plug-in does not behave as expected, if the default credentials are not used or if you have not set your own credentials:

Target	Feature	Behavior if credential is not set
Dgraph	On the Cluster target home page, indicate which Dgraph node is the leader node.	The leader node is not identified.
Dgraph	On the Dgraph target home page, indicate the leader node and the status of the HDFS Agent.	The leader node is not identified, and the status of HDFS Agent is not listed (empty).
Dgraph	Dgraph request search.	An error is issued for the Dgraph target: `Default preferred credentials are not set.`
Dgraph	Dgraph administration operation	For each operation, an error is issued for the Dgraph target: `Default preferred credentials are not set.`
Dgraph	Start up and shut down the Dgraph	An error is issued for the Dgraph target: `Default preferred credentials are not set.`

Target	Feature	Behavior if credential is not set
Dgraph	View log messages for the Dgraph target.	The function is disabled without any warning.
Studio	On the Studio target home page, monitor the Dgraph Gateway status	The status of the Dgraph Gateway is empty.
Dgraph, Studio	View log messages for Cluster target.	The function is disabled without any warning.
Studio	View log messages for Studio target.	The function is disabled without any warning.

Connecting to Studio over a secure port

When Studio's outward-facing port is configured securely (with reverse proxy), the plug-in uses the same port to also connect to Studio in secure mode.

To establish a secure connection, you need to provide the plug-in with the SSL Keystore/Trust Keystore information.

Before installing Big Data Discovery, you can configure options in `bdd.conf` for a secure installation of Studio within WebLogic Server. For information, see the *Installation and Deployment Guide*. That guide also describes how you can set up Studio to use a reverse proxy, after the installation.

If Studio is not deployed securely, the Guided Discovery process of the plug-in determines this fact and does not let you provide SSL information during target discovery process in the plug-in. In such cases, to make sure that the plug-in uses the secure port for Studio, you should manually provide information about SSL for a Studio target in the plug-in by setting the monitoring credentials. For information on setting monitoring credentials within Enterprise Manager plug-in, see the *Enterprise Manager Cloud Control Security Guide*.

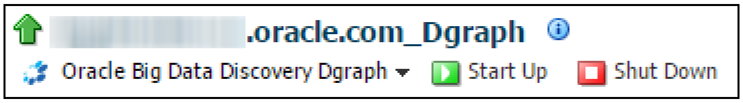
When Studio is configured with SSL, the following requirements apply for monitoring Studio with Enterprise Manager plug-in:

- If Studio is configured in one-way SSL mode, provide only the Trust Keystore to the plug-in.
- If Studio is configured in two-way SSL mode, provide both the Keystore and Trust Keystore to the plug-in. In addition, these two requirements apply to the two-way SSL mode:
  - The password of the private key must be the same as the password of the Keystore.
  - If the Keystore contains more than one key pair, the key pair you want to use must be listed first among all the keys in the Keystore.

Starting and stopping the Dgraph and Studio using Enterprise Manager

You can start up and shut down the Dgraph and Studio from their home pages.

The Dgraph home page in Enterprise Manager has **Start Up** and **Shut Down** commands to manually start and stop the Dgraph and the Dgraph HDFS Agent as necessary:


[redacted].oracle.com_Dgraph

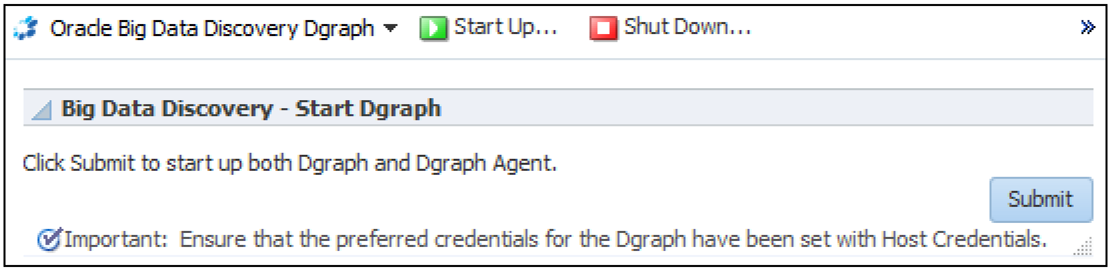
The Dgraph process manages the Dgraph HDFS Agent process, so starting or stopping the Dgraph also starts or stops the Dgraph HDFS Agent.

The Studio home page has a similar pair of **Start Up** and **Shut Down** commands. These commands stop and start both Studio and Dgraph Gateway.

The start and stop commands run as jobs in Enterprise Manager that you can monitor by clicking the job ID. Generally, the commands run quickly and return a Succeeded or Failed status.

Start Up command

Clicking **Start Up** displays the Start Dgraph dialog (from the Dgraph home page) or the Start Studio dialog (from the Studio home page). The Dgraph version looks like this:


Big Data Discovery - Start Dgraph

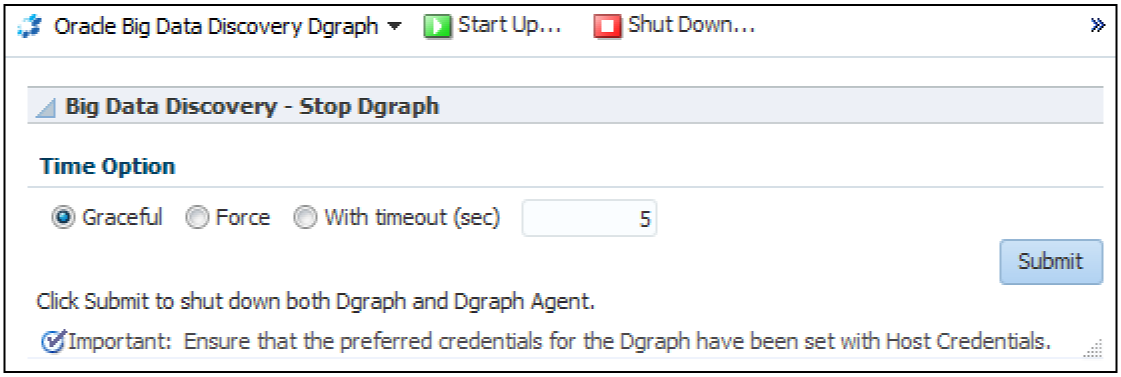
Click Submit to start up both Dgraph and Dgraph Agent.

Important: Ensure that the preferred credentials for the Dgraph have been set with Host Credentials.

Click **Submit** to start the components.

Shut Down command

Clicking **Shut Down** displays the Stop Dgraph dialog (from the Dgraph home page) or the Stop Studio dialog (from the Studio home page). The Dgraph version looks like this::


Big Data Discovery - Stop Dgraph

Time Option

● Graceful ○ Force ○ With timeout (sec) 5

Click Submit to shut down both Dgraph and Dgraph Agent.

Important: Ensure that the preferred credentials for the Dgraph have been set with Host Credentials.

Time Option provides three choices as to how the component should be stopped:

- **Graceful** will shut down the Dgraph or Studio gracefully.
- **Force** will force the Dgraph or Studio to shut down immediately.
- **With timeout** will wait the specified number of seconds for the Dgraph or Studio to shut down gracefully. If the component has not shut down by that time, then it will be shut down immediately.

After choosing a Time Option, click **Submit** to stop the Dgraph and the Dgraph HDFS Agent or Studio and Dgraph Gateway.

Logging for BDD targets

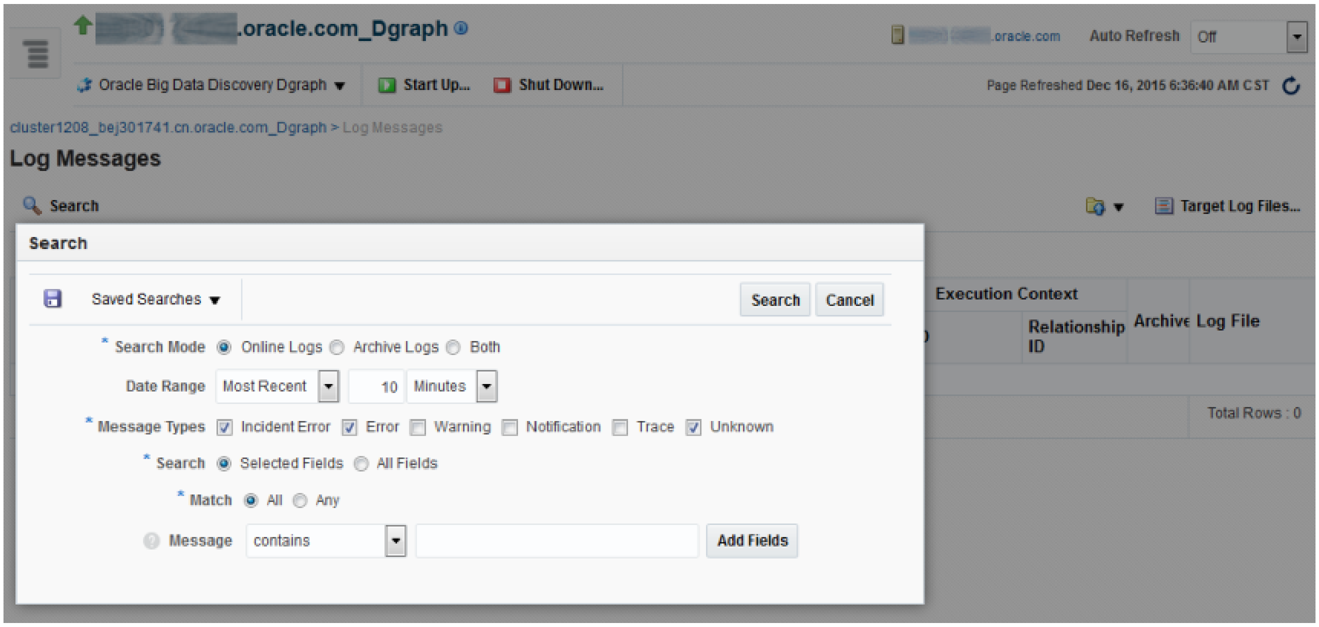
Enterprise Manager provides standard logging controls to list, view, search, and download the log files for the Big Data Discovery targets. You can also group logging messages from each target by host, host IP address, or other parameters.

Viewing log messages

For the Cluster target, to view and search logs for all targets (Cluster, Dgraph, and Studio), select **<target name>>Logs>View Log Message** in Enterprise Manager.

For the Dgraph or Studio target, to view and search logs, use the same command for each target.

The **Log Messages** page displays the standard logging controls for any target in Enterprise Manager, including a Search pane. For example:



[redacted].oracle.com_Dgraph

[redacted].oracle.com Auto Refresh Off

Oracle Big Data Discovery Dgraph ▾ Start Up... Shut Down...

Page Refreshed Dec 16, 2015 6:36:40 AM CST

cluster1208_bej301741.cn.oracle.com_Dgraph > Log Messages

Log Messages

Search

Target Log Files...

Search

Saved Searches ▾ Search Cancel

* Search Mode ● Online Logs ○ Archive Logs ○ Both

Date Range Most Recent 10 Minutes

* Message Types ☑ Incident Error ☑ Error □ Warning □ Notification □ Trace ☑ Unknown

* Search ● Selected Fields ○ All Fields

* Match ● All ○ Any

Message contains Add Fields

Execution Context

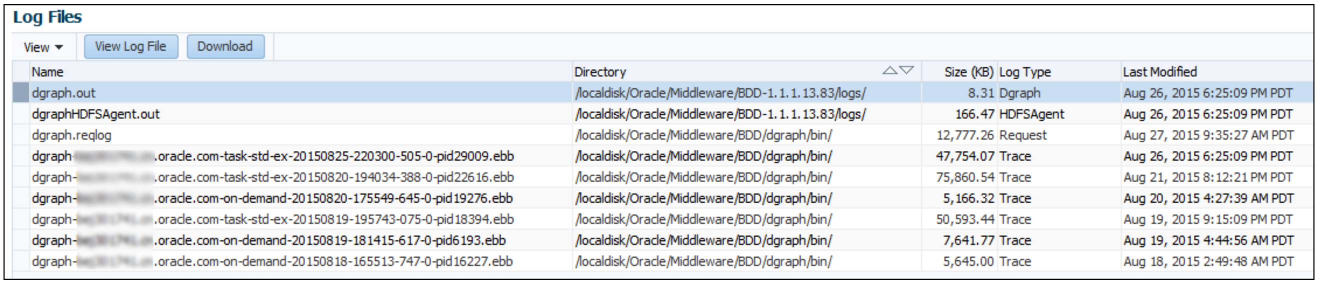
Relationship ID

Archive

Log File

Total Rows : 0

From this page, you can click **Target Log Files** to view a list of the logs for the target and download the logs if desired. For example:

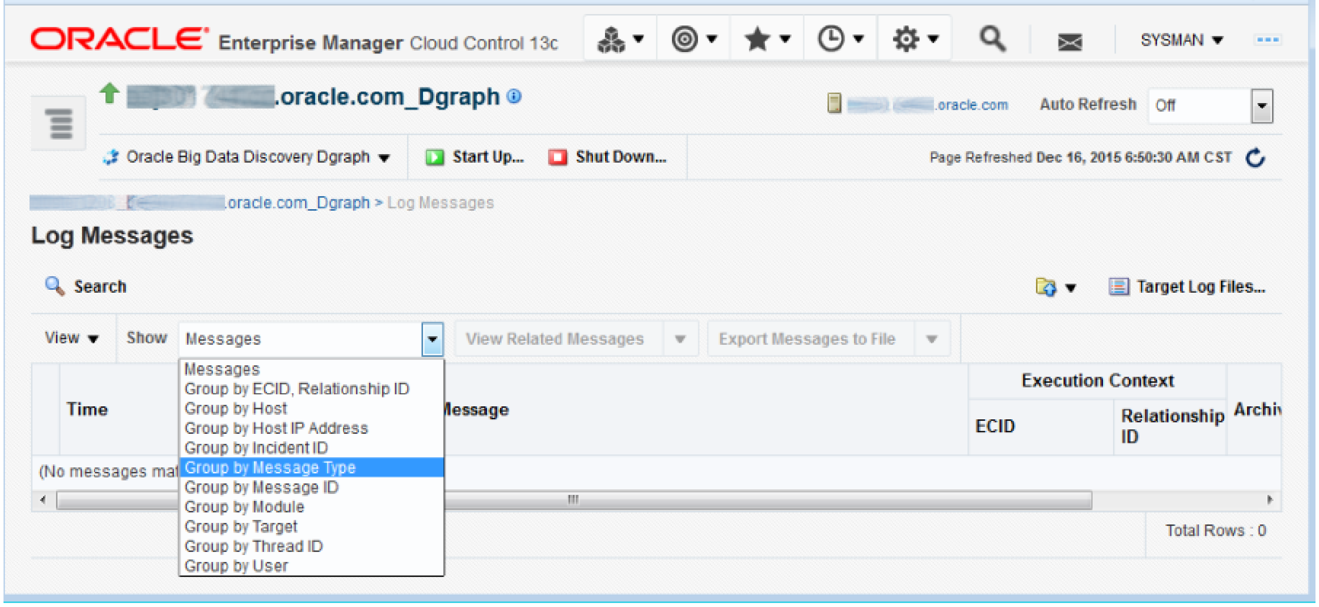


Name	Directory	Size (KB)	Log Type	Last Modified
dgraph.out	/localdisk/Oracle/Middleware/BDD-1.1.1.13.83/logs/	8.31	Dgraph	Aug 26, 2015 6:25:09 PM PDT
dgraphHDFSAgent.out	/localdisk/Oracle/Middleware/BDD-1.1.1.13.83/logs/	166.47	HDFSAgent	Aug 26, 2015 6:25:09 PM PDT
dgraph.reqlog	/localdisk/Oracle/Middleware/BDD/dgraph/bin/	12,777.26	Request	Aug 27, 2015 9:35:27 AM PDT
dgraph-[redacted].oracle.com-task-std-ex-20150825-220300-505-0-pid29009.ebb	/localdisk/Oracle/Middleware/BDD/dgraph/bin/	47,754.07	Trace	Aug 26, 2015 6:25:09 PM PDT
dgraph-[redacted].oracle.com-task-std-ex-20150820-194034-388-0-pid22616.ebb	/localdisk/Oracle/Middleware/BDD/dgraph/bin/	75,860.54	Trace	Aug 21, 2015 8:12:21 PM PDT
dgraph-[redacted].oracle.com-on-demand-20150820-175549-645-0-pid19276.ebb	/localdisk/Oracle/Middleware/BDD/dgraph/bin/	5,166.32	Trace	Aug 20, 2015 4:27:39 AM PDT
dgraph-[redacted].oracle.com-task-std-ex-20150819-195743-075-0-pid18394.ebb	/localdisk/Oracle/Middleware/BDD/dgraph/bin/	50,593.44	Trace	Aug 19, 2015 9:15:09 PM PDT
dgraph-[redacted].oracle.com-on-demand-20150819-181415-617-0-pid6193.ebb	/localdisk/Oracle/Middleware/BDD/dgraph/bin/	7,641.77	Trace	Aug 19, 2015 4:44:56 AM PDT
dgraph-[redacted].oracle.com-on-demand-20150818-165513-747-0-pid16227.ebb	/localdisk/Oracle/Middleware/BDD/dgraph/bin/	5,645.00	Trace	Aug 18, 2015 2:49:48 AM PDT

Grouping messages in the log viewer

For all Big Data Discovery targets, you can group log messages.

To group messages by a parameter (such as by message type), in the **Log Messages** page, select a parameter from the menu. For example:



Configuring log levels for Dgraph targets

The Dgraph has log levels for its log subsystems, which can be changed dynamically.

The logging variables apply to a single Dgraph and not to all Dgraph nodes in a cluster.

For each Dgraph target, you can set the following log levels for the Dgraph log subsystems:

Log subsystem	Description
`background_merging`	Messages about index maintenance activity.

Log subsystem	Description
bulk_ingest	Messages generated by Bulk Load ingest operations
cluster	Messages about ZooKeeper-related cluster operations.
datalayer	Messages about index file usage.
dgraph	Messages related to Dgraph general operations.
eql	Messages generated from the Endeca Query Language engine.
eve	Messages generated from the EVE (Endeca Virtual Engine) query evaluator.
http	Messages about Dgraph HTTP communication operations.
lexer	Messages emitted by the OLT (Oracle Language Technology) subsystem.
splitting	Messages resulting from EVE (Endeca Virtual Engine) splitting tasks.
ssl	Messages generated by the SSL subsystem.
task_scheduler	Messages related to the Dgraph task scheduler.
text_search_rel_rank	Messages related to Relevance Ranking operations during text searches.
text_search_spelling	Messages related to spelling correction operations during text searches.
update	Messages related to updates.
workload_manager	Messages generated from the Dgraph Workload Manager.
ws_request	Messages related to request exchanges between Web services.
xq_web_service	Messages generated from the XQuery-based Web services.

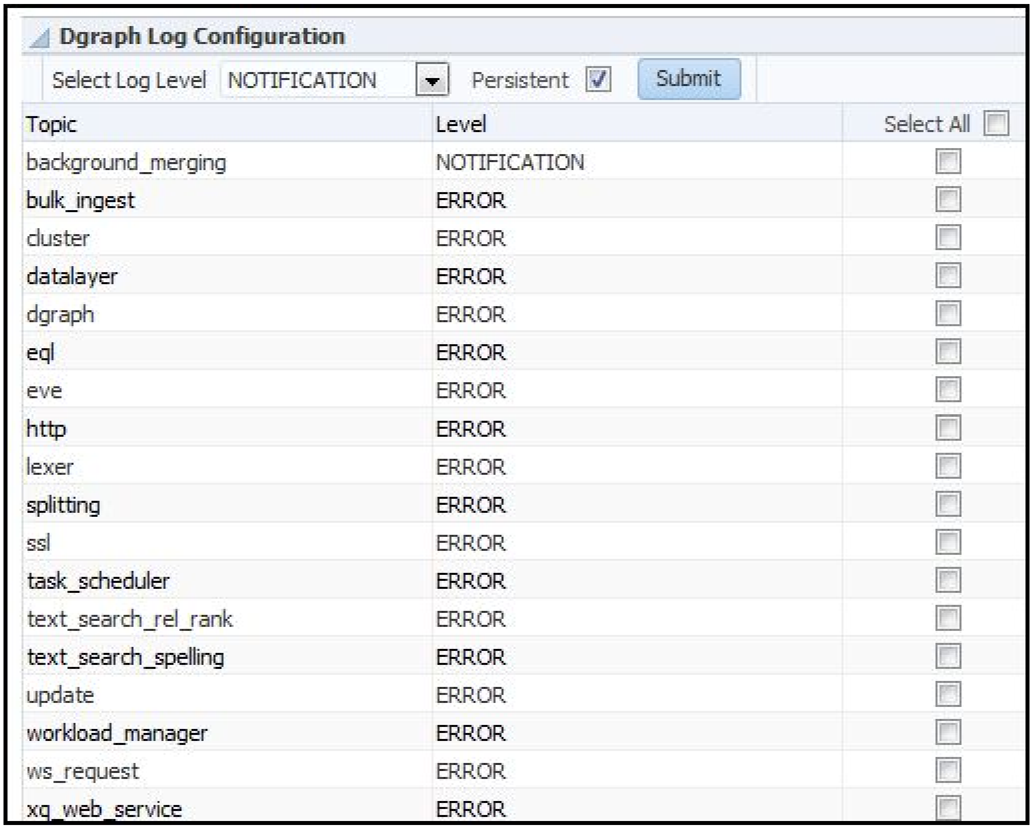
Each subsystem can be set to one of these log levels:

- INCIDENT_ERROR
- ERROR
- WARNING
- NOTIFICATION
- TRACE

To change a log level for a Dgraph target:

1. Log in to Enterprise Manager Cloud Control.
2. Select a Dgraph target.
3. From the Dgraph target menu, select **Administration>Log Configuration**.

   The Dgraph Log Configuration dialog displays with the currently-set log level for each subsystem:



Dgraph Log Configuration

Select Log Level: NOTIFICATION ▾ Persistent ☑ Submit

Topic	Level	Select All □
background_merging	NOTIFICATION	□
bulk_ingest	ERROR	□
cluster	ERROR	□
datalayer	ERROR	□
dgraph	ERROR	□
eql	ERROR	□
eve	ERROR	□
http	ERROR	□
lexer	ERROR	□
splitting	ERROR	□
ssl	ERROR	□
task_scheduler	ERROR	□
text_search_rel_rank	ERROR	□
text_search_spelling	ERROR	□
update	ERROR	□
workload_manager	ERROR	□
ws_request	ERROR	□
xq_web_service	ERROR	□

4. To change the log level for one or more subsystems:

   (a) Mark the checkbox next to the subsystem to be changed.

   (b) In the **Select Log Level** drop-down, select one of the log levels. This log level will apply to all the selected subsystems.

5. Either check or uncheck the **Persistent** checkbox.
   - If checked, the new settings are persisted by being written to the `DGRAPH_LOG_LEVEL` property in `bdd.conf`. This means that the next Dgraph re-start will use the changed the log levels.
   - If unchecked, the changes will not persist into the next Dgraph re-start.

6. Click **Submit**.

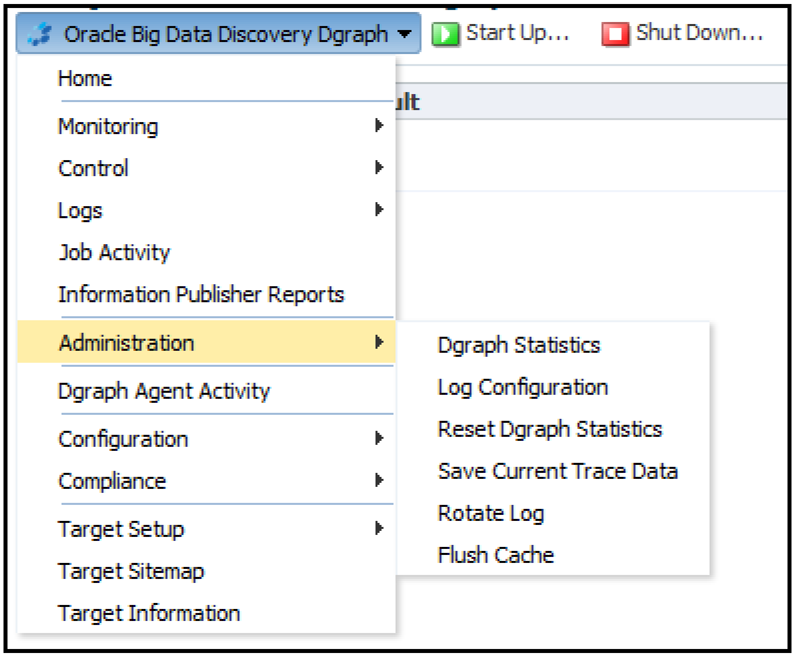
The changes take effect immediately.

Dgraph Administration Operations in Enterprise Manager

This section describes the **Administration** menu of options for the Dgraph target in Enterprise Manager.

The **Administration** menu of the plug-in contains operations for viewing and resetting the statistics page for the Dgraph, as well as operations for saving and downloading Dgraph Tracing Utility data, flushing the cache, and rotating the logs.

You can choose any operation by clicking the Dgraph target, and selecting **Administration**:



Viewing Dgraph statistics

Resetting Dgraph statistics

Flushing the Dgraph cache

Rotating the Dgraph request log

Saving trace data for the Dgraph

Downloading Dgraph trace files

Viewing Dgraph statistics

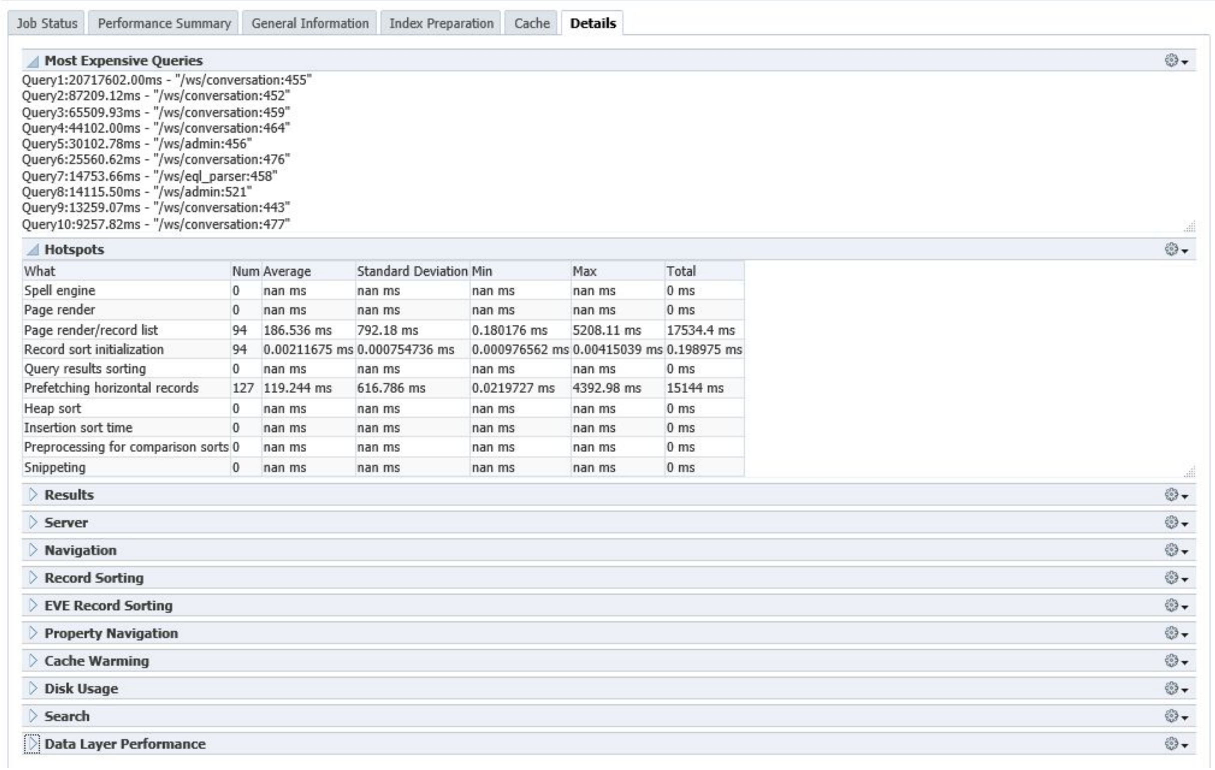
You can view Dgraph statistics for any Dgraph managed by Enterprise Manger Cloud Control.

To view Dgraph statistics:

1. Log in to Enterprise Manager Cloud Control.
2. Select a Dgraph target.

3. From the Dgraph target menu, select **Administration>Dgraph Statistics**.

   A page with six tabs displays. For example:



```
Query1:20717602.00ms - "/ws/conversation:455"
Query2:87209.12ms - "/ws/conversation:452"
Query3:65509.93ms - "/ws/conversation:459"
Query4:44102.00ms - "/ws/conversation:464"
Query5:30102.78ms - "/ws/admin:456"
Query6:25560.62ms - "/ws/conversation:476"
Query7:14753.66ms - "/ws/eql_parser:458"
Query8:14115.50ms - "/ws/admin:521"
Query9:13259.07ms - "/ws/conversation:443"
Query10:9257.82ms - "/ws/conversation:477"
```

What	Num	Average	Standard Deviation	Min	Max	Total
Spell engine	0	nan ms	nan ms	nan ms	nan ms	0 ms
Page render	0	nan ms	nan ms	nan ms	nan ms	0 ms
Page render/record list	94	186.536 ms	792.18 ms	0.180176 ms	5208.11 ms	17534.4 ms
Record sort initialization	94	0.00211675 ms	0.000754736 ms	0.000976562 ms	0.00415039 ms	0.198975 ms
Query results sorting	0	nan ms	nan ms	nan ms	nan ms	0 ms
Prefetching horizontal records	127	119.244 ms	616.786 ms	0.0219727 ms	4392.98 ms	15144 ms
Heap sort	0	nan ms	nan ms	nan ms	nan ms	0 ms
Insertion sort time	0	nan ms	nan ms	nan ms	nan ms	0 ms
Preprocessing for comparison sorts	0	nan ms	nan ms	nan ms	nan ms	0 ms
Snippeting	0	nan ms	nan ms	nan ms	nan ms	0 ms

4. Click the tab and then region you want to examine. The statistics is intended for Oracle Support.

Resetting Dgraph statistics

You can reset Dgraph statistics for any Dgraph managed by Enterprise Manger Cloud Control plug-in.

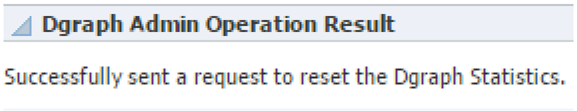
This option runs against a single Dgraph and resets all statistics displayed on the Dgraph Statistics pages. This option is useful if you want to view the statistics information for a single request: you reset statistics, issue a query, and inspect the updated statistics.

To reset the Dgraph statistics:

1. Log in to Enterprise Manager Cloud Control.
2. Select a Dgraph target.
3. From the Dgraph target menu, select **Administration>Reset Dgraph Statistics**.

   A confirmation page displays.

   After you confirm, the following status displays:



Successfully sent a request to reset the Dgraph Statistics.

Flushing the Dgraph cache

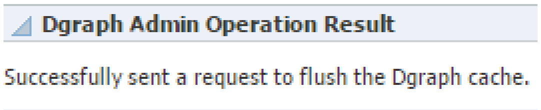
You can flush the cache of any Dgraph managed by Enterprise Manager.

This option runs against a single Dgraph. This option is useful if you are debugging query problems: you can approximate cold-start or post-update performance by clearing the Dgraph cache prior to running a request. To flush the Dgraph's cache:

1. Log in to Enterprise Manager Cloud Control.
2. Select a Dgraph target.
3. From the Dgraph target menu, select **Administration>Flush Cache**.

   A confirmation page displays.

   After you confirm, the following status displays:

   Successfully sent a request to flush the Dgraph cache.

Rotating the Dgraph request log

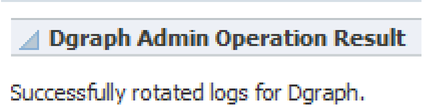
You can roll the Dgraph request log over to a new file for any Dgraph managed by Enterprise Manger Cloud Control plug-in. This option runs against a single Dgraph.

To roll the Dgraph's request log:

1. Log in to Enterprise Manager Cloud Control.
2. Select a Dgraph target.
3. From the Dgraph target menu, select **Administration>Rotate Log**.

   A confirmation page displays.

   After you confirm, the following status displays:

   Successfully rotated logs for Dgraph.

Saving trace data for the Dgraph

You can save trace-level information to a file for any Dgraph managed by Enterprise Manger Cloud Control plug-in.

The Dgraph Tracing Utility runs automatically while the Dgraph is running. It stores the Dgraph target trace data it collects in trace files (`.ebb`).

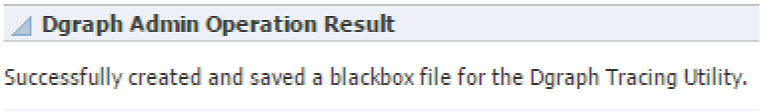
Saving the trace data for the Dgraph is useful when working with Oracle Support to debug and diagnose issues. This option runs against a single Dgraph. The output file is referred to as a blackbox file and is named `on-demand.pid.ebb`.

To save the Dgraph's trace data:

1. Log in to Enterprise Manager Cloud Control.
2. Select a Dgraph target.
3. From the Dgraph target menu, select **Administration>Save Current Trace Data**.

   A confirmation page displays.

   After you confirm, the following status displays:

   Dgraph Admin Operation Result

   Successfully created and saved a blackbox file for the Dgraph Tracing Utility.

Unlike other types of log files, you cannot view the Tracing Utility files in Enterprise Manager. Therefore, after you have saved the current trace data for the Dgraph, you may need to download it, to share with Oracle Support.

Downloading Dgraph trace files

The Dgraph Tracing Utility runs automatically while the Dgraph is running.

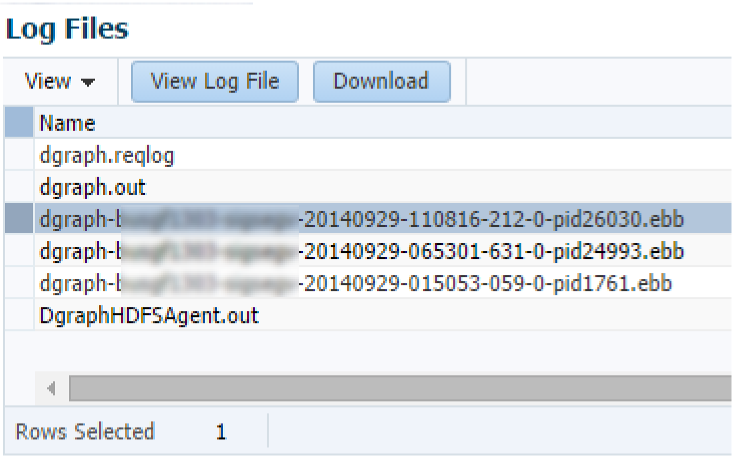
The Tracing Utility stores the Dgraph target trace data it collects in trace `*.ebb` files. The files are intended for use by Oracle Support. Unlike other types of log files, you cannot view the Tracing Utility files in Enterprise Manager. You can, however, download them from the **Log Messages** page.

To download a Tracing Utility file for the Dgraph target:

1. Log in to Enterprise Manager Cloud Control and select a Dgraph target.
2. From the Dgraph target menu, select **Logs>View Log Messages**.

   This opens the **Log Messages** page.
3. In the **Log Messages** page, click **Target Log Files...**.

   This opens the **Log Files** page, which lists the Dgraph target's log files. Tracing Utility files have the `.ebb` file extension and have a **Log Type** of **Trace**.

   Log Files

Name
dgraph.reqlog
dgraph.out
dgraph-[redacted]-20140929-110816-212-0-pid26030.ebb
dgraph-[redacted]-20140929-065301-631-0-pid24993.ebb
dgraph-[redacted]-20140929-015053-059-0-pid1761.ebb
DgraphHDFSAgent.out

   Rows Selected 1

4. Click on a Tracing Utility file to select it, then click **Download**.

   The selected Tracing Utility file is downloaded to your machine.

Part VII

Logging for Studio, Dgraph, and Dgraph Gateway

Chapter 20

Overview of BDD Logging

This topic provides a logging overview of the BDD components.

List of Big Data Discovery logs

Retrieving logs

List of Big Data Discovery logs

This topic provides a list of all the logs generated by a BDD deployment.

The list also includes a summary of where to find logs for each BDD component and tells you how to access logs.

List of BDD logs

Log	Purpose	Default Location
WebLogic Admin Server domain log	Provides a status of the WebLogic domain for the Big Data Discovery deployment. See *WebLogic Domain Log on page 179*.	`$BDD_DOMAIN/servers/AdminServer/logs/bdd_domain.log`
WebLogic Admin Server server log	Contains messages from the WebLogic Admin Server subsystems. For both server logs, see *WebLogic Server Log on page 179*.	`$BDD_DOMAIN/servers/AdminServer/logs/AdminServer.log`
WebLogic Managed Server server log	Contains messages from the WebLogic Managed Server subsystems and applications.	`$BDD_DOMAIN/servers/<serverName>/logs/<serverName>.log`
Dgraph Gateway application log	WebLogic log for the Dgraph Gateway application. See *Dgraph Gateway log entry format on page 180*	`$BDD_DOMAIN/servers/<serverName>/logs/<serverNamem>-diagnostic.log`
Dgraph stdout/stderr log	Contains Dgraph operational messages, including startup messages. See *Dgraph out log on page 173*.	`$BDD_HOME/logs/dgraph.out`
Dgraph request log	Contains entries for Dgraph requests. See *Downloading Dgraph trace files on page 156*.	`$BDD_HOME/dgraph/bin/dgraph.reqlog`

Log	Purpose	Default Location
Dgraph tracing ebb logs	Dgraph Tracing Utility files, which are especially useful for Dgraph crashes. See *Downloading Dgraph trace files on page 156*.	`$BDD_HOME/dgraph/bin/dgraph-<serverName>-*.ebb`
Dgraph HDFS Agent stdout/stderr log	Contains startup messages, as well as messages from operations performed by the Dgraph HDFS Agent (such as ingest operations). See the *Data Processing Guide*.	`$BDD_HOME/logs/dgraphHDFSAgent.out`
Studio application log in Log4j format	Studio application log (in Log4j format). For both Studio application logs, see *About the main Studio log file on page 166*.	`$BDD_DOMAIN/servers/<serverName>/logs/bdd-studio.log`
Studio application log in ODL format	Studio application log (in ODL format).	`$BDD_DOMAIN/servers/<serverName>/logs/bdd-studio-odl.log`
Studio metrics log in Log4j format	Studio metrics log (in Log4j format). For both Studio metrics logs, see *About the metrics log file on page 166*.	`$BDD_DOMAIN/servers/<serverName>/logs/bdd-studio-metrics.log`
Studio metrics log in ODL format	Studio metrics log (in ODL format).	`$BDD_DOMAIN/servers/<serverName>/logs/bdd-studio-metrics-odl.log`
Studio client log in Log4j format	Studio client log (in Log4j format). For both Studio client logs, see *About the Studio client log file on page 169*.	`$BDD_DOMAIN/servers/<serverName>/logs/bdd-studio-client.log`
Studio client log in ODL format	Studio client log (in ODL format).	`$BDD_DOMAIN/servers/<serverName>/logs/bdd-studio-client-odl.log`
Data Processing logs	Contains messages resulting from Data Processing workflows. See the *Data Processing Guide*.	`$BDD_HOME/logs/edp/edp_*.log`
CDH or HDP logs (YARN logs, Spark worker logs, ZooKeeper logs)	YARN logs from CDH and HDP processes that ran Data Processing workflows, as listed in the *Data Processing Guide*. See the Cloudera and Hortonworks documentation for information on the ZooKeeper logs.	Available from the Cloudera Manager and Ambari Web UIs for the component.

Where to find logging information for each component

This table lists how to find detailed logging information for each Big Data Discovery component:

Big Data Discovery Component name	Where to find logging information?
Studio	See *Studio Logging on page 162*.
Data Processing	Data Processing is a component of BDD that runs on CDH or HDP nodes in the BDD deployment. For Data Processing logs, see the *Data Processing Guide*.
Dgraph Gateway (and WebLogic Server logs)	See *Dgraph Gateway Logging on page 177*.
Dgraph	See *Dgraph Logging on page 171*.
Dgraph HDFS Agent	The Dgraph HDFS Agent is responsible for importing and exporting Dgraph data to HDFS. For HDFS Agent logs, see the *Data Processing Guide*.

Ways of accessing logs

You can access the logs for some components of Big Data Discovery through `bdd_admin.sh` and Enterprise Manager:

Method of accessing logs	Logging tasks
Logging options available in `bdd-admin.sh`	Use the `bdd-admin` script for these operations on BDD logs: • *get-logs* • *get-blackbox* • *get-log-levels on page 44* • *set-log-levels on page 45* • *rotate-logs on page 49*
Logging options available in Enterprise Manager for BDD	Use Enterprise Manager to access logs for Studio, Dgraph and the BDD cluster, to configure verbose logging for the Dgraph, and to roll the Dgraph request log. • *Logging for BDD targets on page 149* • *Configuring log levels for Dgraph targets on page 150* • *Rotating the Dgraph request log on page 155*.

Retrieving logs

The `bdd-admin` script's `get-logs` command lets you retrieve all the BDD component logs, or a specified subsection of them.

Full usage information on the `get-logs` command is available in the topic *get-logs on page 47*.

This example shows how to retrieve the most recent Dgraph logs:

1. Change to the `$BDD_HOME/BDD_manager/bin` directory.
2. Use the `get-logs` command with the `-c dgraph` option:

   ```
   ./bdd-admin.sh get-logs -c dgraph /localdisk/logs/dgraph.zip
   ```

   In the example, the Dgraph logs are retrieved and zipped up in the `dgraph.zip` file.

When you unzip the `dgraph.zip` file, a `<hostname>_dgraph.zip` file should be extracted. When you unzip that file, you should see these Dgraph logs:

- `dgraph.out` (Dgraph out log)
- `dgraph.reqlog` (Dgraph request log)
- `dgraph.<num>.trace.log` (Dgraph tracing log, if one exists)
- `<hostname>-dgraph-stats.xml` (Dgraph statistics page)

You can use other -c arguments to get logs from other components.

You can also use the `get-logs` command to retrieve all of the BDD component logs, as in this example:

```
./bdd-admin.sh get-logs -c all /localdisk/logs/all.zip
```

Chapter 21

Studio Logging

Studio logging helps you to monitor and troubleshoot your Studio application.

About logging in Studio

Studio uses the Apache Log4j logging utility.

The Studio log files include:

- A main log file with most of the logging messages
- A second log file for performance metrics logging
- A third log file for client-side logging, in particular JavaScript errors

The log files are generated in both the standard Log4j format, and the ODL (Oracle Diagnostic Logging) format.

You can also use the **Performance Metrics** page of the **Control Panel** to view performance metrics information.

For more information about Log4j, see the *Apache log4j site*, which provides general information about and documentation for Log4j.

ODL log entry format

The following is an example of an ODL-format NOTIFICATION message resulting from creation of a user session in Studio:

```
[2015-08-04T09:39:49.661-04:00] [EndecaStudio] [NOTIFICATION] []
   [com.endeca.portal.session.UserSession] [host: web12.example.com] [nwaddr: 10.152.105.219]
   [tid: [ACTIVE].ExecuteThread: '45' for queue: 'weblogic.kernel.Default (self-tuning)']
   [userId: djones] [ecid: 0000Kvsw8S17ADkpSw4Eyc1LjsrN0000^6,0] UserSession created
```

The format of the ODL log entries (using the above example) and their descriptions are as follows:

ODL log entry field	Description	Example
Timestamp	The date and time when the message was generated. This reflects the local time zone.	`[2015-08-04T09:39:49.661-04:00]`
Component ID	The ID of the component that originated the message. "EndecaStudio" is hard-coded for the Studio component.	`[EndecaStudio]`
Message Type	The type of message (log level): • INCIDENT_ERROR • ERROR • WARNING • NOTIFICATION • TRACE • UNKNOWN	`[NOTIFICATION]`
Message ID	The message ID that uniquely identifies the message within the component. The ID may be null.	`[]`
Module ID	The Java class that prints the message entry.	`[com.endeca.portal.session.UserSession]`
Host name	The name of the host where the message originated.	`[host: web12.example.com]`
Host address	The network address of the host where the message originated	`[nwaddr: 10.152.105.219]`
Thread ID	The ID of the thread that generated the message.	`[tid: [ACTIVE].ExecuteThread: '45' for queue: 'weblogic.kernel.Default (self-tuning)']`
User ID	The name of the user whose execution context generated the message.	`[userId: djones]`
ECID	The Execution Context ID (ECID), which is a global unique identifier of the execution of a particular request in which the originating component participates. Note that	`[ecid: 0000Kvsw8S17ADkpSw4Eyc1LjsrN0000^6,0]`
Message Text	The text of the log message.	`UserSession created`

Log4j log entry format

The following is an example of a Log4j-format INFO message resulting from creation of a user session in Studio:

```
2015-08-05T05:42:09.855-04:00 INFO [UserSession] UserSession created
```

The format of the Log4j log entries (using the above example) and their descriptions are as follows:

Log4j log entry field	Description	Example
Timestamp	The date and time when the message was generated. This reflects the local time zone.	`[2015-08-04T09:39:49.661-04:00]`
Message Type	The type of message (log level): • FATAL • ERROR • WARN • INFO • DEBUG	`[INFO]`
Module ID	The Java class that prints the message entry.	`[UserSession]`
Message Text	The text of the log message.	`UserSession created`

Rotation frequency

The log rotation frequency is set to daily (it is hard-coded, not configurable). However, you can force rotate the logs by running the `bdd-admin` script with the `rotate-logs` command, as in this example:

```
./bdd-admin.sh rotate-logs -c studio -n web009.us.example.com
```

For information on the `rotate-logs` command, see *rotate-logs on page 49*.

About the Log4j configuration XML files

The primary log configuration is managed in `portal-log4j.xml`, which is packed inside the portal application file `WEB-INF/lib/portal-impl.jar`.

The file is in the standard Log4j XML configuration format, and allows you to:

- Create and modify appenders
- Bind appenders to loggers
- Adjust the log verbosity of different classes/packages

By default, `portal-log4j.xml` specifies a log verbosity of INFO for the following packages:

- `com.endeca`

- `com.endeca.portal.metadata`
- `com.endeca.portal.instrumentation`

It does not override any of the default log verbosity settings for other components.

Note: If you adjust the logging verbosity, it is updated for both Log4j and the Java Utility Logging Implementation (JULI). Code using either of these loggers should respect this configuration.

About the main Studio log file

For Studio, the main log file (`bdd-studio.log`) contains all of the log messages.

By default the `bdd-studio.log` is stored in the WebLogic domain in the `$BDD_DOMAIN/<serverName>/logs` directory (where serverName is the name of the Managed Server in which Studio is installed).

The main root logger prints all messages to:

- The console, which typically is redirected to the application server's output log.
- `bdd-studio.log`, the log file in log4j format.
- `bdd-studio-odl.log`, the log file in ODL format. Also stored in `$BDD_DOMAIN/logs`

The main logger does not print messages from the `com.endeca.portal.instrumentation` classes. Those messages are printed to the metrics log file.

About the metrics log file

Studio captures metrics logging, including all log entries from the `com.endeca.portal.instrumentation` classes.

The metrics log files are:

- `bdd-studio-metrics.log`, which is in Log4j format.
- `bdd-studio-metrics-odl.log`, which is in ODL format.

Both metrics log files are created in the same directory as `bdd-studio.log`.

The metrics log file contains the following columns:

Column Name	Description
Total duration (msec)	The total time for this entry (End time minus Start time).
Start time (msec since epoch)	The time when this entry started. For Dgraph Gateway queries and server executions, uses the server's clock. For client executions, uses the client's clock.

Column Name	Description
End time (msec since epoch)	The time when this entry was finished. For Dgraph Gateway queries and server executions, uses the server's clock. For client executions, uses the client's clock.
Session ID	The session ID for the client.
Page ID	If client instrumentation is enabled, the number of full page refreshes or actions the user has performed. Used to help determine how long it takes to load a complete page. Some actions that do not affect the overall state of a page, such as displaying attributes on the **Available Refinements** panel, do not increment this counter.
Gesture ID	The full count of requests to the server.
Portlet ID	This is the ID associated with an individual instance of a component. It generally includes: • The type of component • A unique identifier For example, if a page includes two **Chart** components, the ID can be used to differentiate them.
Entry Type	The type of entry. For example: • `PORTLET_RENDER` - Server execution in response to a full refresh of a component • `DISCOVERY_SERVICE_QUERY` - Dgraph Gateway query • `CONFIG_SERVICE_QUERY` - Configuration service query • `SCONFIG_SERVICE_QUERY` - Semantic configuration service query • `LQL_PARSER_SERVICE_QUERY` - EQL parser service query • `CLIENT` - Client side JavaScript execution • `PORTLET_RESOURCE` - Server side request for resources • `PORTLET_ACTION` - Server side request for an action
Miscellaneous	A URL encoded JSON object containing miscellaneous information about the entry.

Configuring the amount of metrics data to record

To configure the metrics you want to include, you use a setting in `portal-ext.properties`. This setting applies to both the metrics log file and the **Performance Metrics** page.

The metrics logging can include:

- Queries by Dgraph nodes.
- Portlet server executions by component. The server side code is written in Java.

  It handles configuration updates, configuration persistence, and Dgraph queries. The server-side code generates results to send back to the client-side code.

  Server executions include component render, resource, and action requests.
- Component client executions for each component. The client-side code is hosted in the browser and is written in JavaScript. It issues requests to the server code, then renders the results as HTML. The client code also handles any dynamic events within the browser.

By default, only the Dgraph queries and component server executions are included.

You use the `df.performanceLogging` setting in `portal-ext.properties` to configure the metrics to include. The setting is:

```
df.performanceLogging=<metrics to include>
```

Where <*metrics to include*> is a comma-separated list of the metrics to include. The available values to include in the list are:

Value	Description
`QUERY`	If this value is included, then the page includes information for Dgraph queries.
`PORTLET`	If this value is included, then the page includes information on component server executions.
`CLIENT`	If this value is included, then the page includes information on component client executions.

In the default configuration, where only the Dgraph queries and component server executions are included, the value is:

```
df.performanceLogging=QUERY,PORTLET
```

To include all of the available metrics, you would add the `CLIENT` option:

```
df.performanceLogging=QUERY,PORTLET,CLIENT
```

Note that for performance reasons, this configuration is not recommended.

If you make the value empty, then the metrics log file and **Performance Metrics** page also are empty.

```
df.performanceLogging=
```

About the Studio client log file

The Studio client log file collects client-side logging information. In particular, Studio logs JavaScript errors in this file.

The client log files are:

- `bdd-studio-client.log`, which is in Log4j format.
- `bdd-studio-client-odl.log`, which is in ODL format.

Both client log files are created in the same directory as `bdd-studio.log`.

The client logs are intended primarily for Studio developers to troubleshoot JavaScript errors in the Studio Web application. These files are therefore intended for use by Oracle Support only.

Adjusting Studio logging levels

For debugging purposes in a development environment, you can dynamically adjust logging levels for any class hierarchy.

Note: When you adjust the logging verbosity, it is updated for both Log4j and the Java Utility Logging Implementation (JULI). Code using either of these loggers should respect this configuration.

Adjusting Studio logging levels:

1. In the Big Data Discovery header, click the **Configuration Options** icon and select **Control Panel**.
2. Choose **Server>Server Administration** .
3. Click the **Log Levels** tab.
4. On the **Update Categories** tab, locate the class hierarchy you want to modify.
5. From the logging level list, select the logging level.

   Note: When you modify a class hierarchy, all classes that fall under that class hierarchy also are changed.

6. Click **Save**.

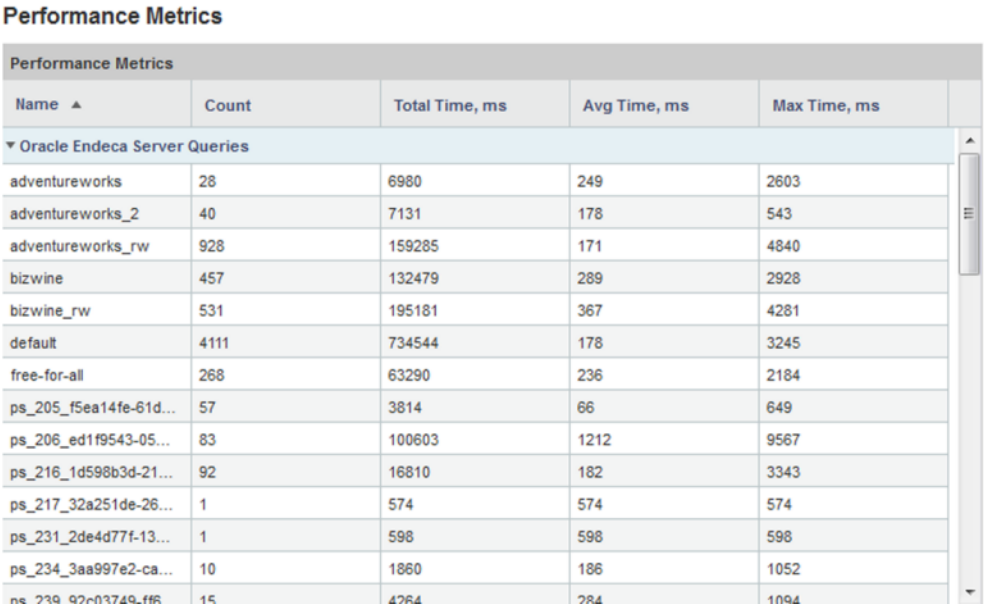
Using the Performance Metrics page to monitor query performance

The **Performance Metrics** page on the **Control Panel** displays information about component and Dgraph Gateway query performance.

It uses the same logging data that is recorded in the metrics log file.

However, unlike the metrics log file, the **Performance Metrics** page uses data stored in memory. Restarting Big Data Discovery clears the **Performance Metrics** data.

For each type of included metric, the table at the top of the page contains a collapsible section.



Performance Metrics

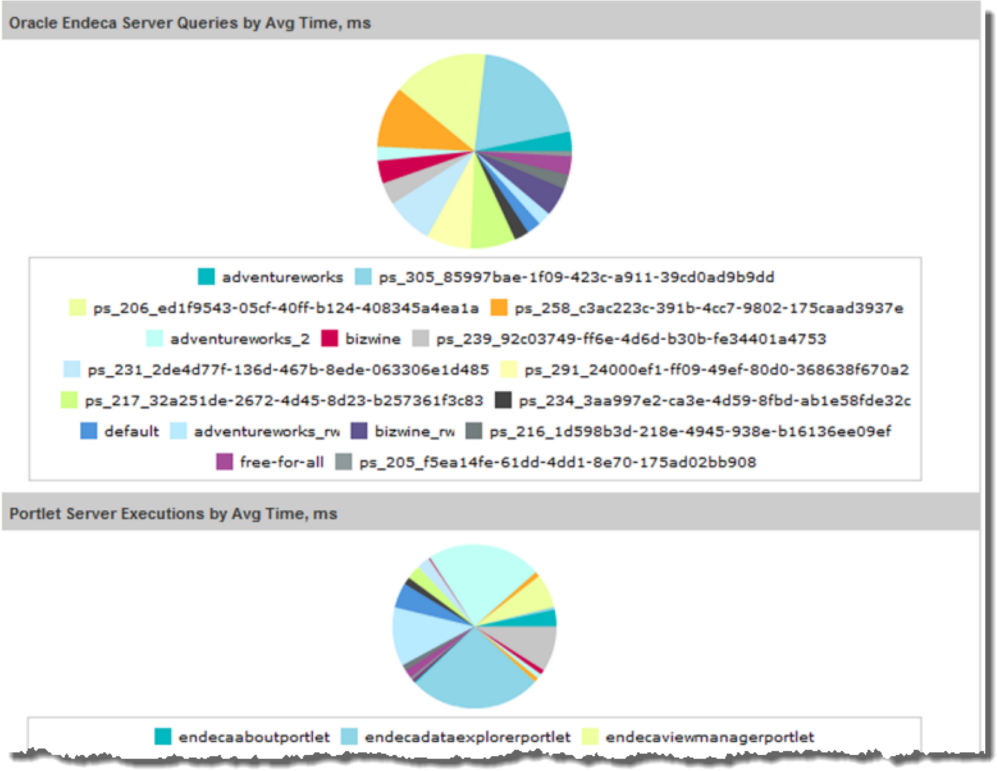
Performance Metrics

Name ▲	Count	Total Time, ms	Avg Time, ms	Max Time, ms
▾ Oracle Endeca Server Queries				
adventureworks	28	6980	249	2603
adventureworks_2	40	7131	178	543
adventureworks_rw	928	159285	171	4840
bizwine	457	132479	289	2928
bizwine_rw	531	195181	367	4281
default	4111	734544	178	3245
free-for-all	268	63290	236	2184
ps_205_f5ea14fe-61d...	57	3814	66	649
ps_206_ed1f9543-05...	83	100603	1212	9567
ps_216_1d598b3d-21...	92	16810	182	3343
ps_217_32a251de-26...	1	574	574	574
ps_231_2de4d77f-13...	1	598	598	598
ps_234_3aa997e2-ca...	10	1860	186	1052
ps_239_92c03749-ff6	15	4264	284	1094

For each data source or component, the table tracks:

- Total number of queries or executions
- Total execution time
- Average execution time
- Maximum execution time

For each type of included metric, there is also a pie chart summarizing the average query or execution time per data source or component.



Oracle Endeca Server Queries by Avg Time, ms

adventureworks
ps_305_85997bae-1f09-423c-a911-39cd0ad9b9dd
ps_206_ed1f9543-05cf-40ff-b124-408345a4ea1a
ps_258_c3ac223c-391b-4cc7-9802-175caad3937e
adventureworks_2
bizwine
ps_239_92c03749-ff6e-4d6d-b30b-fe34401a4753
ps_231_2de4d77f-136d-467b-8ede-063306e1d485
ps_291_24000ef1-ff09-49ef-80d0-368638f670a2
ps_217_32a251de-2672-4d45-8d23-b257361f3c83
ps_234_3aa997e2-ca3e-4d59-8fbd-ab1e58fde32c
default
adventureworks_rw
bizwine_rw
ps_216_1d598b3d-218e-4945-938e-b16136ee09ef
free-for-all
ps_205_f5ea14fe-61dd-4dd1-8e70-175ad02bb908

Portlet Server Executions by Avg Time, ms

endecaaboutportlet
endecadataexplorerportlet
endecaviewmanagerportlet

Note: Dgraph Gateway query performance does not correlate directly to a project page, as a single page often uses multiple Dgraph Gateway queries.

Chapter 22

Dgraph Logging

This section describes the two Dgraph logs.

Dgraph request log

The Dgraph request log (also called the query log) contains one entry for each request processed.

The request log name and storage location is specified by the Dgraph `--log` flag. By default, the name and location of the log file is set to:

```
$BDD_HOME/dgraph/bin/dgraph.reqlog
```

The format of the Dgraph request log consists of the following fields:

- Field 1: Timestamp (yyyy-MM-dd HH:mm:ss.SSS Z).
- Field 2: Client IP Address.
- Field 3: Request ID.
- Field 4: ECID. The ECID (Execution Context ID) is a global unique identifier of the execution of a particular request in which the originating component participates. You can use the ECID to correlate error messages from different components. Note that the ECID comes from the HTTP header, so the ECID value may be null or undefined if the client does not provide it to the Dgraph.
- Field 5: Response Size (bytes).
- Field 6: Total Time (fractional milliseconds).
- Field 7: Processing Time (fractional milliseconds).
- Field 8: HTTP Response Code (0 on client disconnect).
- Field 9: - (unused).
- Field 10: Queue Status. On request arrival, the number of requests in queue (if positive) or the number of available slots at the same priority (if negative).
- Field 11: Thread ID.
- Field 12: HTTP URL (URL encoded).
- Field 13: HTTP POST Body (URL encoded; truncated to 64KBytes, by default; - if empty).
- Field 14: HTTP Headers (URL encoded).

Note that a dash (-) is entered for any field for which information is not available or pertinent. The requests are sorted by their timestamp.

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

Using grep on the Dgraph request log

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

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

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

```
X-Endeca-Portlet-Id:
endecaresultslistportlet_WAR_endecaresultslistportlet_INSTANCE_5RKp_LAYOUT_11601
```

As an example, if you set:

```
PORTLET=endecaresultslistportlet_WAR_endecaresultslistportlet_INSTANCE_5RKp_LAYOUT_11601
```

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

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

The command produces output similar to:

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

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

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

Dgraph out log

The Dgraph out log is where the Dgraph's stdout/stderr output is remapped.

The Dgraph redirects its stdout/stderr output to the log file specified by the Dgraph `--out` flag. By default, the name and location of the file is:

```
$BDD_HOME/logs/dgraph.out
```

You can specify a new out log location by changing the `DGRAPH_OUT_FILE` parameter in the `bdd.conf` file and then restarting the Dgraph.

The Dgraph stdout/stderr log includes startup messages as well as warning and error messages. You can increase the verbosity of the log via the Dgraph `-v` flag. You can also toggle logging verbosity for specified subsystems of the Dgraph, as described in *Setting the Dgraph log levels on page 176*.

Out log format

The format of the Dgraph out log fields are:

- Timestamp
- Component ID
- Message Type
- Log Subsystem
- Job ID
- Message Text

The log entry fields and their descriptions are as follows:

Log entry field	Description	Example
Timestamp	The local date and time when the message was generated, using use the following ISO 8601 extended format: `YYYY-MM-DDTHH:mm:ss.sss(+|-)hh:mm` The hours range is 0 to 23 and milliseconds and offset timezones are mandatory.	`2015-05-18T13:25:30.600-04:00`
Component ID	The ID of the component that originated the message. "DGRAPH" is hard-coded for the Dgraph.	`DGRAPH`
Message Type	The type of message (log level): • `INCIDENT_ERROR` • `ERROR` • `WARNING` • `NOTIFICATION` • `TRACE` • `UNKNOWN`	`WARNING`
Log Subsystem	The log subsystem that generated the message.	`{dgraph}`

Log entry field	Description	Example
Job ID	The ID of the job being executed.	`[0]`
Message Text	The text of the log message.	`Starting HTTP server on port: 7010`

Dgraph log subsystems

The log subsystems that can generate log entries in the Dgraph out log are the following:

- `background_merging` — messages about Dgraph index maintenance activity.
- `bulk_ingest` — messages generated by Bulk Load ingest operations.
- `cluster` — messages about ZooKeeper-related cluster operations.
- `datalayer` — messages about index file usage.
- `dgraph` — messages related to Dgraph general operations.
- `eql` — messages generated from the Endeca Query Language engine.
- `eve` — messages generated from the EVE (Endeca Virtual Engine) query evaluator.
- `http` — messages about Dgraph HTTP communication operations.
- `lexer` — messages from the OLT (Oracle Language Technology) subsystem.
- `splitting` — messages resulting from EVE (Endeca Virtual Engine) splitting tasks.
- `ssl` — messages generated by the SSL subsystem.
- `task_scheduler` — messages related to the Dgraph task scheduler.
- `text_search_rel_rank` — messages related to Relevance Ranking operations during text searches.
- `text_search_spelling` — messages related to spelling correction operations during text searches.
- `update` — messages related to updates.
- `workload_manager` — messages from the Dgraph Workload Manager.
- `ws_request` — messages related to request exchanges between Web services.
- `xq_web_service` — messages generated from the XQuery-based Web services.

All of these subsystems have a default log level of `NOTIFICATION`.

Dgraph start-up arguments

The log entry that begins with "Dynamic graph server" lists the Dgraph start-up flags and arguments. It also lists the Dgraph version, index name and path, PID, HTTP port number, and bulk load port number.

Out log example

The following snippets from a Dgraph out log show the format of the entries:

```
...
2015-05-28T10:15:33.638-04:00    DGRAPH    NOTIFICATION      {cluster}    [0]    Established
coordinator session 0x14d05dbc9bd3d7d with timeout 60000
```

```
2015-05-28T10:15:33.808-04:00    DGRAPH    NOTIFICATION      {database}    [0]    Mounting database
0
...
2015-05-28T23:46:23.592-04:00    DGRAPH    NOTIFICATION      {bulk_ingest}    [0]    Start ingest
for collection: edp_cli_edp_b0cd0b74
2015-05-28T23:46:23.592-04:00    DGRAPH    NOTIFICATION      {bulk_ingest}    [0]    Starting a bulk
ingest operation
2015-05-28T23:46:24.032-04:00    DGRAPH    NOTIFICATION      {bulk_ingest}    [0]    batch 0 finish
BatchUpdating status Success
2015-05-28T23:46:24.032-04:00    DGRAPH    NOTIFICATION      {bulk_ingest}    [0]    Ending bulk
ingest at client's request - finalizing changes
2015-05-28T23:46:24.593-04:00    DGRAPH    NOTIFICATION      {bulk_ingest}    [0]    Bulk ingest
completed: Added 4 records and rejected 0 records.
```

The `bulk_ingest` entries show the ingest of a small data set.

Setting the Dgraph log levels

The `DGRAPH_LOG_LEVEL` property in `bdd.conf` sets the log levels for the Dgraph log subsystems at start-up time.

If you do not explicitly set the log levels (i.e., if the `DGRAPH_LOG_LEVEL` property is empty), all the log subsystems will use the `NOTIFICATION` log level.

The syntax of the property is:

```
DGRAPH_LOG_LEVEL="subsystem1 level1|subsystem2 level2|subsystemN levelN"
```

where:

- *subsystem* is a Dgraph log subsystem name, as listed in *Dgraph log subsystems on page 175*.
- *level* is one of these log levels:
  - `INCIDENT_ERROR`
  - `ERROR`
  - `WARNING`
  - `NOTIFICATION`
  - `TRACE`

The pipe character is required if you are setting more than one subsystem/level combination.

To set the Dgraph log levels:

1. Modify the `DGRAPH_LOG_LEVEL` property in `bdd.conf` to set the required log levels.

   Be sure you modify the `bdd.conf` version that is in the `$BDD_HOME/BDD_manager/conf` directory.

2. Run the `bdd-admin` script with the `publish-config` command to update the configuration file of your BDD cluster.

   For details on this command, see *publish-config on page 36*.

3. Restart the Dgraph by running the `bdd-admin` script with the `restart` command.

   For details on this command, see *restart on page 30*.

Keep in mind that you can dynamically change the Dgraph log levels by running the `bdd-admin` script with the `set-log-levels` command, as in this example:

```
./bdd-admin.sh set-log-levels -c dgraph -s eql,task_scheduler -l warning
```

The new log level may persist into the next Dgraph re-start, depending on whether the command's `--non-persistent` option is used:

- If `--non-persistent` is used, the change will not persist into the next Dgraph re-start, at which time the log levels in the `DGRAPH_LOG_LEVEL` property are used.
- If `--non-persistent` is omitted, the new setting is persisted by being written to the `DGRAPH_LOG_LEVEL` property in `bdd.conf`. This means that the next Dgraph re-start will use the changed the log levels in the `bdd.conf` file.

For details on the `set-log-levels` command, see *set-log-levels on page 45*.

Chapter 23

Dgraph Gateway Logging

This section describes the logging of the Dgraph Gateway process in the WebLogic Server domain.

Dgraph Gateway logs

Dgraph Gateway uses the Apache Log4j logging utility for logging and its messages are written to WebLogic Server logs.

The BDD installation creates a WebLogic domain, whose name is set by the `WEBLOGIC_DOMAIN_NAME` parameter of the `bdd.conf` file. The WebLogic domain has both an Admin Server and a Managed Server. The Admin Server is named **AdminServer** while the Managed Server has the same name as the host machine. Both the Dgraph Gateway and Studio are deployed into the Managed Server.

There are two sets of logs for the two different servers:

- The Admin Server logs are in the `$BDD_DOMAIN/servers/AdminServer/logs` directory.
- The Managed Server logs are in the `$BDD_DOMAIN/servers/<serverName>/logs` directory .

There are three types of logs:

- WebLogic Domain Log
- WebLogic Server Log
- Application logs

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

By default, these log files are located in the `$DOMAIN_HOME/servers/AdminServer/logs` directory (for the Admin Server) or one of the `$DOMAIN_HOME/servers/<serverName>/logs` directories (for a Managed Server).

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

WebLogic Domain Log

The WebLogic domain log is generated only for the Admin Server. This domain log is intended to provide a central location from which to view the overall status of the domain.

The name of the domain log is:

```
$BDD_DOMAIN/servers/AdminServer/logs/<bdd_domain>.log
```

The domain log is located in the `$DOMAIN_HOME/servers/AdminServer/logs` directory.

For more information on the WebLogic domain and server logs, see the "Server Log Files and Domain Log Files" topic in this page:
http://docs.oracle.com/cd/E24329_01/web.1211/e24428/logging_services.htm#WLLOG124

WebLogic Server Log

A WebLogic server log is generated for the Admin Server and for each Managed Server instance.

The default path of the Admin Server server log is:

```
$BDD_DOMAIN/servers/AdminServer/logs/AdminServer.log
```

The default path of the server log for a Managed Server is:

```
$BDD_DOMAIN/servers/<serverName>/logs/<serverName>.log
```

For example, if "web001.us.example.com" is the name of the Managed Server, then its server log is:

```
$BDD_DOMAIN/servers/web001.us.example.com/logs/web001.us.example.com.log
```

Application logs

Application logs are generated by the deployed applications. In this case, Dgraph Gateway and Studio are the applications.

For Dgraph Gateway, its application log is at:

```
$BDD_DOMAIN/servers/<serverName>/logs/<serverName>-diagnostic.log
```

For example, if "web001.us.example.com" is the name of the Managed Server, then the Dgraph Gateway application log is:

```
$BDD_DOMAIN/servers/web001.us.example.com/logs/web001.us.example.com-diagnostic.log
```

For Studio, its application log is at:

```
$BDD_DOMAIN/servers/<serverName>/logs/bdd-studio.log
```

For example, if "web001.us.example.com" is the name of the Managed Server, then its application log is:

```
$BDD_DOMAIN/servers/web001.us.example.com/logs/bdd-studio.log
```

The directory also stores other Studio metric log files, which are described in *About the metrics log file on page 166*.

Logs to check when problems occur

For Dgraph Gateway problems, you should check the WebLogic server log for the Managed Server and the Dgraph Gateway application log:

```
$BDD_DOMAIN/servers/<serverName>/logs/<serverName>.log
```

```
and
$BDD_DOMAIN/servers/<serverName>/logs/<serverName>-diagnostic.log
```

For Studio issues, check the WebLogic server log for the Managed Server and the Dgraph Gateway application log:

```
$BDD_DOMAIN/servers/<serverName>/logs/<serverName>.log
and
$BDD_DOMAIN/servers/<serverName>/logs/bdd-studio.log
```

Dgraph Gateway log entry format

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

The following is an example of an error message:

```
[2014-08-21T15:09:08.711+08:00] [EndecaServer] [ERROR] [OES-000091]
[com.endeca.opmodel.ws.ControlServletContextListener] [host: YYZHU-CA] [nwaddr: 10.192.251.139]
[tid: [ACTIVE].ExecuteThread: '24' for queue: 'weblogic.kernel.Default (self-tuning)'] [userId:
YYZHU]
[ecid: 0000KVrPS^C1FgUpM4^Aye1JxPgK000000,0] OES-000091: Could not find properties file:
C:\WebLogic\Oracle\MIDDLE~1\USER_P~1\domains\BASE_D~1\config\EndecaServer.properties
```

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

Log entry field	Description	Example
Timestamp	The date and time when the message was generated. This reflects the local time zone.	`[2014-08-21T15:09:08.711+08:00]`
Component ID	The ID of the component that originated the message. "EndecaServer" is hard-coded for the Dgraph Gateway.	`[EndecaServer]`
Message Type	The type of message (log level): • INCIDENT_ERROR • ERROR • WARNING • NOTIFICATION • TRACE • UNKNOWN	`[ERROR]`
Message ID	The message ID that uniquely identifies the message within the component. The ID consists of the prefix `OES` (representing the component), followed by a dash, then a number.	`[OES-000091]`
Module ID	The Java class that prints the message entry.	`[com.endeca.opmodel.ws.ControlServletContextListener]`

Log entry field	Description	Example
Host name	The name of the host where the message originated.	[host: web05.example.com]
Host address	The network address of the host where the message originated	[nwaddr: 10.192.251.139]
Thread ID	The ID of the thread that generated the message.	[tid: [ACTIVE].ExecuteThread: '24' for queue: 'weblogic.kernel.Default (self-tuning)']
User ID	The name of the user whose execution context generated the message.	[userId: YYZHU]
ECID	The Execution Context ID (ECID), which is a global unique identifier of the execution of a particular request in which the originating component participates.	[ecid: 0000KVrPS^C1FgUpM4^Aye1JxPgK000000,0]
Message Text	The text of the log message.	OES-000091: Could not find properties file: ...]

Log entry information

This topic describes some of the information that is found in log entries.

For Dgraph Gateways in cluster-mode, this logged information can help you trace the life cycle of requests.

Note that all Dgraph Gateway ODL log entries are prefixed with OES followed by the number and text of the message, as in this example:

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

Logged request type and content

When a new request arrives at the server, the SOAP message in the request is analyzed. From the SOAP body, the request type of each request (such as allocateBulkLoadPort) is determined and logged. Complex requests (like Conversation) will be analyzed further, and detailed information will be logged as needed. Note that this information is logged if the log level is DEBUG.

For example, a Conversation request is sent to Server1. After being updated, the logs on the server might have entries such as these:

```
OES-000239: Receive request 512498665 of type 'Conversation'. This request does the
   following queries: [RecordCount, RecordList]
OES-000002: Timing event: start 512498665 ...
OES-000002: Timing event: DGraph start 512498665 ...
OES-000002: Timing event: DGraph end 512498665 ...
OES-000002: Timing event: end 512498665 ...
```

As shown in the example, when Server1 receives a request, it will choose a node from the routing table and tunnel the request to that node. The routed request will be processed on that node. In the Dgraph request log, the request can also be tracked via the request ID in the HTTP header.

Log ingest timestamp and result

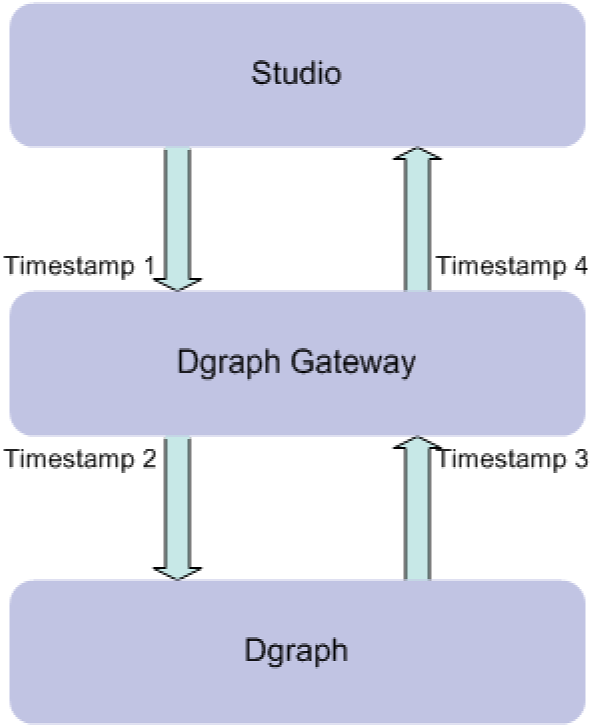
For ingest operations, a start and end timestamp is logged. At the end of the operation, the ingest results are also logged (number of added records, number of deleted records, number of updated records, number of replaced records, number of added or updated records).

Log entries would look like these examples:

```
OES-000002: Timing event: start ingest into Dgraph "http://host:7010"
OES-000002: Timing event: end ingest into Dgraph "http:/
/host:7010" (1 added, 1 deleted, 0 replaced, 0 updated, 0 added or updated)
```

Total request and Dgraph processing times

Four calculated timestamps in the logs record the time points of a query as it moves from Studio to the Dgraph and back. The query path is shown in this illustration:



The four timestamps are:

1. Timestamp1: Dgraph Gateway begins to process the request from Studio
2. Timestamp2: Dgraph Gateway forwards the request to the Dgraph
3. Timestamp3: Dgraph Gateway receives the response from the Dgraph
4. Timestamp4: Dgraph Gateway finishes processing the request

To determine the total time cost of the request, the timestamp differences are calculated and logged:

- (Timestamp4 - Timestamp1) is the total request processing time in Dgraph Gateway.
- (Timestamp3 - Timestamp2) is the Dgraph processing time.

The log entries will look similar to these examples:

```
OES-000240: Total time cost(Request processing) of request 512498665 : 1717 ms
OES-000240: Total time cost(Dgraph processing) of request 512498665 : 424 ms
```

Logging properties file

Dgraph Gateway has a default Log4j configuration file that sets its logging properties.

The file is named `EndecaServerLog4j.properties` and is located in the `$DOMAIN_HOME/config` directory.

The default version of the file is as follows:

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

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

# ODL-format Log Appender
log4j.appender.ODL=com.endeca.util.ODLAppender
log4j.appender.ODL.MaxSize=1048576000
log4j.appender.ODL.MaxSegmentSize=104857600
log4j.appender.ODL.encoding=UTF-8
log4j.appender.ODL.MaxDaysToRetain=7

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

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

The file has the following properties:

Logging property	Description
`log4j.rootLogger=WARN`, `stdout`, `ODL`	The level of the root logger is defined as WARN and attaches the Console Appender (stdout) and ODL-format Log Appender (ODL) to it.
`log4j.appender.stdout=org.apache.log4j.ConsoleAppender`	Defines stdout as a `Log4j ConsoleAppender`
`org.apache.log4j.PatternLayout`	Sets the `PatternLayout` class for the stdout layout.

Logging property	Description
`log4j.appender.stdout.layout.ConversionPattern`	Defines the log entry conversion pattern as: • **%d** is the date of the logging event. • **%p** outputs the priority of the logging event. • **%c** outputs the category of the logging event. • **%L** outputs the line number from where the logging request was issued. • **%m** outputs the application-supplied message associated with the logging event while **%n** is the platform-dependent line separator character. For other conversion characters, see: *https://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/PatternLayout.html*
`log4j.appender.ODL=com.endeca.util.ODLAppender`	Defines ODL as an ODL Appender. ODL (Oracle Diagnostics Logging) is the logging format for Oracle applications.
`log4j.appender.ODL.MaxSize`	Sets the maximum amount of disk space to be used by the `<ServerName>-diagnositic.log` file and the logging rollover files. The default is 1048576000 (about 1GB). Older log files are deleted to keep the total log size under the given limit.
`log4j.appender.ODL.MaxSegmentSize`	Sets the maximum size (in bytes) of the log file. When the `<ServerName>-diagnositic.log` file reaches this size, a rollover file is created. The default is 104857600 (about 10 MB).
`log4j.appender.ODL.encoding`	Sets character encoding the log file. The default UTF-8 value prints out UTF-8 characters in the file.

Logging property	Description
`log4j.appender.ODL.MaxDaysToRetain`	Sets how long (in days) older log file should be kept. Files that are older than the given days are deleted. Files are deleted only when there is a log rotation. As a result, files may not be deleted for some time after the retention period expires. The value must be a positive integer. The default is 7 days.
`log4j.logger.com.endeca`	Sets the default log level for the Dgraph Gateway log messages. ERROR is the default log level.
`log4j.logger.org.apache.zookeeper`	Sets the default log level for the ZooKeeper client logger (i.e., not for the ZooKeeper server that is running on the Hadoop environment. WARN is the default log level.

Rotation frequency

The log rotation frequency is set to daily (it is hard-coded, not configurable). This means that a new log file is created either when the log file reaches a certain size (the `MaxSegmentSize` setting) or when a particular time is reached (it is 00:00 UTC for Dgraph Gateway).

However, you can force rotate the logs by running the `bdd-admin` script with the `rotate-logs` command, as in this example:

```
./bdd-admin.sh rotate-logs -c gateway -n web009.us.example.com
```

For information on the `rotate-logs` command, see *rotate-logs on page 49*.

Log levels

This topic describes the log levels that can be set in the `EndecaServerLog4j.properties` file.

The WebLogic logger for Dgraph Gateway is configured with the type of information written to log files, by specifying the log level. When you specify the type, WebLogic returns all messages of that type, as well as the messages that have a higher severity. For example, if you set the message type to `WARN`, WebLogic also returns messages of type `FATAL` and `ERROR`.

The `EndecaServerLog4j.properties` file lists two packages for which you can set a logging level:

- `log4j.logger.com.endeca` for BDD-related events.
- `log4j.logger.org.apache.zookeeper` for the Dgraph Gateway ZooKeeper client.

There are two ways of changing a log level:

- Manually, by opening the properties file in a text editor and changing the level. With this method, you use a Java log level from the table below.

- Dynamically, by using the `bdd-admin` script with the `set-log-levels` command. With this method, you use an ODL log level from the table below.

This example shows how you can manually change a log level setting:

```
log4j.logger.com.endeca=INFO
```

In the example, the log level for the Endeca logger is set to INFO.

Logging levels

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

Java Log Level	ODL Log Level	Meaning
`OFF`	N/A	Has the highest possible rank and is used to turn off logging.
`FATAL`	`INCIDENT_ERROR`	Indicates a serious problem that may be caused by a bug in the product and that should be reported to Oracle Support. In general, these messages describe events that are of considerable importance and which will prevent normal program execution.
`ERROR`	`ERROR`	Indicates a serious problem that requires immediate attention from the administrator and is not caused by a bug in the product.
`WARN`	`WARNING`	Indicates a potential problem that should be reviewed by the administrator.
`INFO`	`NOTIFICATION`	A message level for informational messages. This level typically indicates a major lifecycle event such as the activation or deactivation of a primary sub-component or feature. This is the default level.
`DEBUG`	`TRACE`	Debug information for events that are meaningful to administrators, such as public API entry or exit points.

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

Dynamically changing log levels

You can use the `bdd-admin` script with the `set-log-levels` command to change the log level of the `log4j.logger.com.endeca` package. The command takes one of the ODL levels and converts it to its Java-level equivalent before writing it to the properties file. Note that this command cannot change the setting of the `log4j.logger.org.apache.zookeeper` package. For usage information, see *set-log-levels on page 45*.

At any time, you can use the `bdd-admin` script with the `get-log-levels` command to retrieve the setting of the `log4j.logger.com.endeca` package. For usage information, see *get-log-levels on page 44*.

Customizing the HTTP access log

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

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

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

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

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

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

To customize the HTTP access log:

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

   ```
   date time time-taken cs-method cs-uri sc-status
   ```

9. Click **Save**.
10. In the **Change Center of the Administration Console**, click **Activate Changes**.
11. Restart WebLogic Server by running the `bdd-admin` script with the `restart` command. For example:

    ```
    ./bdd-admin.sh restart -c bddServer -n web05.us.example.com
    ```

    For information on the `restart` command, see *restart on page 30*.

Index