

# Oracle® Big Data Discovery

Administrator's Guide

Version 1.4.0 • October 2016

# Copyright and disclaimer

Copyright © 2015, 2017, Oracle and/or its affiliates. All rights reserved.

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

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

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

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

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

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

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

# Table of Contents

<b>Copyright and disclaimer</b> .....	<b>2</b>
<b>Preface</b> .....	<b>8</b>
About this guide .....	8
Audience .....	8
Conventions .....	8
Contacting Oracle Customer Support .....	9

## Part I: Overview of Big Data Discovery

<b>Chapter 1: Overview of BDD Components</b> .....	<b>11</b>
Studio .....	11
Data Processing .....	11
Dgraph .....	12
<b>Chapter 2: The BDD Cluster</b> .....	<b>13</b>
What is a BDD cluster? .....	13
Cluster architecture .....	13
The Admin Server .....	14
The bdd user .....	14

## Part II: Administering Big Data Discovery

<b>Chapter 3: The bdd-admin Script Reference</b> .....	<b>16</b>
About the bdd-admin script .....	16
Lifecycle management commands .....	19
start .....	19
stop .....	20
restart .....	22
System management commands .....	23
autostart .....	23
backup .....	24
restore .....	27
publish-config .....	30
bdd .....	31
hadoop .....	31
kerberos .....	32
cert .....	33
database .....	34
update-model .....	34
flush .....	35

reshape-nodes . . . . .	36
enable-components . . . . .	36
disable-components . . . . .	37
Diagnostics commands . . . . .	37
get-blackbox . . . . .	38
status . . . . .	38
get-stats . . . . .	39
reset-stats . . . . .	40
get-log-levels . . . . .	41
set-log-levels . . . . .	42
get-logs . . . . .	44
rotate-logs . . . . .	47
<b>Chapter 4: Updating Configuration . . . . .</b>	<b>49</b>
Updating bdd.conf . . . . .	49
Configuration properties that can be modified . . . . .	50
Updating BDD's Hadoop configuration . . . . .	54
Updating the Hadoop client configuration files . . . . .	54
Setting the Hue URI . . . . .	54
Upgrading Hadoop . . . . .	55
Updating BDD's Kerberos configuration . . . . .	57
Enabling Kerberos . . . . .	57
Changing the location of the Kerberos krb5.conf file . . . . .	59
Updating the Kerberos keytab file . . . . .	59
Updating the Kerberos principal . . . . .	60
Updating component database configuration . . . . .	60
Refreshing TLS/SSL certificates . . . . .	61
<b>Chapter 5: Adding and Removing BDD Nodes . . . . .</b>	<b>63</b>
Adding new Dgraph nodes . . . . .	63
Adding new Data Processing nodes . . . . .	65
Removing Data Processing nodes . . . . .	66
<b>Chapter 6: Backing Up and Restoring BDD . . . . .</b>	<b>67</b>
Backing up BDD . . . . .	67
Performing a full BDD restoration . . . . .	68
Restoring BDD to a new cluster . . . . .	70
Troubleshooting MySQL database restorations . . . . .	72
<b>Part III: Administering the Dgraph</b>	
<b>Chapter 7: Dgraph Overview . . . . .</b>	<b>75</b>
The Dgraph databases . . . . .	75
Moving the Dgraph databases to HDFS . . . . .	77
The Dgraph cluster . . . . .	80
Dgraph memory consumption . . . . .	81
The Dgraph Tracing Utility . . . . .	82

Dgraph statistics	82
<b>Chapter 8: Adjusting Dgraph Settings</b>	<b>83</b>
Changing the Dgraph memory limit	83
Setting the Dgraph cache size	84
Using Linux ulimit settings for merges	85
Setting up cgroups for the Dgraph	85
<b>Chapter 9: Dgraph and Dgraph HDFS Agent Flags</b>	<b>87</b>
Dgraph flags	87
Dgraph HDFS Agent flags	92
<b>Part IV: Administering Studio</b>	
<b>Chapter 10: Managing Data Sources</b>	<b>95</b>
About database connections and JDBC data sources	95
Creating data connections	95
Deleting data connections	96
Creating a data source	96
Editing a data source	97
Deleting a data source	97
<b>Chapter 11: Configuring Studio Settings</b>	<b>98</b>
Studio settings in BDD	98
Changing the Studio setting values	100
Modifying the Studio session timeout value	100
Changing the Studio database password	101
Viewing the Server Administration Page information	101
<b>Chapter 12: Configuring Data Processing Settings</b>	<b>102</b>
List of Data Processing Settings	102
Changing the data processing settings	103
<b>Chapter 13: Running a Studio Health Check</b>	<b>105</b>
<b>Chapter 14: Viewing Project Usage Summary Reports</b>	<b>106</b>
About the project usage logs	106
About the System Usage page	107
Using the System Usage page	108
<b>Chapter 15: Configuring the Locale and Time Zone</b>	<b>111</b>
Locales and their effect on the user interface	111
How Studio determines the locale to use	112
Locations where the locale may be set	112
Scenarios for selecting the locale	112
Selecting the default locale	113
Configuring a user's preferred locale	114
Setting the default time zone	115

<b>Chapter 16: Configuring Settings for Outbound Email Notifications</b> .....	<b>117</b>
Configuring the email server settings .....	117
Configuring the sender name and email address for notifications .....	118
Setting up the Account Created and Password Changed notifications .....	118
<b>Chapter 17: Managing Projects from the Control Panel</b> .....	<b>120</b>
Configuring the project type .....	120
Assigning users and user groups to projects .....	121
Certifying a project .....	121
Making a project active or inactive .....	121
Deleting projects .....	122
<b>Part V: Controlling User Access to Studio</b>	
<b>Chapter 18: Configuring User-Related Settings</b> .....	<b>124</b>
Configuring authentication settings for users .....	124
Configuring the password policy .....	125
Restricting the use of specific screen names and email addresses .....	126
<b>Chapter 19: Creating and Editing Studio Users</b> .....	<b>127</b>
About user roles and access privileges .....	127
Creating a new Studio user .....	131
Editing a Studio user .....	132
Deactivating, reactivating, and deleting Studio users .....	133
<b>Chapter 20: Integrating with an LDAP System to Manage Users</b> .....	<b>134</b>
About using LDAP .....	134
Configuring the LDAP settings and server .....	135
Authenticating against LDAP over TLS/SSL .....	139
Preventing encrypted LDAP passwords from being stored in BDD .....	140
Assigning roles based on LDAP user groups .....	140
<b>Chapter 21: Setting Up Single Sign-On (SSO)</b> .....	<b>142</b>
About using single sign-on .....	142
Overview of the process for configuring SSO with Oracle Access Manager .....	142
Configuring the reverse proxy module in OHS .....	143
Registering the Webgate with the Oracle Access Manager server .....	144
Testing the OHS URL .....	145
Configuring Big Data Discovery to integrate with SSO via Oracle Access Manager .....	146
Configuring the LDAP connection for SSO .....	146
Configuring the Oracle Access Manager SSO settings .....	147
Completing and testing the SSO integration .....	148
<b>Part VI: Logging for Studio, Dgraph, and Dgraph Gateway</b>	
<b>Chapter 22: Overview of BDD Logging</b> .....	<b>151</b>
List of Big Data Discovery logs .....	151

---

Gathering information for diagnosing problems .....	153
Retrieving logs .....	156
Rotating logs .....	156
<b>Chapter 23: Studio Logging .....</b>	<b>157</b>
About logging in Studio .....	157
About the Log4j configuration XML files .....	159
About the main Studio log file .....	160
About the metrics log file .....	160
Configuring the amount of metrics data to record .....	161
About the Studio client log file .....	162
Adjusting Studio logging levels .....	163
Using the Performance Metrics page to monitor query performance .....	163
<b>Chapter 24: Dgraph Logging .....</b>	<b>166</b>
Dgraph request log .....	166
Dgraph out log .....	167
Dgraph log levels .....	170
Setting the Dgraph log levels .....	171
<b>Chapter 25: Dgraph Gateway Logging .....</b>	<b>173</b>
Dgraph Gateway logs .....	173
Dgraph Gateway log entry format .....	175
Log entry information .....	176
Logging properties file .....	178
Setting the Dgraph Gateway log level .....	181
Customizing the HTTP access log .....	182

## Preface

Oracle Big Data Discovery is a set of end-to-end visual analytic capabilities that leverage the power of Apache Spark to turn raw data into business insight in minutes, without the need to learn specialist big data tools or rely only on highly skilled resources. The visual user interface empowers business analysts to find, explore, transform, blend and analyze big data, and then easily share results.

## 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
<b>User Interface Elements</b>	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.
<i>Variable</i>	This formatting is used for variable values. For variables within a code sample, the formatting is <i>Variable</i> .
File Path	This formatting is used for file names and paths.

### Path variable conventions

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

Path variable	Meaning
<code>\$ORACLE_HOME</code>	Indicates the absolute path to your Oracle Middleware home directory, where BDD and WebLogic Server are installed.



Path variable	Meaning
\$BDD_HOME	Indicates the absolute path to your Oracle Big Data Discovery home directory, \$ORACLE_HOME/BDD- <i>&lt;version&gt;</i> .
\$DOMAIN_HOME	Indicates the absolute path to your WebLogic domain home directory. For example, if your domain is named <i>bdd- &lt;version&gt;_domain</i> , then \$DOMAIN_HOME is \$ORACLE_HOME/user_projects/domains/ <i>bdd- &lt;version&gt;_domain</i> .
\$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**



## Chapter 1

# Overview of BDD Components

---

BDD is made up of a number of distinct components.

[Studio](#)

[Data Processing](#)

[Dgraph](#)

## Studio

Studio is Big Data Discovery's front-end web application. It provides tools that you can use to create and manage data sets and projects, as well as administrator tools for managing end user access and other settings.

## Transform Service

The Transform Service processes user-defined changes to data sets, called *transformations*, on behalf of Studio. It enables users to preview the effects their transformations will have on their data before saving them.

## Data Processing

Data Processing collectively refers to a set of processes and jobs that discover, sample, profile, and enrich source data.

For the most part, all aspects of the Data Processing component of BDD, including configuration, behavior, treatment of data types and logs, are described in the *Data Processing Guide*. However, a few settings which you modify in Studio's Control Panel are described in this guide.

## Workflow Manager Service

The Workflow Manager Service acts as an intermediary between Spark and the BDD clients: Studio and Data Processing CLI. The service receives data set workflow requests from its BDD clients, and delegates the sequence of Spark jobs needed for each workflow, such as sampling, discovery or transformations, to run in YARN. The Spark jobs run asynchronously of each other and the service notifies Studio of the job status. The Workflow Manager also delegates jobs to other components in BDD, such as the Dgraph and the Dgraph HDFS Agent.

For information on modifying the Workflow Manager configuration, see the *Data Processing Guide*.

## Data Processing CLI

The Data Processing Command Line Interface (CLI) provides a way to manually launch Data Processing workflows and invoke the Hive Table Detector (see below). You can also configure it to run as a cron job. The DP CLI submits its data set workflow requests to the Workflow Manager Service, which then delegates the jobs to run as needed.

## Hive Table Detector

The Hive Table Detector is a Data Processing component that monitors the Hive database for new or deleted tables, and launches Data Processing workflows as needed. It's invoked by the CLI, either manually by the Hive administrator or automatically by a cron job.

## Dgraph

The Dgraph indexes the data sets produced by Data Processing and stores them in databases. It also responds to end user queries for the indexed data, which are routed to it by the Dgraph Gateway.

## Dgraph Gateway

The Dgraph Gateway is a Java-based interface that routes requests to the Dgraph instances and provides caching and business logic. It also handles cluster services for the Dgraph instances by leveraging Hadoop ZooKeeper.

## Dgraph HDFS Agent

The Dgraph HDFS Agent acts as a data transport layer between the Dgraph and the HDFS environment. It exports records to HDFS on behalf of the Dgraph, and imports records from HDFS during data ingest operations.

The HDFS Agent is automatically installed on the same nodes as the Dgraph.



## Chapter 2

# The BDD Cluster

---

*What is a BDD cluster?*

*Cluster architecture*

*The Admin Server*

*The bdd user*

## What is a BDD cluster?

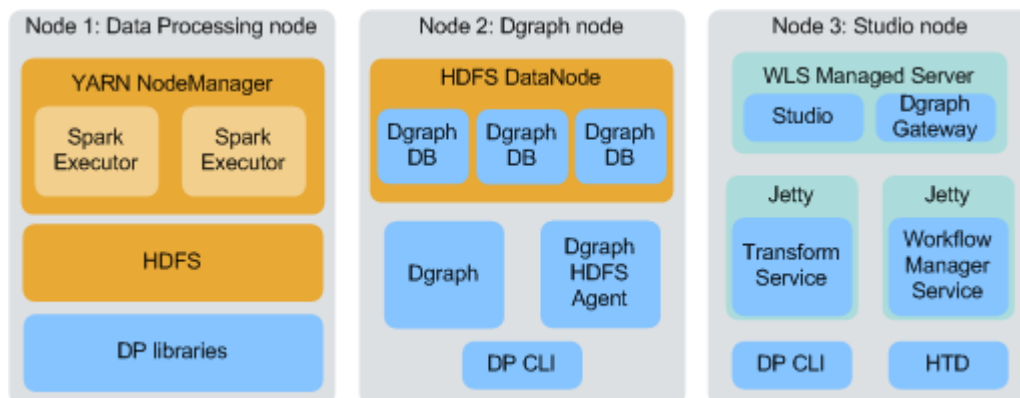
A BDD cluster is an on-premise installation of Big Data Discovery, on either commodity hardware or an engineered system like Oracle Big Data Appliance.

Typically, the term refers to a BDD installation running on multiple nodes in a production environment, although the software can also be installed in a single or dual node configuration for demo or development purposes, respectively.

The BDD cluster depends on and integrates with a separate Hadoop cluster, which provides data loading, processing, and storage functionality. BDD runs on a subset of nodes in the Hadoop cluster and communicates with others on a regular basis. For example, the BDD Data Processing libraries are installed on qualified Hadoop YARN NodeManagers, and BDD nodes running the Dgraph Gateway connect to Hadoop ZooKeeper nodes regularly for assistance managing the Dgraph.

## Cluster architecture

BDD supports many different cluster configurations, although most will contain nodes similar to the following.



The diagram above depicts the three basic types of BDD nodes. Note that it doesn't include nodes in your Hadoop cluster that BDD requires but doesn't run on, like those running ZooKeeper and your Hadoop cluster manager.

- Node 1 is running BDD Data Processing libraries, along with the YARN NodeManager service, Spark on YARN, and HDFS, which DP requires to function. A typical production cluster would contain multiple Data Processing nodes. Oracle recommends at least three to maintain high availability.
- Node 2 is running the Dgraph, the Dgraph HDFS Agent, the DP CLI, and the Hadoop HDFS DataNode service, which is required because the Dgraph databases are stored on HDFS. (Note that the Dgraph databases could also be stored on an NFS (network file system), in which case HDFS would not be required on Dgraph nodes. For more information, see [The Dgraph databases on page 75](#).) A typical production cluster would contain multiple such nodes to ensure high availability of the Dgraph.
- Node 3 is running Studio and the Dgraph Gateway inside a WebLogic Managed Server; the Transform Service and Workflow Manager Service, each inside a Jetty container; the DP CLI; and the Hive Table Detector. A typical cluster would contain one or more Studio nodes, depending on the number of end users making concurrent queries. Note that in a cluster with multiple Studio nodes, the Workflow Manager Service and Hive Table Detector would each be installed on only one of them.

As previously mentioned, you aren't bound to the configurations described above. You can co-locate different BDD and Hadoop components on the same node, and your cluster can contain any number of each type of node. More information on cluster configuration and component co-location is available in the *Installation Guide*.

Furthermore, you can add and remove Dgraph and Data Processing nodes as needed post-install. For instructions, see [Adding and Removing BDD Nodes on page 62](#). You can also add and remove non-BDD nodes from your Hadoop cluster without impacting BDD.

## The Admin Server

One node in a BDD cluster serves as the Administration Server, or Admin Server. This is a WebLogic Server entity that plays an important role in BDD.

In addition to controlling the WebLogic Managed Servers that Studio and the Dgraph Gateway run in, the Admin Server serves as a central point of control for the BDD cluster. Among other things, all script-based administrative tasks—including starting and stopping components, as well as backing up and restoring BDD data—are performed from this node.

This document only describes the role of the Admin Server within the BDD cluster. For information on how it's used in the context of WebLogic Server, see [Role of the Administration Server](#) in WebLogic Server's documentation.

## The bdd user

The `bdd` user is the Linux user that installed BDD and runs all BDD processes. Although this user might have a different name in your installation, this document refers to it as the `bdd` user for simplicity.

When performing administrative tasks like running the `bdd-admin` script and the DP CLI, you must be logged in as the `bdd` user. This document will specify when an operation needs to be performed by the `bdd` user.

Because the `bdd` user performs tasks that affect all nodes in your BDD cluster and some in your Hadoop cluster, it requires special permissions. These are described in the *Installation Guide*.

# **Part II**

## **Administering Big Data Discovery**



## Chapter 3

# The bdd-admin Script Reference

---

You can use the `bdd-admin` script to administer your BDD cluster from the command line. This section describes the script and its commands.

[About the bdd-admin script](#)

[Lifecycle management commands](#)

[System management commands](#)

[Diagnostics 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 by the `bdd` user. This user must have the following:

- Passwordless `sudo` enabled on all nodes in the cluster
- The same UID on all nodes in the cluster

`bdd-admin` has the following syntax:

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

When you run the script, you must specify a command. This determines the operation it will perform. 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`
- `publish-config`
- `reshape-nodes`

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 options to further control the script's behavior. For example, you can run most commands on all nodes or one or more specific ones. The options each command supports are described later in this chapter.

The commands `bdd-admin` supports are described below.

## Lifecycle management commands

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

Command	Description
<code>start</code>	Starts components.
<code>stop</code>	Stops components.
<code>restart</code>	Restarts components.

## System management commands

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

Command	Description
<code>autostart</code>	Enables/disables autostart for components. Components that have autostart enabled will automatically restart after their hosts are rebooted.
<code>backup</code>	Backs up your cluster's data and metadata to a single tar file.
<code>restore</code>	Restores your cluster's data and metadata from a backup tar file.
<code>publish-config</code>	Publishes updated BDD, Hadoop, Kerberos, and database configuration to all BDD nodes. Can also be used to refresh TLS/SSL certificates on secured Hadoop clusters.
<code>update-model</code>	Either updates the model files for Data Enrichment modules, or restores them to their original states.
<code>flush</code>	Flushes component caches.
<code>reshape-nodes</code>	Adds or removes Data Processing nodes from your BDD cluster.
<code>enable-components</code>	For use by Oracle Support, only. Enables components that are currently disabled.
<code>disable-components</code>	For use by Oracle Support, only. Disables components that are currently enabled.

## 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.
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

The `start` command starts components.



**Note:** `start` can't be run if `stop`, `restart`, `backup`, `restore`, `publish-config`, or `reshape-nodes` are currently running.

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

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

`start` supports the following options.

Option	Description
<code>-c, --component &lt;component(s)&gt;</code>	<p>A comma-separated list of the components to start:</p> <ul style="list-style-type: none"> <li><code>agent</code>: Dgraph HDFS Agent</li> <li><code>dgraph</code>: Dgraph</li> <li><code>dp</code>: Hive Table Detector cron job</li> <li><code>bddServer</code>: Studio and Dgraph Gateway</li> <li><code>transform</code>: Transform Service</li> <li><code>clustering</code>: Clustering Service (if enabled)</li> <li><code>wm</code>: Workflow Manager Service</li> </ul> <p>Note the following:</p> <ul style="list-style-type: none"> <li>Starting <code>bddServer</code> requires the WebLogic Server username and password if the <code>BDD_WLS_USERNAME</code> and <code>BDD_WLS_PASSWORD</code> environment variables aren't set.</li> <li><code>agent</code> can't be started if ZooKeeper isn't running.</li> </ul>
<code>-n, --node &lt;hostname(s)&gt;</code>	<p>A comma-separated list of the nodes to run on. Each must be defined in <code>\$BDD_HOME/BDD_manager/conf/bdd.conf</code>.</p>

If no options are specified, the script starts 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

The `stop` command stops components.



**Note:** Never use `SIGKILL`, `kill -9`, or any other OS command to stop BDD components. Always use `bdd-admin` with the `stop` command. If you need to stop a component immediately, run `stop` with `-t 0`.

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

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



**Note:** `stop` can't be run if `start`, `restart`, `backup`, `restore`, `publish-config`, or `reshape-nodes` is currently running.

`stop` supports the following options.

Option	Description
<code>-t, --timeout &lt;minutes&gt;</code>	<p>The amount of time to wait (in minutes) before terminating the component(s).</p> <p>If this value is 0, the script forces the component(s) to shut down immediately. If it's greater than 0, the script waits the specified amount of time for the component(s) to shut down gracefully, then terminates them if they don't.</p> <p>If this option isn't specified, the script shuts the component(s) down gracefully, which may take a very long time.</p>

Option	Description
-c, --component <component(s)>	A comma-separated list of the components to stop: <ul style="list-style-type: none"> <li>• agent: Dgraph HDFS Agent</li> <li>• dgraph: Dgraph</li> <li>• dp: Hive Table Detector cron job</li> <li>• bddServer: Studio and Dgraph Gateway</li> <li>• transform: Transform Service</li> <li>• clustering: Clustering Service (if enabled)</li> <li>• wm: Workflow Manager Service</li> </ul> Note that when <code>stop</code> runs on the <code>bddServer</code> component (or all components), it will prompt for the WebLogic Server username and password if the <code>BDD_WLS_USERNAME</code> and <code>BDD_WLS_PASSWORD</code> environment variables aren't set.
-n, --node <hostname(s)>	A comma-separated list of the nodes to run on. Each must be defined in <code>\$BDD_HOME/BDD_manager/conf/bdd.conf</code> .

If no options are specified, the script stops all supported components gracefully.

## Stopping Data Processing

Running `stop` on the `dp` and `wm` components has different effects on Data Processing:

- Running it on the `dp` component disables the Hive Table Detector cron job, if it's currently enabled. This prevents the Detector from launching new Data Processing jobs, but doesn't affect current jobs or prevent new ones from being run by the Workflow Manager Service.
- Running it on the `wm` component stops the Workflow Manager Service and cancels all currently-running Data Processing jobs. If a timeout value is included, the script waits for the specified amount of time before canceling current jobs; otherwise, it waits for them to complete normally before stopping the Workflow Manager Service, which may take a long time.

Be aware that no Data Processing jobs can run while the Workflow Manager Service is stopped. If you stop it but not the Hive Table Detector, the Detector will continue to run but will be ineffective since it submits all jobs to the Workflow Manager Service.

## 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 Workflow Manager Service to shut down gracefully, then terminates any that are still running:

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

## restart

The `restart` command restarts components regardless of whether they're currently running or stopped.



**Note:** `restart` can't be run if `start`, `stop`, `backup`, `restore`, `publish-config`, or `reshape-nodes` is currently running.

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

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

`restart` supports the following options.

Option	Description
<code>-t, --timeout &lt;minutes&gt;</code>	<p>The amount of time to wait (in minutes) before terminating the component(s).</p> <p>If this value is 0, the script forces the component(s) to shut down immediately. If it's greater than 0, the script waits the specified amount of time for the component(s) to shut down gracefully, then terminates them if they don't.</p> <p>If this option isn't specified, the script shuts the component(s) down gracefully, which may take a very long time. If a component is down, a timeout value should be specified or the script will hang.</p>
<code>-c, --component &lt;component(s)&gt;</code>	<p>A comma-separated list of the components to restart:</p> <ul style="list-style-type: none"> <li>• <code>agent</code>: Dgraph HDFS Agent</li> <li>• <code>dgraph</code>: Dgraph</li> <li>• <code>dp</code>: Hive Table Detector cron job</li> <li>• <code>bddServer</code>: Studio and Dgraph Gateway</li> <li>• <code>transform</code>: Transform Service</li> <li>• <code>clustering</code>: Clustering Service (if enabled)</li> <li>• <code>wm</code>: Workflow Manager Service</li> </ul> <p>Note the following:</p> <ul style="list-style-type: none"> <li>• Restarting <code>bddServer</code> requires the WebLogic Server username and password if the <code>BDD_WLS_USERNAME</code> and <code>BDD_WLS_PASSWORD</code> environment variables aren't set.</li> <li>• <code>agent</code> can't be restarted if ZooKeeper isn't running.</li> </ul>
<code>-n, --node &lt;hostname(s)&gt;</code>	<p>A comma-separated list of the nodes to run on. Each must be defined in <code>\$BDD_HOME/BDD_manager/conf/bdd.conf</code>.</p>

If no options are specified, the script restarts all supported components gracefully.

## Restarting Data Processing

Running `restart` on the `dp` and `wm` components has different effects on Data Processing:

- Running it on `dp` disables and reenables the Hive Table Detector cron job.
- Running it on `wm` cancels all current Data Processing jobs, and stops and restarts the Workflow Manager Service. If a timeout value is included, the script waits for the specified amount of time before canceling current jobs; otherwise, it waits for them to complete normally, which may take a long time.

Be aware that when the Workflow Manager Service is restarted, no new Data Processing jobs can run until it's up again.

## 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*

*reshape-nodes*

*enable-components*

*disable-components*

### autostart

The `autostart` command enables and disables autostart for components. Components that have autostart enabled 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>]
```

`autostart` requires one of the following operations.

Operation	Description
<code>on</code>	Enables autostart for the specified component(s).
<code>off</code>	Disables autostart for the specified component(s).
<code>status</code>	Returns the status of autostart for the specified component(s).

`autostart` also supports the following options.

Option	Description
<code>-c, --component &lt;component(s)&gt;</code>	A comma-separated list of the components to run on: <ul style="list-style-type: none"> <li><code>agent</code>: Dgraph HDFS Agent</li> <li><code>dgraph</code>: Dgraph</li> <li><code>bddServer</code>: Studio and Dgraph Gateway</li> <li><code>transform</code>: Transform Service</li> <li><code>clustering</code>: Clustering Service (if enabled)</li> <li><code>wm</code>: Workflow Manager Service</li> </ul>
<code>-n, --node &lt;hostname(s)&gt;</code>	A comma-separated list of the nodes to run on. Each must be defined in <code>\$BDD_HOME/BDD_manager/conf/bdd.conf</code> .

If no options are specified, the script runs on all supported components.

## Examples

The following command enables autostart for all supported components:

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

The following command returns the status of autostart for the HDFS Agent running on the `web009.us.example.com` node:

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

## backup

The `backup` command creates a backup of the cluster's data and metadata to a single TAR file that can later be used to restore it.



**Note:** `backup` can't be run if `start`, `stop`, `restart`, `restore`, `publish-config`, or `reshape-nodes` is currently running.



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

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

Where <file> is the absolute path to the backup TAR file. This must not exist and its parent directory must be writable.

backup supports 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 performs a hot backup.  For more information on hot and cold backups, see <a href="#">Hot vs. cold backups on page 27</a> .
-r, --repeat <num>	The number of times to repeat the backup process if verification fails. This is only used for hot backups.  If this option isn't specified, the script makes one attempt to back up the cluster. If it fails, the script must be rerun.  For more information, see <a href="#">Verification on page 27</a> .
-l, --local-tmp <path>	The absolute path to the temporary directory on the Admin Server used during the backup operation. If this option isn't specified, the location defined by BACKUP_LOCAL_TEMP_FOLDER_PATH in \$BDD_HOME/BDD_manager/conf/bdd.conf is used.
-d, --hdfs-tmp <path>	The absolute path to the temporary directory in HDFS used during the backup operation. If this option isn't specified, the location defined by BACKUP_HDFS_TEMP_FOLDER_PATH in \$BDD_HOME/BDD_manager/conf/bdd.conf is used.
-v, --verbose	Enables debugging messages.

If no options are specified, the script makes one attempt to perform a hot backup and doesn't output debugging messages.

For detailed instructions on backing up the cluster, see [Backing up BDD on page 67](#).

## Prerequisites

Before running backup, verify the following:

- You can provide the script with the usernames and passwords for all component databases. You can either enter this information at runtime or set the following environment variables. Note that if you have HDP, you must also provide the username and password for Ambari.
  - BDD\_STUDIO\_JDBC\_USERNAME: The username for the Studio database
  - BDD\_STUDIO\_JDBC\_PASSWORD: The password for the Studio database
  - BDD\_WORKFLOW\_MANAGER\_JDBC\_USERNAME: The username for the Workflow Manager Service database

- `BDD_WORKFLOW_MANAGER_JDBC_PASSWORD`: The password for the Workflow Manager Service database
- `BDD_HADOOP_UI_USERNAME`: The username for Ambari (HDP only)
- `BDD_HADOOP_UI_PASSWORD`: The password for Ambari (HDP only)
- You have an Oracle or MySQL database. Hypersonic isn't supported.
- 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. The Instant Client isn't supported.
- For Oracle databases, the `ORACLE_HOME` environment variable must be 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`.
- The temporary directories used during the backup operation contain enough free space. For more information, see [Space requirements on page 26](#) below.

## Backed-up data

The following data are included in the backup:

- The Dgraph databases
- The databases used by Studio and the Workflow Manager Service
- The user sandbox data in the directory defined by `SANDBOX_PATH` in `$BDD_HOME/BDD_manager/conf/bdd.conf`
- The HDFS sample data in `$SANDBOX_PATH/edp/data/.swampData`
- `$BDD_HOME/BDD_manager/conf/bdd.conf`
- The Hadoop server certificates (if TLS/SSL is enabled)
- Studio configuration from `portal-ext.properties` and `esconfig.properties`
- The DP CLI black- and white-lists (`cli_blacklist.txt` and `cli_whitelist.txt`)
- The OPSS files `cwallet.sso` and `system-jzn-data.xml`

Note that transient data, like state in Studio, is not backed up. This information will be lost if the cluster is restored.

## Space requirements

When the script runs, it verifies that the temporary directories it uses contain enough free space. These requirements only need to be met for the duration of the backup operation.

- The destination of the backup TAR file must contain enough space to store the Dgraph databases, `$HDFS_DP_USER_DIR`, and the `edpDataDir` (defined in `edp.properties`) at the same time.
- The `local-tmp` directory on the Admin Server also requires enough space to store all three items simultaneously.
- The `hdfs-tmp` directory in HDFS must contain enough free space to accommodate the largest of these items, as it will only store them one at a time.

If these requirements aren't met, the script will fail.

## Hot vs. cold backups

`backup` can perform both hot and cold backups:

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

## Verification

Because hot backups are performed while the cluster is running, it's possible for the data in the backups of the Studio and Dgraph databases and sample files to become inconsistent. For example, something could be added to a Dgraph database after the database was backed up, which would make the data in those locations different.

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, it can be configured 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 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

The `restore` command restores your BDD data and metadata from a backup TAR file created by the `backup` command.



**Note:** `restore` can't be run if `start`, `stop`, `restart`, `backup`, `publish-config`, or `reshape-nodes` is currently running.

To restore the backed up cluster, run the following from the Admin Server on the target cluster:

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

Where `<file>` is the absolute path to the backup file.

restore supports the following options.

Option	Description
-f, --full	Performs a full restoration, which restores all BDD data and configuration information. If this option isn't specified, the script will perform a data-only restoration, which doesn't include configuration. For more information, see <a href="#">Restoration types on page 29</a> below.
-l, --local-tmp <path>	The absolute path to the temporary directory on the Admin Server used during the restore operation. If this option isn't specified, BACKUP_LOCAL_TEMP_FOLDER_PATH will be used.  Before restoring, verify that this location contains enough free space. For more information, see <a href="#">Space requirements on page 29</a> below.
-d, --hdfs-tmp <path>	The absolute path to the temporary directory in HDFS used during the restore operation. If this option isn't specified, BACKUP_HDFS_TEMP_FOLDER_PATH will be used.  Before restoring, verify that this location contains enough free space. For more information, see <a href="#">Space requirements on page 29</a> below.
-v, --verbose	Enables debugging messages.

Note that `restore` makes a copy of the current Dgraph databases directory in `DGRAPH_INDEX_DIR/.snapshot/old_copy`, which should be deleted if the restored version is kept.

For detailed instructions on restoring your cluster, see [Performing a full BDD restoration on page 68](#) and [Restoring BDD to a new cluster on page 70](#).

## Prerequisites

Before running `restore`, verify the following:

- You have an existing backup TAR file created by the `backup` command.
- You can provide the script with the usernames and passwords for all component databases. You can either enter this information at runtime or set the following environment variables. Note that if you have HDP, you must also provide the username and password for Ambari.
  - `BDD_STUDIO_JDBC_USERNAME`: The username for the Studio database
  - `BDD_STUDIO_JDBC_PASSWORD`: The password for the Studio database
  - `BDD_WORKFLOW_MANAGER_JDBC_USERNAME`: The username for the Workflow Manager Service database
  - `BDD_WORKFLOW_MANAGER_JDBC_PASSWORD`: The password for the Workflow Manager Service database
  - `BDD_HADOOP_UI_USERNAME`: The username for Ambari (HDP only)
  - `BDD_HADOOP_UI_PASSWORD`: The password for Ambari (HDP only)
- Both the source and target clusters have the same minor version of BDD; for example, 1.4.0.37.xxxx.

- Both clusters have the same type of database, either Oracle or MySQL. Hypersonic isn't supported.
- The database client is installed on the Admin Server. For MySQL databases, this should be MySQL client. For Oracle databases, it should be Oracle Database Client, installed with a type of Administrator. The Instant Client isn't supported.
- For Oracle databases, the `ORACLE_HOME` environment variable must be 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`.
- For MySQL databases, the `lower_case_table_names` system variable has the same value on both clusters. If it doesn't, be sure to change it accordingly on the current cluster or the restoration will fail. For more information, see [Troubleshooting MySQL database restorations on page 72](#).
- The temporary directories used during the restore operation contain enough free space. For more information, see [Space requirements on page 29](#) below.

## Restoration types

`restore` supports two types of restoration: data-only and full.

**Data-only restorations** are performed by default. They restore the following to the target cluster:

- The Dgraph databases
- The databases used by Studio and the Workflow Manager Service
- The user sandbox data in the location defined by `HDFS_DP_USER_DIR` in `bdd.conf`
- The HDFS sample data in `$HDFS_DP_USER_DIR/edp/data/.collectionData`

Note that data-only restorations don't include any configuration information. Because of this, they can be performed on any BDD cluster that meets the criteria described in [Prerequisites on page 28](#). It can be different from the one that was originally backed up and can even have a different topology than the original. For example, you can restore data from an eight-node cluster to a new six-node one.

**Full restorations** restore the data listed above *and* the following configuration data:

- `$BDD_HOME/BDD_manager/conf/bdd.conf`
- The Hadoop TLS/SSL certificates (if TLS/SSL is enabled)
- Studio configuration from `portal-ext.properties` and `esconfig.properties`
- The DP CLI blacklist and whitelist (`cli_blacklist.txt` and `cli_whitelist.txt`)
- The OPSS files `cwallet.sso` and `system-jzn-data.xml`

Because full restorations include configuration information, they can only be performed on the original cluster that was backed up.

## Space requirements

When the script runs, it verifies that the temporary directories it uses contain enough free space. These requirements only need to be met for the duration of the restore operation.

- The `local-tmp` directory on the Admin Server must contain enough space to store the Dgraph databases, `$HDFS_DP_USER_DIR`, and the `edpDataDir` (defined in `edp.properties`) at the same time.

- The `hdfs-tmp` directory in HDFS must contain free space equal to the largest of these items, as it will only store them one at a time.

If these requirements aren't met, the script will fail.

## Examples

The following command performs a data-only restoration using the `/tmp/bdd_backup1.tar` file:

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

The following command performs a full restoration using the `/tmp/bdd_backup1.tar` file:

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

## publish-config

The `publish-config` command publishes configuration changes to your BDD cluster.



**Note:** `publish-config` can't be run if `start`, `stop`, `restart`, `backup`, `restore`, or `reshape-nodes` is currently running.

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

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

After `publish-config` runs, the cluster must be restarted for the changes to take effect.



**Important:** When making changes to your cluster's configuration, *always* use the post-install version of `bdd.conf` in `$BDD_HOME/BDD_manager/conf/`, and *not* the original one used during the installation. The post-install version contains a number of new properties added by the installer, which will be lost if you use the original one.

`publish-config` requires one of the following configuration types.

Configuration type	Description
<code>bdd &lt;path&gt;</code>	Publishes an updated version of <code>bdd.conf</code> specified by <code>&lt;path&gt;</code> to all BDD nodes. See <a href="#">bdd on page 31</a> for more information.
<code>hadoop [option &lt;arg&gt;]</code>	Publishes Hadoop configuration changes to all BDD nodes and performs any other operations defined by the specified options. See <a href="#">hadoop on page 31</a> for more information.
<code>kerberos [operation] &lt;option &lt;arg&gt;&gt;</code>	Publishes the specified Kerberos principal, <code>krb5.conf</code> file, or keytab file to all BDD nodes. See <a href="#">kerberos on page 32</a> for more information.
<code>cert</code>	Refreshes the certificates on BDD clusters secured with TLS/SSL. See <a href="#">cert on page 33</a> for more information.
<code>database</code>	Updates a component's database settings, including its username, password, and JDBC URL. See <a href="#">database on page 34</a> for more information.

## bdd

The `bdd` configuration type publishes a modified version of `bdd.conf` to all BDD nodes. This updates the configuration of the entire cluster.

To update the cluster configuration, edit a *copy* of `$BDD_HOME/BDD_manager/conf/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`. Note that it's recommended to edit a *copy* of `bdd.conf` to preserve the original in case the changes need to be reverted.



**Important:** When making changes to your cluster's configuration, *always* use the post-install version of `bdd.conf` in `$BDD_HOME/BDD_manager/conf/`, and *not* the original one used during the installation. The post-install version contains a number of new properties added by the installer, which will be lost if you use the original one.

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`. This file can be used to revert the configuration changes, if necessary.

The script then copies the modified version of `bdd.conf` to all BDD nodes in the cluster. When it completes, the cluster must be restarted for the changes to take affect.



**Note:** When `bdd` runs, any component log levels you've set on specific nodes using the `set-log-levels` command will be overwritten by the `DGRAPH_LOG_LEVELS` and `ENDECA_SERVER_LOG_LEVEL` properties in the updated file.

For more information on updating your cluster configuration, see [Updating bdd.conf on page 49](#).

## hadoop

The `hadoop` configuration type makes changes to BDD's Hadoop configuration.

Depending on the specified options, `hadoop` can:

- Publish new or updated Hadoop client configuration files to your BDD cluster.
- Reset the `HUE_URI` property in `$BDD_HOME/BDD_manager/conf/bdd.conf` (HDP only).
- Switch to a different version of your Hadoop distribution without reinstalling BDD. Note, however, that it can't be used to switch to a different Hadoop distribution.



**Note:** The script requires the username and password for your Hadoop cluster manager if the `BDD_HADOOP_UI_USERNAME` and `BDD_HADOOP_UI_PASSWORD` environment variables aren't set.

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

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

hadoop supports the following options.

Option	Description
<code>-u, --hueuri &lt;host:port&gt;</code>	HDP clusters only. Sets the <code>HUE_URI</code> property in <code>bdd.conf</code> to the specified URI.
<code>-l, --clientlibs &lt;path[,path]&gt;</code>	Regenerates the Hadoop fat jar from a comma-separated list of client libraries. <code>&lt;path[,path]&gt;</code> must be a comma-separated list of the new libraries. This can be used to switch to a different version of your Hadoop distribution.  This must be run with <code>--sparkjar</code> .
<code>-j, --sparkjar &lt;file&gt;</code>	Sets the location of the Spark on YARN jar in all BDD configuration files to the specified path. <code>&lt;file&gt;</code> must be the absolute path to the Spark on YARN jar on the Hadoop nodes. This can be used to switch to a different version of your Hadoop distribution.  Note that unless the location of your Hadoop installation has changed, you can use the value of <code>SPARK_ON_YARN_JAR</code> in <code>bdd.conf</code> . Be sure to double-check the path, just in case.  This must be run with <code>--clientlibs</code> .

If no options are specified, 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 54](#)
- [Upgrading Hadoop on page 55](#)

## kerberos

The `kerberos` configuration type updates to BDD's Kerberos configuration.

Depending on the specified options, `kerberos` can do the following:

- Enable Kerberos
- Update the location of `krb5.conf` 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>
```



kerberos requires one of the following operations.

Operation	Description
on	Enables Kerberos. The <code>-k</code> , <code>-t</code> , and <code>-p</code> options must also be specified.
config	Updates BDD's Kerberos configuration. At least one option must be specified.  This is the command's default behavior, so this operation is optional. You can only use this if Kerberos is already enabled.

kerberos supports the following options.

Option	Description
<code>-k, --krb5 &lt;file&gt;</code>	Updates the location of <code>krb5.conf</code> in all BDD configuration files. <code>&lt;file&gt;</code> must be the new absolute path to the file.  <code>krb5.conf</code> must be moved to its new location on all BDD nodes before running this option.
<code>-t, --keytab &lt;file&gt;</code>	Publishes the specified keytab file to all BDD nodes. <code>&lt;path&gt;</code> must be the absolute path to the new keytab file.  The script renames this file <code>bdd.keytab</code> and copies it to <code>\$BDD_HOME/common/kerberos</code> .
<code>-p, --principal &lt;principal&gt;</code>	Publishes the specified principal to all BDD nodes. This option can't be used 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](#).

## cert

The `cert` configuration type refreshes BDD's TLS/SSL certificates for the HDFS, YARN, Hive, and KMS services.

Before running this command, you must export the updated certificates from your Hadoop nodes and copy them to the directory on the Admin Server defined by `HADOOP_CERTIFICATES_PATH` in `$BDD_HOME/BDD_manager/conf/bdd.conf`.

To refresh the certificates, run:

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

When the script runs, it imports the certificates to the custom truststore file, then copies the truststore to `$BDD_HOME/common/security/cacerts` on all BDD nodes.

For more information on refreshing your certificates, see [Refreshing TLS/SSL certificates on page 61](#).

## database

The `database` configuration type updates a component's database configuration, including its username, password, and JDBC URL.

To update a component's database configuration, run the following from the Admin Server:

```
./bdd-admin.sh publish-config database <component> [option]
```

`database` requires one of the following components.

Component	Description
wm	Workflow Manager Service

`database` supports the following options.

Option	Description
<code>-u, --username &lt;value&gt;</code>	Sets the database username to the specified value.
<code>-p, --password [value]</code>	Sets the database password to the specified value. If no value is provided, the script will prompt for one.
<code>-j, --jdbc &lt;value&gt;</code>	Sets the database JDBC URL to the specified value.

Note that you can specify multiple options at once. For example, the following command updates the Workflow Manager Service's database username and password:

```
./bdd-admin.sh publish-config database wm -u workflow -p
Enter password:
[2016/08/30 05:26:12 -0400] [Admin Server] Updating database settings...
[2016/08/30 05:26:12 -0400] [Admin Server] Refreshing settings in file...Success!
[2016/08/30 05:26:17 -0400] [Admin Server] Distributing updated file to all nodes...Success!
[2016/08/30 05:26:21 -0400] [Admin Server] Successfully updated database settings.
```

For more information on updating component database configuration, see [Updating component database configuration on page 60](#).

## update-model

The `update-model` command updates or resets the models used by some of the Data Enrichment modules.

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]
```

`update-model` requires one of the following model types.

Model type	Description
geonames	The model for the GeoTagger Data Enrichment modules.

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

[path] is the absolute path to the location of the files to update the model with. This argument is optional. You must move these files to a single directory on the Admin Server before running the script.

If [path] is included, the script creates a jar from the files in the specified directory, then replaces the current jar on the YARN worker nodes with the new one. If [path] isn't included, the script resets the specified model to its original state.

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

## Examples

The following command updates the Sentiment Analysis model using the contents of the /share/model/sentiment directory:

```
./bdd-admin.sh update-model sentiment /share/models/sentiment
```

The following command resets the tfidf model:

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

## flush

The flush command flushes component caches.

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

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

flush supports the following options.

Option	Description
-c, --component <component(s)>	A comma-separated list of the component caches to flush: <ul style="list-style-type: none"> <li>dgraph: Dgraph</li> <li>gateway: Dgraph Gateway</li> </ul> When debugging query issues, cold-start or post-update performance can be approximated by cleaning the Dgraph cache before running a request.
-n, --node <hostname(s)>	A comma-separated list of the nodes to run on. Each must be defined in \$BDD_HOME/BDD_manager/conf/bdd.conf.

If no options are specified, the script flushes 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
```

## reshape-nodes

The `reshape-nodes` command adds and removes Data Processing nodes from your BDD cluster.



**Note:** `reshape-nodes` can't be run if `start`, `stop`, `restart`, `backup`, `restore`, or `publish-config` is currently running.

When the script runs, it queries your Hadoop cluster manager (Cloudera Manager, Ambari, or MCS) for the list of YARN NodeManager nodes that support Data Processing, determines whether any have been added or removed, and updates your BDD cluster accordingly. For example, if you add a qualified YARN NodeManager, the script automatically installs Data Processing on it.

To add or remove Data Processing nodes from your cluster, run the following from the Admin Server:

```
./bdd-admin.sh reshape-nodes
```

`reshape-nodes` doesn't support any options.

For more information on reshaping your cluster, see [Adding and Removing BDD Nodes on page 62](#).

## enable-components

The `enable-components` command enables components that are currently disabled. Note that this command can only be used for certain components.



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

To enable a component, run the following from the Admin Server:

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

`enable-components` supports the following options.

Option	Description
<code>-c, --component &lt;component(s)&gt;</code>	A comma-separated list of the component(s) to enable: <ul style="list-style-type: none"> <li><code>clustering</code></li> </ul>

If no option is specified, the script enables all supported components.

When the script runs, it enables the specified component(s) by updating the relevant properties in `$BDD_HOME/BDD_manager/conf/bdd.conf`, then starts them. They can then be controlled with other `bdd-admin` commands like `start` and `stop`.

Components enabled by the `enable-components` command can later be disabled by the `disable-components` command. For more information, see [disable-components on page 37](#).

## Examples

The following command enables the Clustering Service:

```
./bdd-admin.sh enable-components -c clustering
```

## disable-components

The `disable-components` command disables specific components that are currently enabled. Note that this can only be used on components that were enabled by the `enable-components` command.



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

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

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

`disable-components` supports the following options.

Option	Description
<code>-c, --component &lt;component(s)&gt;</code>	A comma-separated list of the component(s) to disable: <ul style="list-style-type: none"> <li><code>clustering</code></li> </ul>

If no option is specified, the script disables all supported components.

When the script runs, it stops the specified component(s), then disables them by updating the relevant properties in `$BDD_HOME/BDD_manager/conf/bdd.conf`.

Components disabled by the `disable-components` command can later be re-enabled by the `enable-components` command. For more information, see [enable-components on page 36](#).

## Examples

The following command disables the Clustering Service:

```
./bdd-admin.sh disable-components -c clustering
```

## 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](#)

[status](#)

[get-stats](#)

[reset-stats](#)[get-log-levels](#)[set-log-levels](#)[get-logs](#)[rotate-logs](#)

## get-blackbox

The `get-blackbox` command generates the Dgraph's on-demand tracing blackbox file and 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>]
```

`get-blackbox` supports the following options.

Option	Description
<code>-n, --node &lt;hostname(s)&gt;</code>	A comma-separated list of the nodes the script will run on. Each must be defined in <code>\$BDD_HOME/BDD_manager/conf/bdd.conf</code> .

If no options are specified, the script generates 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

The `status` command checks component statuses and the overall health of the BDD cluster.

`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>]
```

status supports the following options.

Option	Description
-c, --component <component(s)>	A comma-separated list of the components to run on: <ul style="list-style-type: none"> <li>agent: Dgraph HDFS Agent</li> <li>dgraph: Dgraph</li> <li>dp: Hive Table Detector</li> <li>gateway: Dgraph Gateway</li> <li>studio: Studio</li> <li>transform: Transform Service</li> <li>clustering: Clustering Service (if enabled)</li> <li>wm: Workflow Manager Service</li> </ul>
-n, --node <hostname(s)>	A comma-separated list of the nodes to run on. Each must be defined in \$BDD_HOME/BDD_manager/conf/bdd.conf.
--health-check	Returns the health of the cluster and the Hive Table Detector. When specified, the -c or -n options can't be included.  If the healthcheck fails, information on what went wrong can be found in the Studio and Data Processing logs.

If no options are specified, the script returns 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

The get-stats command obtains Dgraph statistics.



**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>
```

Where <dest> is the absolute path to the directory the script will output the requested statistics to. When the script completes, this location will contain a file named <hostname>-<timestamp>-dgraph-stats.xml.

get-stats supports the following options.

Option	Description
-c, --component <component(s)>	A comma-separated list of the components to run on: <ul style="list-style-type: none"> <li>dgraph: Dgraph</li> </ul>
-n, --node <hostname(s)>	A comma-separated list of the nodes to run on. Each must be defined in \$BDD_HOME/BDD_manager/conf/bdd.conf.

If no options are specified, the script obtains the statistics for all Dgraph instances in the cluster.

For more information on Dgraph statistics, see [Dgraph statistics on page 82](#).

## 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

The reset-stats command resets the Dgraph statistics.



**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>]
```

reset-stats supports the following options.

Option	Description
-c, --component <component(s)>	A comma-separated list of the components to run on: <ul style="list-style-type: none"> <li>dgraph: Dgraph</li> </ul>
-n, --node <hostname(s)>	A comma-separated list of the nodes to run on. Each must be defined in \$BDD_HOME/BDD_manager/conf/bdd.conf.



If no options are specified, the script resets the statistics for all Dgraph instances in the cluster.

For more information on Dgraph statistics, see [Dgraph statistics on page 82](#).

## 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

The `get-log-levels` command returns the list of component logs and their current levels.

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

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

`get-log-levels` supports the following options.

Option	Description
<code>-c, --component &lt;component(s)&gt;</code>	<p>A comma-separated list of the components to run on:</p> <ul style="list-style-type: none"> <li><code>dgraph</code>: Dgraph</li> <li><code>dp</code>: DP CLI</li> <li><code>gateway</code>: Dgraph Gateway</li> </ul> <p>Note the following:</p> <ul style="list-style-type: none"> <li>The <code>dgraph</code> component returns the current levels of all Dgraph out log subsystems. For more information, see <a href="#">Dgraph out log on page 167</a>.</li> <li>The <code>dp</code> component returns the current log levels for the DP CLI, and not the Workflow Manager Service or Data Processing. For information on obtaining the log levels for these components, see the <i>Data Processing Guide</i>.</li> </ul>
<code>-n, --node &lt;hostname(s)&gt;</code>	A comma-separated list of the nodes to run on. Each must be defined in <code>\$BDD_HOME/BDD_manager/conf/bdd.conf</code> .

If no options are specified, the script returns 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
  database                 : ERROR
  datalayer               : ERROR
  dgraph                  : ERROR
  eql                     : ERROR
  eql_feature             : TRACE
  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


The `set-log-levels` command sets component log levels and updates their configuration files so that the changes persist when the components are restarted.

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

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

`set-log-levels` supports the following options.

Option	Description
<code>-c, --component &lt;component(s)&gt;</code>	<p>A comma-separated list of the components to run on:</p> <ul style="list-style-type: none"> <li><code>dgraph</code>: Dgraph</li> <li><code>dp</code>: DP CLI</li> <li><code>gateway</code>: Dgraph Gateway</li> </ul> <p>Note that the <code>dp</code> component sets the log level for the DP CLI, and not the Workflow Manager Service or Data Processing. For information on settings log levels for these components, see the <i>Data Processing Guide</i>.</p>

Option	Description
<p><code>-s, --subsystem</code>  <code>&lt;subsystem(s)&gt;</code></p>	<p>A comma-separated list of the Dgraph out log subsystems to run on:</p> <ul style="list-style-type: none"> <li>• background_merging</li> <li>• bulk_ingest</li> <li>• cluster</li> <li>• database</li> <li>• datalayer</li> <li>• dgraph (Note that this is different from the <code>dgraph</code> component.)</li> <li>• eql</li> <li>• eql_feature</li> <li>• eve</li> <li>• hdfs_fuse (Note that this isn't currently used.)</li> <li>• http</li> <li>• lexer</li> <li>• splitting</li> <li>• ssl</li> <li>• task_scheduler</li> <li>• text_search_rel_rank</li> <li>• text_search_spelling</li> <li>• update</li> <li>• workload_manager</li> <li>• ws_request</li> <li>• xq_web_service</li> </ul> <p>This option can only be specified when running on the <code>dgraph</code> component. If the script runs on the <code>dgraph</code> component and this option isn't specified, it runs on all supported subsystems.</p> <p> <b>Note:</b> When setting the levels of Dgraph log subsystems, the script also updates the <code>DGRAPH_LOG_LEVELS</code> property in <code>\$BDD_HOME/BDD_manager/conf/bdd.conf</code> accordingly. When setting log levels on specific nodes, it only updates <code>bdd.conf</code> on those nodes. These settings will be overwritten if the <code>publish-config</code> command is run.</p> <p>For more information on the Dgraph out log and its subsystems, see <a href="#">Dgraph out log on page 167</a>.</p>

Option	Description
-l, --level <level>	<p>The log level to set for the components:</p> <ul style="list-style-type: none"> <li>• INCIDENT_ERROR</li> <li>• ERROR</li> <li>• WARNING</li> <li>• NOTIFICATION</li> <li>• NOTIFICATION:16 (Dgraph only)</li> <li>• TRACE</li> <li>• TRACE:16 (Dgraph only)</li> <li>• TRACE:32 (Dgraph only)</li> </ul> <p>Only one log level can be specified. If this option is omitted, the script sets all specified logs to NOTIFICATION.</p> <p>Note that the NOTIFICATION:16, TRACE:16, and TRACE:32 log levels are only supported by the dgraph component.</p>
--non-persistent	<p>Indicates that the log levels should be reset when the components are restarted. When specified, the script doesn't update the component configuration files.</p> <p>This option is only available for the dgraph and gateway components. Data Processing log levels are always persistent.</p>
-n, --node <hostname(s)>	<p>A comma-separated list of the nodes to run on. Each must be defined in \$BDD_HOME/BDD_manager/conf/bdd.conf.</p>

If no options are specified, the script sets 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 the Dgraph log subsystems cluster and datalayer to WARNING:

```
./bdd-admin.sh set-log-levels -c dgraph -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-persistent
```

## get-logs

The get-logs command collects requested log files 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>
```

Where <file> defines the absolute path to the output zip file. This file must not exist and must include the .zip file extension.

get-logs supports the following options.

Option	Description
-t, --time <hours>	When specified, the script returns the logs that were modified within the last <hours> hours.  If this option is omitted, the script returns the most recently updated log file for each component.

Option	Description
<p><code>-c, --component &lt;component(s)&gt;</code></p>	<p>A comma-separated list of the component logs to collect:</p> <ul style="list-style-type: none"> <li>• <code>agent</code>: Dgraph HDFS Agent logs</li> <li>• <code>all</code>: All component logs</li> <li>• <code>clustering</code>: Clustering Service (if enabled)</li> <li>• <code>dgraph</code>: Dgraph logs</li> <li>• <code>dg-on-crash</code>: Dgraph on-crash tracing logs</li> <li>• <code>dg-on-demand</code>: Dgraph on-demand tracing logs</li> <li>• <code>dp</code>: DP CLI logs</li> <li>• <code>gateway</code>: Dgraph Gateway logs</li> <li>• <code>spark</code>: Spark logs</li> <li>• <code>studio</code>: Studio logs</li> <li>• <code>transform</code>: Transform Service</li> <li>• <code>weblogic</code>: WebLogic Server logs</li> <li>• <code>wm</code>: Workflow Manager Service</li> <li>• <code>zk-log</code>: ZooKeeper logs</li> <li>• <code>zk-transaction</code>: ZooKeeper transaction logs</li> </ul> <p>Note the following:</p> <ul style="list-style-type: none"> <li>• The <code>spark</code>, <code>zk-log</code>, and <code>zk-transaction</code> components will prompt for the username and password for Cloudera Manager/Ambari/MCS if the <code>BDD_HADOOP_UI_USERNAME</code> and <code>BDD_HADOOP_UI_PASSWORD</code> environment variables aren't set.</li> <li>• The <code>dg-on-demand</code> log is only generated when the <code>get-blackbox</code> command is run. This means that if the <code>-t</code> option is specified, <code>get-logs</code> only returns the <code>dg-on-demand</code> log if <code>get-blackbox</code> was run during the specified time frame. And if the <code>-t</code> option is omitted, <code>get-logs</code> won't return the <code>dg-on-demand</code> log if <code>get-blackbox</code> has never been run.</li> <li>• The <code>dp</code> component returns the logs for the DP CLI, and not Data Processing. For information on obtaining the Data Processing logs, see the <i>Data Processing Guide</i>.</li> </ul>
<p><code>-n, --node &lt;hostname(s)&gt;</code></p>	<p>A comma-separated list of the nodes to run on. Each must be defined in <code>\$BDD_HOME/BDD_manager/conf/bdd.conf</code>.</p>

If no options are specified, the script obtains the most recently updated logs for all components except `dg-on-crash`, `dg-on-demand`, and `zk-transaction`.

## 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

The `rotate-logs` command rotates component logs.



**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>]
```

`rotate-logs` supports the following options.

Option	Description
<code>-c, --component &lt;component(s)&gt;</code>	A comma-separated list of the component logs to rotate: <ul style="list-style-type: none"> <li>agent: Dgraph HDFS Agent logs</li> <li>dgraph: Dgraph logs</li> <li>gateway: Dgraph Gateway logs</li> <li>studio: Studio logs</li> <li>transform: Transform Service</li> <li>weblogic: WebLogic Server logs</li> <li>clustering: Clustering Service (if enabled)</li> <li>wm: Workflow Manager Service</li> </ul>
<code>-n, --node &lt;hostname(s)&gt;</code>	A comma-separated list of the nodes to run on. Each must be defined in <code>\$BDD_HOME/BDD_manager/conf/bdd.conf</code> .

If no options are specified, the script rotates 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
```





The following sections describe how to update BDD's configuration.

[Updating bdd.conf](#)

[Updating BDD's Hadoop configuration](#)

[Updating BDD's Kerberos configuration](#)

[Updating component database configuration](#)

[Refreshing TLS/SSL certificates](#)

## Updating bdd.conf

You can update BDD's configuration by editing `bdd.conf`, then running the `bdd-admin` script to distribute your changes to the rest of the cluster.



**Important:** When making changes to your cluster's configuration, *always* use the post-install version of `bdd.conf` in `$BDD_HOME/BDD_manager/conf/`, and *not* the original one used during the installation. The post-install version contains a number of new properties added by the installer, which will be lost if you use the original one.

Be aware that it's not recommended to modify all properties in `bdd.conf`. In particular, you should avoid changing properties related to cluster topology, like `<COMPONENT>_PORT` and `<COMPONENT>_SERVERS`. Additionally, some Studio and Data Processing settings should be configured through either Studio or the Data Processing configuration files. For more information, see [Configuring Studio Settings on page 97](#), [Configuring Data Processing Settings on page 101](#), and the [Data Processing Guide](#).

Also note that 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` and `ENDECA_SERVER_LOG_LEVEL` properties in the updated file.

When the script runs, it backs up the original version of `$BDD_HOME/BDD_manager/conf/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_HOME/BDD_manager/conf/bdd.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`.

- Restart your cluster so the changes take effect:

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

### Configuration properties that can be modified

## Configuration properties that can be modified

The table below describes the properties in `$BDD_HOME/BDD_manager/conf/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
AGENT_OUT_FILE	The path to the HDFS Agent's stdout/stderr file.
BACKUP_HDFS_TEMP_FOLDER_PATH	The absolute path to the default temporary folder on the Admin Server used during backup and restore operations. Note that this can also be overridden on a case-by-case basis by the <code>bdd-admin</code> script.
BACKUP_LOCAL_TEMP_FOLDER_PATH	The absolute path to the default temporary folder on HDFS used during backup and restore operations. Note that this can also be overridden on a case-by-case basis by the <code>bdd-admin</code> script.
CLUSTERING_SERVICE_*	These properties can't be modified directly. For information on enabling and disabling the Clustering Service, see <a href="#">enable-components on page 36</a> and <a href="#">disable-components on page 37</a> .
COMPANY_SECURITY_*	These properties can't be modified directly. For information on updating user authentication settings, see <a href="#">Configuring authentication settings for users on page 124</a> .
DETECTOR_*	Hive Table Detector properties shouldn't be modified directly, although you can change the Detector's behavior by editing its cron job. For more information, see the <i>Data Processing Guide</i> .
DGRAPH_ADDITIONAL_ARG	Defines one or more flags to start the Dgraph with. This property is only intended for use by Oracle Support.  Note that you cannot include flags that map to properties in <code>bdd.conf</code> . For more information on Dgraph flags, see <a href="#">Dgraph flags on page 87</a> .

Property	Description
DGRAPH_CACHE	The Dgraph cache size, in MB. 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_CGROUP_NAME, DGRAPH_ENABLE_CGROUP	Can be edited in certain circumstances. For instructions, see <a href="#">Setting up cgroups for the Dgraph on page 85</a> .
DGRAPH_HDFS_MOUNT_DIR	Can be edited in certain circumstances. For instructions, see <a href="#">Moving the Dgraph databases to HDFS on page 77</a> .
DGRAPH_INDEX_DIR	The path to the Dgraph databases directory. You must prepare the database files in the new location before changing the value of this property. See <a href="#">Moving the Dgraph databases to HDFS on page 77</a> .
DGRAPH_LOG_LEVEL	<p>Optional. Defines the log levels for the Dgraph's out log subsystems. This must be formatted as:</p> <pre>subsystem1 level1 subsystem2,subsystem3 level2 subsystemN levelN</pre> <p>For example:</p> <pre>DGRAPH_LOG_LEVEL =bulk_ingest WARNING cluster ERROR dgraph, eql, eve INCIDENT_ERROR</pre> <p>You can include as many subsystems as you want. Any you don't include will be set to NOTIFICATION. If you enter an unsupported or improperly formatted value, it will default to NOTIFICATION.</p> <p>For more information on the Dgraph's out log subsystems and their supported levels, see <a href="#">Dgraph out log on page 167</a>.</p>
DGRAPH_OUT_FILE	The path to the Dgraph's stdout/stderr file.
DGRAPH_SERVERS	Can be edited in certain circumstances. For instructions, see <a href="#">Adding new Dgraph nodes on page 63</a> and <a href="#">Moving the Dgraph databases to HDFS on page 77</a> .

Property	Description
DGRAPH_THREADS	<p>The number of threads the Dgraph starts with. Oracle recommends the following:</p> <ul style="list-style-type: none"> <li>For machines running only the Dgraph, the number of threads should be equal to the number of CPU cores on the machine.</li> <li>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.</li> </ul> <p>Be sure that the number you use is in compliance with the licensing agreement.</p>
DGRAPH_USE_MOUNT_HDFS	Can be edited in certain circumstances. For instructions, see <a href="#">Moving the Dgraph databases to HDFS on page 77</a> .
DGRAPH_USE_NFS_MOUNT	Can be edited in certain circumstances. For instructions, see <a href="#">Moving the Dgraph databases to HDFS on page 77</a> .
DP_ADDITIONAL_JARS	Defines custom SerDe JARs to be added to Data Processing workflows. For more information, see the <i>Data Processing Guide</i> .
ENABLE_AUTOSTART	Shouldn't be modified directly. For information on enabling autostart for components, see <a href="#">autostart on page 23</a> .
ENABLE_CLUSTERING_SERVICE	These properties can't be modified directly. For information on enabling and disabling the Clustering Service, see <a href="#">enable-components on page 36</a> and <a href="#">disable-components on page 37</a> .
ENABLE_ENRICHMENTS	See the <i>Data Processing Guide</i> .
ENABLE_HIVE_TABLE_DETECTOR	Shouldn't be modified after install time. For information on modifying the Hive Table Detector's behavior, see the <i>Data Processing Guide</i> .
ENABLE_KERBEROS	Can be edited in certain circumstances. For instructions, see <a href="#">Enabling Kerberos on page 57</a> .
ENDECA_SERVER_LOG_LEVEL	For instructions on setting Dgraph Gateway log levels, see <a href="#">Setting the Dgraph Gateway log level on page 181</a> .
HADOOP_CERTIFICATES_PATH	Shouldn't be modified directly. See <a href="#">Refreshing TLS/SSL certificates on page 61</a> .
HADOOP_CLIENT_LIB_PATHS	Shouldn't be modified directly. See <a href="#">Updating the Hadoop client configuration files on page 54</a> .
HIVE_DATABASE_NAME	Can't be modified directly. If you want to change the database used by the Hive Table Detector, you must modify its cron job. For more information, see the <i>Data Processing Guide</i> .

Property	Description
HUE_URI	Shouldn't be modified directly. See <a href="#">Setting the Hue URI on page 54</a> .
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.
KERBEROS_KEYTAB_PATH	Shouldn't be modified directly. See <a href="#">Updating the Kerberos keytab file on page 59</a> .
KERBEROS_PRINCIPAL	Shouldn't be modified directly. See <a href="#">Updating the Kerberos principal on page 60</a> .
KERBEROS_TICKET_LIFETIME	The amount of time that the Dgraph's Kerberos ticket is valid. This should be given as a number followed by a supported unit of time: s, m, h, or d. For example, 10h (10 hours), or 10m (10 minutes).
KERBEROS_TICKET_REFRESH_INTERVAL	The interval (in minutes) at which the Dgraph's Kerberos ticket is refreshed. For example, if set to 60, the Dgraph's ticket would be refreshed every 60 minutes, or every hour.
KRB5_CONF_PATH	Shouldn't be modified directly. See <a href="#">Changing the location of the Kerberos krb5.conf file on page 59</a> .
LANGUAGE	For information on changing the language used by Data Processing, see the <i>Data Processing Guide</i> .
LDAP_*	For instructions on configuring an LDAP for Studio, see <a href="#">Integrating with an LDAP System to Manage Users on page 133</a> .
MAX_INPUT_SPLIT_SIZE	See the <i>Data Processing Guide</i> .
MAX_RECORDS	See the <i>Data Processing Guide</i> .
RECORD_SEARCH_THRESHOLD	See the <i>Data Processing Guide</i> .
STUDIO_ADMIN_*	Shouldn't be modified directly. See <a href="#">Editing a Studio user on page 132</a> .
STUDIO_JDBC_CACHE	For instructions on enabling and disabling database caching for Studio, see the <i>Installation Guide</i> .
VALUE_SEARCH_THRESHOLD	See the <i>Data Processing Guide</i> .

## 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](#)

[Upgrading Hadoop](#)

### 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/MCS, then updates the following:

- All Hadoop properties in `$BDD_HOME/BDD_manager/conf/bdd.conf`
- The following properties in Studio's `portal-ext.properties` file:
  - `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 [-t <minutes>]
```

### Setting the Hue URI

If you have HDP, you can use the `bdd-admin` script to update the URI of the node running Hue in `$BDD_HOME/BDD_manager/conf/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 [-t <minutes>]
```

## Upgrading Hadoop

If you want to 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 could upgrade from CDH 5.4 to CDH 5.5, but not to HDP 2.3.

To upgrade Hadoop:

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`

If you have MapR, you may need to reinstall and reconfigure the MapR Client if a different version needs to be used with the new version of MapR. The MapR Client must be installed and added to the `$PATH` on all Dgraph, Studio, and Transform Service nodes that aren't part of your MapR cluster. For instructions on installing the Client, see [Installing the MapR Client](#) in MapR's documentation.

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 CDH, 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 HDP, 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/`
- If you have MapR, copy the following directories from your Hadoop nodes to the Admin Server:
  - `/opt/mapr/spark/spark-<version>/lib`
  - `/opt/mapr/hive/hive-<version>/lib`
  - `/opt/mapr/zookeeper/zookeeper-<version>`
  - `/opt/mapr/zookeeper/zookeeper-<version>/lib`
  - `/opt/mapr/hadoop/hadoop-<version>/share/hadoop/common`
  - `/opt/mapr/hadoop/hadoop-<version>/share/hadoop/common/lib`
  - `/opt/mapr/hadoop/hadoop-<version>/share/hadoop/hdfs`
  - `/opt/mapr/hadoop/hadoop-<version>/share/hadoop/hdfs/lib`
  - `/opt/mapr/hadoop/hadoop-<version>/share/hadoop/mapreduce`
  - `/opt/mapr/hadoop/hadoop-<version>/share/hadoop/mapreduce/lib`
  - `/opt/mapr/hadoop/hadoop-<version>/share/hadoop/tools/lib`



- /opt/mapr/hadoop/hadoop-**<version>**/share/hadoop/yarn
- /opt/mapr/hadoop/hadoop-**<version>**/share/hadoop/yarn/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>
```

Where:

- **<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. Unless the location of your Hadoop installation has changed, you can use the value of `SPARK_ON_YARN_JAR` in `$BDD_HOME/BDD_manager/conf/bdd.conf`. Be sure to double-check the path, just in case.

7. Restart your cluster so the changes take effect:

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

## 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>`, where `<bdd>` is the name of the `bdd` user.

- `/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 `bdd` user. Their group must be the HDFS super users group, which is defined by the `dfs.permissions.supergroup` configuration parameter. The default value is `supergroup`.

3. Add the `bdd` user 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 `bdd` user 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. If your Dgraph databases are stored on HDFS, you must also enable Kerberos for the Dgraph. On the Admin Server, make a copy of `$BDD_HOME/BDD_manager/conf/bdd.conf` and edit the following properties in the copy:

Property	Description
<code>KERBEROS_TICKET_REFRESH_INTERVAL</code>	The interval (in minutes) at which the Dgraph's Kerberos ticket is refreshed. For example, if set to 60, it would be refreshed every 60 minutes, or every hour.
<code>KERBEROS_TICKET_LIFETIME</code>	The amount of time that the Dgraph's Kerberos ticket is valid. This should be given as a number followed by a supported unit of time: <code>s</code> , <code>m</code> , <code>h</code> , or <code>d</code> . For example, <code>10h</code> (10 hours), or <code>10m</code> (10 minutes).

Then go to `$BDD_HOME/BDD_manager/bin` and run:

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

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

9. 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.

10. If you use HDP, publish the change you made to `core-site.xml`:

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

11. Restart your cluster for the changes to take effect:

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

Once Kerberos is enabled, you can use the `bdd-admin` script to update its configuration as needed. For more information, see [kerberos on page 32](#).

## 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 32](#).

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 [-t <minutes>]
```

## 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 32](#).

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 [-t <minutes>]
```

## 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 name of the principal in BDD's configuration files. Note that 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 32](#).

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 [-t <minutes>]
```

## Updating component database configuration

You can use the `bdd-admin` script to update the configuration for the Workflow Manager Service database, including its username, password, and JDBC URL.

The following sections describe how to update each of these values. Additional information is available in [database on page 34](#).

### Updating the database username

To update the username for the Workflow Manager Service database, go to `$BDD_HOME/BDD_manager/bin` on the Admin Server and run:

```
./bdd-admin.sh publish-config database wm -u <value>
```

Where `<value>` is the new username.

## Updating the database password

To update the password for the Workflow Manager Service database, go to `$BDD_HOME/BDD_manager/bin` on the Admin Server and run:

```
./bdd-admin.sh publish-config database wm -p [value]
```

Where `[value]` is the new password.

If you don't want the new password to be visible, you can omit `[value]` and enter it when prompted:

```
./bdd-admin.sh publish-config database wm -p
Enter password:
```

## Updating the database JDBC URL

To update the JDBC URL for the Workflow Manager Service database, go to `$BDD_HOME/BDD_manager/bin` on the Admin Server and run:

```
./bdd-admin.sh publish-config database wm -j <value>
```

Where `<value>` is the new JDBC URL.

## Refreshing TLS/SSL certificates

If you have TLS/SSL enabled for BDD, you can use the `bdd-admin` script to refresh your certificates, when needed.

For more information on refreshing your TLS/SSL certificates with `bdd-admin`, see [cert on page 33](#).

Before beginning this procedure, verify that the password for `$JAVA_HOME/jre/lib/security/cacerts` is set to `changeit`.

To refresh your TLS/SSL certificates:

1. Export the public key certificates from all Hadoop nodes running TLS/SSL- secured HDFS, YARN, Hive, and/or KMS.

You can do this with the following command:

```
keytool -exportcert -alias <alias> -keystore <keystore_filename> -file <export_filename>
```

Where:

- `<alias>` is the certificate's alias.
  - `<keystore_filename>` is the absolute path to your keystore file. You can find this in Cloudera Manager/Ambari/MCS.
  - `<export_filename>` is the name of the file to export the keystore to.
2. Copy all of the exported certificates to the directory on the Admin Server defined by `HADOOP_CERTIFICATES_PATH` in `$BDD_HOME/BDD_manager/conf/bdd.conf`.
  3. On the Admin Server, go to `$BDD_HOME/BDD_manager/bin` and run:

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

When the script runs, it imports the certificates to the custom truststore file, then copies the truststore to `$BDD_HOME/common/security/cacerts` on all BDD nodes.



## Chapter 5

# Adding and Removing BDD Nodes

The following sections describe how to add and remove nodes from your BDD cluster.

[Adding new Dgraph nodes](#)

[Adding new Data Processing nodes](#)

[Removing Data Processing nodes](#)

## Adding new Dgraph nodes

You can add new Dgraph nodes to BDD to expand your Dgraph cluster.



**Note:** You can also add new Data Processing nodes; for more information, see [Adding new Data Processing nodes on page 65](#). You can't add more WebLogic Server nodes without reinstalling.

To add a new Dgraph node:

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

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

2. Select a node in your cluster to move the Dgraph to.

If your databases are on HDFS/MapR-FS, this must be an HDFS DataNode.

3. If BDD is currently installed on the selected node, verify that the following directories are present and copy over any that are missing:

- `$BDD_HOME/common/edp`
- `$BDD_HOME/dataprocessing`
- `$BDD_HOME/dgraph`
- `$BDD_HOME/logs/edp`

If BDD isn't installed on the selected node:

- (a) Create a new `$BDD_HOME` directory on the node.
- (b) Set the permissions of `$BDD_HOME` to 755 and the owner to the `bdd` user.
- (c) Copy the following directories from an existing Dgraph node to the new one:
  - `$BDD_HOME/BDD_manager`
  - `$BDD_HOME/common`
  - `$BDD_HOME/dataprocessing`
  - `$BDD_HOME/dgraph`

- \$BDD\_HOME/logs
- \$BDD\_HOME/uninstall
- \$BDD\_HOME/version.txt

(d) Create a symlink \$ORACLE\_HOME/BDD pointing to \$BDD\_HOME.

4. If you have MapR and the new Dgraph node isn't part of your MapR cluster, install the MapR Client on it.

For instructions, see [Installing the MapR Client](#) in MapR's documentation.

5. If your databases are on HDFS, install the HDFS NFS Gateway service (called the MapR NFS in MapR) on the new node.

For instructions, refer to the documentation for your Hadoop distribution.

6. If you have to host the Dgraph on the same node as Spark (or any other memory-intensive process), set up cgroups so that the Dgraph will have access to the resources it requires.

For instructions, see [Setting up cgroups for the Dgraph on page 85](#).

7. Clean up the ZooKeeper index.

8. On the Admin Server, copy \$BDD\_HOME/BDD\_manager/conf/bdd.conf to a new location. Open the *copy* in a text editor and update the following properties:

Property	Description
DGRAPH_SERVERS	The hostnames of all Dgraph servers. Add the new node to this list. Be sure to use its FQDN.
DGRAPH_THREADS	The number of threads the Dgraph starts with. Verify that this setting is still accurate. It should be the number of CPU cores on the Dgraph nodes minus the number required to run HDFS and any other Hadoop services.
DGRAPH_CACHE	The size of the Dgraph cache. Verify that this setting is still accurate. It should either be 50% of the machine's RAM or the total amount of free memory, whichever is larger.
DGRAPH_ENABLE_CGROUP	Enables cgroups for the Dgraph. This must be set to <code>TRUE</code> if you created a Dgraph cgroup. You must also set <code>DGRAPH_CGROUP_NAME</code> .
DGRAPH_CGROUP_NAME	The name of the cgroup that controls the Dgraph. This is required if <code>DGRAPH_ENABLE_CGROUP</code> is set to <code>TRUE</code> .
NFS_GATEWAY_SERVERS	The hostnames of all NFS Gateway nodes. If you installed the NFS Gateway service on the new node, add its FQDN to this list.

9. To populate your configuration changes to the rest of the cluster, go to \$BDD\_HOME/BDD\_manager/bin and run:

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

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



## 10. Start your cluster:

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

## Adding new Data Processing nodes

You can add new Data Processing nodes to your BDD cluster to increase your processing power.



**Note:** You can also add more Dgraph nodes; for more information, see [Adding new Dgraph nodes on page 63](#). You can't add more WebLogic Server nodes without reinstalling.

To do this, you add one or more qualified YARN NodeManager nodes to your Hadoop cluster, then run the `bdd-admin` script with the `reshape-nodes` command. The script queries your Hadoop cluster manager (Cloudera Manager, Ambari, or MCS) for the newly-added nodes and automatically installs Data Processing on them. When the script completes, the new nodes are up and ready to accept new jobs.



**Note:** The `bdd-admin` script requires the username and password for the Hadoop cluster manager to query it. It will prompt you for this information if the `BDD_HADOOP_UI_USERNAME` and `BDD_HADOOP_UI_PASSWORD` environment variables aren't set.

To add a new Data Processing node:

1. Add one or more YARN NodeManager nodes to your Hadoop cluster. To support Data Processing, the following Hadoop components must be installed on each:
  - Spark on YARN
  - YARN
  - HDFS/MapR-FS

For instructions on adding new YARN NodeManager nodes, refer to the documentation for your Hadoop distribution.

2. If you have TLS/SSL enabled, export the public key certificates for the new YARN node(s), then copy them to the directory on the Admin Server defined by `HADOOP_CERTIFICATES_PATH` in `$BDD_HOME/BDD_manager/conf/bdd.conf`.

You can export the certificates by running the following from the new YARN node(s):

```
keytool -exportcert -alias <alias> -keystore <keystore_filename> -file <export_filename>
```

Where:

- `<alias>` is the certificate's alias.
  - `<keystore_filename>` is the absolute path to your keystore file. You can find this in your Hadoop manager.
  - `<export_filename>` is the name of the file you want to export the keystore to.
3. On the Admin Server, go to `$BDD_HOME/BDD_manager/bin` and run:

```
./bdd-admin.sh reshape-nodes
```

4. Enter the username and password for your Hadoop cluster manager, if prompted.

## Removing Data Processing nodes

You can remove Data Processing nodes from your BDD cluster, if necessary.

To do this, you remove one or more of the YARN NodeManager nodes running Data Processing from your Hadoop cluster, then run the `bdd-admin` script with the `reshape-nodes` command. The script queries your Hadoop cluster manager (Cloudera Manager, Ambari, or MCS) for the removed node(s) and updates BDD's configuration accordingly.



**Note:** The `bdd-admin` script requires the username and password for the Hadoop cluster manager to query it. It will prompt you for this information if the `BDD_HADOOP_UI_USERNAME` and `BDD_HADOOP_UI_PASSWORD` environment variables aren't set.

To remove a Data Processing node:

1. Remove one or more of the YARN NodeManager nodes running Data Processing from your Hadoop cluster.

For instructions, refer to the documentation for your Hadoop distribution.

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

```
./bdd-admin.sh reshape-nodes
```

3. Enter the username and password for your Hadoop cluster manager, if prompted.



## Chapter 6

# Backing Up and Restoring BDD

---

The `bdd-admin` script provides commands for backing up and restoring your data.

*Backing up BDD*

*Performing a full BDD restoration*

*Restoring BDD to a new cluster*

*Troubleshooting MySQL database restorations*

## Backing up BDD

You can back up your BDD data and metadata to a TAR file that you can later use to restore your cluster.



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

Before you back up your cluster, verify that:

- You can provide the script with the usernames and passwords for all component databases. You can either enter this information at runtime or set the following environment variables. Note that if you have HDP, you must also provide the username and password for Ambari.
  - `BDD_STUDIO_JDBC_USERNAME`: The username for the Studio database
  - `BDD_STUDIO_JDBC_PASSWORD`: The password for the Studio database
  - `BDD_WORKFLOW_MANAGER_JDBC_USERNAME`: The username for the Workflow Manager Service database
  - `BDD_WORKFLOW_MANAGER_JDBC_PASSWORD`: The password for the Workflow Manager Service database
  - `BDD_HADOOP_UI_USERNAME`: The username for Ambari (HDP only)
  - `BDD_HADOOP_UI_PASSWORD`: The password for Ambari (HDP only)
- You have an Oracle or MySQL database. Hypersonic isn't supported.
- 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. The Instant Client isn't supported.
- For Oracle databases, the `ORACLE_HOME` environment variable must be 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`.

- The temporary directories used during the backup operation contain enough free space. For more information, see [Space requirements on page 26](#).

For more information on backup, see [backup on page 24](#). For instructions on restoring your cluster, see [Performing a full BDD restoration on page 68](#).

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, as the script might take a long time to finish and the output will keep you informed of its current status.

3. If you haven't set the environment variables listed above, enter the usernames and password for the component databases when prompted. If you have HDP, you must also provide the username and password for Ambari.

## Performing a full BDD restoration

You can use the `bdd-admin` script to perform a full restoration, which restores all BDD data, metadata, and configuration information to your cluster.

Because this includes configuration data, **you must restore to the same cluster that was backed up**. If you want to restore to a different cluster, see [Restoring BDD to a new cluster on page 70](#).

Before restoring your cluster, verify that:

- You are restoring to the same cluster that was backed up.
- You have a backup TAR file created by the `bdd-admin` script's `backup` command.
- You can provide the `bdd-admin` script with the usernames and passwords for all component databases. You can either enter this information at runtime or set the following environment variables. Note that if you have HDP, you must also provide the username and password for Ambari.
  - `BDD_STUDIO_JDBC_USERNAME`: The username for the Studio database
  - `BDD_STUDIO_JDBC_PASSWORD`: The password for the Studio database
  - `BDD_WORKFLOW_MANAGER_JDBC_USERNAME`: The username for the Workflow Manager Service database
  - `BDD_WORKFLOW_MANAGER_JDBC_PASSWORD`: The password for the Workflow Manager Service database
  - `BDD_HADOOP_UI_USERNAME`: The username for Ambari (HDP only)
  - `BDD_HADOOP_UI_PASSWORD`: The password for Ambari (HDP only)

- Both the source and target clusters have the same minor version of BDD; for example, 1.4.0.37.xxxx.
- 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. The Instant Client isn't supported.
- For Oracle databases, the `ORACLE_HOME` environment variable must be 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`.
- For MySQL databases, the `lower_case_table_names` system variable has the same value on both clusters. If it doesn't, be sure to change it accordingly on the current cluster or the restoration will fail. For more information, see [Troubleshooting MySQL database restorations on page 72](#).
- The temporary directories used during the restore operation contain enough free space. For more information, see [Space requirements on page 29](#).

For more information on the `restore` command, see [restore on page 27](#).



**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 [-t <minutes>]
```

3. Run the `restore` command:

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

Where `<file>` is the absolute path to the backup TAR file you want to restore from.

4. Enter the usernames and passwords for the component databases, if prompted. If you have HDP, you must also provide the username and password for Ambari.
5. When restoration completes, restart your cluster:

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

6. Verify that the restoration was successful:

- (a) On the Admin Server, go to `$BDD_HOME/BDD_manager/bin` and perform a healthcheck:

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

- (b) Log in to Studio and verify that the project data is consistent with the original cluster.
- (c) Load a new data set to make sure file upload is working.
- (d) Perform a simple transform.

The restore operation created a copy of your Dgraph database directory in `DGRAPH_INDEX_DIR/.snapshot/old_copy`. You should delete this if you decide to keep the restored version of the Dgraph databases.

If you have a MySQL database and the restoration failed during database migration, see [Troubleshooting MySQL database restorations on page 72](#).

## Restoring BDD to a new cluster

You can use the `bdd-admin` script to perform a data-only restoration, which restores all BDD data and metadata. Because this method excludes configuration information, you can use it to restore your current cluster to a new one.

A data-only restoration restores the following data to the target cluster:

- The Dgraph databases
- The databases used by Studio and the Workflow Manager Service
- The user sandbox data in the location defined by `HDFS_DP_USER_DIR` in `bdd.conf`
- The HDFS sample data in `$HDFS_DP_USER_DIR/edp/data/.collectionData`

You can perform a data-only restoration on any cluster that meets the criteria listed below—it can be different from the one that was originally backed up and can even have a different topology than the original. For example, you can restore a backup of an eight-node cluster to a new six-node cluster, provided it meets the criteria listed below.

Before restoring, verify that:

- You can provide the `bdd-admin` script with the usernames and passwords for all component databases. You can either enter this information at runtime or set the following environment variables. Note that if you have HDP, you must also provide the username and password for Ambari.
  - `BDD_STUDIO_JDBC_USERNAME`: The username for the Studio database
  - `BDD_STUDIO_JDBC_PASSWORD`: The password for the Studio database
  - `BDD_WORKFLOW_MANAGER_JDBC_USERNAME`: The username for the Workflow Manager Service database
  - `BDD_WORKFLOW_MANAGER_JDBC_PASSWORD`: The password for the Workflow Manager Service database
  - `BDD_HADOOP_UI_USERNAME`: The username for Ambari (HDP only)
  - `BDD_HADOOP_UI_PASSWORD`: The password for Ambari (HDP only)
- Both the source and target clusters have the same minor version of BDD; for example, 1.4.0.37.xxxx.
- Both clusters have the same type of database, either Oracle or MySQL. Hypersonic isn't supported.
- 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. The Instant Client isn't supported.
- For Oracle databases, the `ORACLE_HOME` environment variable must be 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`.

- For MySQL databases, the `lower_case_table_names` system variable has the same value on both clusters. If it doesn't, be sure to change it accordingly on the current cluster or the restoration will fail. For more information, see [Troubleshooting MySQL database restorations on page 72](#).
- The temporary directories used during the restore operation contain enough free space. For more information, see [Space requirements on page 29](#).

To restore BDD to a new cluster:

1. On the Admin Server of the source cluster, go to `$BDD_HOME/BDD_manager/bin` and stop BDD:

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

2. Back up the source cluster:

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

Where `<file>` is the absolute path to the backup TAR file. This file must not exist and its parent directory must be writeable.

Once the backup completes, you can optionally restart the source cluster.

3. Copy the backup file you want to use to the target cluster.

The location you put it in is arbitrary as you will specify this information at runtime.

4. On the Admin Server of the target cluster, go to `$BDD_HOME/BDD_manager/bin`.

5. Stop the cluster, if it's running:

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

6. Optionally, create a backup of the target cluster.

7. Restore the backup of the source cluster to the target cluster:

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

Where `<file>` is the absolute path to the source cluster's backup TAR file.

8. Enter the component usernames and passwords, if prompted. If you have HDP, you must also provide the username and password for Ambari.

9. When the script completes, restart the target cluster:

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

10. Verify that the restoration was successful:

- (a) On the Admin Server, go to `$BDD_HOME/BDD_manager/bin` and perform a healthcheck:

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

- (b) Log in to Studio and verify that the project data is consistent with the original cluster.

- (c) Load a new data set to make sure file upload is working.

- (d) Perform a simple transform.

11. Migrate your Hive metadata and MetaStore data to the target cluster.

For instructions, refer to the documentation for your Hadoop distribution.

12. Verify that the Hive table migration was successful:
  - (a) Open Hue in a browser and click **Metastore Manager** at the top of the page.
  - (b) Select a table from the **Tables** list on the right and open it.
  - (c) Click the **Browse Data** icon on the top right.
  - (d) Verify that the data is the same as in the source cluster.

The restore operation created a copy of your Dgraph database directory in `DGRAPH_INDEX_DIR/.snapshot/old_copy`. You should delete this if you decide to keep the restored version of the Dgraph databases.

If you have a MySQL database and the restoration failed during database migration, see [Troubleshooting MySQL database restorations on page 72](#).

## Troubleshooting MySQL database restorations

A restore operation may fail when restoring MySQL component databases.

Below are some common errors you may encounter when restoring a BDD cluster and steps for fixing them. Once you've resolved the issue, you can rerun the restoration process.

### Identifier case sensitivity

The following error indicates that the source and target databases have different values for `lower_case_table_names`:

```
There is a discrepancy on lower_case_table_names between the current studio database and backed up one!
You backed up studio databaslower_case_table_names is 1 and are trying to restore to a database whose value of lower_case_table_names is 0.
```

To resolve this issue:

1. Stop `mysql`.
2. Open `/etc/my.cnf` and add the following line to the `[mysqld]` section:

```
lower_case_table_names=1
```

3. Restart `mysql`.
4. Verify the change:

```
mysqladmin -u root -p variables | grep lower_case_table_name
```

### Packet too large

The following error indicates that the MySQL server received a packet larger than `max_allowed_packet`:

```
2006 (HY000) at line 742: MySQL server has gone away
```

To resolve this issue:

1. Stop `mysql`.
2. Open `/etc/my.cnf` and add the following line to the `[mysqld]` section:



```
max_allowed_packet=32m
```

3. Restart mysql.

4. Verify the change:

```
mysqladmin -u root -p variables | grep max_allowed_packet=32m
```

# **Part III**

## **Administering the Dgraph**



## Chapter 7

# Dgraph Overview

---

The Dgraph uses data structures and algorithms to provide real-time responses to client requests for analytic processing and data summarization.

The data the Dgraph queries is stored in a set of *databases* (formerly called indexes). When a Studio user wants to view a specific data set, the Dgraph retrieves it from the appropriate database and returns the results.

The Dgraph works closely with both the Dgraph Gateway and the Dgraph HDFS Agent.

- The Dgraph Gateway routes Studio user requests to the Dgraph for processing. It uses session affinity to ensure that the same Dgraph instance responds to all queries from a given user session. It is also responsible for appointing a Dgraph instance as leader of each database.
- The Dgraph HDFS Agent acts as a data transport layer between the Dgraph and HDFS. It exports records to HDFS on behalf of the Dgraph and imports them from HDFS during data ingest operations.

The following sections provide more information about the Dgraph and how it functions.

[The Dgraph databases](#)

[The Dgraph cluster](#)

[Dgraph memory consumption](#)

[The Dgraph Tracing Utility](#)

[Dgraph statistics](#)

## The Dgraph databases

The Dgraph stores the data it queries in databases (formerly called indexes).

The databases are stored in the Dgraph databases directory, which is defined by the `DGRAPH_INDEX_DIR` property in the `$BDD_HOME/BDD_manager/conf/bdd.conf` file. This directory also contains three internal, system-created databases that are used by Studio:

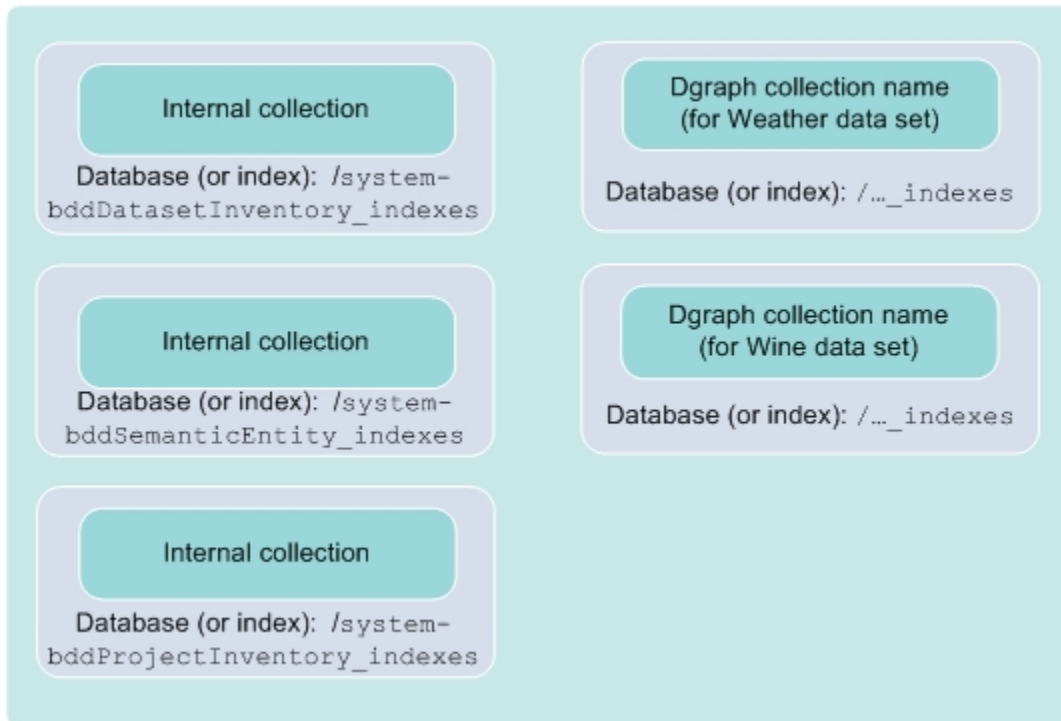
- `system-bddProjectInventory_indexes`
- `system-bddDatasetInventory_indexes`
- `system-bddSemanticEntity_indexes`

The Dgraph automatically creates a database for each new data set added by Studio or the DP CLI. By default, each database is named `<dataset>_indexes`, where `<dataset>` is the name of the original data set:

```
edp_cli_edp_256b0c6b-cacf-478c-80bf-b5332f4f37ae_indexes
```

For example, if you created two data sets called `Wine` and `Weather` in Studio, the Dgraph databases directory would contain five databases (one for each of the two data sets you created, plus the three internal ones). There might also be other databases that were created by committing transformed data sets.

#### Dgraph Databases Directory



### Database directory location

The Dgraph database directory must be stored in a location that all Dgraph nodes can access. The following filesystem types are supported:

- HDFS (Hadoop Distributed File System), or MapR-FS (for MapR clusters). This is recommended for production environments, as it's the best high availability option. For instructions on moving your databases to HDFS post-install, see [Moving the Dgraph databases to HDFS on page 77](#).
- NFS (network file system). This option provides some high availability, making it suitable for production environments. All Dgraph nodes must have read and write access to the NFS.
- Local storage. This option doesn't provide high availability, and is therefore only recommended for small demo or development environments.

If the Dgraph databases are on HDFS, the Dgraph can start if HDFS is down, but won't be able to accept requests. A background thread will try to connect to HDFS once per second until a connection is established.

Additionally, if you have HDFS data at rest encryption enabled, you can keep your databases in special directories called *encryption zones*. All files within an encryption zone are transparently encrypted and decrypted on the client side, meaning decrypted data is never stored in HDFS.

More information about database locations is available in the *Installation Guide*.

## Database logging

When a Dgraph instance mounts a database, an entry similar to the following is written to the Dgraph out log:

```
DGRAPH NOTIFICATION {database} [0] Mounting database edp_cli_edp_256b0c6b-cacf-478c-80bf
```

Note that the entry is made by the Dgraph database log subsystem.

The database name also appears in other BDD component messages. For example, the name of a DP workflow in a YARN log will contain the database name:

```
EDP: ProvisionDataSetFromHiveConfig{hiveDatabaseName=default, hiveTableName=warrantyclaims,
newCollectionId=MdexCollectionIdentifier{databaseName
=edp_cli_edp_256b0c6b-cacf-478c-80bf-b5332f4f37ae,
collectionName=edp_cli_edp_256b0c6b-cacf-478c-80bf-b5332f4f37ae}}
```

You should also see database names in the logs for Studio, Dgraph HDFS Agent, Workflow Manager, and Transform Service.

### [Moving the Dgraph databases to HDFS](#)

## Moving the Dgraph databases to HDFS

If your Dgraph databases are currently stored on NFS, you can move them to HDFS.



**Note:** This procedure is supported for MapR, which uses MapR-FS instead of HDFS. Although this document only refers to HDFS for simplicity, all information also applies to MapR-FS unless specified otherwise.

Because HDFS is a distributed file system, storing your databases there provides increased high availability for the Dgraph. It also increases the amount of data your databases can contain.

When its databases are stored on HDFS, the Dgraph has to run on HDFS DataNodes. If it isn't currently installed on DataNodes, you must move its binaries over when you move its databases.



**Important:** The DataNode service should be the only Hadoop service running on the Dgraph nodes. In particular, you shouldn't co-locate the Dgraph with Spark, as both require a lot of resources. If you *have* to host the Dgraph on nodes running Spark or other Hadoop services, use cgroups to ensure it has access to sufficient resources. For more information, see [Setting up cgroups for the Dgraph on page 85](#).

To move your Dgraph databases to HDFS:

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

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

2. Copy your Dgraph databases from their current location to the new one on HDFS.

The `bdd` user must have read and write access to the new location.

If you have MapR, the new location must be mounted with a volume and the `bdd` user must have permission to create and delete snapshots from it.

If you have HDFS data at rest encryption enabled, the new location must be an encryption zone.

3. If the Dgraph isn't currently installed on HDFS DataNodes, select one or more in your Hadoop cluster to move it to.

If other BDD components are currently installed on the selected nodes, verify that the following directories are present on each, and copy over any that are missing.

- `$BDD_HOME/common/edp`
- `$BDD_HOME/dataprocessing`
- `$BDD_HOME/dgraph`
- `$BDD_HOME/logs/edp`

If no BDD components are installed on the selected nodes:

- (a) Create a new `$BDD_HOME` directory on each node. Its permissions must be 755 and its owner must be the `bdd` user.
- (b) Copy the following directories from an existing Dgraph node to the new ones:
  - `$BDD_HOME/BDD_manager`
  - `$BDD_HOME/common`
  - `$BDD_HOME/dataprocessing`
  - `$BDD_HOME/dgraph`
  - `$BDD_HOME/logs`
  - `$BDD_HOME/uninstall`
  - `$BDD_HOME/version.txt`
- (c) Create a symlink `$ORACLE_HOME/BDD` pointing to `$BDD_HOME`.
- (d) Optionally, remove the `/dgraph` directory from the old Dgraph nodes, as it's no longer needed.

Leave any other BDD directories as they may still be useful.

4. To enable the Dgraph to access its databases in HDFS, install the HDFS NFS Gateway service (called MapR NFS in MapR) on all Dgraph nodes.

For instructions, refer to the documentation for your Hadoop distribution.

5. If you have MapR, mount MapR-FS to the local mount point, `$BDD_HOME/dgraph/hdfs_root`.

You can do this by adding an NFS mount point to `/etc/fstab` on each new Dgraph node. This ensures MapR-FS will be mounted automatically when your system starts. Note that you'll have to remove this manually if you uninstall BDD.

6. If you have to host the Dgraph on the same node as Spark or any other Hadoop processes (in addition to the HDFS DataNode process), create cgroups to isolate the resources used by Hadoop and the Dgraph.

For instructions, see [Setting up cgroups for the Dgraph on page 85](#).

7. For best performance, configure short-circuit reads in HDFS.

This enables the Dgraph to access local files directly, rather than having to use the HDFS DataNode's network sockets to transfer the data. For instructions, refer to the documentation for your Hadoop distribution.

8. Clean up the ZooKeeper index.

9. On the Admin Server, copy `$BDD_HOME/BDD_manager/conf/bdd.conf` to a new location. Open the *copy* in a text editor and update the following properties:

Property	Description
DGRAPH_INDEX_DIR	<p>The absolute path to the new location of the Dgraph databases directory on HDFS.</p> <p>If you have MapR, this location must be mounted as a volume, and the <code>bdd</code> user must have permission to create and delete snapshots from it.</p> <p>If you have HDFS data at rest encryption enabled, this location must be an encryption zone.</p>
DGRAPH_SERVERS	A comma-separated list of the FQDNs of the new Dgraph nodes. All must be HDFS DataNodes.
DGRAPH_THREADS	The number of threads the Dgraph starts with. This should be the number of CPU cores on the Dgraph nodes minus the number required to run HDFS and any other Hadoop services running on the new Dgraph nodes.
DGRAPH_CACHE	The size of the Dgraph cache. This should be either 50% of the machine's RAM or the total amount of free memory, whichever is larger.
DGRAPH_USE_MOUNT_HDFS	Determines whether the Dgraph mounts HDFS when it starts. Set this to <code>TRUE</code> .
DGRAPH_HDFS_MOUNT_DIR	<p>The absolute path to the local directory where the Dgraph mounts the HDFS root directory. This location must exist and be empty, and must have read, write, and execute permissions for the <code>bdd</code> user.</p> <p>It's recommended that you use the default location, <code>\$BDD_HOME/dgraph/hdfs_root</code>, which was created by the installer and should meet these requirements.</p>
KERBEROS_TICKET_REFRESH_INTERVAL	Only required if you have Kerberos enabled. The interval (in minutes) at which the Dgraph's Kerberos ticket is refreshed. For example, if set to 60, the Dgraph's ticket would be refreshed every 60 minutes, or every hour.
KERBEROS_TICKET_LIFETIME	Only required if you have Kerberos enabled. The amount of time that the Dgraph's Kerberos ticket is valid. This should be given as a number followed by a supported unit of time: <code>s</code> , <code>m</code> , <code>h</code> , or <code>d</code> . For example, <code>10h</code> (10 hours), or <code>10m</code> (10 minutes).
DGRAPH_ENABLE_CGROUP	Only required if you set up cgroups for the Dgraph. This must be set to <code>TRUE</code> if you created a Dgraph cgroup.
DGRAPH_CGROUP_NAME	Only required if you set up cgroups for the Dgraph. The name of the cgroup that controls the Dgraph.

Property	Description
NFS_GATEWAY_SERVERS	Only required if you're using the NFS Gateway. A comma-separated list of the FQDNs of the nodes running the NFS Gateway service. This should include all Dgraph nodes.
DGRAPH_USE_NFS_MOUNT	If you're using the NFS Gateway, set this property to <code>TRUE</code> .

10. To populate your configuration changes to the rest of the cluster, go to `$BDD_HOME/BDD_manager/bin` and run:

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

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

11. Start your cluster:

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

## The Dgraph cluster

The Dgraph nodes form their own cluster within the BDD cluster.

The Dgraph cluster provides high availability for query processing—if one node fails, the others will continue processing. It also increases throughput, as the query load is spread across the cluster without requiring more storage.

Your Dgraph cluster can contain any number of nodes, although for best performance, Oracle recommends having at least three.

### Leader and follower nodes

Each node within the Dgraph cluster can have two roles: leader and follower.

- A **leader node** receives and processes updates to a specific Dgraph database (that is, for a specific data set). Updates include full or incremental updates to the database, as well as administration or configuration changes. A given Dgraph can be the leader for multiple databases.

After updating its database, a leader Dgraph notifies the others of the new version so they can begin using it. Note that all Dgraph nodes will continue using the previous version of the database to complete processing that started against it.

- **Follower nodes** are all other Dgraph nodes that aren't the leader for a particular database. They can process read-only queries against that database, but can't write to it.

All nodes start up without any databases mounted. If a Dgraph gets a Web service request involving a database that it has not mounted, it tries to mount it as a follower node, unless it has already been appointed leader by the Dgraph Gateway. The Dgraph automatically mounts new databases when they're created.

A leader is selected for a Dgraph database by the Dgraph Gateway the first time a write operation (for example, a transformation from Studio) for that database comes in. Until that point, it doesn't have a leader. Once a leader has been appointed for a Dgraph database, it remains the leader for as long as it's running.



## Session affinity

The Dgraph Gateway routes client requests to the Dgraph nodes using session affinity. 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 Dgraph Gateway can't locate the session ID, it relies on a round-robin strategy for deciding which Dgraph node the request should be routed to.

Note that session affinity is enabled by default, via the `endeca-session-id-key` and `endeca-session-id-type` properties in the request headers.

## Role of ZooKeeper

Hadoop ZooKeeper provides a number of services for the Dgraph, including configuration and state management, synchronization, and event notification. These mechanisms continue to work when Dgraph nodes, or the connections between them, fail. For example, if a leader Dgraph fails, ZooKeeper informs the Dgraph Gateway, which in turn starts the automatic leader re-election process.

The Dgraph can be started if ZooKeeper isn't running, although it won't be able to process any requests. The Dgraph HDFS Agent, on the other hand, can't start when ZooKeeper is down.

To ensure Dgraph high availability, ZooKeeper should be running on an odd number of nodes; the minimum recommendation is three. This prevents ZooKeeper from becoming a single point of failure. Note that these nodes don't have to be Dgraph nodes.

## Dgraph memory consumption

By default, the Dgraph is allowed to use up to 80% of the RAM available on its host machine. This prevents it from running into out-of-memory performance issues. The Dgraph also uses a considerable amount of virtual memory, which it needs for ingesting data and executing queries. This is an expected behavior and can be observed with system diagnostic tools.

If the Dgraph reaches a memory consumption limit, it will cancel queries, beginning with the one consuming the most memory. Each time 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 retains the physical memory it's using indefinitely, unless it's running on the same node as other memory-intensive processes, in which case it will release a significant portion fairly quickly. Because of this, depending on your requirements and available resources, you may want to host the Dgraph on dedicated nodes.

In some cases, you will be required to host the Dgraph on nodes with other processes; for example, if your databases are on HDFS, the Dgraph must be hosted on HDFS DataNodes. Oracle recommends limiting the number of processes running on these nodes. In particular, you shouldn't host the Dgraph on the same node as Spark. You should also use Linux cgroups (control groups) to ensure the Dgraph has access to the resources it requires; for more information, see [Setting up cgroups for the Dgraph on page 85](#).

You can also set a custom limit on the amount of memory the Dgraph can consume using the `--memory limit` flag. For more information, see [Changing the Dgraph memory limit on page 83](#).

## The Dgraph Tracing Utility

The Dgraph Tracing Utility is a Dgraph diagnostic program that collects the Dgraph target trace data, which is used by Oracle Support to troubleshoot the Dgraph.

The Tracing Utility is automatically started and stopped along with the Dgraph. It writes the data it collects to \*.ebb files in the \$DGRAPH\_HOME/bin directory. You can also manually generate and save the trace data with the bdd-admin script's `get-blackbox` command, as described in [get-blackbox on page 38](#).

## 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 can be used to tune your Dgraph or sent to Oracle Support for troubleshooting issues.

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 or reset the Dgraph statistics page with the bdd-admin script. For more information, see [get-stats on page 39](#) and [reset-stats on page 40](#).



## Chapter 8

# Adjusting Dgraph Settings

---

The following sections describe how to adjust Dgraph settings.

[Changing the Dgraph memory limit](#)

[Setting the Dgraph cache size](#)

[Using Linux ulimit settings for merges](#)

[Setting up cgroups for the Dgraph](#)

## Changing the Dgraph memory limit

You can set a custom limit on the amount of memory the Dgraph can consume with the `--memory-limit` flag.



**Note:** This flag is intended for use by Oracle Support, only.

When this flag is set, then the amount of memory required by the Dgraph to process all current queries can't exceed its value. Once this limit is reached, the Dgraph begins cancelling queries in the same way it does with the default limit.

You can set the `--memory-limit` flag by adding it to the `DGRAPH_ADDITIONAL_ARG` property in `$BDD_HOME/BDD_manager/conf/bdd.conf`.

Using a value of 0 means there is no limit set on the amount of memory the Dgraph can use. In this case, 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 the Dgraph to be restarted manually.



**Note:** If you use the `--cmem` flag, cache size (`--cmem`) should always be smaller than memory limit (the `--memory-limit` flag). In addition, if you enter an invalid value to the `--memory-limit` flag, the Dgraph will ignore that flag and instead use the default memory limit value.

For information on all Dgraph flags, see [Dgraph flags on page 87](#).

To change the memory limit:

1. On the Admin Server, go to `$BDD_HOME/BDD_manager/conf` directory and make a copy of the `bdd.conf` file in a different location.
2. Open the *copy* and add the `--memory-limit` flag to the `DGRAPH_ADDITIONAL_ARG` property. Be sure to save the file before closing.
3. Go to `$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`.

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 bdd.conf on page 49](#).

4. Restart the Dgraph so your changes take effect.

## Setting the Dgraph cache size

The Dgraph cache size must be large enough for the to Dgraph operate smoothly under a normal query load.



**Note:** While the Dgraph typically operates within 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. When the cache size reaches this threshold, the Dgraph evicts entries that consume its cache more aggressively, to reduce its cache memory usage to the configured limit. This behavior is not configurable.

This means that an occasional spike in Dgraph cache usage shouldn't be cause for alarm. You should only consider adjusting the Dgraph cache size after observing Dgraph performance over longer periods of time.

You can adjust the size of the Dgraph cache by gradually changing the `DGRAPH_CACHE` value in the `$BDD_HOME/BDD_manager/conf/bdd.conf`. (This property maps to the Dgraph `--cmem` flag.) You can then use the `bdd-admin` script to publish your changes to the entire cluster. For more information, see [publish-config on page 30](#).



**Note:** Cache size (`--cmem`) should always be smaller than memory limit (the `--memory-limit` flag). In addition, if you enter an invalid value to the `--cmem` flag, the Dgraph will ignore that flag and instead use the default cache size.

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.

Note that the percentage of memory used for the Dgraph cache is determined by the total amount of RAM on the machine, but is taken out of the percentage allocated to the Dgraph, which by default is 80%. For example, if the machine has 100MB of available RAM, the Dgraph would have access to 80MB, 50MB of which would be available for the cache.

## Using Linux ulimit settings for merges

For purposes of merging generation files for the internal Dgraph databases, it is recommended that you set the following Linux `ulimit`.

Parameter	Setting
<code>-v, -m</code>	unlimited. For the <code>-v</code> option, this sets no limit on the maximum amount of virtual memory available to a process. For the <code>-m</code> option, it sets no limit on the maximum resident set size. This can help prevent problems when the Dgraph is merging the generation files for its internal Dgraph databases.
<code>-n</code>	65536. This sets the maximum number of open file descriptors to 64K, which is especially important if the Dgraph and Hadoop are running on the same node. This parameter should have been set at install time.
<code>-u</code>	A soft limit of 65536 and a hard limit of unlimited. This essentially sets no limit on the maximum number of processes. This parameter should have been set at install time.

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

```
ERROR 04/03/15 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 database merging tasks.

## Setting up cgroups for the Dgraph

Control groups, or cgroups, is a Linux kernel feature that enables you to allocate resources like CPU time and system memory to specific processes or groups of processes. If you need to host the Dgraph on nodes running Spark, you must use cgroups to ensure sufficient resources are available to it.



**Note:** Because the Dgraph and Spark are both memory-intensive processes, hosting them on the same nodes is not recommended and should only be done if absolutely necessary. Although you can use the `--memory-limit` flag to set Dgraph memory consumption, Spark isn't aware of this and will continue to use as much memory as it needs, regardless of other processes.

To do this, you must enable cgroups in Hadoop and create one for YARN to limit the CPU percentage and amount of memory it can consume. Then, create a separate cgroup for the Dgraph to allocate appropriate amounts of memory and swap space to it.

To set up cgroups:

1. If your system doesn't currently have the `libcgroup` package, install it as root.  
This creates `/etc/cgconfig.conf`, which configures cgroups.
2. Enable the `cgconfig` service to run automatically:

```
chkconfig cgconfig on
```

3. Create a cgroup for YARN. This must be done within Hadoop. For instructions, refer to the documentation for your Hadoop distribution.

The YARN cgroup should limit the amounts of CPU and memory allocated to all YARN containers. The appropriate limits to set depend on your system and the amount of data you will process. At a minimum, you should reserve the following for the Dgraph:

- 10GB of RAM
- 2 CPU cores

The number of CPU cores YARN is allowed to use must be specified as a percentage. For example, on a quad-core machine, YARN should only get two of cores, or 50%. On an eight-core machine, YARN could get up to four of them, or 75%. When setting this amount, remember that allocating more cores to the Dgraph will boost its performance.

4. Create a cgroup for the Dgraph by adding the following to `cgconfig.conf`:

```
# Create a Dgraph cgroup named "dgraph"
group dgraph {
  # Specify which users can edit this group
  perm {
    admin {
      uid = $BDD_USER;
    }
    # Specify which users can add tasks for this group
    task {
      uid = $BDD_USER;
    }
  }
  # Set the memory and swap limits for this group
  memory {
    # Set memory limit to 10GB
    memory.limit_in_bytes = 10000000000;

    # Set memory + swap limit to 12GB
    memory.memsw.limit_in_bytes = 12000000000;
  }
}
```

Where `$BDD_USER` is the name of the `bdd` user.



**Note:** The values given for `memory.limit_in_bytes` and `memory.memsw.limit_in_bytes` above are the *absolute minimum* requirements. You should use higher values, if possible.

5. Restart `cfconfig` to enable your changes.



## Chapter 9

# Dgraph and Dgraph HDFS Agent Flags

The following sections describe the flags used by the Dgraph and the Dgraph HDFS Agent.

[Dgraph flags](#)

[Dgraph HDFS Agent flags](#)

## 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 30](#).






**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 ( <code>dgraph.out</code> ) or the file set by the <code>--out</code> flag.
<code>--backlog-timeout</code>	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.
<code>--bulk_load_port</code>	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 <code>--bulk_load_port</code> flag. This flag maps to the <code>DGRAPH_BULKLOAD_PORT</code> property in <code>bdd.conf</code> .

Flag	Description
<code>--cluster_identity</code>	<p>Specifies the cluster identity of the Dgraph running on this node. The syntax is:</p> <pre>protocol:hostname:dgraph_port:bulkload_port:export_port:mpp_port</pre> <p>This must be the same as the cluster identity specified for the HDFS Agent <code>--cluster_identity</code> flag.</p>
<code>--cmem</code>	<p>Specify the maximum memory usage (in MB) for the Dgraph cache. Note that cache size (<code>--cmem</code>) should always be smaller than memory limit (<code>--memory-limit</code>). For more information, see <a href="#">Setting the Dgraph cache size on page 84</a>.</p> <p>This flag maps to the <code>DGRAPH_CACHE</code> property in <code>bdd.conf</code>.</p>
<code>--export_port</code>	<p>Specifies the port on which the Dgraph listens for requests from the HDFS Agent.</p> <p>This should be the same as the number specified for the HDFS Agent <code>--export_port</code> flag. It should be different from the numbers specified for both the <code>--port</code> and <code>--bulk_load_port</code> flags.</p> <p>This flag maps to the <code>AGENT_EXPORT_PORT</code> property in <code>bdd.conf</code>.</p>
<code>--help</code>	<p>Prints the help message and exits. The help message includes usage information for each Dgraph flag.</p>
<code>--host</code>	<p>Specifies the name of the Dgraph's host server.</p> <p>This flag maps to the <code>DGRAPH_SERVERS</code> property in <code>bdd.conf</code>.</p>
<code>--log</code>	<p>Specifies the path to the Dgraph request log file. The default file used is <code>dgraph.reqlog</code>.</p>
<code>--log-level</code>	<p>Specifies the log level for the Dgraph log subsystems. For information on setting this flag, see <a href="#">Setting the Dgraph log levels on page 171</a>.</p> <p>This flag maps to the <code>DGRAPH_LOG_LEVEL</code> property in <code>bdd.conf</code>.</p>



Flag	Description
--memory-limit	<p>Specifies the maximum amount of memory (in MB) the Dgraph is allowed to use for processing. Note that cache size (<code>--cmem</code>) should always be smaller than memory limit (<code>--memory-limit</code>).</p> <p>If you do not use this flag, the memory limit is by default set to 80% of the machine's available RAM.</p> <p>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.</p> <p>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.</p> <p>For a summary of how Dgraph allocates and utilizes memory, see <a href="#">Dgraph memory consumption on page 81</a>.</p>
--mount_hdfs	<p>Specifies that the Dgraph should mount HDFS in a CDH or HDP environment. The target HDFS is specified by <code>&lt;hdfs config&gt;</code> which is the Hadoop HDFS configuration file (usually named <code>hdfs-site.xml</code>) and <code>&lt;core config&gt;</code> which is the Hadoop core configuration file (usually named <code>core-site.xml</code>).</p>
--mount-maprfs	<p>Specifies that the Dgraph should mount MapR-FS. <code>&lt;cluster&gt;</code> specifies the name of MapR cluster, while <code>&lt;path&gt;</code> is the index path on MapR-FS.</p>
--mppPort	<p>Specifies the port on this machine used for the Distributed Dgraph connection.</p> <p>This flag maps to the <code>DGRAPH_MPP_PORT</code> property in <code>bdd.conf</code>.</p>
--net-timeout	<p>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.</p>
--out	<p>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, <code>dgraph.out</code>.</p> <p>This file must be different from the one specified by the HDFS Agent's <code>--out</code> flag.</p> <p>This flag maps to the <code>DGRAPH_OUT_FILE</code> property in <code>bdd.conf</code>.</p>
--pidfile	<p>Specifies the file the Dgraph's process ID (PID) will be written to. The default filename is <code>dgraph.pid</code>.</p>
--port	<p>Specifies the port used by the Dgraph's host server.</p> <p>This flag maps to the <code>DGRAPH_WS_PORT</code> property in <code>bdd.conf</code>.</p>

Flag	Description
<code>--search_char_limit</code>	Specifies the maximum number of characters that a text search term can contain. The default value is 132.
<code>--search_max</code>	Specifies the maximum number of terms that a text search query can contain. The default value is 10.
<code>--snip_cutoff</code>	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.
<code>--snip_disable</code>	Globally disables snippeting.
<code>--sslcafile</code>	 <b>Note:</b> 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.
<code>--sslcertfile</code>	 <b>Note:</b> 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.
<code>--stat-brel</code>	 <b>Note:</b> 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.
<code>--threads</code>	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 <code>DGRAPH_THREADS</code> property in <code>bdd.conf</code> .
<code>--version</code>	Prints version information and then exits. The version information includes the Oracle Big Data Discovery version number and the internal Dgraph identifier.

Flag	Description
<code>--wildcard_max</code>	Specifies the maximum number of terms that can match a wildcard term in a wildcard query that contains punctuation, such as <code>ab*c.def*</code> . The default is 100.
<code>--zookeeper</code>	Specifies a comma-separated list of ZooKeeper servers. The syntax for each ZooKeeper server is: <pre>&lt;hostname&gt;:&lt;port&gt;</pre> This must be the same as the value specified for the HDFS Agent <code>--zookeeper</code> flag.
<code>--zookeeper_auth</code>	Obtains the ZooKeeper authentication password from standard in. Note the following about this flag: <ul style="list-style-type: none"> <li>The "ZooKeeper authentication password" corresponds to individual node-level access using ACL described here (Dgraph uses the digest scheme): <a href="https://zookeeper.apache.org/doc/r3.1.2/zookeeperProgrammers.html#sc_ZooKeeperAccessControl">https://zookeeper.apache.org/doc/r3.1.2/zookeeperProgrammers.html#sc_ZooKeeperAccessControl</a></li> </ul> It has nothing to do with Kerberos or the ability of the Dgraph to establish a session with ZooKeeper. <ul style="list-style-type: none"> <li>It is imperative that all Dgraphs, Dgraph Gateway, and Dgraph HDFS Agent are using the same "Zookeeper authentication password" because they will not be able to access needed information created by other components if they are using different passwords. If the Dgraph cannot access information in ZooKeeper due to a wrong password, it is a fatal error.</li> </ul>
<code>--zookeeper_index</code>	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 <code>--zookeeper_index</code> flag.  This flag maps to the <code>ZOOKEEPER_INDEX</code> property in <code>bdd.conf</code> .

## How the Dgraph handles command line arguments

At start-up time, the Dgraph parses the flags and their arguments from the command line. It verifies that each flag is a valid flag and, if it is valid, that the flag has a valid argument.

Invalid flag names or arguments are handles as follows:

- If the name of the flag is invalid, the flag is ignored and the Dgraph attempts to start up. A warning is also logged in the `dgraph.out` log file, as in this example in which the `--threads` flag was misspelled:

:

```
DGRAPH WARNING {dgraph} [0] Invalid option: treads. Option ignored.
```

- If the name of the flag is valid but it has an invalid argument, the Dgraph will not start up. An error, similar to the following example, is logged in the `dgraph.out` log file:

```
DGRAPH INCIDENT_ERROR {dgraph} [0] bad argument for --threads
```

In either case, correct the invalid flag or argument and re-start the Dgraph.

## 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
<code>--agent_port</code>	Sets the port on which the Dgraph HDFS Agent is listening for HTTP requests. Note that there is no Dgraph version of this flag.
<code>--export_port</code>	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 <code>--export_port</code> flag.
<code>--port</code>	Specifies the port on which the Dgraph is listening for HTTP requests. This port number must be the same as specified for the Dgraph <code>--port</code> flag.
<code>--bulk_load_port</code>	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 <code>--bulk_load_port</code> flag.
<code>--cluster_identity</code>	Specifies the cluster identity of the Dgraph running on this node. The syntax is: <pre>protocol:hostname:dgraph_port:bulkload_port:export_port:mpp_port</pre> This cluster identity must be the same as specified for the Dgraph <code>--cluster_identity</code> flag.
<code>--out</code>	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 <code>--out</code> flag.
<code>--principal</code>	For Kerberos support, specifies the name of the principal.
<code>--keytab</code>	For Kerberos support, specifies the path to the principal's keytab.
<code>--krb5conf</code>	For Kerberos support, specifies the path to the <code>krb5.conf</code> configuration file.
<code>--hadoop_truststore</code>	To support TLS-enabled Hadoop services, specifies the location of the Hadoop trust store.

Dgraph HDFS Agent flag	Description
--zookeeper	Specifies the host and port on which ZooKeeper is running. The syntax is: <pre data-bbox="615 394 1453 436">host:port</pre> (with a semicolon separating the host name and port). This host:port must be the same as specified for the Dgraph --zookeeper flag.
--zookeeper_index	Specifies the index of the cluster in the ZooKeeper ensemble. This index must be the same as specified for the Dgraph --zookeeper_index 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 7010
--bulk_load_port 7019 --zookeeper web04.example.com:2181 --zookeeper_index cluster1
--cluster_identity http:web04.example.com:7010:7019:7101:7029 --out /tmp/agent.log
```

# **Part IV**

## **Administering Studio**



## Chapter 10

# Managing Data Sources

---

You can add, configure, and delete database connections and JDBC data sources in Studio.

[About database connections and JDBC data sources](#)

[Creating data connections](#)

[Deleting data connections](#)

[Creating a data source](#)

[Editing a data source](#)

[Deleting a data source](#)

## 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, follow the steps below.

To create a data connection:

1. Log in to Studio as an administrator.
2. Click **Configuration Options** and then **Control Panel**. Then navigate to **Big Data Discovery** and then **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** and then **Control Panel**. Then navigate to **Big Data Discovery** and then **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** and then **Control Panel**. Then navigate to **Big Data Discovery** and then **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, follow the steps below.

To delete a data source:

1. Log in to Studio as an administrator.
2. Click **Configuration Options** and then **Control Panel**. Then navigate to **Big Data Discovery** and then **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**.



## Configuring Studio Settings

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

[Studio settings in BDD](#)

[Changing the Studio setting values](#)

[Modifying the Studio session timeout value](#)

[Changing the Studio database password](#)

[Viewing the Server Administration Page information](#)

### Studio settings in BDD

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
<code>df.bddSecurityManager</code>	The fully-qualified class name to use for the BDD Security Manager. If empty, the Security Manager is disabled.
<code>df.clientLogging</code>	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.
<code>df.countApproxEnabled</code>	Specifies a Boolean value to indicate that components perform approximate record counts rather than precise record counts. A value of <code>true</code> indicates that Studio displays approximate record counts using the <code>COUNT_APPROX</code> aggregation in an EQL query. A value of <code>false</code> indicates precise record counts using the <code>COUNT</code> aggregation. Setting this to <code>true</code> increases the performance of refinement queries in Studio. The default value is <code>false</code> .
<code>df.dataSourceDirectory</code>	The directory used to store keystore and certificate files for secured data.

Setting	Description
df.defaultAccessForDerivedDataSets	Controls whether new data sets created by <b>Export</b> or <b>Create new data set</b> are set to <b>Private</b> (restricted to the creator and all Studio Administrators) or made publically available at various access levels. Defaults to <b>Public (Default Access)</b> .
df.defaultCurrencyList	A comma-separated list of currency symbols to add to the ones currently available.
df.helpLink	Used to configure the path to the documentation for this release. Used for links to specific information in the documentation.
df.mapLocation	<p>The URL for the Oracle MapViewer eLocation service. The eLocation service is used for the text location search on the <b>Map</b> 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.</p> <p>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.</p>
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.mdexCacheManager	Internal use only.
df.notificationsMaxDaysToStore	The maximum number of days to store notifications. This is a setting to prune notifications from displaying in the <b>Notifications</b> window. It is a global limit that applies to all Studio users. Notifications that are older than this value are automatically deleted.
df.notificationsMaxToStore	The maximum number of notifications to store per user. This is a setting to prune notifications from displaying in the <b>Notifications</b> window. Notifications that exceed this value are automatically deleted. The default number of notifications 300.

Setting	Description
<code>df.stringTruncationLimit</code>	The maximum number of characters to display for a string value. You can override this value when configuring the display of a string value in an individual component. The default number is 10000 characters.
<code>df.sunburstAnimationEnabled</code>	Toggles animation and dynamic refinements for the <b>Chart Pie / Sunburst</b> component.
<code>df.performanceLogging</code>	This property can only be modified from the <code>portal-ext.properties</code> file.

## Changing the Studio setting values

To set the values of Studio settings, you modify the fields on the **Studio Settings** page.



**Note:** Take care when modifying these settings, as incorrect values can cause problems with your Studio instance. Also, if a setting on this page was specified in `portal-ext.properties` file, then you cannot change the setting from this page. You must set it in the file. (This is uncommon.)

To change the Studio setting values:

1. From the Control Panel, select **Big Data Discovery** and then **Studio Settings**.
2. Click **Update Settings**.
3. To apply the changes, restart Studio.

## Modifying the Studio session timeout value

The timeout notification that appears in the header of Studio can be controlled by the `session.timeout` property in `portal-ext.properties`.

By default, the `portal-ext.properties` file does not have the `session.timeout` property. Instead, the default timeout value is set in the `web.xml` file (which is in `endeca-portal.war`), on the WebLogic Server that is running Studio. The `session.timeout` property, when added to `portal-ext.properties`, will override the `web.xml` setting.

To modify the Studio session timeout value:

1. Stop Studio.

For example, you can run the `stop` command of `bdd-admin`:

```
./bdd-admin stop -c bddServer
```

2. On the server running WebLogic, open `$DOMAIN_HOME/config/studio/portal-ext.properties` and add the following property (or modify it if it has already been added):

```
session.timeout=30
```

Note that the `session-timeout` parameter takes an integer value in minutes.

### 3. Restart Studio.

For example, you can run the `start` command of `bdd-admin`:

```
./bdd-admin start -c bddServer
```

## Changing the Studio database password

As described in the *Installation Guide*, Studio requires a relational database to store configuration and state, including component configuration, user permissions, and system settings. Before BDD installation, an administrator creates the Studio database with a corresponding username and password.

To change the database password:

#### 1. Change the password in the database server.

For example, in MySQL, the command is similar to:

```
SET PASSWORD FOR 'studio'@'%' = PASSWORD('bdd');
```

For specific details, see the database documentation for the particular database type the administrator installed (Oracle 11g, 12c, or MySQL).

#### 2. Change it in WebLogic Server.

(a) In the WebLogic Administration Console for the BDD domain, go to **Services** and then **Data Sources**.

(b) Delete the existing `BDDStudioPool`.

(c) Create a new `BDDStudioPool` with the updated password.

For additional details, see the [WebLogic Administration Console Online Help](#).

#### 3. Restart Studio.

You can use the WebLogic Administration Console under **Environment** and then **Deployment** or use `bdd-admin` to restart the BDD Server.

## Viewing the Server Administration Page information

The features on the **Server Administration** page primarily provide debugging information for the Studio framework, and the features are intended for Oracle Support.



## Chapter 12

# Configuring Data Processing Settings

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

[List of Data Processing Settings](#)

[Changing the data processing settings](#)

## 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*.



**Important:** Except where noted, editing the Data Processing settings is not supported in Big Data Discovery Cloud Service.

Hadoop Setting	Description
<code>bdd.enableEnrichments</code>	<p>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 <code>true</code> runs all the data enrichment modules and <code>false</code> does not run them. You cannot enable an individual enrichment. The default value is <code>true</code>.</p> <p> <b>Note:</b> Editing this setting is supported in BDD Cloud Service.</p>

Hadoop Setting	Description
<code>bdd.sampleSize</code>	<p>Specifies the maximum number of records in the sample size of a data set. This is a global setting controls both the sample size for all files uploaded using Studio, and it also controls the sample size resulting from transform operations such as Join, Aggregate, and FilterRows.</p> <p>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.</p> <p>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.)</p> <p> <b>Note:</b> Editing this setting is supported in BDD Cloud Service.</p>

## Data Processing Topology

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

Hadoop Setting	Description
<b>Hadoop Admin Console</b>	The hostname and Admin Console port of the machine that acts as the Master for your Hadoop cluster.
<b>Name Node</b>	The NameNode internal Web server and port.
<b>Hive metastore Server</b>	The Hive metastore listener and port.
<b>Hive Server</b>	The Hive server listener and port.
<b>Hue Server</b>	The Hue Web interface server and port.
<b>Database Name</b>	The name of the Hive database that stores the source data for Studio data sets.
<b>Sandbox</b>	The HDFS directory in which to store the Parquet files created when users export data from Big Data Discovery. The default value is <code>/user/bdd</code> .

## 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** and then **Data Processing Settings**.
3. For each setting, update the value as necessary.
4. Click **Update Settings**.

The changes are applied immediately.





## Chapter 13

---

# Running a Studio Health Check

You check the health and basic functionality of Studio by running a health check URL in a Web browser. This operation is typically only run after major changes to the BDD setup, such as upgrading and patching.

You do not need machine access or command line access to run the health check URL. This is especially useful if you do not have machine access and therefore access to a command prompt to run `bdd-admin`.

The health check URL provides a more complete Studio check than running the `bdd-admin status` command. The `bdd-admin` command pings the Studio instance to see whether it is running or not. Whereas, the health check URL does the following:

- Checks that the Studio database is accessible.
- Uploads a file to HDFS.
- Creates a Hive table from that file.
- Ingests a data set from that Hive table.
- Queries the data set to ensure it returns results.

To run a Studio health check:

1. Start a web browser and type the following health check URL:  
`http://<Studio Host Name>:<Studio port>/bdd/health.`  
For example: `http://abcd01.us.oracle.com:7003/bdd/health.`
2. Optionally, check the **Notifications** panel to watch the progress of the check if you are signed into Studio.

The check should return `200 OK` to the browser if the health check succeeds.



## Chapter 14

# Viewing Project Usage Summary Reports

---

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

[About the project usage logs](#)

[About the System Usage page](#)

[Using the System Usage page](#)

## 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.

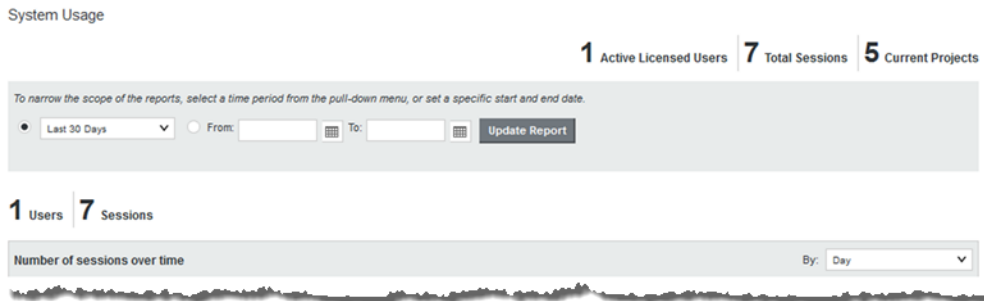
To change the age of the entries to delete, add the following setting to `portal-ext.properties`:

```
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.



The page is divided into the following sections:

Section	Description
<b>Summary totals</b>	At the top right of the page are the total number of: <ul style="list-style-type: none"> <li>• Users in the system</li> <li>• Sessions that have occurred</li> <li>• Projects</li> </ul>
<b>Date range fields</b>	Contains fields to set the range of dates for which to display report data.
<b>Current number of users and sessions</b>	Lists the number of users that were logged in and the number of sessions for the date range that you specify.
<b>Number of sessions over time</b>	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.
<b>User Activity</b>	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: <ul style="list-style-type: none"> <li>• A specific user, or all users</li> <li>• A specific project, or all projects</li> <li>• Whether to display the top or bottom values (most or least sessions)</li> <li>• The number of values to display</li> </ul>

Section	Description
<b>Project Usage</b>	<p>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.</p> <p>At the top of the report are lists to select:</p> <ul style="list-style-type: none"> <li>• A specific project, or all projects</li> <li>• Whether to display the top or bottom values (most or least sessions)</li> <li>• The number of values to display</li> </ul>
<b>System</b>	Contains a pie chart that shows the relative number of sessions by browser type and version for the selected date range.

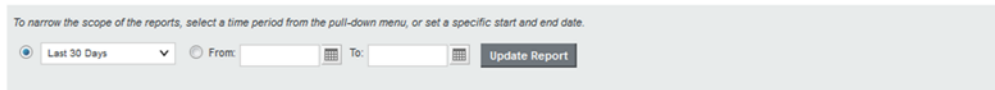
## Using the System Usage page

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.

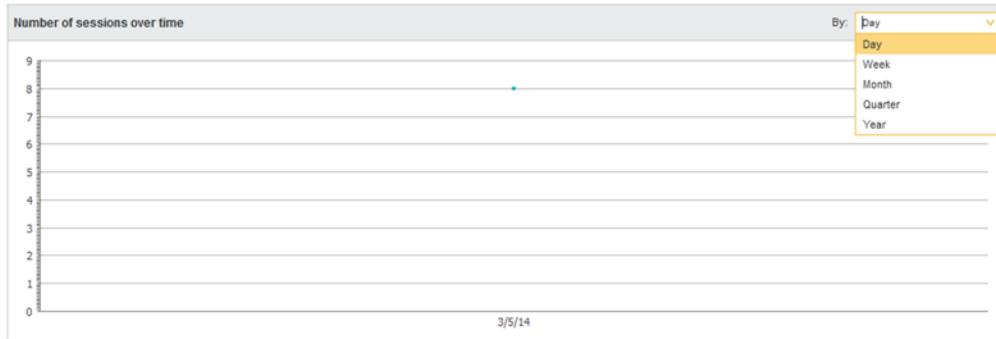
By default, the page is set to display data from the last 30 days.



- (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**.

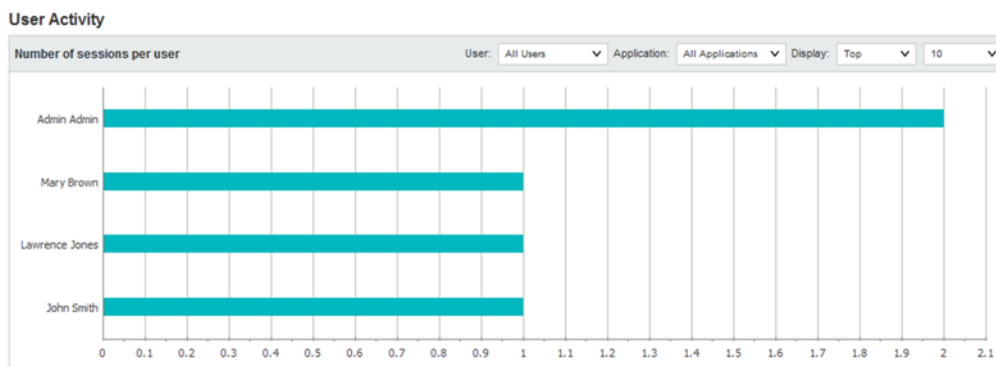
2. For the **Number of sessions over time** report, you can control the date/time unit used to display the results.

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



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.



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.

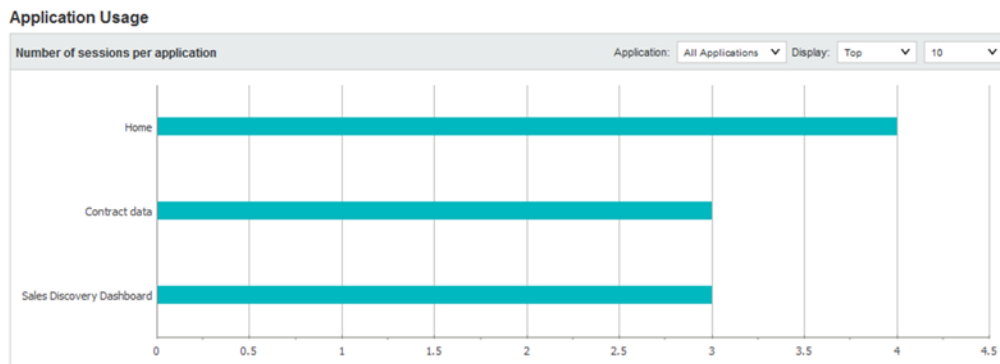
- (b) To narrow the report to a specific project, from the **Project** list, select the project.

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.

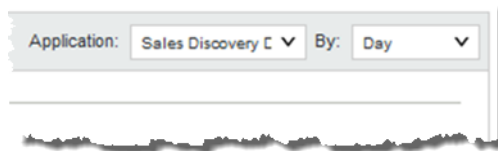


You can narrow the report to show values for a specific project, and change the number of values displayed.

- (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 15

# Configuring the Locale and Time Zone

---

The user interface of Studio and project data can be displayed in different locales and different time zones.

*Locales and their effect on the user interface*

*How Studio determines the locale to use*

*Selecting the default locale*

*Configuring a user's preferred locale*

*Setting the default time zone*

## 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

Studio supports the following languages:

- Chinese - Simplified
- English - US
- English - UK
- 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 determines 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	<p>The locale for a new user is initially set to <b>Use Browser Locale</b>, which indicates to use the current browser locale.</p> <p>This value can be changed to a specific locale.</p> <p>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.</p>
A non-logged-in user navigates to Big Data Discovery	<p>For a non-logged-in user, Big Data Discovery first tries to use the locale from the cookie.</p> <p>If there is no cookie, or the cookie is invalid, then Big Data Discovery tries to use the browser locale.</p> <p>If the current browser locale is not one of the supported locales, then the default locale is used.</p>



Scenario	How the locale is determined
A registered user logs in	<p>When a user logs in, Big Data Discovery first checks the locale configured for their user account.</p> <ul style="list-style-type: none"> <li>If the user's locale is set to <b>Use Browser Locale</b>, then Big Data Discovery tries to use the locale from the cookie.</li> </ul> <p>If there is no cookie, or if the cookie is invalid, then Big Data Discovery tries to use the browser locale.</p> <p>If the current browser locale is not a supported locale, then the default locale is used.</p> <ul style="list-style-type: none"> <li>If the user account is configured with a locale value other than <b>Use Browser Locale</b>, then Big Data Discovery uses that locale, and also updates the cookie with that locale.</li> </ul>
A non-logged-in user uses the user menu option to select a different locale	<p>When a non-logged-in user selects a locale, Big Data Discovery updates the cookie with the new locale.</p> <p>Note that this locale change is only applied locally. It is not applied to all non-logged-in users.</p>
A logged-in user uses the user menu option to select a different locale	<p>When a logged-in user selects a locale, Big Data Discovery updates both the user's account and the cookie with the selected locale.</p>

## Selecting the default locale

Studio is configured with a default locale that 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:

- From the **Control Panel**, select **Platform Settings** and then **Display Settings**.
- From the **Locale** list, select a default locale.

### Display Settings

---

Locale

United States - English ▼

Time Zone

(UTC) Coordinated Universal Time ▼

- 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** and then **My Account**.

### My Account

\* Required

#### User Details

Screen Name:\*

Password:

Email Address:\*

Retype Password:

First Name:\*

#### Display Settings

Locale:

Middle Name:

Time Zone:

Last Name:\*

Role: Administrator

► INHERITED ROLES

Cancel

Save

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** and then **Users**.

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

**Add/Edit User**

**User Details**

\*Required

Screen Name:\*  
rwiggum

Email Address:\*  
ralph.wiggum@ssotest.com

First Name:\*  
Ralph

Password:\*  
[Empty]

Middle Name:  
[Empty]

Retype Password:\*  
[Empty]

Last Name:\*  
Wiggum

Role:\*  
Restricted User

Locale:\*  
United States - English

▶ INHERITED ROLES

▶ PROJECTS

Cancel Save

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

## Setting the default time zone

Studio is configured with a default time zone that you can update from the **Control Panel**. By default, the time zone is set to UTC. You might want to set it to your local time zone to reflect accurate time stamps in the **Notifications** panel.

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

To set the default time zone:

1. From the **Control Panel**, select **Platform Settings** and then **Display Settings**.

- From the **Time Zone** list, select a default time zone.

### **Display Settings**

---

#### **Locale**

United States - English ▼

#### **Time Zone**

(UTC ) Coordinated Universal Time ▼

- Click **Save**.



## Chapter 16

# 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*

*Configuring the sender name and email address for notifications*

*Setting up the Account Created and Password Changed notifications*

## 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 **Platform Settings** and then **Email Settings**.
3. Click the **Sender** 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	<input type="text" value="acme.com.s7a1.pstmp.com"/>
Outgoing Port	<input type="text" value="25"/>
Use a Secure Network Connection	<input type="checkbox"/>
User Name	<input type="text" value="user_user@acme.com"/>
Password	<input type="password" value="*****"/>

- (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 and email address to display on outbound notifications.

To configure the sender name and email address:

1. From the **Control Panel**, select **Platform Settings** and then **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** and then **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**.



# 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](#)

[Assigning users and user groups to projects](#)

[Certifying a project](#)

[Making a project active or inactive](#)

[Deleting 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	<ul style="list-style-type: none"><li>The project Creator and Studio Administrators are the only users with access</li><li>The <b>All Big Data Discovery users</b> group is set to <b>No Access</b></li></ul> Projects are Private by default. Access must be granted by the Creator or by a Studio Administrator.
Public	<ul style="list-style-type: none"><li>The <b>All Big Data Discovery users</b> group is set to <b>Project Restricted Users</b></li></ul> Public projects grant view access to Studio users.
Shared	The project has been modified in any of the following ways: <ul style="list-style-type: none"><li>Users other than the Creator are added to the project</li><li>User Groups other than <b>All Big Data Discovery admins</b> and <b>All Big Data Discovery users</b> are added to the project</li><li>The <b>All Big Data Discovery users</b> group is set to <b>Project Authors</b></li></ul> 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** and then **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 **Sharing** page (under **Project Settings**) or from the project details panel in the Catalog. For details, see "Assigning project roles" in the *Studio User's 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** and then **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** and then **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** and then **Projects**.
2. Click the **Actions** link for the project you want to remove.
3. Click **Delete**.

# **Part V**

## **Controlling User Access to Studio**



## Chapter 18

# Configuring User-Related Settings

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

[Configuring authentication settings for users](#)

[Configuring the password policy](#)

[Restricting the use of specific screen names and email addresses](#)

## 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** and then **Credentials**.
3. On the **Credentials** page, click the **Authentication** tab.

### Credentials

Reserved Credentials Authentication

How do users authenticate?

By Email Address ▾

Allow users to automatically login?

Allow users to request forgotten passwords?

Configure Authentication

4. From the **How do users authenticate?** list, select the name used to log in.  
To enable 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 **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.
6. 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. From **Configuration Options**, select **User Settings** and then **Password Policies**.

The **Password Policies** page displays.

### Password Policies

**i** 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

#### 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  **?**
- Expiration Enabled **?**
- Maximum Age  **?**
- Warning Time  **?**
- Grace Limit  **?**

- (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** and then **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 19

---

# 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 user roles and access privileges](#)*

*[Creating a new Studio user](#)*

*[Editing a Studio user](#)*

*[Deactivating, reactivating, and deleting Studio users](#)*

## About user roles and access 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
<b>Administrator</b>	<p>Administrators have full access to all features in Studio.</p> <p>Administrators can:</p> <ul style="list-style-type: none"> <li>• Access the <b>Control Panel</b></li> <li>• Create and delete data sets and projects</li> <li>• Transform data within a project</li> <li>• View, configure, and manage all projects</li> </ul>
<b>Power User</b>	<p>Power users can:</p> <ul style="list-style-type: none"> <li>• Create and delete data sets and projects</li> <li>• Transform data within a project</li> <li>• Export data to HDFS and create new data sets</li> <li>• View, configure, and manage projects for which they have a project role</li> <li>• Edit their account information</li> </ul> <p>Power users cannot:</p> <ul style="list-style-type: none"> <li>• Access the <b>Control Panel</b></li> </ul>
<b>User</b>	<p>Users can:</p> <ul style="list-style-type: none"> <li>• Create and delete data sets and projects</li> <li>• Transform data within a project</li> <li>• View, configure, and manage projects for which they have a project role</li> <li>• Edit their account information</li> </ul> <p>Users cannot:</p> <ul style="list-style-type: none"> <li>• Access the <b>Control Panel</b></li> <li>• Export data to HDFS</li> </ul>



Role	Description
<b>Restricted User</b>	<p>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.</p> <p>Restricted users can:</p> <ul style="list-style-type: none"> <li>• Create new projects</li> <li>• View data sets in the Catalog</li> <li>• View, configure, and manage projects for which they have a project role</li> </ul> <p>Restricted users cannot:</p> <ul style="list-style-type: none"> <li>• Edit their account information</li> <li>• Access the <b>Control Panel</b></li> <li>• Create new data sets</li> <li>• Transform data within a project</li> <li>• Export data to HDFS</li> </ul>



**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
<b>Project Author</b>	<p>Project authors can:</p> <ul style="list-style-type: none"> <li>• Configure and manage a project</li> <li>• Add or remove users and user groups</li> <li>• Assign user and user group roles</li> <li>• Transform project data</li> <li>• Export project data</li> </ul> <p>Project authors cannot:</p> <ul style="list-style-type: none"> <li>• Create new data sets</li> <li>• Access the Big Data Discovery Control Panel</li> </ul>

Role	Description
<b>Project Restricted User</b>	<p>Project Restricted Users can:</p> <ul style="list-style-type: none"> <li>• View a project and navigate through the configured pages</li> <li>• Add and configure project pages and components</li> </ul> <p>Project restricted users cannot:</p> <ul style="list-style-type: none"> <li>• Access <b>Project Settings</b></li> <li>• Create new data sets</li> <li>• Transform data</li> <li>• Export project data</li> </ul>

### Data set access levels

In addition to the global feature access and project level access controlled by user roles and project roles, some deployments may require access controls at the data set level. Since data sets are a fundamental component of Big Data Discovery, this requires granting or denying access to data sets on a case-by-case basis.



**Note:** You cannot set permissions to "Default Access" or "No Access" for individual users, only for user groups.

Access Level	Description
<b>No Access</b> (User Groups only)	The user group cannot access the data set. The data set does not show up for this user or group in the Catalog.
<b>Default Access</b> (User Groups only)	The user group has default access to the data set. The "default" access level is set via the <code>df.defaultAccessForDerivedDataSets</code> setting on the Studio Settings page in the Control Panel.
<b>Read-only</b>	<p>Users with Read access to a data set can</p> <ul style="list-style-type: none"> <li>• See the data set in search results or by browsing the Catalog</li> <li>• Explore the data set</li> <li>• Add the data set to a project and modify it within the project</li> </ul>
<b>Read/Write</b>	<p>In addition to Read permissions, users with Write access to a data set can</p> <ul style="list-style-type: none"> <li>• Modify data set metadata such as description, searchable tags, and global attribute metadata</li> <li>• Manage access to the data set</li> </ul>

Users have No Access to any data set uploaded from a file by another user; only the file uploader and Studio Administrators have access, and both have the Read/Write permissions level.

As an example of using these access levels, you may wish to restrict default data set access "Read-only" and assign the "Default Access" level to all non-Administrative user groups. This gives all users the ability to add data sets to a project and modify them there. You can then create a "Data Curators" group that has Read/Write access to data sets in order to configure attribute metadata and data set details globally to make it easier for your users to navigate the Catalog. The group effectively becomes an additional level of permissions on top of whatever other access its users have.



**Important:** A user without any access to a data set can still explore the data they are a Project Restricted User or Project Author on a project that uses the data set. Project Authors can use the Transform operations to create a duplicate data set and gain access to the new data set. Similarly, a user with Read-only access to a data set can create a project using that data set and then execute transformations against the data if the default data set permissions include Write access. If you are working with sensitive information, consider this when assigning project roles and data set permissions.

## 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** and then **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 user roles and access privileges on page 127](#).

10. From the **Projects** section at the bottom of the dialog, to assign the user to projects:

Projects	Description	Project Role	Type
<input type="checkbox"/> asimoRegressionTests-11h4...		Project Restricted User	Private
<input type="checkbox"/> asimoSmokeTests-ie-14h14...		Project Restricted User	Private
<input type="checkbox"/> asimoRegressionTests-11h4...		Project Restricted User	Private
<input type="checkbox"/> asimoSmokeTests-ie-14h14...		Project Restricted User	Private
<input type="checkbox"/> 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** and then **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 20

---

# Integrating with an LDAP System to Manage Users

If you have an LDAP system, users can use their LDAP credentials to log in to Big Data Discovery. You can also configure BDD to communicate with the LDAP server over TLS/SSL.

*[About using LDAP](#)*

*[Configuring the LDAP settings and server](#)*

*[Authenticating against LDAP over TLS/SSL](#)*

*[Preventing encrypted LDAP passwords from being stored in BDD](#)*

*[Assigning roles based on LDAP user groups](#)*

## 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.

Users in LDAP must be contained in LDAP groups for Studio to properly map roles and permissions.

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	<p>Specifies a user-based import. This is the default value.</p> <p>User-based batch import uses the import search filter configured in the <b>User Mapping</b> section of the <b>LDAP</b> tab.</p> <p>For user-first import, Big Data Discovery:</p> <ol style="list-style-type: none"> <li>1. Uses the user import search filter to run an LDAP search query.</li> <li>2. Imports the resulting list of users, including all of the LDAP groups the user belongs to.</li> </ol> <p>The group import search filter is ignored.</p>
group	<p>Specifies a group-based import.</p> <p>Group-based import uses the import search filter configured in the <b>Group Mapping</b> section of the <b>LDAP</b> tab.</p> <p>For group-based import, Big Data Discovery:</p> <ol style="list-style-type: none"> <li>1. Uses the group import search filter to run an LDAP search query.</li> <li>2. Imports the resulting list of groups, including all of the users in those groups.</li> </ol> <p>The user import search filter is ignored.</p>

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 Studio header, click **Configuration Options** and select **Control Panel**.
2. Click **Credentials**.

3. Click **Authentication** and then **Configure Authentication** button.

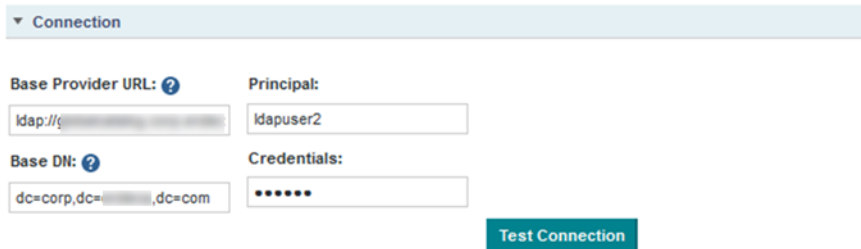
The **Configure Authentication** dialog displays with the **LDAP** tab selected.



4. To enable LDAP authentication, select **Enabled**.
5. To require users to log in only using an LDAP account, select **Required**.

If this is selected, then any users that you create manually in Studio cannot log in. To allow users you create manually to log in, deselect this option.

6. In **Provider type**, select the type of LDAP server you are connecting to.
7. Expand **Connection** and specify settings for the basic connection to LDAP:



Field	Description
<b>Base Provider URL</b>	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.
<b>Base DN</b>	The Base Distinguished Name for your LDAP directory. For a commercial organization, it may look something like: <code>dc=companynamehere,dc=com</code>
<b>Principal</b>	The user name of the administrator account for your LDAP system. This ID is used to synchronize user accounts to and from LDAP.



Field	Description
<b>Credentials</b>	The password for the administrative user.

8. After providing the connection information, click **Test Connection** to test the connection to the LDAP server.
9. Expand **User Mapping** and specify values for the following settings:

**▼ User Mapping**

Authentication Search Filter:  Password:  First Name:

(&(objectClass=person)(sAMA userPassword givenName

Import Search Filter: Screen Name: Last Name:

(&(objectClass=person)(\objec sAMAccountName sn

Email Address: Full Name: Group:

userprincipalname cn memberOf

**Test Users**

- (a) Use the search filter fields to configure the filters for finding and identifying users in your LDAP directory.

Field	Description
<b>Authentication Search Filter</b>	<p>The search criteria for user logins.</p> <p>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.</p> <p>By default, users log in using their email address. If you have changed this setting, you must modify the search filter here.</p> <p>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:</p> <pre>(cn=@screen_name@)</pre>

Field	Description
<b>Import Search Filter</b>	<p>The search filter to use for batch import of users.</p> <p>This filter is used if:</p> <ul style="list-style-type: none"> <li>You enable batch import of LDAP users</li> <li>In <code>portal-ext.properties</code>, <code>ldap.import.method</code> is set to <code>user</code></li> </ul> <p>Depending on the LDAP server, there are different ways to identify the user.</p> <p>The default setting (<code>objectClass=inetOrgPerson</code>) usually is fine, but to search for only a subset of users or for users that have different object classes, you can change this.</p>

(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**.

10. Under **Group Mapping**, map your LDAP groups.

▼ Group Mapping

<b>Import Search Filter:</b>	<b>Description:</b>
<input style="width: 95%;" type="text" value="(objectClass=group)"/>	<input style="width: 95%;" type="text" value="sAMAccountName"/>
<b>Group Name:</b>	<b>User:</b>
<input style="width: 95%;" type="text" value="cn"/>	<input style="width: 95%;" type="text" value="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.

11. The **Options** section is used to configure importing and exporting of LDAP user data and to select the password policy:

The screenshot shows a configuration window with a section titled "Options". Under this section, there are three checkboxes:
 

- 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.

## Authenticating against LDAP over TLS/SSL

To have Big Data Discovery Studio authenticate users against LDAP over TLS/SSL, export a certificate from your LDAP server and copy it to the `cacerts` keystore on the machine running Studio.

If your root Certificate Authority cert is issued internally by the company or if you have configured a self-signed certificate for your LDAP server, follow the steps below to export and copy it to the Java trust store on the machine running BDD Studio. If you are using a well-known commercial SSL CA certificate, it should already be present in the server's trust store and no further configuration is required.

To configure LDAP over TLS/SSL:

1. On your LDAP server, export the Root Certificate Authority certificate to DER encoded binary X.509 `.cer` file format.
2. Copy the exported `.cer` file to the `$BDD_HOME/common/security/cacerts` directory on the machine running BDD Studio.
3. Import the certificate to the `cacerts` keystore:

```
$JAVA_HOME/jre/bin/keytool -import -trustcacerts -keystore $BDD_HOME/common/security/cacerts -storepass <password> -noprompt -<alias> MyRootCA -file <keystore_filepath>
```

Where:

- `<password>` is the `cacerts` password. By default this is `changeit`.
- `<alias>` is the certificate's alias.

- `<keystore_filepath>` is the absolute path to the `.cer` file you copied over in Step 2.
4. Test your changes.

## 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 Studio.
2. Add the following settings to `portal-ext.properties`:

```
ldap.password.cache.hashing=false
ldap.auth.required=true
auth.pipeline.enable.liferay.check=false
```

3. Restart Studio.

Studio no longer stores the encrypted LDAP passwords for authenticated users. If the LDAP system is unavailable, Studio 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** and then **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 user roles and access privileges on page 127](#).

7. Click **Save**.

The group is added to the **User Groups** list.



## Chapter 21

---

# Setting Up Single Sign-On (SSO)

You can provide user access by integrating with an SSO system.

*[About using single sign-on](#)*

*[Overview of the process for configuring SSO with Oracle Access Manager](#)*

*[Configuring the reverse proxy module in OHS](#)*

*[Registering the Webgate with the Oracle Access Manager server](#)*

*[Testing the OHS URL](#)*

*[Configuring Big Data Discovery to integrate with SSO via Oracle Access Manager](#)*

*[Completing and testing the SSO integration](#)*

## 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 143](#).
6. Install the Webgate module into the Oracle HTTP Server. See [Registering the Webgate with the Oracle Access Manager server on page 144](#).
7. In Big Data Discovery, configure the LDAP connection for your SSO implementation. See [Configuring the LDAP connection for SSO on page 146](#).
8. In Big Data Discovery, configure the Oracle Access Manager SSO settings. See [Configuring the Oracle Access Manager SSO settings on page 147](#).
9. Configure Big Data Discovery's web server settings to use the OHS server. See [Completing and testing the SSO integration on page 148](#).
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 ( <code>http://host:port</code> ) 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** and then **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 **Provider type** 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 135](#).
7. Configure the user roles for your user groups as described in [Assigning roles based on LDAP user groups on page 140](#).
8. To save the LDAP connection information, click **Save**.

## 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.



To configure the SSO settings:

1. From the **Control Panel**, select **Platform Settings** and then **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`.

- 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:

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/
```

- 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:

- 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**

## **Logging for Studio, Dgraph, and Dgraph Gateway**



## Overview of BDD Logging

This topic provides a logging overview of the BDD components.

[List of Big Data Discovery logs](#)

[Gathering information for diagnosing problems](#)

[Retrieving logs](#)

[Rotating 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 <a href="#">Dgraph Gateway logs on page 173</a> .	\$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 <a href="#">Dgraph Gateway logs on page 173</a> .	\$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 <a href="#">Dgraph Gateway log entry format on page 175</a>	\$BDD_DOMAIN/servers/<serverName>/logs/<serverName>-diagnostic.log
Dgraph stdout/stderr log	Contains Dgraph operational messages, including startup messages. See <a href="#">Dgraph out log on page 167</a> .	\$BDD_HOME/logs/dgraph.out

Log	Purpose	Default Location
Dgraph request log	Contains entries for Dgraph requests. See <a href="#">Dgraph request log on page 166</a> .	\$BDD_HOME/dgraph/bin/dgraph.reqlog
Dgraph tracing ebb logs	Dgraph Tracing Utility files, which are especially useful for Dgraph crashes. See <a href="#">get-blackbox on page 38</a> .	\$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 <i>Data Processing Guide</i> .	\$BDD_HOME/logs/dgraphHDFSAGENT.out
Studio application log in Log4j format	Studio application log (in Log4j format). For both Studio application logs, see <a href="#">About the main Studio log file on page 160</a> .	\$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 <a href="#">About the metrics log file on page 160</a> .	\$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 <a href="#">About the Studio client log file on page 162</a> .	\$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 from Data Processing workflows. See the <i>Data Processing Guide</i> .	/tmp/edp/log/edp_*.log
Workflow Manager logs	Contains messages from Workflow Manager operations. See the <i>Data Processing Guide</i> .	\$BDD_HOME/logs/workflowmanager/diagnostic.log
Transform Service logs	Contains messages from transformation operations. See the <i>Data Processing Guide</i> .	\$BDD_HOME/logs/transformservice/<date>.stderrout.log



Log	Purpose	Default Location
Hadoop logs (YARN, Spark worker, and ZooKeeper logs)	YARN logs from the Spark processes that ran Data Processing workflows, as listed in the <i>Data Processing Guide</i> . See the Hadoop distribution documentation for information on the ZooKeeper logs.	Available from the Hadoop cluster manager.

## 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 <a href="#">Studio Logging on page 156</a> .
Data Processing	Data Processing is a component of BDD that runs on the YARN NodeManager nodes in the BDD deployment. For Data Processing logs, see the <i>Data Processing Guide</i> .
Dgraph Gateway (and WebLogic Server logs)	See <a href="#">Dgraph Gateway Logging on page 172</a> .
Dgraph	See <a href="#">Dgraph Logging on page 165</a> .
Dgraph HDFS Agent	The Dgraph HDFS Agent is responsible for importing and exporting Dgraph data to HDFS. For HDFS Agent logs, see the <i>Data Processing Guide</i> .

## Ways of accessing logs

You can access the logs for some components of Big Data Discovery through these commands of the `bdd_admin.sh` script:

- [get-logs](#)
- [set-log-levels on page 42](#)
- [rotate-logs on page 47](#)

## Gathering information for diagnosing problems

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

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

## Dgraph Gateway information

There are four areas of information to gather for Dgraph Gateway problems.

### 1: WebLogic standard Logs

Get the full contents of the following logs:

- WebLogic Admin Server server log
- WebLogic Admin Server domain log
- WebLogic Managed Server log
- Dgraph Gateway application log

For the name of the logs, see [List of Big Data Discovery logs on page 151](#).

### 2: Config file

Get the `config.xml` in the `$BDD_DOMAIN/config` directory.

### 3: Thread dumps

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

The **jps -mlv** command lists all running JVMs. You can use this format to obtain the WebLogic PID:

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

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

```
jps -mlv | fgrep weblogic
7769 weblogic.Server -Xms1024m -Xmx4096m -XX:MaxPermSize=1024m -Dweblogic.Name=AdminServer
...
```

In the example, 7769 is the WebLogic JVM PID.

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

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

For example:

```
jstack -l 7769 jstack.weblogic.outIt
```

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

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

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

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

### 4: Heap dumps

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

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

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

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

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

## Dgraph information

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

### 1: Logs for performance issues

Collect the following information:

- `dgraph.reqlog` log from the `$BDD_HOME/logs` directory
- `WebLogic.access.log`
- Dgraph blackbox file, from the `bdd-admin get-blackbox` command
- Dgraph Statistics output, from the `bdd-admin get-stats` command
- BDD version, from the `bdd-admin --version` command
- Dgraph version, from the `dgraph --version` command

### 2: Logs for Dgraph crashes

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

- `dgraph.out` log from the `$BDD_HOME/logs` directory
- Dgraph core dump file

### 3: Logs for other correctness issues

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

- Dgraph databases for the data sets (the Dgraph databases are stored in the directory specified by the `DGRAPH_INDEX_DIR` property in the `bdd.conf` file)
- `dgraph.reqlog` log from the `$BDD_HOME/logs` directory
- `dgraph.out` log from the `$BDD_HOME/logs` directory
- Dgraph version, from the `dgraph --version` command

### 4: ZooKeeper logs

The ZooKeeper log and the ZooKeeper transaction logs are valuable to help diagnose Dgraph problems that may result from leader/follower issues. These logs can be retrieved as part of the `bdd-admin get-logs` command output.

### 5: Changing Dgraph flags

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

Dgraph flags are set by various properties in the `bdd.conf` file. For example, the `DGRAPH_THREADS` property sets the number of threads for the Dgraph. The `DGRAPH_ADDITIONAL_ARG` property is especially useful as it allows you to add new flags, such as the `-v` flag. For details on changing these properties, see [Configuration properties that can be modified on page 50](#).

## Topology information

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

- Hardware specifications and configuration of the machines.
- Description of the Dgraph Gateway and Dgraph topology (number and names of servers in the BDD cluster and number of Dgraph nodes).
- Description of which Dgraph Gateway nodes and Dgraph nodes are affected.
- Network topology.

## 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 44](#).

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
```

## Rotating logs

Dgraph Gateway and Studio logs are hardcoded to rotate daily. You can force rotate logs by running the `bdd-admin` script with the `rotate-logs` command.

For 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 47](#).



## Chapter 23

# Studio Logging

---

Studio logging helps you to monitor and troubleshoot your Studio application.

[About logging in Studio](#)

[About the Log4j configuration XML files](#)

[About the main Studio log file](#)

[About the metrics log file](#)

[Configuring the amount of metrics data to record](#)

[About the Studio client log file](#)

[Adjusting Studio logging levels](#)

[Using the Performance Metrics page to monitor query performance](#)

## 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. The log rotation frequency is set to daily (it is hard-coded, not configurable).

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: 0000Kvsw8S17ADkpSw4EyclLjsrN0000^6,0] 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): <ul style="list-style-type: none"> <li>• INCIDENT_ERROR</li> <li>• ERROR</li> <li>• WARNING</li> <li>• NOTIFICATION</li> <li>• TRACE</li> <li>• UNKNOWN</li> </ul>	[ 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: 0000Kvsw8S17ADkpSw4EyclLjsrN0000^6,0 ]
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): <ul style="list-style-type: none"> <li>• FATAL</li> <li>• ERROR</li> <li>• WARN</li> <li>• INFO</li> <li>• DEBUG</li> </ul>	[ INFO ]
Module ID	The Java class that prints the message entry.	[UserSession]
Message Text	The text of the log message.	UserSession created

## 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
<b>Total duration (msec)</b>	The total time for this entry (End time minus Start time).
<b>Start time (msec since epoch)</b>	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.
<b>End time (msec since epoch)</b>	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.
<b>Session ID</b>	The session ID for the client.



Column Name	Description
<b>Page ID</b>	<p>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.</p> <p>Some actions that do not affect the overall state of a page, such as displaying attributes on the <b>Available Refinements</b> panel, do not increment this counter.</p>
<b>Gesture ID</b>	The full count of requests to the server.
<b>Portlet ID</b>	<p>This is the ID associated with an individual instance of a component.</p> <p>It generally includes:</p> <ul style="list-style-type: none"> <li>• The type of component</li> <li>• A unique identifier</li> </ul> <p>For example, if a page includes two <b>Chart</b> components, the ID can be used to differentiate them.</p>
<b>Entry Type</b>	<p>The type of entry. For example:</p> <ul style="list-style-type: none"> <li>• PORTLET_RENDER - Server execution in response to a full refresh of a component</li> <li>• DISCOVERY_SERVICE_QUERY - Dgraph Gateway query</li> <li>• CONFIG_SERVICE_QUERY - Configuration service query</li> <li>• SCONFIG_SERVICE_QUERY - Semantic configuration service query</li> <li>• LQL_PARSER_SERVICE_QUERY - EQL parser service query</li> <li>• CLIENT - Client side JavaScript execution</li> <li>• PORTLET_RESOURCE - Server side request for resources</li> <li>• PORTLET_ACTION - Server side request for an action</li> </ul>
<b>Miscellaneous</b>	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** and then **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.

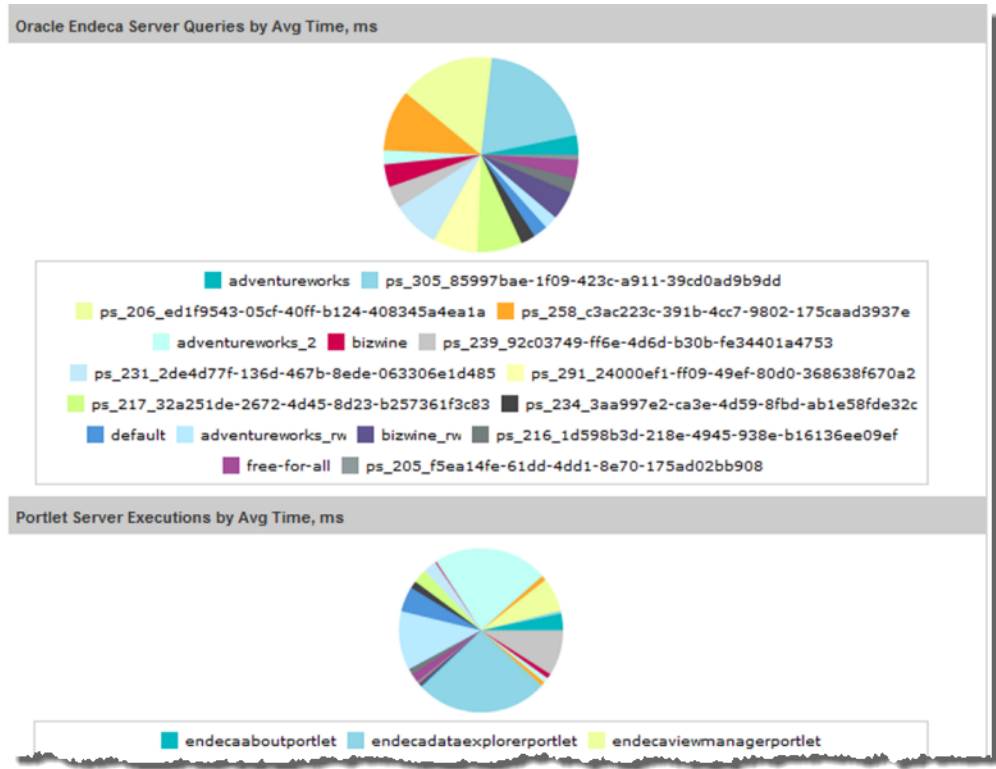
**Performance Metrics**

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-f6...	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.



**Note:** Dgraph Gateway query performance does not correlate directly to a project page, as a single page often uses multiple Dgraph Gateway queries.



## Chapter 24

# Dgraph Logging

---

This section describes the Dgraph logs.

*Dgraph request log*

*Dgraph out log*

## 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 a Dgraph database. 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.

## Dgraph 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	<p>The local date and time when the message was generated, using use the following ISO 8601 extended format:</p> <pre>YYYY-MM-DDTHH:mm:ss.sss(+ -)hh:mm</pre> <p>The hours range is 0 to 23 and milliseconds and offset timezones are mandatory.</p>	2016-03-18T13:25:30.600-04:00
Component ID	The ID of the component that originated the message. "DGRAPH" is hard-coded for the Dgraph.	DGRAPH
Message Type	<p>The type of message (log level):</p> <ul style="list-style-type: none"> <li>• INCIDENT_ERROR</li> <li>• ERROR</li> <li>• WARNING</li> <li>• NOTIFICATION</li> <li>• TRACE</li> <li>• UNKNOWN</li> </ul>	WARNING
Log Subsystem	The log subsystem that generated the message.	{dgraph}
Job ID	The ID of the job being executed.	[0]



Log entry field	Description	Example
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 database maintenance activity.
- `bulk_ingest` — messages generated by Bulk Load ingest operations.
- `cluster` — messages about ZooKeeper-related cluster operations.
- `database` — messages about Dgraph database operations.
- `datalayer` — messages about Dgraph database file usage.
- `dgraph` — messages related to Dgraph general operations.
- `eql` — messages generated from the EQL, the engine for the BDD query language.
- `eql_feature` — messages providing usage information for certain EQL features.
- `eve` — messages generated from the BDD query evaluator.
- `http` — messages about Dgraph HTTP communication operations.
- `lexer` — messages from the OLT (Oracle Language Technology) subsystem.
- `splitting` — messages resulting from BDD query evaluator's 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 startup information

The first log entry (that begins with "Starting Dgraph") lists the Dgraph version, startup flags and arguments, and path to the Dgraph databases directory. Later entries log additional start-up information, such as the amount of RAM and the number of logical CPUs on the system, the CPU cache topology, the created Web services, HTTP port number, and Bulk Load port number.

## Dgraph shutdown information

As part of the Dgraph shutdown process, the shutdown details are logged, including the total amount of time for the shutdown. For example (note that timestamps have been removed for ease of reading):

```
DGRAPH NOTIFICATION {dgraph} [0] Shutdown request received at Wed Oct 5 16:17:42
2016. Shutdown will complete when all outstanding
jobs are complete.
DGRAPH WARNING {cluster} [0] Lost connection to ZooKeeper: ZooKeeper connection lost
(zk error -4)
DGRAPH NOTIFICATION {cluster} [0] Finished closing zk connection
DGRAPH NOTIFICATION {database} [0] Finished unmounting everything.
DGRAPH NOTIFICATION {dgraph} [0] All dgraph transactions completed at Wed Oct 5
16:17:43 2016, exiting normally (pid=14789)
DGRAPH NOTIFICATION {database} [0] Finished unmounting everything.
DGRAPH NOTIFICATION {dgraph} [0] Overall shutdown took 922 ms
```

## Out log ingest example

The following snippets from a Dgraph out log show the entry format for an ingest operation. Note that timestamps have been removed for ease of reading.

```
DGRAPH NOTIFICATION {cluster} [0] Promoting to leader on database edp_cli_edp_c23fdc4c
DGRAPH NOTIFICATION {database} [0] Mounting database edp_cli_edp_c23fdc4c
DGRAPH NOTIFICATION {dgraph} [0] Initial DL version: 2
DGRAPH NOTIFICATION {bulk_ingest} [0] MessageParser created
DGRAPH NOTIFICATION {bulk_ingest} [0] Start ingest for collection: edp_cli_edp_c23fdc4c
for database edp_cli_edp_c23fdc4c
DGRAPH NOTIFICATION {bulk_ingest} [0] Starting bulk ingest operation REPLACE_RECORDS
for database edp_cli_edp_c23fdc4c
DGRAPH NOTIFICATION {bulk_ingest} [0] Ending bulk ingest for database edp_cli_edp_c23fdc4c
at client's request
DGRAPH NOTIFICATION {bulk_ingest} [0] Bulk ingest completed: Added 351 records and
rejected 0 records, for database edp_cli_edp_c23fdc4c
DGRAPH NOTIFICATION {bulk_ingest} [0] Final statistics: 0.060 MiB, 1.008 records seconds,
0.060 MiB/s records throughput, 1.182 total seconds,
0.051 MiB/s total throughput for database
edp_cli_edp_c23fdc4c
```

The `bulk_ingest` entries show the ingest of a data set with 351 records.

### [Dgraph log levels](#)

#### [Setting the Dgraph log levels](#)

## Dgraph log levels

This topic describes the Dgraph log levels.

The Dgraph uses Oracle Diagnostic Logging (ODL) for logging. The Dgraph loggers are configured with the amount and type of information written to log files, by specifying the log level. When you specify the log level, the logger returns all messages of that type, as well as the messages that have a higher severity. For example, if you set the log level to `WARNING`, the logger also returns `INCIDENT_ERROR` and `ERROR` messages.

The following table lists the Dgraph log levels and their descriptions.

Dgraph log level	Description
INCIDENT_ERROR	A serious problem that may be caused by a bug in the product and that should be reported to Oracle Support. \
ERROR	A serious problem that requires immediate attention from the administrator and is not caused by a bug in the product.
WARNING	A potential problem that should be reviewed by the administrator.
NOTIFICATION	A major lifecycle event such as the activation or deactivation of a primary sub-component or feature.
NOTIFICATION:16	A finer level of granularity for reporting normal events.
TRACE	Trace or debug information for events that are meaningful to administrators, such as public API entry or exit points.
TRACE:16	Detailed trace or debug information that can help Oracle Support diagnose problems with a particular subsystem.
TRACE:32	Very detailed trace or debug information that can help Oracle Support diagnose problems with a particular subsystem.

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

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

## Setting 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 subsystems 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 out log on page 167](#).

- *level* is one of these log levels:
  - INCIDENT\_ERROR
  - ERROR
  - WARNING
  - NOTIFICATION
  - NOTIFICATION:16
  - TRACE
  - TRACE:16
  - TRACE:32

The pipe character is required if you are setting more than one subsystem/level combination.

To set the Dgraph log levels:

1. Make a copy of `$BDD_HOME/BDD_manager/conf/bdd.conf` in a different directory.
2. Modify the `DGRAPH_LOG_LEVEL` property in the copy to set the required log levels.
3. 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 30](#).

4. Restart the Dgraph by running the `bdd-admin` script with the `restart` command.

For details on this command, see [restart on page 22](#).

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_HOME/BDD_manager/conf/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 42](#).



This section describes the Dgraph Gateway logs.

[Dgraph Gateway logs](#)

[Dgraph Gateway log entry format](#)

[Log entry information](#)

[Logging properties file](#)

[Setting the Dgraph Gateway log level](#)

[Customizing the HTTP access log](#)

## 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](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 160](#).

## 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:

```
[2016-03-29T06:23:05.360-04:00] [EndecaServer] [ERROR] [OES-000066]
[com.endeca.features.ws.ConfigPortImpl] [host: bus04.example.com] [nwaddr: 10.152.104.14]
[tid: [ACTIVE].ExecuteThread: '7' for queue: 'weblogic.kernel.Default (self-tuning)']
[userId: nsmith] [ecid: 0000LF1tV0X7y0kpSwXBic1My_Qv00002I,0] OES-000066: Service error:
java.lang.Exception: OES-000188: Error contacting the config service on dgraph
http://bus04.example.com:7010: Database 'default_edp_f9332e56-2c29-4b77-bbf0-25730a5368bc'
does not exist.
```

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.	[2016-03-29T06:23:05.360-04:00]
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): <ul style="list-style-type: none"> <li>INCIDENT_ERROR</li> <li>ERROR</li> <li>WARNING</li> <li>NOTIFICATION</li> <li>TRACE</li> </ul>	[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-000066]
Module ID	The Java class that prints the message entry.	[com.endeca.features.ws.ConfigPortImpl]

Log entry field	Description	Example
Host name	The name of the host where the message originated.	[host: bus04.example.com]
Host address	The network address of the host where the message originated	[nwaddr: 10.152.104.14]
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: nsmith]
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^Aye1JxPgK00000,0]
Message Text	The text of the log message.	OES-000066: Service error: ...]

## 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
```

## Logging levels

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

ODL Log Level	Meaning
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	Indicates a serious problem that requires immediate attention from the administrator and is not caused by a bug in the product.
WARNING	Indicates a potential problem that should be reviewed by the administrator.



ODL Log Level	Meaning
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.
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 `NOTIFICATION` level to get most of the messages, and change to a less verbose level in production.

## 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

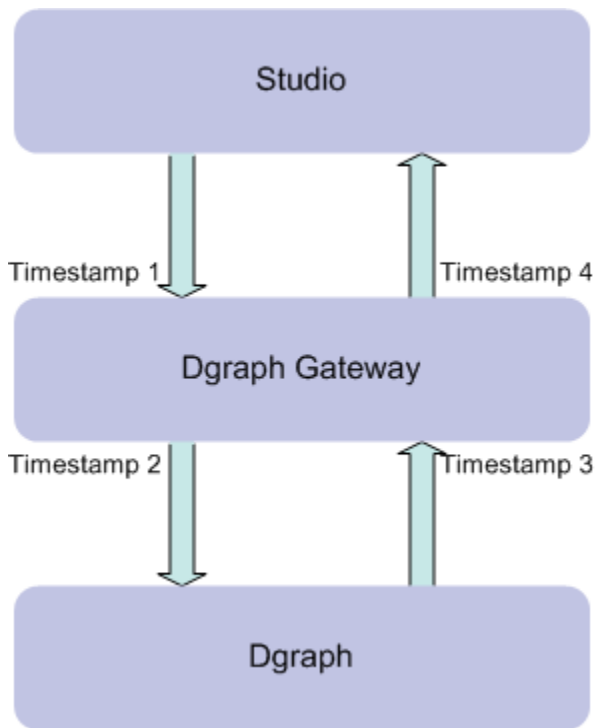
For ingest operations, a start and end timestamp is logged. At the end of the operation, the ingest results are also logged (number of added records, number of deleted records, number of updated records, number of replaced records, number of added or updated records).

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 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).

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.server.logging.ODLAppender
log4j.appender.ODL.MaxSize=1048576000
log4j.appender.ODL.MaxSegmentSize=104857600
log4j.appender.ODL.encoding=UTF-8
log4j.appender.ODL.MaxDaysToRetain=7

# Zookeeper client log level
log4j.logger.org.apache.zookeeper=WARN

```

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

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.
log4j.appender.stdout.layout.ConversionPattern	<p>Defines the log entry conversion pattern as:</p> <ul style="list-style-type: none"> <li>• <b>%d</b> is the date of the logging event.</li> <li>• <b>%p</b> outputs the priority of the logging event.</li> <li>• <b>%c</b> outputs the category of the logging event.</li> <li>• <b>%L</b> outputs the line number from where the logging request was issued.</li> <li>• <b>%m</b> outputs the application-supplied message associated with the logging event while <b>%n</b> is the platform-dependent line separator character.</li> </ul> <p>For other conversion characters, see: <a href="https://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/PatternLayout.html">https://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/PatternLayout.html</a></p>

Logging property	Description
<code>log4j.appender.ODL=com.endeca.util.ODLAppender</code>	Defines ODL as an ODL Appender. ODL (Oracle Diagnostics Logging) is the logging format for Oracle applications.
<code>log4j.appender.ODL.MaxSize</code>	Sets the maximum amount of disk space to be used by the <code>&lt;ServerName&gt;-diagnostic.log</code> file and the logging rollover files. The default is 1048576000 (about 1GB). Older log files are deleted to keep the total log size under the given limit.
<code>log4j.appender.ODL.MaxSegmentSize</code>	Sets the maximum size (in bytes) of the log file. When the <code>&lt;ServerName&gt;-diagnostic.log</code> file reaches this size, a rollover file is created. The default is 104857600 (about 10 MB).
<code>log4j.appender.ODL.encoding</code>	Sets character encoding the log file. The default UTF-8 value prints out UTF-8 characters in the file.
<code>log4j.appender.ODL.MaxDaysToRetain</code>	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.
<code>log4j.logger.org.apache.zookeeper</code>	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.

## Changing the ZooKeeper client log level

You can change the ZooKeeper client log level to another setting, as in this example:

```
log4j.logger.org.apache.zookeeper=INFO
```

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

- OFF
- FATAL
- ERROR

- WARN
- INFO
- DEBUG

## Setting the Dgraph Gateway log level

Use the `bdd-admin` script with the `set-log-levels` command to set the log level for the Dgraph Gateway.

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 `WARNING`, WebLogic also returns messages of type `ERROR` and `INCIDENT_ERROR`.

The `ENDECA_SERVER_LOG_LEVEL` property in `bdd.conf` sets the log level for the Dgraph Gateway at start-up time. The `set-log-levels` command lets you change the current log-level setting. This change can be persisted for subsequent re-starts of the Dgraph Gateway.

The `set-log-levels` command syntax is:

```
./bdd-admin.sh set-log-levels --component gateway --level <level> [--non-persistent]
```

where:

- `--component` (abbreviated `-c`) specifies `gateway` as the component to be modified.
- `--level` (abbreviated `-l`) specifies the new log level. *level* is one of these log levels:
  - `INCIDENT_ERROR`
  - `ERROR`
  - `WARNING`
  - `NOTIFICATION`
  - `TRACE`

The new log level may persist into the next Dgraph Gateway 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 Gateway re-start, at which time the log level in the `ENDECA_SERVER_LOG_LEVEL` property is used.
- If `--non-persistent` is omitted, the new setting is persisted by being written to the `ENDECA_SERVER_LOG_LEVEL` property in `bdd.conf`. This means that the next Dgraph Gateway re-start will use the changed the log level in the `bdd.conf` file.

For additional usage information, see [set-log-levels on page 42](#).

To set the Dgraph Gateway log level:

1. Navigate to the `$BDD_HOME/BDD_manager/bin` directory.
2. Run the `bdd-admin` script with the `set-log-levels` command. For example:

```
./bdd-admin.sh set-log-levels --component gateway --level WARNING
```

Note that the `set-log-levels` command cannot change the setting of the `log4j.logger.org.apache.zookeeper` package. For information on setting this package, see [Changing the ZooKeeper client log level on page 180](#).

## 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** and then **HTTP**.
6. On the **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 22](#).

# Index

## A

Admin Server, about 14

## B

backing up Big Data Discovery 67

bdd-admin 16

  autostart 23

  backup 24

  disable-components 37

  enable-components 36

  flush 35

  get-blackbox 38

  get-log-levels 41

  get-logs 44

  get-stats 39

  publish-config 30

  publish-config, bdd 31

  publish-config, cert 33

  publish-config, database 34

  publish-config, hadoop 31

  publish-config, kerberos 32

  reset-stats 40

  reshape-nodes 36

  restart 22

  restore 27

  rotate-logs 47

  set-log-levels 42

  start 19

  status 38

  stop 20

  update-model 34

bdd.conf

  properties that can be modified 50

  updating 49

bdd user, about 14

## C

cgroups 85

## D

database configuration, updating 60

data connections

  about 95

  creating 95

  deleting 96

  editing 95

Data Enrichment models, updating 34

Data Processing, about 11

Data Processing nodes

  adding 65

  removing 66

Data Source Library

  data connections, creating 95

  data connections, deleting 96

  data connections, editing 95

  data sources, creating 96

  data sources, deleting 97

  data sources, editing 97

data sources

  about 95

  creating 96

  deleting 97

  details, displaying 97

  editing 97

Dgraph

  about 12, 75

  adding nodes 63

  cache size 84

  cgroups 85

  databases 75

  databases, moving 77

  flags 87

  flushing the cache 35

  HDFS data at rest encryption 76

  log levels 170

  memory consumption 81

  modifying memory limit 83

  out log 167

  request log 166

  setting log level 171

  statistics 82

  Tracing Utility 82

Dgraph Gateway

  about 12

  flushing the cache 35

  logging configuration 178

  logs 173

  setting log level 181

Dgraph HDFS Agent

  about 12

  flags 92

DP CLI, about 12

## E

email notifications

  Account Created Notification, configuring 118

  Password Changed Notification,

  configuring 118

  sender, configuring 118

  server, configuring 117

## F

framework settings

  list of 98



**G**

gathering information for diagnosing problems 153

**H**

## Hadoop

- client configuration files, updating 54
- Hue URI, setting 54
- upgrading 55

## Hadoop settings

- configuring 103
- list of 102

Hive Table Detector, about 12

HTTP access log 182

Hue URI, setting 54

**K**

## Kerberos

- enabling 57
- keytab file, updating 59
- krb5.conf, changing the location of 59
- principal, updating 60

**L**

## LDAP integration

- preventing passwords from being stored 140
- roles, assigning based on groups 140
- server connection, configuring 135
- settings, configuring 135

## locales

- configuring the default 113
- configuring user preferred 114
- effect of selection 111
- list of supported 111
- locations where set 112
- scenarios for determining 112

## logging

- list of available logs 151
- Log4j configuration files, about 159
- main Studio log file 160
- metrics data, configuring 161
- metrics log file, about 160
- Performance Metrics page 163
- Studio client log 162
- verbosity, adjusting from the Control Panel 163

## logs

- Dgraph Gateway 173
- Dgraph out 167
- Dgraph request 166
- retrieving 156
- rotate-logs 156
- rotating 156
- WebLogic HTTP access log 182

**M**

MySQL restoration, troubleshooting 72

**O**

Oracle MapViewer settings in Studio 98

**P**

## passwords

- existing user, changing for 132
- new user, setting for 131
- password policy, configuring 125

Performance Metrics page 163

## project roles

- about 129
- types of 129

## projects

- certifying 121
- deleting 122
- existing user, changing membership 132
- making active or inactive 121
- new user, assigning membership to 132
- project type, configuring 120
- roles 129

**R**

## restoring BDD

- data-only 70
- full 68

## roles

- existing user, changing 132
- groups, assigning to for LDAP 140
- new user, assigning 131
- project roles 129
- user roles, editing 127
- user roles, list of 127

**S**

## single sign-on

See SSO

## SSO

- about 142
- LDAP connection, configuring 146
- OHS URL, testing 145
- Oracle Access Manager settings, configuring in Big Data Discovery 147
- overview of the integration process 142
- portal-ext.properties, configuring 148
- reverse proxy configuration, WebLogic Server 143
- Webgate, registering with Oracle Access Manager 144

## Studio

- about 11
- creating users 131
- database password 101
- Data Processing settings 102
- email configuration 117
- framework settings 98
- health check 105

- Hue integration, enabling 54
- locales 111
- logging 157
- session timeout, modifying 100
- setting time zone 115

- Studio settings
  - configuring 100

- system backup 67

- System Usage
  - sections, about 107
  - usage logs, adding entries 106
  - using 108

## T

- time zone, Studio 115

- TLS/SSL certificates, refreshing 61

- Transform Service, about 11

## U

- users

- authentication settings, configuring 124

- creating 131

- deactivating 133

- deleting 133

- editing 132

- email addresses, listing restricted 126

- reactivating 133

- screen names, listing restricted 126

## W

- WebLogic logs

- AdminServer 173

- HTTP access log 182

- Workflow Manager Service, about 11

## Z

- ZooKeeper client log level 180