

Oracle® Big Data Discovery Cloud Service

Administrator's Guide

E65370-05

November 2016

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

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.

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, then 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.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about 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 unless otherwise set forth in an applicable agreement between you and Oracle. 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, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface	vii
About this guide	vii
Audience	vii
Conventions	vii
Contacting Oracle Customer Support	viii
 Part I Administering Studio	
 1 Managing Data Sources	
About database connections and JDBC data sources	1-1
Creating data connections	1-2
Deleting data connections	1-2
Creating a data source	1-2
Editing a data source	1-3
Deleting a data source	1-3
 2 Configuring Studio Settings	
Studio settings in BDDCS	2-1
Changing the Studio setting values	2-2
Modifying the Studio session timeout value	2-2
Changing the Studio database password	2-3
Viewing the Server Administration Page information	2-3
 3 Configuring Data Processing Settings	
List of Data Processing Settings	3-1
Changing the data processing settings	3-3
 4 Running a Studio Health Check	
 5 Viewing Project Usage Summary Reports	
About the project usage logs	5-1
About the System Usage page	5-1

Using the System Usage page	5-3
6 Configuring the Locale and Time Zone	
Locales and their effect on the user interface.....	6-1
How Studio determines the locale to use.....	6-2
Locations where the locale may be set	6-2
Scenarios for selecting the locale.....	6-2
Selecting the default locale	6-3
Configuring a user's preferred locale.....	6-4
Setting the default time zone	6-6
7 Managing Projects from the Control Panel	
Configuring the project type	7-1
Assigning users and user groups to projects.....	7-2
Certifying a project	7-2
Making a project active or inactive	7-3
Deleting projects	7-3
Part II Controlling User Access to Studio	
8 Configuring User-Related Settings	
Configuring authentication settings for users	8-1
Configuring the password policy.....	8-2
Restricting the use of specific screen names and email addresses	8-3
9 Creating and Editing Studio Users	
About user roles and access privileges.....	9-1
Creating a new Studio user	9-4
Editing a Studio user	9-5
Deactivating, reactivating, and deleting Studio users.....	9-6
Part III Logging for Studio, Dgraph, and Dgraph Gateway	
10 Overview of BDD Logging	
List of Big Data Discovery logs.....	10-1
Troubleshooting errors in Big Data Discovery Cloud Service	10-3
11 Studio Logging	
About logging in Studio	11-1
About the Log4j configuration XML files.....	11-3
About the main Studio log file.....	11-4
About the metrics log file	11-4
Configuring the amount of metrics data to record	11-5

About the Studio client log file	11-6
Adjusting Studio logging levels	11-7
Using the Performance Metrics page to monitor query performance	11-7
12 Dgraph Logging	
Dgraph request log	12-1
Dgraph out log	12-3
Dgraph log levels.....	12-6
FUSE out log.....	12-7
13 Dgraph Gateway Logging	
Dgraph Gateway logs.....	13-1
Dgraph Gateway log entry format	13-3
Log entry information	13-4
Logging properties file.....	13-6

Index

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
User Interface Elements	This formatting is used for graphical user interface elements such as pages, dialog boxes, buttons, and fields.
<code>Code Sample</code>	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> .
<code>File Path</code>	This formatting is used for file names and paths.

Symbol conventions

The following table describes symbol conventions used in this document.

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

Path variable conventions

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

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

Contacting Oracle Customer Support

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

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

Part I

Administering Studio

Managing Data Sources

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

About database connections and JDBC data sources

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

Creating data connections

To create a data connection, follow the steps below.

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.

Creating a data source

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

Editing a data source

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

Deleting a data source

To delete a data source, follow the steps below.

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 > Control Panel** and navigate to **Big Data Discovery > Data Source Library**.
3. Click **+ Connection**.
4. On the **New data connection** dialog, provide the name, URL, and authentication information for the data connection.
5. Click **Save**.

Deleting data connections

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

To delete a data connection:

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

Creating a data source

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

To create a data source:

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

Studio does not control the order of the records. The SQL statement can indicate the order of records to import using an ORDER BY clause.

7. In the text area, enter the SQL query to retrieve the records for the data source, then click **Next**.

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

8. Click **Save**.

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

Editing a data source

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

Displaying details for a data source

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

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

Editing a data source

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

Deleting a data source

To delete a data source, follow the steps below.

To delete a data source:

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

Configuring Studio Settings

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

[Studio settings in BDDCS](#)

In BDD Cloud Service, you can edit these Studio settings. The other settings, even though visible, should not be modified.

[Changing the Studio setting values](#)

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

[Modifying the Studio session timeout value](#)

The timeout notification that appears in the header of Studio is controlled by two settings: `session.timeout` in `portal-ext.properties` and the `web.xml` settings in the WebLogic Server running Studio.

[Changing the Studio database password](#)

Studio requires a relational database to store configuration and state, including component configuration, user permissions, and system settings. When you create a BDDCS instance, the Studio database with a corresponding username and password is created for you.

[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.

Studio settings in BDDCS

In BDD Cloud Service, you can edit these Studio settings. The other settings, even though visible, should not be modified.

Setting	Description
<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.

Setting	Description
<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 display 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.defaultCurrencyList</code>	A comma-separated list of currency symbols to add to the ones currently available.
<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.

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 > 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 is controlled by two settings: `session.timeout` in `portal-ext.properties` and the `web.xml` settings in the WebLogic Server running Studio.

The values for these settings should be the same. In other words, if you set the timeout to 30 minutes in `portal-ext.properties`, it should match in `web.xml`. By default, `session.timeout=30` in `portal-ext.properties`.

To modify the Studio session timeout value :

1. Stop Studio, by locating the Studio Java application in the WebLogic Server Admin console on the BDCS node, and stopping it.
2. On the same WebLogic server, open `$DOMAIN_HOME/config/studio/portal-ext.properties` and modify the following settings:

```
session.timeout=30
```


3. Restart Studio in the WebLogic Server Admin Console.
4. On the same WebLogic Server, modify the Studio timeout in `web.xml` to match step 2.

If you are not familiar with modifying this file, see the WebLogic Server Administration documentation.

Changing the Studio database password

Studio requires a relational database to store configuration and state, including component configuration, user permissions, and system settings. When you create a BDDCS instance, the Studio database with a corresponding username and password is created for you.

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 > 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.

4. Change it in the BDDCS dashboard.

- a. Select the **Update Credentials** option in the BDDCS dashboard in Cloud MyServices
- b. Specify a new username and password.

For more information, see the *BDDCS Getting Started Guide*.

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.

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

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

Changing the data processing settings

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

List of Data Processing Settings

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

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

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

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>	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> .

Note: Editing this setting is supported in BDD Cloud Service.

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. 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> <div>Note: Editing this setting is supported in BDD Cloud Service.</div>
<code>bdd.maxSplitSize</code>	<p>The maximum partition size for Spark jobs measured in MB. This controls the size of the blocks of data handled by Data Processing jobs. Partition size directly affects Data Processing performance — when partitions are smaller, more jobs run in parallel and cluster resources are used more efficiently. This improves both speed and stability.</p> <p>The default is set by the <code>MAX_INPUT_SPLIT_SIZE</code> property in the <code>bdd.conf</code> file (which is 32, unless changed by the user). The 32MB is amount should be sufficient for most clusters, with a few exceptions:</p> <ul style="list-style-type: none">• If your Hadoop cluster has a very large processing capacity and most of your data sets are small (around 1GB), you can decrease this value.• In rare cases, when data enrichments are enabled the enriched data set in a partition can become too large for its YARN container to handle. If this occurs, you can decrease this value to reduce the amount of memory each partition requires. <p>Note that this property overrides the HDFS block size used in Hadoop.</p>

Data Processing Topology

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

Hadoop Setting	Description
Hadoop Admin Console	The hostname and Admin Console port of the machine that acts as the Master for your Hadoop cluster.
Name Node	The NameNode internal Web server and port.

Hadoop Setting	Description
Hive metastore Server	The Hive metastore listener and port.
Hive Server	The Hive server listener and port.
Hue Server	The Hue Web interface server and port.
Cluster OLT Home	The OLT home directory in the BDD cluster. The BDD installer detects this value and populates the setting.
Database Name	The name of the Hive database that stores the source data for Studio data sets.
EDP Data Directory	The directory that contains the contents of the <code>edp_cluster_*.zip</code> file on each worker node.
Sandbox	The HDFS directory in which to store the avro files created when users export data from Big Data Discovery. The default value is <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 > Data Processing Settings**.
3. For each setting, update the value as necessary.
4. Click **Update Settings**.

The changes are applied immediately.

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 set up 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.

Viewing Project Usage Summary Reports

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

[About the project usage logs](#)

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

[About the System Usage page](#)

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

[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.

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.

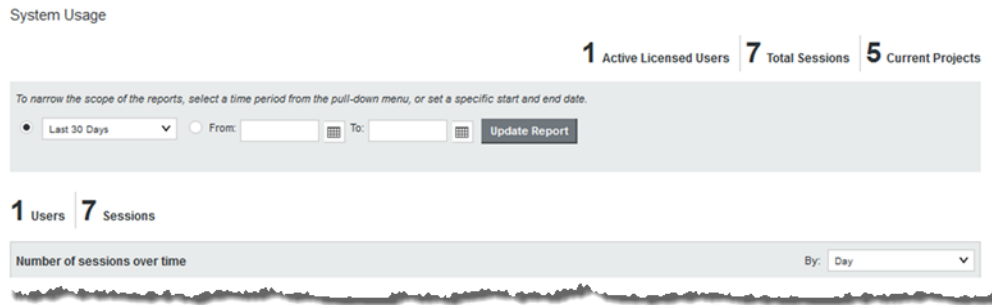
To change the age of the entries to delete, add the following setting to `portal-ext.properties`:

```
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
Summary totals	At the top right of the page are the total number of: <ul style="list-style-type: none"> • Users in the system • Sessions that have occurred • Projects
Date range fields	Contains fields to set the range of dates for which to display report data.
Current number of users and sessions	Lists the number of users that were logged in and the number of sessions for the date range that you specify.
Number of sessions over time	Report showing the number of sessions that have been active for the date range that you specify Includes a list to set the date unit to use for the chart.
User Activity	Report that initially shows the top 10 number of sessions per user for the selected date range across all projects. You can click on any bars in this chart to drill down into the reporting data. At the top of the report are lists to select: <ul style="list-style-type: none"> • A specific user, or all users • A specific project, or all projects • Whether to display the top or bottom values (most or least sessions) • The number of values to display
Project Usage	Report that initially shows the top 10 number of sessions per project for the selected date range across all projects. You can click on any bars in this chart to drill down into the reporting data. At the top of the report are lists to select: <ul style="list-style-type: none"> • A specific project, or all projects • Whether to display the top or bottom values (most or least sessions) • The number of values to display
System	Contains a pie chart that shows the relative number of sessions by browser type and version for the selected date range.

Using the System Usage page

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

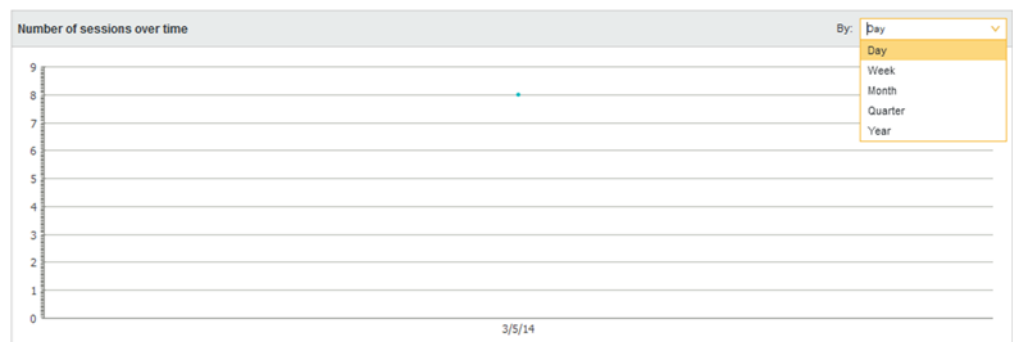
To use the **System Usage** page:

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

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

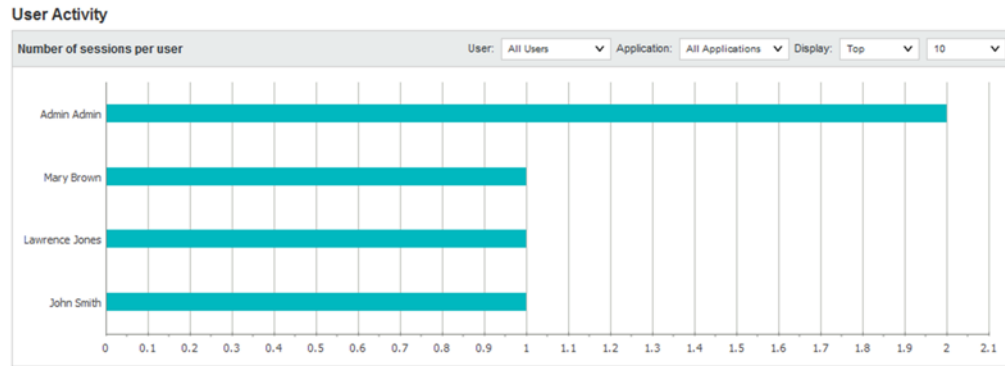
- a. To select a different time frame, from the list, select the time frame to use.
 - b. To select a specific range of dates, click the other radio button, then in the **From** and **To** date fields, provide the start and end dates.
 - c. After selecting a time frame or range of dates, to update the 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.

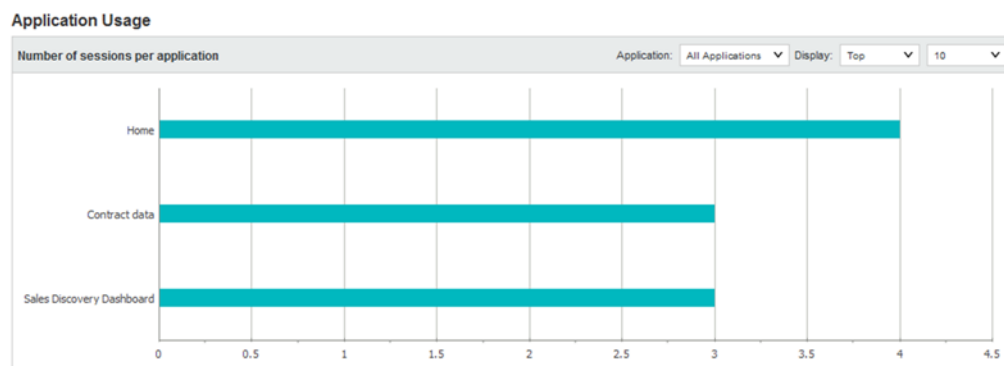
- b. To narrow the report to a specific project, from the **Project** list, select the project.

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.

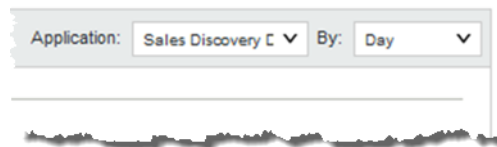


You can narrow the report to show values for a specific project, and change the number of values displayed.

- 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.

A screenshot of a web-based filter interface. It features a light gray header bar with two dropdown menus. The first dropdown is labeled 'Application:' and has 'Sales Discovery C' selected. The second dropdown is labeled 'By:' and has 'Day' selected. Below the header bar is a thin horizontal line, and underneath that is a dark, irregular, torn-paper-like border.

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.

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

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

How Studio determines the locale to use

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

Selecting the default locale

Studio is configured with a default locale that you can update from the **Control Panel**.

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.

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.

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

The locale is set in different locations.

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.

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 Use Browser Locale, 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>
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"> • If the user's locale is set to Use Browser Locale, then Big Data Discovery tries to use the locale from the cookie. <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"> • If the user account is configured with a locale value other than Use Browser Locale, then Big Data Discovery uses that locale, and also updates the cookie with that locale.
A non-logged-in user uses the user menu option to select a different locale	<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:

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

Display Settings

Locale

United States - English ▼

Time Zone

(UTC) Coordinated Universal Time ▼

3. Click **Save**.

Configuring a user's preferred locale

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

To configure the preferred locale for a user:

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

My Account

* Required

User Details

Screen Name:*

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 > Users**.
 - c. Locate the user and click **Actions > Edit**.

Add/Edit User

User Details

*Required

Screen Name:*	Email Address:*
<input type="text" value="rwiggum"/>	<input type="text" value="ralph.wiggum@ssotest.com"/>
First Name:*	Password:*
<input type="text" value="Ralph"/>	<input type="password"/>
Middle Name:	Retype Password:*
<input type="text"/>	<input type="password"/>
Last Name:*	Role:*
<input type="text" value="Wiggum"/>	<input type="text" value="Restricted User"/>
	Locale:*
	<input type="text" value="United States - English"/>

► INHERITED ROLES

► PROJECTS

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 > Display Settings**.

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

Display Settings

Locale

United States - English ▼

Time Zone

(UTC) Coordinated Universal Time ▼

3. 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

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

Assigning users and user groups to projects

You can manage access to projects from the **Project Settings > Sharing** page or from the project details panel in the Catalog. For details, see "Assigning project roles" in the *Studio User's Guide*.

Certifying a project

Big Data Discovery administrators can certify a project.

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**.

Deleting projects

From the **Control Panel**, Big Data Discovery administrators can delete 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">The project Creator and Studio Administrators are the only users with accessThe All Big Data Discovery users group is set to No Access <p>Projects are Private by default. Access must be granted by the Creator or by a Studio Administrator.</p>
Public	<ul style="list-style-type: none">The All Big Data Discovery users group is set to Project Restricted Users <p>Public projects grant view access to Studio users.</p>

Project Type	Description
Shared	<p>The project has been modified in any of the following ways:</p> <ul style="list-style-type: none">• Users other than the Creator are added to the project• User Groups other than All Big Data Discovery admins and All Big Data Discovery users are added to the project• The All Big Data Discovery users group is set to Project Authors <p>Projects are set to Shared to indicate changes from the default Public or Private permissions.</p>

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

To change the project type for a project:

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

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

5. Click **Save**.

Assigning users and user groups to projects

You can manage access to projects from the **Project Settings > Sharing** page or from the project details panel in the Catalog. For details, see "Assigning project roles" in the *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 > Projects**.
2. Click the **Actions** link for the project, then click **Edit**.
3. On the project configuration page, to certify the project, select the **Certified** check box.
4. Click **Save**.

Making a project active or inactive

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

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

To make a project active or inactive:

1. In the Studio header, click the **Configuration Options** icon and select **Control Panel**.
2. Select **User Settings > Projects**
3. Click the **Actions** link for the project, then click **Edit**.
4. To make the project inactive, deselect the **Active** check box. If the project is inactive, then to make the project active, check the **Active** check box.
5. Click **Save**.

Deleting projects

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

To delete a project:

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

Part II

Controlling User Access to Studio

Configuring User-Related Settings

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

Configuring authentication settings for users

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

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.

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.

Configuring authentication settings for users

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

To configure the authentication settings for users:

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

Credentials

The screenshot shows the 'Credentials' page with the 'Authentication' tab selected. It includes a dropdown menu for 'How do users authenticate?' set to 'By Email Address', checkboxes for 'Allow users to automatically login?' (checked) and 'Allow users to request forgotten passwords?' (unchecked), and a 'Configure Authentication' button.

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

Configuring the password policy


The password policy sets the requirements for creating and setting Studio passwords. These options do not apply to Studio passwords managed by an LDAP system.

To configure the password policy:

1. Select **Configuration Options > User Settings > Password Policies**.

The **Password Policies** page displays.



Password Policies

 You are using LDAP's password policy. Please change your LDAP password policy settings if you wish to use a local password policy.

Options Syntax Checking

☒ Syntax Checking Enabled
☒ Allow Dictionary Words
Minimum Length

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.

The screenshot shows the 'Security' settings panel. It includes the following fields:

- History Enabled**: A checked checkbox with a help icon.
- History Count**: A dropdown menu set to '6' with a help icon.
- Expiration Enabled**: A checked checkbox with a help icon.
- Maximum Age**: A dropdown menu set to '2 Weeks' with a help icon.
- Warning Time**: A dropdown menu set to '1 Day' with a help icon.
- Grace Limit**: A text input field set to '0' with a help icon.

- b. From the **History Count** list, select the number of previous passwords to save and prevent the user from using.

For example, if you select 6, then users cannot use their last 6 passwords.
5. To enable password expiration:
 - a. Select the **Expiration Enabled** check box.

You should not enable expiration if users cannot change their passwords in Big Data Discovery.
 - b. From the **Maximum Age** list, select the amount of time before a password expires.
 - c. From the **Warning Time** list, select the amount of time before the expiration to begin displaying warnings to the user.
 - d. In the **Grace Limit** field, type the number of times a user can log in using an expired password.
6. Click **Save**.

Restricting the use of specific screen names and email addresses

If needed, you can configure lists of screen names and email addresses that should not be used for Studio users.

To restrict the user of specific screen names and email addresses:

1. In the Studio header, click the **Configuration Options** icon and select **Control Panel**.
2. Select **Platform Settings > Credentials**.
3. On the **Reserved Credentials** tab, in the **Screen Names** text area, type the list of screen names that cannot be used.

Put each screen name on a separate line.
4. In the **Email Addresses** text area, type the list of email addresses that cannot be used.

Put each email address on a separate line.

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

Each Studio user is assigned a user role. The user role determines a user's access to features within Studio.

Creating a new Studio user

If you are not using LDAP, you may want to create Studio users manually.

Editing a Studio user

The **Users** page also allows you to edit a user's account.

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.

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
Administrator	Administrators have full access to all features in Studio. Administrators can: <ul style="list-style-type: none">• Access the Control Panel• Create and delete data sets and projects• Transform data within a project• View, configure, and manage all projects
Power User	Power users can: <ul style="list-style-type: none">• Create and delete data sets and projects• Transform data within a project• Export data to HDFS and create new data sets• View, configure, and manage projects for which they have a project role• Edit their account information Power users cannot: <ul style="list-style-type: none">• Access the Control Panel
User	Users can: <ul style="list-style-type: none">• Create and delete data sets and projects• Transform data within a project• View, configure, and manage projects for which they have a project role• Edit their account information Users cannot: <ul style="list-style-type: none">• Access the Control Panel• Export data to HDFS
Restricted User	This is the default user role for new users. It has the most restricted privileges and is essentially a read-only role. This is the default user role for new users. Restricted users can: <ul style="list-style-type: none">• Create new projects• View data sets in the Catalog• View, configure, and manage projects for which they have a project role Restricted users cannot: <ul style="list-style-type: none">• Edit their account information• Access the Control Panel• Create new data sets• Transform data within a project• Export data to HDFS

Note: Power Users, Users, and Restricted Users have no project roles by default, but they can access any projects that grant roles to the **All Big Data Discovery users** group. They can also access projects for which they have a project role, outlined below.

Project Roles

Project roles grant access privileges to project content and configuration. You can assign project roles to individual users or to user groups, and they define access to a given project regardless of a user's user role in Big Data Discovery Studio. The roles are:

Role	Description
Project Author	<p>Project authors can:</p> <ul style="list-style-type: none"> • Configure and manage a project • Add or remove users and user groups • Assign user and user group roles • Transform project data • Export project data <p>Project authors cannot:</p> <ul style="list-style-type: none"> • Create new data sets • Access the Big Data Discovery Control Panel
Project Restricted User	<p>Project Restricted Users can:</p> <ul style="list-style-type: none"> • View a project and navigate through the configured pages • Add and configure project pages and components <p>Project restricted users cannot:</p> <ul style="list-style-type: none"> • Access Project Settings • Create new data sets • Transform data • Export project data

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
No Access (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.
Default Access (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.

Access Level	Description
Read-only	Users with Read access to a data set can <ul style="list-style-type: none">• See the data set in search results or by browsing the Catalog• Explore the data set• Add the data set to a project and modify it within the project
Read/Write	In addition to Read permissions, users with Write access to a data set can <ul style="list-style-type: none">• Modify data set metadata such as description, searchable tags, and global attribute metadata• Manage access to the data set

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 > 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](#).

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

▼ PROJECTS

Available Projects ▼

	Projects	Description	Project Role
<input type="checkbox"/>	asimoRegressionTests-11h4...		Project Restricted User ▼
<input type="checkbox"/>	asimoSmokeTests-ie-14h14...		Project Restricted User ▼
<input type="checkbox"/>	asimoRegressionTests-11h4...		Project Restricted User ▼
<input type="checkbox"/>	asimoSmokeTests-ie-14h14...		Project Restricted User ▼
<input type="checkbox"/>	alanTest		Project Restricted User ▼

- a. Select the check box next to each project you want the new user to be a member of.
 - b. For each project, from the **Role** list, select the project role to assign to the user.
11. Click **Save**.

The user is added to the list of users.

Editing a Studio user

The **Users** page also allows you to edit a user's account.

From the **Users** page, to edit a user:

1. In the Studio header, click the **Configuration Options** icon and select **Control Panel**.
2. Select **User Settings > Users**
3. Click the **Actions** button next to the user.
4. Click **Edit**.
5. To change the user's password:
 - a. In the **Password** field, type the new password.
 - b. In the **Retype Password** field, re-type the new password.
6. To change the user role, from the **Role** list, select the new role.
7. Under **Projects**, to add a user as an project member:
 - a. Make sure the list is set to **Available Projects**. These are projects the user is not yet a member of.
 - b. Select the check box next to each project you want to add the user to.
 - c. For each project, from the **Role** list, select the project role to assign to the user.
8. Under **Projects**, to change the project role for or remove the user from a project:
 - a. From the list, select **Assigned Projects**.

The list shows the projects the user is currently a member of.
 - b. To change the user's project role, from the **Role** drop-down list, select the new project role.
 - c. To remove the user from a project, deselect the check box.
9. Click **Save**.

Deactivating, reactivating, and deleting Studio users

From the **Users** page of the **Control Panel**, you can make an active user inactive. You can also reactivate or delete inactive users.

Note that you cannot make your own user account inactive, and you cannot delete an active user.

From the **Users** page, to change the status of a user account:

1. To make an existing user inactive:
 - a. In the users list, select the check box for the user you want to deactivate.
 - b. Click **Deactivate**.

Big Data Discovery prompts you to confirm that you want to deactivate the user.

The user is then removed from the list of active users.

Note that inactive users are not removed from Big Data Discovery.

2. To reactivate or delete an inactive user:

- a. Click the **Advanced** link below the user search field.

Big Data Discovery displays additional user search fields.

- b. From the **Active** list, select **No**.

Note that if you change the **Match type** to **Any**, you must also provide search criteria in at least one of the other fields.

- c. Click **Search**.

The users list displays only the inactive users.

- d. Select the check box for the user you want to reactivate or delete.

- e. To reactivate the user, click **Restore**.

- f. To delete the user, click **Delete**.

Part III

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](#)

This topic provides a list of all the logs generated by a BDD deployment.

[Troubleshooting errors in Big Data Discovery Cloud Service](#)

While most logs are available to the service's users, in most cases it is faster to troubleshoot issues by contacting your Oracle Cloud Support team.

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 Dgraph Gateway logs .	\$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 Dgraph Gateway logs .	\$BDD_DOMAIN/servers/AdminServer/logs/AdminServer.log
WebLogic Managed Server server log	Contains messages from the WebLogic Managed Server subsystems and applications.	\$BDD_DOMAIN/servers/<serverName>/logs/<serverName>.log
Dgraph Gateway application log	WebLogic log for the Dgraph Gateway application. See Dgraph Gateway log entry format	\$BDD_DOMAIN/servers/<serverName>/logs/<serverName>-diagnostic.log
Dgraph stdout/stderr log	Contains Dgraph operational messages, including startup messages. See Dgraph out log .	\$BDD_HOME/logs/dgraph.out

Log	Purpose	Default Location
Dgraph request log	Contains entries for Dgraph requests. See Dgraph request log .	\$BDD_HOME/ dgraph/bin/ dgraph.reqlog
Dgraph tracing ebb logs	Dgraph Tracing Utility files, which are especially useful for Dgraph crashes.	\$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
FUSE stdout/stderr log	Contains FUSE operational messages. See FUSE out log .	\$BDD_HOME/logs/ hdfs_fuse_client.out
Studio application log in Log4j format	Studio application log (in Log4j format). For both Studio application logs, see About the main Studio log file .	\$BDD_DOMAIN/servers/ <serverName>/logs/ bdd-studio.log
Studio application log in ODL format	Studio application log (in ODL format).	\$BDD_DOMAIN/servers/ <serverName>/logs/ bdd-studio-odl.log
Studio metrics log in Log4j format	Studio metrics log (in Log4j format). For both Studio metrics logs, see About the metrics log file .	\$BDD_DOMAIN/servers/ <serverName>/logs/ bdd-studio- metrics.log
Studio metrics log in ODL format	Studio metrics log (in ODL format).	\$BDD_DOMAIN/servers/ <serverName>/logs/ bdd-studio-metrics- odl.log
Studio client log in Log4j format	Studio client log (in Log4j format). For both Studio client logs, see About the Studio client log file .	\$BDD_DOMAIN/servers/ <serverName>/logs/ bdd-studio- client.log
Studio client log in ODL format	Studio client log (in ODL format).	\$BDD_DOMAIN/servers/ <serverName>/logs/ bdd-studio-client- odl.log
Data Processing logs	Contains messages resulting from Data Processing workflows. See the <i>Data Processing Guide</i> .	\$BDD_HOME/logs/edp/ edp_*.log
Transform Service logs	Contains messages from transformation operations. See the <i>Data Processing Guide</i> .	\$BDD_HOME/logs/ transformservice/ <data>.stderrout.log

Log	Purpose	Default Location
CDH logs (YARN, Spark worker, and ZooKeeper logs)	YARN logs from CDH processes that ran Data Processing workflows, as listed in the <i>Data Processing Guide</i> . See the Cloudera and Hortonworks documentation for information on the ZooKeeper logs.	Available from the Cloudera Manager Web UI for the component.

Where to find logging information for each component

This table lists how to find detailed logging information for each Big Data Discovery component:

Big Data Discovery Component name	Where to find logging information?
Studio	See Studio Logging .
Data Processing	Data Processing is a component of BDD that runs on CDH nodes in the BDD deployment. For Data Processing logs, see the <i>Data Processing Guide</i> .
Dgraph Gateway (and WebLogic Server logs)	See Dgraph Gateway Logging .
Dgraph	See Dgraph Logging .
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> .

Troubleshooting errors in Big Data Discovery Cloud Service

While most logs are available to the service's users, in most cases it is faster to troubleshoot issues by contacting your Oracle Cloud Support team.

The Big Data Discovery Cloud Service issues these types of messages:

- **General information messages** indicate changes in the system status or other changes that you may need to be aware of. They are cleared once you refresh the page, or perform any actions in the console.
- **General error messages** appear for intermittent errors. They are cleared when you refresh the page, or go to another page in the console. You can also close the general error messages.
- **Operational error messages** indicate that an operation in the service console did not succeed. Such error messages persist at the top of the page until you start a new operation, such as start, stop, or back up, or until the new operation succeeds.

All errors in BDDCS that appear in the service console include a Reference ID in the format:

Reference Id: XX-YY-ZZ

If you encounter an error with Big Data Discovery Cloud Service, copy the Reference ID and provide it to the Oracle Cloud Support team to assist them in locating the related log files so that they can quickly fix the issue.

Studio Logging

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

[About logging in Studio](#)

Studio uses the Apache Log4j logging utility.

[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`.

[About the main Studio log file](#)

For Studio, the main log file (`bdd-studio.log`) contains all of the log messages.

[About the metrics log file](#)

Studio captures metrics logging, including all log entries from the `com.endeca.portal.instrumentation` classes.

[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.

[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.

[Adjusting Studio logging levels](#)

For debugging purposes in a development environment, you can dynamically adjust logging levels for any class hierarchy.

[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.

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"> • INCIDENT_ERROR • ERROR • WARNING • NOTIFICATION • TRACE • UNKNOWN 	[NOTIFICATION]
Message ID	The message ID that uniquely identifies the message within the component. The ID may be null.	[]
Module ID	The Java class that prints the message entry.	[com.endeca.portal.session.UserSession]
Host name	The name of the host where the message originated.	[host: web12.example.com]
Host address	The network address of the host where the message originated	[nwaddr: 10.152.105.219]

ODL log entry field	Description	Example
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: 0000Kvsw8S17ADkpSw4EyclLjsrN 0000^6,0]
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"> FATAL ERROR WARN INFO DEBUG 	[INFO]
Module ID	The Java class that prints the message entry.	[UserSession]
Message Text	The text of the log message.	UserSession created

About the Log4j configuration XML files

The primary log configuration is managed in `portal-log4j.xml`, which is packed inside the portal application file `WEB-INF/lib/portal-impl.jar`.

The file is in the standard Log4j XML configuration format, and allows you to:

- Create and modify appenders
- Bind appenders to loggers
- Adjust the log verbosity of different classes/packages

By default, `portal-log4j.xml` specifies a log verbosity of INFO for the following packages:

- `com.endeca`
- `com.endeca.portal.metadata`
- `com.endeca.portal.instrumentation`

It does not override any of the default log verbosity settings for other components.

Note: If you adjust the logging verbosity, it is updated for both Log4j and the Java Utility Logging Implementation (JULI). Code using either of these loggers should respect this configuration.

About the main Studio log file

For Studio, the main log file (`bdd-studio.log`) contains all of the log messages.

By default the `bdd-studio.log` is stored in the WebLogic domain in the `$BDD_DOMAIN/<serverName>/logs` directory (where `serverName` is the name of the Managed Server in which Studio is installed).

The main root logger prints all messages to:

- The console, which typically is redirected to the application server's output log.
- `bdd-studio.log`, the log file in log4j format.
- `bdd-studio-odl.log`, the log file in ODL format. Also stored in `$BDD_DOMAIN/logs`

The main logger does not print messages from the `com.endeca.portal.instrumentation` classes. Those messages are printed to the metrics log file.

About the metrics log file

Studio captures metrics logging, including all log entries from the `com.endeca.portal.instrumentation` classes.

The metrics log files are:

- `bdd-studio-metrics.log`, which is in Log4j format.
- `bdd-studio-metrics-odl.log`, which is in ODL format.

Both metrics log files are created in the same directory as `bdd-studio.log`.

The metrics log file contains the following columns:

Column Name	Description
Total duration (msec)	The total time for this entry (End time minus Start time).
Start time (msec since epoch)	The time when this entry started. For Dgraph Gateway queries and server executions, uses the server's clock. For client executions, uses the client's clock.
End time (msec since epoch)	The time when this entry was finished. For Dgraph Gateway queries and server executions, uses the server's clock. For client executions, uses the client's clock.
Session ID	The session ID for the client.
Page ID	If client instrumentation is enabled, the number of full page refreshes or actions the user has performed. Used to help determine how long it takes to load a complete page. Some actions that do not affect the overall state of a page, such as displaying attributes on the Available Refinements panel, do not increment this counter.
Gesture ID	The full count of requests to the server.
Portlet ID	This is the ID associated with an individual instance of a component. It generally includes: <ul style="list-style-type: none"> • The type of component • A unique identifier For example, if a page includes two Chart components, the ID can be used to differentiate them.
Entry Type	The type of entry. For example: <ul style="list-style-type: none"> • PORTLET_RENDER - Server execution in response to a full refresh of a component • DISCOVERY_SERVICE_QUERY - Dgraph Gateway query • CONFIG_SERVICE_QUERY - Configuration service query • SCONFIG_SERVICE_QUERY - Semantic configuration service query • LQL_PARSER_SERVICE_QUERY - EQL parser service query • CLIENT - Client side JavaScript execution • PORTLET_RESOURCE - Server side request for resources • PORTLET_ACTION - Server side request for an action
Miscellaneous	A URL encoded JSON object containing miscellaneous information about the entry.

Configuring the amount of metrics data to record

To configure the metrics you want to include, you use a setting in `portal-ext.properties`. This setting applies to both the metrics log file and the **Performance Metrics** page.

The metrics logging can include:

- Queries by Dgraph nodes.
- Portlet server executions by component. The server side code is written in Java.

It handles configuration updates, configuration persistence, and Dgraph queries. The server-side code generates results to send back to the client-side code.

Server executions include component render, resource, and action requests.

- Component client executions for each component. The client-side code is hosted in the browser and is written in JavaScript. It issues requests to the server code, then renders the results as HTML. The client code also handles any dynamic events within the browser.

By default, only the Dgraph queries and component server executions are included.

You use the `df.performanceLogging` setting in `portal-ext.properties` to configure the metrics to include. The setting is:

```
df.performanceLogging=<metrics to include>
```

Where *<metrics to include>* is a comma-separated list of the metrics to include. The available values to include in the list are:

Value	Description
QUERY	If this value is included, then the page includes information for Dgraph queries.
PORTLET	If this value is included, then the page includes information on component server executions.
CLIENT	If this value is included, then the page includes information on component client executions.

In the default configuration, where only the Dgraph queries and component server executions are included, the value is:

```
df.performanceLogging=QUERY,PORTLET
```

To include all of the available metrics, you would add the `CLIENT` option:

```
df.performanceLogging=QUERY,PORTLET,CLIENT
```

Note that for performance reasons, this configuration is not recommended.

If you make the value empty, then the metrics log file and **Performance Metrics** page also are empty.

```
df.performanceLogging=
```

About the Studio client log file

The Studio client log file collects client-side logging information. In particular, Studio logs JavaScript errors in this file.

The client log files are:

- `bdd-studio-client.log`, which is in Log4j format.
- `bdd-studio-client-odl.log`, which is in ODL format.

Both client log files are created in the same directory as `bdd-studio.log`.

The client logs are intended primarily for Studio developers to troubleshoot JavaScript errors in the Studio Web application. These files are therefore intended for use by Oracle Support only.

Adjusting Studio logging levels

For debugging purposes in a development environment, you can dynamically adjust logging levels for any class hierarchy.

Note: When you adjust the logging verbosity, it is updated for both Log4j and the Java Utility Logging Implementation (JULI). Code using either of these loggers should respect this configuration.

Adjusting Studio logging levels:

1. In the Big Data Discovery header, click the **Configuration Options** icon and select **Control Panel**.
2. Choose **Server > Server Administration**.
3. Click the **Log Levels** tab.
4. On the **Update Categories** tab, locate the class hierarchy you want to modify.
5. From the logging level list, select the logging level.

Note: When you modify a class hierarchy, all classes that fall under that class hierarchy also are changed.

6. Click **Save**.

Using the Performance Metrics page to monitor query performance

The **Performance Metrics** page on the **Control Panel** displays information about component and Dgraph Gateway query performance.

It uses the same logging data that is recorded in the metrics log file.

However, unlike the metrics log file, the **Performance Metrics** page uses data stored in memory. Restarting Big Data Discovery clears the **Performance Metrics** data.

For each type of included metric, the table at the top of the page contains a collapsible section.

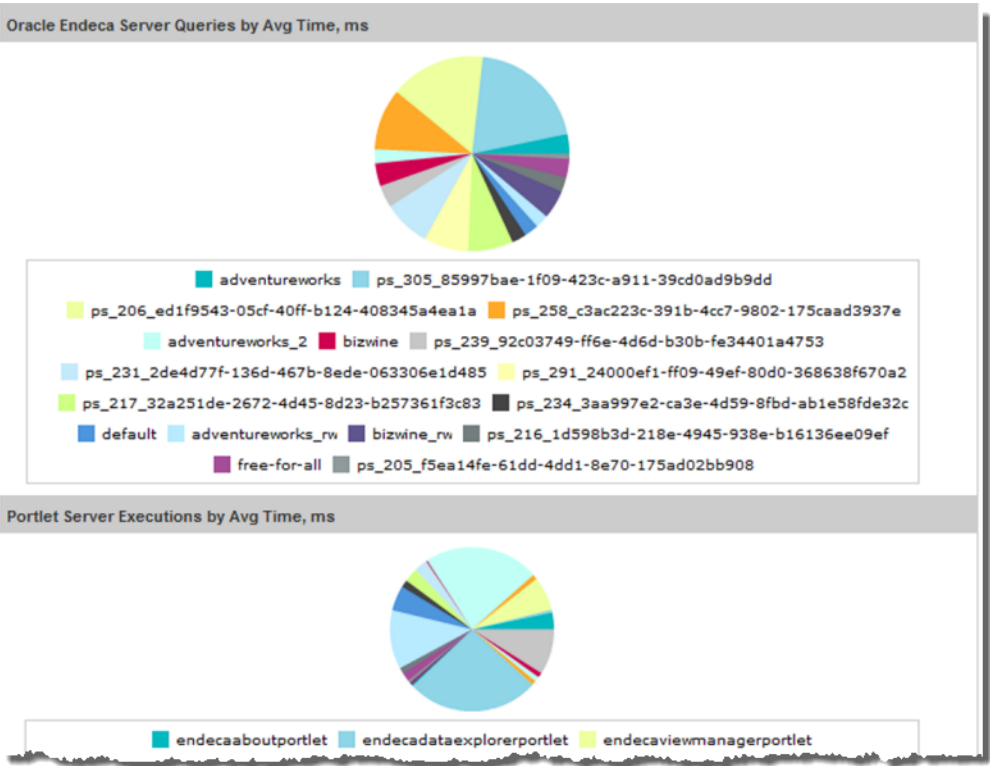
Performance Metrics

Performance Metrics					
Name ▲	Count	Total Time, ms	Avg Time, ms	Max Time, ms	
▼ Oracle Endeca Server Queries					
adventureworks	28	6980	249	2603	
adventureworks_2	40	7131	178	543	
adventureworks_rw	928	159285	171	4840	
bizwine	457	132479	289	2928	
bizwine_rw	531	195181	367	4281	
default	4111	734544	178	3245	
free-for-all	268	63290	236	2184	
ps_205_f5ea14fe-61d...	57	3814	66	649	
ps_206_ed1f9543-05...	83	100603	1212	9567	
ps_216_1d598b3d-21...	92	16810	182	3343	
ps_217_32a251de-26...	1	574	574	574	
ps_231_2de4d77f-13...	1	598	598	598	
ps_234_3aa997e2-ca...	10	1860	186	1052	
ps_239_92c03749-ff...	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.

Dgraph Logging

This section describes the Dgraph logs.

Dgraph request log

The Dgraph request log (also called the query log) contains one entry for each request processed.

Dgraph out log

The Dgraph out log is where the Dgraph's stdout/stderr output is remapped.

FUSE out log

The FUSE out log is where the FUSE client's stdout/stderr output is remapped.

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:  
endecareultslistportlet_WAR_endecareultslistportlet_INSTANCE_5RKp_LAYOUT_11601
```

As an example, if you set:

```
PORTLET=endecareultslistportlet_WAR_endecareultslistportlet_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	The local date and time when the message was generated, using use the following ISO 8601 extended format: <code>YYYY-MM-DDTHH:mm:ss.sss(+ -)hh:mm</code> The hours range is 0 to 23 and milliseconds and offset timezones are mandatory.	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

Log entry field	Description	Example
Message Type	The type of message (log level): <ul style="list-style-type: none"> • INCIDENT_ERROR • ERROR • WARNING • NOTIFICATION • TRACE • UNKNOWN 	WARNING
Log Subsystem	The log subsystem that generated the message.	{dgraph}
Job ID	The ID of the job being executed.	[0]
Message Text	The text of the log message.	Starting HTTP server on port: 7010

Dgraph log subsystems

The log subsystems that can generate log entries in the Dgraph out log are the following:

- `background_merging` — messages about Dgraph 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 (Endeca Query Language) engine.
- `eql_feature` — messages providing usage information for certain EQL features.
- `eve` — messages generated from the EVE (Endeca Virtual Engine) query evaluator.
- `http` — messages about Dgraph HTTP communication operations.
- `lexer` — messages from the OLT (Oracle Language Technology) subsystem.
- `splitting` — messages resulting from EVE (Endeca Virtual Engine) splitting tasks.
- `ssl` — messages generated by the SSL subsystem.
- `task_scheduler` — messages related to the Dgraph task scheduler.
- `text_search_rel_rank` — messages related to Relevance Ranking operations during text searches.
- `text_search_spelling` — messages related to spelling correction operations during text searches.

- `update` — messages related to updates.
- `workload_manager` — messages from the Dgraph Workload Manager.
- `ws_request` — messages related to request exchanges between Web services.
- `xq_web_service` — messages generated from the XQuery-based Web services.

All of these subsystems have a default log level of `NOTIFICATION`.

Dgraph 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 Tue Jun 21
13:21:53

                                2016.  Shutdown will complete when all
outstanding
                                jobs are complete.
DGRAPH    NOTIFICATION {database} [0] Finished unmounting everything.
DGRAPH    WARNING      {cluster}  [0] Lost connection to ZooKeeper: ZooKeeper
connection lost
                                (zk error -4)
DGRAPH    NOTIFICATION {cluster} [0] Finished closing zk connection
DGRAPH    NOTIFICATION {dgraph}  [0] All dgraph transactions completed at Tue Jun
21
                                13:21:54 2016, exiting normally (pid=3605)
DGRAPH    NOTIFICATION {dgraph}  [0] Overall shutdown took 324 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_f475de43
DGRAPH    NOTIFICATION {database}    [0] Mounting database edp_f475de43
DGRAPH    NOTIFICATION {dgraph}      [0] Initial DL version: 2
DGRAPH    NOTIFICATION {bulk_ingest} [0] MessageParser constructor, parserCounter
incremented,
                                is now 1
DGRAPH    NOTIFICATION {bulk_ingest} [0] Start ingest for collection: edp_f475de43
for
                                database edp_f475de43
DGRAPH    NOTIFICATION {bulk_ingest} [0] Starting a bulk ingest operation for
database edp_f475de43
DGRAPH    NOTIFICATION {bulk_ingest} [0] batch 0 finish BatchUpdating status
Success for
                                database edp_f475de43
DGRAPH    NOTIFICATION {bulk_ingest} [0] Ending bulk ingest at client's request
for
```

```

changes
DGRAPH NOTIFICATION {bulk_ingest} [0] Bulk ingest completed: Added 9983 records
and
rejected 0 records, for database
edp_f475de43
DGRAPH NOTIFICATION {bulk_ingest} [0] Ingest end - 9.411MB in 13.022sec =
0.723MB/sec for database edp_f475de43

```

The `bulk_ingest` entries show the ingest of a data set with 9983 records.

[Dgraph log levels](#)

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

FUSE out log

The FUSE out log is where the FUSE client's stdout/stderr output is remapped.

When configured to run, the FUSE client redirects its stdout/stderr output to the following log file:

```
$BDD_HOME/logs/hdfs_fuse_client.out
```

Note that you cannot change the log level configuration for this log.

FUSE log entry format

The format of the FUSE log fields are:

- Timestamp
- 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> <p>YYYY-MM-DDTHH:mm:ss.sss(+ -)hh:mm</p> <p>The hours range is 0 to 23 and milliseconds and offset timezones are mandatory.</p>	2016-03-23T07:11:39.173-04:00
Message Type	<p>The type of message (log level):</p> <ul style="list-style-type: none"> • INCIDENT_ERROR • ERROR • WARNING • NOTIFICATION • TRACE • UNKNOWN 	WARNING

Log entry field	Description	Example
Log Subsystem	The log subsystem that generated the message. "hdfs_fuse" is hard-coded for the FUSE client.	{hdfs_fuse}
Job ID	The ID of the job being executed.	[0]
Message Text	The text of the log message.	FileNotFoundException: Path /bdd_dgraph_indexv43/ Claim_indexes/committed/ Endeca.422.703 does not exist.

Dgraph Gateway Logging

This section describes the Dgraph Gateway logs.

[Dgraph Gateway logs](#)

Dgraph Gateway uses the Apache Log4j logging utility for logging and its messages are written to WebLogic Server logs.

[Dgraph Gateway log entry format](#)

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

[Log entry information](#)

This topic describes some of the information that is found in log entries.

[Logging properties file](#)

Dgraph Gateway has a default Log4j configuration file that sets its logging properties.

Dgraph Gateway logs

Dgraph Gateway uses the Apache Log4j logging utility for logging and its messages are written to WebLogic Server logs.

The BDD installation creates a WebLogic domain, whose name is set by the `WEBLOGIC_DOMAIN_NAME` parameter of the `bdd.conf` file. The WebLogic domain has both an Admin Server and a Managed Server. The Admin Server is named **AdminServer** while the Managed Server has the same name as the host machine. Both the Dgraph Gateway and Studio are deployed into the Managed Server.

There are two sets of logs for the two different servers:

- The Admin Server logs are in the `$BDD_DOMAIN/servers/AdminServer/logs` directory.
- The Managed Server logs are in the `$BDD_DOMAIN/servers/<serverName>/logs` directory.

There are three types of logs:

- WebLogic Domain Log
- WebLogic Server Log
- Application logs

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

By default, these log files are located in the `$DOMAIN_HOME/servers/AdminServer/logs` directory (for the Admin Server) or one of the `$DOMAIN_HOME/servers/<serverName>/logs` directories (for a Managed Server).

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

WebLogic Domain Log

The WebLogic domain log is generated only for the Admin Server. This domain log is intended to provide a central location from which to view the overall status of the domain.

The name of the domain log is:

```
$BDD_DOMAIN/servers/AdminServer/logs/<bdd_domain>.log
```

The domain log is located in the `$DOMAIN_HOME/servers/AdminServer/logs` directory.

For more information on the WebLogic domain and server logs, see the "Server Log Files and Domain Log Files" topic in this page: http://docs.oracle.com/cd/E24329_01/web.1211/e24428/logging_services.htm#WLLOG124

WebLogic Server Log

A WebLogic server log is generated for the Admin Server and for each Managed Server instance.

The default path of the Admin Server server log is:

```
$BDD_DOMAIN/servers/AdminServer/logs/AdminServer.log
```

The default path of the server log for a Managed Server is:

```
$BDD_DOMAIN/servers/<serverName>/logs/<serverName>.log
```

For example, if "web001.us.example.com" is the name of the Managed Server, then its server log is:

```
$BDD_DOMAIN/servers/web001.us.example.com/logs/web001.us.example.com.log
```

Application logs

Application logs are generated by the deployed applications. In this case, Dgraph Gateway and Studio are the applications.

For Dgraph Gateway, its application log is at:

```
$BDD_DOMAIN/servers/<serverName>/logs/<serverName>-diagnostic.log
```

For example, if "web001.us.example.com" is the name of the Managed Server, then the Dgraph Gateway application log is:

```
$BDD_DOMAIN/servers/web001.us.example.com/logs/web001.us.example.com-diagnostic.log
```

For Studio, its application log is at:

```
$BDD_DOMAIN/servers/<serverName>/logs/bdd-studio.log
```

For example, if "web001.us.example.com" is the name of the Managed Server, then its application log is:

```
$BDD_DOMAIN/servers/web001.us.example.com/logs/bdd-studio.log
```

The directory also stores other Studio metric log files, which are described in [About the metrics log file](#).

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"> INCIDENT_ERROR ERROR WARNING NOTIFICATION TRACE 	[ERROR]

Log entry field	Description	Example
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]
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^Aye1JxPgK 000000,0]
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.

ODL Log Level	Meaning
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.
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

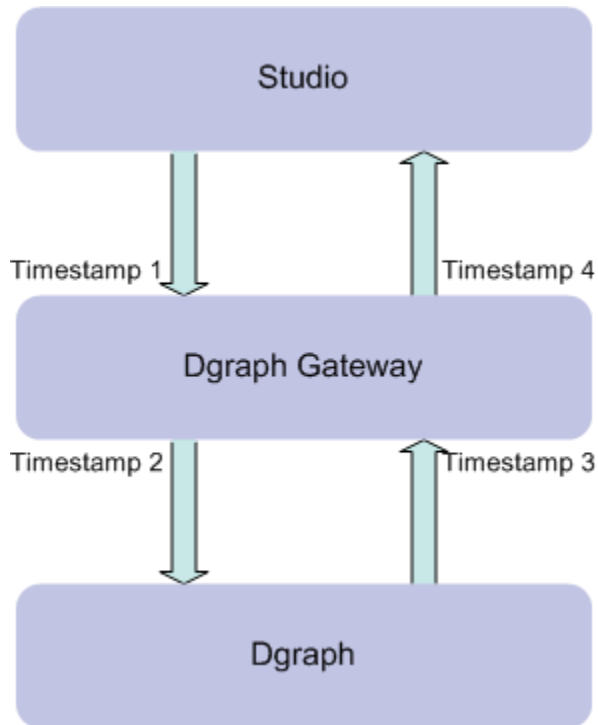
For ingest operations, a start and end timestamp is logged. At the end of the operation, the ingest results are also logged (number of added records, number of deleted records, number of updated records, number of replaced records, number of added or updated records).

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
<code>log4j.rootLogger=WARN, stdout, ODL</code>	The level of the root logger is defined as WARN and attaches the Console Appender (stdout) and ODL-format Log Appender (ODL) to it.
<code>log4j.appender.stdout=org.apache.log4j.ConsoleAppender</code>	Defines stdout as a Log4j ConsoleAppender
<code>org.apache.log4j.PatternLayout</code>	Sets the PatternLayout class for the stdout layout.

Logging property	Description
<code>log4j.appender.stdout.layout.ConversionPattern</code>	<p>Defines the log entry conversion pattern as:</p> <ul style="list-style-type: none">• %d is the date of the logging event.• %p outputs the priority of the logging event.• %c outputs the category of the logging event.• %L outputs the line number from where the logging request was issued.• %m outputs the application-supplied message associated with the logging event while %n is the platform-dependent line separator character. <p>For other conversion characters, see: https://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/PatternLayout.html</p>
<code>log4j.appender.ODL=com.endeca.util.ODLAppender</code>	<p>Defines ODL as an ODL Appender. ODL (Oracle Diagnostics Logging) is the logging format for Oracle applications.</p>
<code>log4j.appender.ODL.MaxSize</code>	<p>Sets the maximum amount of disk space to be used by the <code><ServerName>-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.</p>
<code>log4j.appender.ODL.MaxSegmentSize</code>	<p>Sets the maximum size (in bytes) of the log file. When the <code><ServerName>-diagnostic.log</code> file reaches this size, a rollover file is created. The default is 104857600 (about 10 MB).</p>
<code>log4j.appender.ODL.encoding</code>	<p>Sets character encoding the log file. The default UTF-8 value prints out UTF-8 characters in the file.</p>
<code>log4j.appender.ODL.MaxDaysToRetain</code>	<p>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.</p>

Logging property	Description
<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

Index

D

data connections

- about, [1-1](#)
- creating, [1-2](#)
- deleting, [1-2](#)
- editing, [1-2](#)

Data Source Library

- data connections, creating, [1-2](#)
- data connections, deleting, [1-2](#)
- data connections, editing, [1-2](#)
- data sources, creating, [1-2](#)
- data sources, deleting, [1-3](#)
- data sources, editing, [1-3](#)

data sources

- about, [1-1](#)
- creating, [1-2](#)
- deleting, [1-3](#)
- details, displaying, [1-3](#)
- editing, [1-3](#)

Dgraph

- log levels, [12-6](#)
- out log, [12-3](#)
- request log, [12-1](#)

Dgraph Gateway

- logging configuration, [13-6](#)
- logs, [13-1](#)

F

framework settings

- that you can edit, [2-1](#)

FUSE out log, [12-7](#)

H

Hadoop settings

- configuring, [3-3](#)
- list of, [3-1](#)

L

locales

locales (*continued*)

- configuring the default, [6-3](#)
- configuring user preferred, [6-4](#)
- effect of selection, [6-1](#)
- list of supported, [6-1](#)
- locations where set, [6-2](#)
- scenarios for determining, [6-2](#)

logging

- about, [10-3](#)
- list of available logs, [10-1](#)
- Log4j configuration files, about, [11-3](#)
- main Studio log file, [11-4](#)
- metrics data, configuring, [11-5](#)
- metrics log file, about, [11-4](#)
- Performance Metrics page, [11-7](#)
- Studio client log, [11-6](#)
- verbosity, adjusting from the Control Panel, [11-7](#)

logs

- Dgraph Gateway, [13-1](#)
- Dgraph out, [12-3](#)
- Dgraph request, [12-1](#)
- FUSE out, [12-7](#)

P

passwords

- existing user, changing for, [9-6](#)
- new user, setting for, [9-5](#)
- password policy, configuring, [8-2](#)

Performance Metrics page, [11-7](#)

project roles

- about, [9-3](#)
- types of, [9-3](#)

projects

- certifying, [7-2](#)
- deleting, [7-3](#)
- existing user, changing membership, [9-6](#)
- making active or inactive, [7-3](#)
- new user, assigning membership to, [9-5](#)
- project type, configuring, [7-1](#)
- roles, [9-3](#)

R

reporting errors in BDDCS, [10-3](#)
roles

- existing user, changing, [9-6](#)
- new user, assigning, [9-5](#)
- project roles, [9-3](#)
- user roles, editing, [9-1](#)
- user roles, list of, [9-1](#)

S

Studio

- creating users, [9-4](#)
- Data Processing settings, [3-1](#)
- database password, [2-3](#)
- editable framework settings, [2-1](#)
- health check, [4-1](#)
- locales, [6-1](#)
- logging, [11-1](#)
- setting time zone, [6-6](#)

Studio settings

- configuring, [2-2](#)

Studio, timeout, [2-2](#)

System Usage

- sections, about, [5-2](#)
- usage logs, adding entries, [5-1](#)
- using, [5-3](#)

System Usage (*continued*)

T

time zone, Studio, [6-6](#)

U

users

- authentication settings, configuring, [8-1](#)
- creating, [9-4](#)
- deactivating, [9-6](#)
- deleting, [9-6](#)
- editing, [9-5](#)
- email addresses, listing restricted, [8-3](#)
- reactivating, [9-6](#)
- screen names, listing restricted, [8-3](#)

W

WebLogic logs

- AdminServer, [13-1](#)

Z

ZooKeeper

- client log level, [13-9](#)