

Cloudera Data Management



Important Notice

(c) 2010-2015 Cloudera, Inc. All rights reserved.

Cloudera, the Cloudera logo, Cloudera Impala, and any other product or service names or slogans contained in this document are trademarks of Cloudera and its suppliers or licensors, and may not be copied, imitated or used, in whole or in part, without the prior written permission of Cloudera or the applicable trademark holder.

Hadoop and the Hadoop elephant logo are trademarks of the Apache Software Foundation. All other trademarks, registered trademarks, product names and company names or logos mentioned in this document are the property of their respective owners. Reference to any products, services, processes or other information, by trade name, trademark, manufacturer, supplier or otherwise does not constitute or imply endorsement, sponsorship or recommendation thereof by us.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Cloudera.

Cloudera may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Cloudera, the furnishing of this document does not give you any license to these patents, trademarks copyrights, or other intellectual property. For information about patents covering Cloudera products, see <http://tiny.cloudera.com/patents>.

The information in this document is subject to change without notice. Cloudera shall not be liable for any damages resulting from technical errors or omissions which may be present in this document, or from use of this document.

Cloudera, Inc.
1001 Page Mill Road Bldg 2
Palo Alto, CA 94304
info@cloudera.com
US: 1-888-789-1488
Intl: 1-650-362-0488
www.cloudera.com

Release Information

Version: Navigator 2.3.x
Date: July 14, 2015

Table of Contents

About Cloudera Data Management.....	5
Auditing.....	7
Audit Log Properties.....	7
Service Auditing Properties.....	9
<i>Auditing Impala Operations.....</i>	<i>13</i>
Audit Events and Audit Reports.....	14
<i>Viewing Audit Events.....</i>	<i>15</i>
<i>Filtering Audit Events.....</i>	<i>15</i>
<i>Creating Audit Event Reports.....</i>	<i>16</i>
<i>Editing Audit Event Reports.....</i>	<i>16</i>
<i>Downloading Audit Event Reports.....</i>	<i>16</i>
<i>Audit Event Fields.....</i>	<i>17</i>
Downloading HDFS Directory Access Permission Reports.....	20
Metadata.....	21
Metadata Search.....	23
<i>Search Syntax.....</i>	<i>23</i>
<i>Search Properties.....</i>	<i>23</i>
Accessing Metadata.....	26
<i>Navigator Metadata UI.....</i>	<i>26</i>
<i>Navigator API.....</i>	<i>29</i>
Modifying Custom Metadata.....	29
Policies.....	35
Policy Expressions.....	36
Lineage Diagrams.....	44
Displaying a Template Lineage Diagram.....	46
Displaying an Instance Lineage Diagram.....	48
Displaying the Template Lineage Diagram for an Instance Lineage Diagram.....	49
Downloading a Lineage File.....	49
Impala Lineage Properties.....	61
Schema.....	62

<i>Displaying Hive, Sqoop, and Impala Table Schema.....</i>	<i>62</i>
<i>Displaying Pig Table Schema.....</i>	<i>62</i>
<i>Displaying HDFS Dataset Schema.....</i>	<i>63</i>

About Cloudera Data Management

This guide describes how to perform data management using Cloudera Navigator. Data management activities include auditing access to data residing in HDFS and Hive metastores, reviewing and updating metadata, and discovering the lineage of data objects.

- **Important:** This feature is available only with a Cloudera Enterprise license; it is not available in Cloudera Express. For information on Cloudera Enterprise licenses, see [Managing Licenses](#).

Cloudera Navigator is a fully integrated data management and security tool for the Hadoop platform. Data management and security capabilities are critical for enterprise customers that are in highly regulated industries and have stringent compliance requirements.

Cloudera Navigator provides three categories of functionality:

- **Auditing data access and verifying access privileges** - The goal of auditing is to capture a complete and immutable record of all activity within a system. While Hadoop has historically lacked centralized cross-component audit capabilities, products such as Cloudera Navigator add secured, real-time audit components to key data and access frameworks. Cloudera Navigator allows administrators to configure, collect, and view audit events, to understand who accessed what data and how. Cloudera Navigator also allows administrators to generate reports that list the HDFS access permissions granted to groups.

Cloudera Navigator tracks access permissions and actual accesses to all entities in HDFS, Hive, HBase, Impala, Sentry, and Solr, and the Cloudera Navigator Metadata Server itself to help answer questions such as - who has access to which entities, which entities were accessed by a user, when was an entity accessed and by whom, what entities were accessed using a service, which device was used to access, and so on. Cloudera Navigator auditing supports tracking access to:

- HDFS entities accessed by HDFS, Hive, HBase, Impala, and Solr services
- HBase and Impala
- Hive metadata
- Sentry
- Solr
- Cloudera Navigator Metadata Server
- **Searching metadata and visualizing lineage** - Cloudera Navigator metadata management features allow DBAs, data modelers, business analysts, and data scientists to search for, amend the properties of, and tag data entities.

In addition, to satisfy risk and compliance audits and data retention policies, it supports the ability to answer questions such as: where did the data come from, where is it used, and what are the consequences of purging or modifying a set of data entities. Cloudera Navigator supports tracking the lineage of HDFS files, datasets, and directories, Hive tables and columns, MapReduce and YARN jobs, Hive queries, Impala queries, Pig scripts, Oozie workflows, Spark jobs, and Sqoop jobs.

- **Securing data and simplifying storage and management of encryption keys** - Data encryption and key management provide a critical layer of protection against potential threats by malicious actors on the network or in the data center. It is also a requirement for meeting key compliance initiatives and ensuring the integrity of your enterprise data.

The following Cloudera Navigator components enable compliance initiatives that require at-rest data encryption and key management:

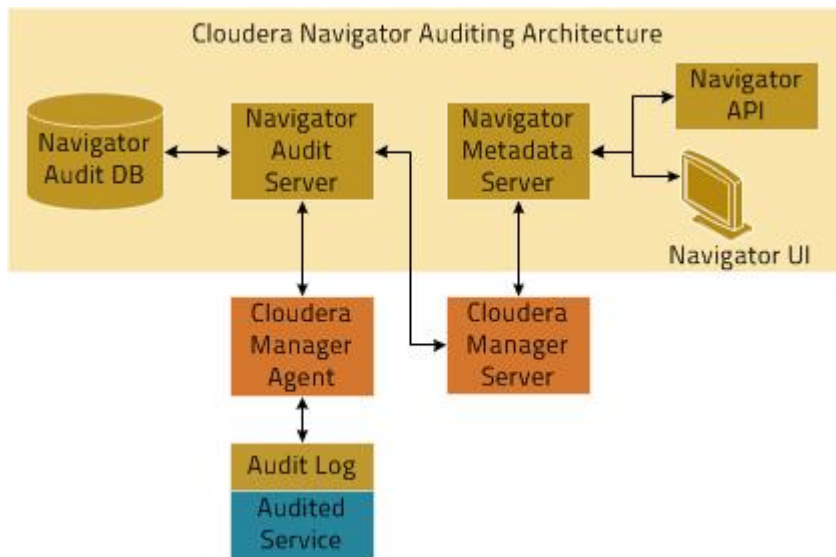
- [Cloudera Navigator Key Trustee Server](#) is an enterprise-grade virtual safe-deposit box that stores and manages cryptographic keys and other security artifacts.
- [Cloudera Navigator Key HSM](#) allows Cloudera Navigator Key Trustee Server to seamlessly integrate with a hardware security module (HSM).

About Cloudera Data Management

- [Cloudera Navigator Encrypt](#) transparently encrypts and secures data at rest without requiring changes to your applications and ensures there is minimal performance lag in the encryption or decryption process.

Auditing

Cloudera Navigator auditing provides data auditing and access features. The Cloudera Navigator auditing architecture is illustrated below.



When Cloudera Navigator auditing is configured, plug-ins that enable collection and filtering of audit events are added to the HDFS, HBase, and Hive (that is, the HiveServer2 and Beeswax servers) services. The plug-ins write the audit events to an audit log on the local filesystem. Cloudera Impala and Sentry collection and filter audit events and write them directly in an audit log file.

The Cloudera Manager Agent monitors the audit log files and sends these events to the Navigator Audit Server. The Cloudera Manager Agent retries any event that it fails to transmit. As there is no in-memory transient buffer involved, once the audit events are written to the audit log file, they are guaranteed to be delivered (as long as filesystem is available). The Cloudera Manager Agent keeps track of current audit event offset in the audit log that it has successfully transmitted, so on any crash/restart it picks up the audit event from the last successfully sent position and resumes. Audit logs are rotated and the Cloudera Manager Agent follows the rotation of the log. The Agent also takes care of purging old audit logs once they have been successfully transmitted to the Navigator Audit Server. If a plug-in fails to write audit event to audit log file, it can either drop the event or shut down the process in which they are running (depending on the configured queue policy).

The Navigator Audit Server performs the following functions:

- Tracking and coalescing events
- Storing events to the audit database

Audit Log Properties

A service **Enable Audit Collection** property controls whether the Cloudera Manager Agent tracks a service's audit log file. A validation check is performed for all lifecycle actions (stop/start/restart). If the Enable Collection flag is selected and the Audit Log Directory property *is not set*, the validator displays a message that says that the Audit Log Directory property must be set to enable auditing.

The following properties apply to a service audit log file:

- **Audit Log Directory** - The directory in which audit log files are written. By default, this property is not set if Cloudera Navigator is not installed.

▪ **Note:** If the value of this property is changed, and service is restarted, then the Cloudera Manager Agent will start monitoring the new log directory for audit events. In this case it is possible that not all events are published from the old audit log directory. To avoid loss of audit events, when this property is changed, perform the following steps:

1. Stop the service.
2. Copy audit log files and (for Impala only) the `impalad_audit_wal` file from the old audit log directory to the new audit log directory. This needs to be done on all the hosts where Impala Daemons are running.
3. Start the service.

- **Maximum Audit Log File Size** - The maximum size of the audit log file before a new file is created. The unit of the file size is service dependent:
 - **HDFS, HBase, Hive, Navigator Metadata Server, Sentry, Solr** - MiB
 - **Impala** - lines (queries)
- **Number of Audit Logs to Retain** - Maximum number of rolled over audit logs to retain. The logs will not be deleted if they contain audit events that have not yet been propagated to the Audit Server.

Enabling Audit Collection

1. Do one of the following:
 - Click a supported service.
 - Do one of the following:
 - Select **Clusters > Cloudera Management Service > Cloudera Management Service**.
 - On the Status tab of the Home page, in **Cloudera Management Service** table, click the **Cloudera Management Service** link.
2. Click the **Configuration** tab.
3. Select **Scope > ServiceName (Service-Wide)**.
4. Select **Category > Cloudera Navigator**.
5. Select the **Enable Audit Collection** checkbox.
6. Click **Save Changes** to commit the changes.
7. Restart the service.

Configuring Audit Logs

1. Do one of the following:
 - Service - Click a supported service.
 - Navigator Metadata Server
 - Do one of the following:
 - Select **Clusters > Cloudera Management Service > Cloudera Management Service**.
 - On the Status tab of the Home page, in **Cloudera Management Service** table, click the **Cloudera Management Service** link.
2. Click the **Configuration** tab.
3. Select the scope according to the service:
 - All services except Impala - Select **Scope > ServiceName (Service-Wide)**.

- Impala - Select **Scope** > **Impala Daemon**.
 - Navigator Metadata Server - Select **Scope** > **Navigator Metadata Server**.
4. Select **Category** > **Logs**.
 5. Configure the log properties. For Impala, preface each log property with **Impala Daemon**.
 6. Click **Save Changes** to commit the changes.
 7. Restart the service.

Service Auditing Properties

Each service (with exceptions noted) that supports auditing has the following properties:

- **Enable Audit Collection** - See [Audit Log Properties](#) on page 7.
- **Audit Event Filter** - A set of rules that capture properties of auditable events and actions to be performed when an event matches those properties. The Cloudera Manager Agent uses this property to filter events out *before* they are sent to Cloudera Navigator. This property is not supported for Sentry, Solr, or the Cloudera Navigator Metadata Server. The default filter settings discard the following events:
 - **HDFS** - generated by the internal Cloudera and Hadoop users (`cloudera-scm`, `hdfs`, `hbase`, `hive`, `impala`, `mapred`, `solr`, `spark`, and `dr.who`), events generated by the `hdfs` user running the `listStatus`, `listCachePools`, `listCacheDirectives`, and `getFileinfo` operations, and that affect files in the `/tmp` directory.
 - **HBase** - that affect the `-ROOT-`, `.META.`, and `acl` tables
 - **Hive** - generated by Hive MapReduce jobs in the `/tmp` directory
 - **Impala** - no default filter.
- **Audit Event Tracker** - A set of rules for tracking and coalescing events. This feature is used to define equivalency between different audit events. Tracking works by keeping a reference to events when they first appear, and comparing other incoming events against the tracked events according to the rules defined. When events match, according to a set of configurable parameters, only one entry in the audit list is generated for all the matching events. This property is not supported for the Cloudera Navigator Metadata Server.
- **Audit Queue Policy** - The action to take when the audit event queue is full. The options are Drop or Shutdown. When a queue is full and the queue policy of the service is Shutdown, before shutting down the service, *N* audits will be discarded, where *N* is the size of the Cloudera Navigator Audit Server queue.

▪ **Note:** If the queue policy is Shutdown, the Impala service is shut down only if Impala is unable to write to the audit log file. It is possible that an event may not appear in the audit event log due to an error in transfer to the Cloudera Manager Agent or database. In such cases Impala will not shut down and will keep writing to the log file. When the transfer problem is fixed the events will be transferred to the database.

This property is not supported for the Cloudera Navigator Metadata Server.

The Audit Event Filter and Audit Event Tracker rules for filtering and coalescing events are expressed as JSON objects.

You can edit these rules using a rule editor:

HDFS-1 (Service-Wide)
View as JSON

➤ Action: discard Fields: <code>username: (?:(cloudera-scm hbase mapred hive dr.who solr impala spark)(?:/.+)?</code>	- +
➤ Action: discard Fields: <code>username: (?:(hdfs)(?:/.+)?, operation: (?:(listStatus listCachePools listCacheDirectives getFileinfo)</code>	- +
➤ Action: discard Fields: <code>src: /tmp(?:/.*)?</code>	- +

Default action Accept

or in a JSON text field:


HDFS-1 (Service-Wide)
View Editor

```

{
  "rules": [
    {
      "action": "discard",
      "fields": [
        {
          "name": "username",
          "match": "(?:cloudera-scm|hbase|mapred|hive|dr.who|solr|impala|spark)(?:/.+)?"
        }
      ]
    },
    {
      "action": "discard",
      "fields": [
        {
          "name": "username",
          "match": "(?:hdfs)(?:/.+)?"
        },
        {
          "name": "operation",
          "match": "(?:listStatus|listCachePools|listCacheDirectives|getFileinfo)"
        }
      ]
    },
    {
      "action": "discard",
      "fields": [
        {
          "name": "src",
          "match": "/tmp(?:/.+)?"
        }
      ]
    }
  ],
  "defaultAction": "accept",
  "comment": [
    "Default filter for HDFS services.",
    "Discards events generated by the internal Cloudera and/or HDFS users",
    "(cloudera-scm, hbase, mapred, hive, dr.who, solr, impala, and spark),",
    "'ls' actions performed by the hdfs user,",
    "and events that affect files in the /tmp directory."
  ]
}

```

For information on the structure of the objects, and the properties for which you can set filters, display the description on the configuration page as follows:

1. In the Cloudera Manager Admin Console, go to a service that supports auditing.
2. Click the **Configuration** tab.
3. Select **Scope > Service (Service-Wide)**.
4. Select **Category > Cloudera Navigator** category.
5. In **Audit Event Tracker** row, click . For example, the Hive properties are:
 - userName: the user performing the action.
 - ipAddress: the IP from where the request originated.
 - operation: the Hive operation being performed.
 - databaseName: the databaseName for the operation.
 - tableName: the tableName for the operation.

Configuring Service Auditing Properties

Required Role: **Navigator Administrator** **Full Administrator**

Follow this procedure for all cluster services that support auditing. In addition, for Impala and Solr auditing, perform the steps in [Configuring Impala Daemon Logging](#) on page 11, [Enabling Solr Auditing](#) on page 11.

1. Go to a service that supports auditing.
2. Click the **Configuration** tab.
3. Select **Scope** > **Service (Service-Wide)**.
4. Select **Category** > **Cloudera Navigator** category.
5. Edit the properties.
6. Click **Save Changes** to commit the changes.
7. Restart the service.

Configuring Impala Daemon Logging

Required Role: **Configurator** **Cluster Administrator** **Full Administrator**

To control whether the Impala Daemon role logs to the audit log:

1. Click the Impala service.
2. Click the **Configuration** tab.
3. Select **Scope** > **Impala Daemon**.
4. Select **Category** > **Logs**.
5. Edit the **Enable Impala Audit Event Generation**.
6. Click **Save Changes** to commit the changes.
7. Restart the service.

To set the log file size:

1. Click the Impala service.
2. Select **Scope** > **Impala Daemon**.
3. Select **Category** > **Logs**.
4. Set the **Impala Daemon Maximum Audit Log File Size** property.
5. Click **Save Changes** to commit the changes.
6. Restart the service.

Enabling Solr Auditing

Required Role: **Configurator** **Cluster Administrator** **Full Administrator**

Solr auditing is disabled by default. To enable auditing:

1. Enable Sentry authorization for Solr following the procedure in [Enabling Sentry Authorization for Solr](#).
2. Go to the Solr service.
3. Click the **Configuration** tab.
4. Select **Scope** > **Solr Service (Service-Wide)**.
5. Select **Category** > **Policy File Based Sentry** category.
6. Select or deselect the **Enable Sentry Authorization** checkbox.
7. Select **Category** > **Cloudera Navigator** category.
8. Select or deselect the **Enable Audit Collection** checkbox. See [Audit Log Properties](#) on page 7.
9. Click **Save Changes** to commit the changes.
10. Restart the service.

Enabling and Disabling Navigator Metadata Server Auditing

Required Role: **Navigator Administrator** **Full Administrator**

Navigator Metadata Server auditing is enabled by default. To enable or disable auditing:

1. Do one of the following:
 - Select **Clusters > Cloudera Management Service > Cloudera Management Service**.
 - On the Status tab of the Home page, in **Cloudera Management Service** table, click the **Cloudera Management Service** link.
2. Click the **Configuration** tab.
3. Select **Scope > Navigator Metadata Server**
4. Select **Category > Cloudera Navigator** category.
5. Select or deselect the **Enable Audit Collection** checkbox.
6. Click **Save Changes** to commit the changes.
7. Restart the service.

Audit Logging to Syslog

Required Role: Navigator Administrator Full Administrator

The Audit Server logs all audit records into a [Log4j](#) logger called `auditStream`. The log messages are logged at the TRACE level, with the attributes of the audit records. By default, the `auditStream` logger is inactive because the logger level is set to FATAL. It is also connected to a [NullAppender](#), and does not forward to other appenders (additivity set to false).

To record the audit stream, configure the `auditStream` logger with the desired appender. For example, the standard [SyslogAppender](#) allows you to send the audit records to a remote syslog.

The Log4j `SyslogAppender` supports only UDP. An example syslog configuration would be:

```
$ModLoad imudp
$UDPServerRun 514
# Accept everything (even DEBUG messages) local2.* /my/audit/trail.log
```

It is also possible to attach [other appenders](#) to the `auditStream` to provide other integration behaviors.

You can audit events to syslog in two formats: JSON and RSA EnVision. To configure audit logging to syslog, do the following:

1. Do one of the following:
 - Select **Clusters > Cloudera Management Service > Cloudera Management Service**.
 - On the Status tab of the Home page, in **Cloudera Management Service** table, click the **Cloudera Management Service** link.
2. Click the **Configuration** tab.
3. Locate the Navigator Audit Server Logging Advanced Configuration Snippet property by typing its name in the **Search** box.
4. Depending on the format type, enter:

```
log4j.logger.auditStream = TRACE,SYSLOG
log4j.appender.SYSLOG = org.apache.log4j.net.SyslogAppender
log4j.appender.SYSLOG.SyslogHost = hostname
log4j.appender.SYSLOG.Facility = Local2
log4j.appender.SYSLOG.FacilityPrinting = true
```

To configure the specific stream type, enter:

Format	Properties
JSON	<code>log4j.additivity.auditStream = false</code>
RSA EnVision	<code>log4j.additivity.auditStreamEnVision = false</code>

5. Click **Save Changes** to commit the changes.

Example Log Messages

Format	Log Message Example
JSON	Jul 23 11:05:15 hostname local2: { "type": "HDFS", "allowed": "true", "time": "1374602714758", "service": "HDFS-1", "user": "root", "ip": "10.20.93.93", "op": "mkdirs", "src": "/audit/root", "perms": "rwxr-xr-x" }
RSA EnVision	Cloudera Navigator 1 type="Hive",allowed="false",time="1382551146763", service="HIVE-1",user="systest",impersonator="",ip="/10.20.190.185",op="QUERY", opText="select count(*) from sample_07",db="default",table="sample_07",path="/user/hive/warehouse/sample_07",objType="TABLE"

If a particular field is not applicable for that audit event, it is omitted from the message.

Auditing Impala Operations

To monitor how Impala data is being used within your organization, ensure that your Impala authorization and authentication policies are effective, and detect attempts at intrusion or unauthorized access to Impala data, you can use the auditing feature in Impala 1.2.1 and higher:

- Enable auditing by including the option `-audit_event_log_dir=directory_path` in your `impalad` startup options for a cluster not managed by Cloudera Manager, or [configuring Impala Daemon logging in Cloudera Manager](#). The log directory must be a local directory on the server, not an HDFS directory.
- Decide how many queries will be represented in each log files. By default, Impala starts a new log file every 5000 queries. To specify a different number, [configure Impala Daemon logging in Cloudera Manager](#).
- Configure the Cloudera Navigator product to collect and consolidate the audit logs from all the hosts in the cluster.
- Use Cloudera Navigator or Cloudera Manager to filter, visualize, and produce reports based on the audit data. (The Impala auditing feature works with Cloudera Manager 4.7 to 5.1 and Cloudera Navigator 2.1 and higher.) Check the audit data to ensure that all activity is authorized and/or detect attempts at unauthorized access.

Durability and Performance Considerations for Impala Auditing

The auditing feature only imposes performance overhead while auditing is enabled.

Because any Impala host can process a query, enable auditing on all hosts where the Impala Daemon role runs. Each host stores its own log files, in a directory in the local filesystem. The log data is periodically flushed to disk (through an `fsync()` system call) to avoid loss of audit data in case of a crash.

The runtime overhead of auditing applies to whichever host serves as the coordinator for the query, that is, the host you connect to when you issue the query. This might be the same host for all queries, or different applications or users might connect to and issue queries through different hosts.

To avoid excessive I/O overhead on busy coordinator hosts, Impala syncs the audit log data (using the `fsync()` system call) periodically rather than after every query. Currently, the `fsync()` calls are issued at a fixed interval, every 5 seconds.

By default, Impala avoids losing any audit log data in the case of an error during a logging operation (such as a disk full error), by immediately shutting down the Impala Daemon role on the host where the auditing problem occurred.

Format of the Audit Log Files

The audit log files represent the query information in JSON format, one query per line. Typically, rather than looking at the log files themselves, you use the Cloudera Navigator product to consolidate the log data from all Impala hosts and filter and visualize the results in useful ways. (If you do examine the raw log data, you might run the files through a JSON pretty-printer first.)

All the information about schema objects accessed by the query is encoded in a single nested record on the same line. For example, the audit log for an `INSERT ... SELECT` statement records that a select operation

occurs on the source table and an insert operation occurs on the destination table. The audit log for a query against a view records the base table accessed by the view, or multiple base tables in the case of a view that includes a join query. Every Impala operation that corresponds to a SQL statement is recorded in the audit logs, whether the operation succeeds or fails. Impala records more information for a successful operation than for a failed one, because an unauthorized query is stopped immediately, before all the query planning is completed.

The information logged for each query includes:

- Client session state:
 - Session ID
 - User name
 - Network address of the client connection
- SQL statement details:
 - Query ID
 - Statement Type - DML, DDL, and so on
 - SQL statement text
 - Execution start time, in local time
 - Execution Status - Details on any errors that were encountered
 - Target Catalog Objects:
 - Object Type - Table, View, or Database
 - Fully qualified object name
 - Privilege - How the object is being used (`SELECT`, `INSERT`, `CREATE`, and so on)

Which Operations Are Audited

The kinds of SQL queries represented in the audit log are:

- Queries that are prevented due to lack of authorization.
- Queries that Impala can analyze and parse to determine that they are authorized. The audit data is recorded immediately after Impala finishes its analysis, before the query is actually executed.

The audit log does not contain entries for queries that could not be parsed and analyzed. For example, a query that fails due to a syntax error is not recorded in the audit log. The audit log also does not contain queries that fail due to a reference to a table that does not exist, if you would be authorized to access the table if it did exist.

Certain statements in the `impala-shell` interpreter, such as `CONNECT`, `SUMMARY`, `PROFILE`, `SET`, and `QUIT`, do not correspond to actual SQL queries, and these statements are not reflected in the audit log.

Reviewing the Audit Logs

You typically do not review the audit logs in raw form. The Cloudera Manager Agent periodically transfers the log information into a back-end database where it can be examined in consolidated form. See [Audit Events and Audit Reports](#) on page 14.

Audit Events and Audit Reports

Required Role: **Auditing Viewer** **Full Administrator**

An **audit event** is an event that describes an action of accessing a service. An **audit report** is a collection of audit events that satisfy a set of filters.



Audit events are recorded by the [Cloudera Navigator Audit Server](#). Audit report metadata is recorded by the [Cloudera Navigator Metadata Server](#).

Viewing Audit Events


1. [Start and log into the Cloudera Navigator data management component UI.](#)
2. Click the **Audits** tab. The Audit Events report displays all audit events that occurred during the last hour.

Filtering Audit Events

Specifying a Time Range

1. Click the date-time range at the top right of the audits page.
2. Do one of the following:
 - Click a **Last *n* hours** link.
 - Specify a custom range:
 1. Click **Custom range**.
 2. In the Selected Range endpoints, click each endpoint and specify a date and time in the date control fields.
 - **Date** - Click the down arrow ▼ to display a calendar and select a date, or click a field and click the spinner arrows  or up and down arrow keys.
 - **Time** - Click the hour, minute, and AM/PM fields and click the spinner arrows  or up and down arrow keys to specify the value.
 - Move between fields using the right and left arrow keys.
3. Click **Apply**.


Adding a Filter

1. Do one of the following:
 - Click the  icon that displays next to a field when you hover in one of the event entries.
 - Click the **Filters** link. The Filters pane displays.
 1. Click **Add New Filter** to add a filter.
 2. Choose a [field](#) in the drop-down list. You can search by fields such as username, service name, or operation. The fields vary depending on the service or role. The service name of the Navigator Metadata Server is Navigator.
 3. Choose an operator in the operator drop-down list.
 4. Type a field value in the value text field. To match a substring, use the `like` operator and specify `%` around the string. For example, to see all the audit events for files created in the folder `/user/joe/out` specify `Source like %/user/joe/out%`.

A filter control with field, operation, and value fields is added to the list of filters.

2. Click **Apply**. A field, operation, and value breadcrumb is added above the list of audit events and the list of events displays all events that match the filter criteria.

Removing a Filter

1. Do one of the following:
 - Click the **x** next to the filter above the list of events. The list of events displays all events that match the filter criteria.
 - Click the **Filters** link. The Filters pane displays.
 1. Click the  at the right of the filter.

2. Click **Apply**. The filter is removed from above the list of audit event and the list of events displays all events that match the filter criteria.

Creating Audit Event Reports

1. [Start and log into the Cloudera Navigator data management component UI](#).
2. Click the **Audits** tab. The Audit Events report displays all audit events that occurred during the last hour.
3. Do one of the following:
 - Save a filtered version of the Audit Events report:
 1. Optionally specify [filters](#).
 2. Click **Save As Report**.
 - Create a new report:
 1. Click **Create New Report**.
4. Enter a report name.
5. In the **Default time range** field, specify a relative time range. If you had specified a custom absolute time range before selecting **Save As Report**, the *custom absolute time range is discarded*.
6. Optionally add [filters](#).
7. Click **Save**.

Editing Audit Event Reports

1. [Start and log into the Cloudera Navigator data management component UI](#).
2. Click the **Audits** tab. The Audit Events report displays all audit events that occurred during the last hour.
3. In the left pane, click a report name.
4. Click **Edit Report**.
5. In the **Default time range** field, specify a relative time range. If you had specified a custom absolute time range before selecting **Save As Report**, the *custom absolute time range is discarded*.
6. Optionally add [filters](#).
7. Click **Save**.

Downloading Audit Event Reports

You can download audit event reports in the Audit UI or using the Audit API. An audit event contains the following fields: `timestamp`, `service`, `username`, `ipAddress`, `command`, `resource`, `allowed`, `[operationText]`, `serviceValues`. The contents of the `resource` and `serviceValues` fields depends on the type of the service. In addition, Hive, Hue, Impala, and Sentry events have the `operationText` field, which contains the operation string. See [Audit Event Fields](#) on page 17.

Downloading Audit Event Reports Using the Audit UI

1. [Start and log into the Cloudera Navigator data management component UI](#).
2. Click the **Audits** tab. The Audit Events report displays all audit events that occurred during the last hour.
3. Do one of the following:
 - Add [filters](#).
 - In the left pane, click a report name.
4. Select **Export** > *format*, where *format* is CSV or JSON.

Downloading Audit Events Using the Audit API

You can filter and download audit events using the [Cloudera Navigator Data Management Component API](#).

Hive Audit Events Using the Audit API

To download the audits events for a service named hive using the API, issue the request

```
curl
http://Navigator_Metadata_Server_host:port/api/v5/audits/?query=service%3Dhive&startTime=1431025200000&endTime=1431032400000\
&limit=5&offset=0&format=JSON&attachment=false -X GET -u username:password
```

startTime and endTime are required parameters and must be specified in [epoch time](#) in milliseconds.

The request could return the following JSON items:

```
[ {
  "timestamp" : "2015-05-07T20:34:39.923Z",
  "service" : "hive",
  "username" : "hdfs",
  "ipAddress" : "12.20.199.170",
  "command" : "QUERY",
  "resource" : "default:sample_08",
  "operationText" : "INSERT OVERWRITE \n  TABLE sample_09 \nSELECT \n
sample_07.code,sample_08.description \n  FROM sample_07 \n  JOIN sample_08 \n  WHERE
sample_08.code = sample_07.code",
  "allowed" : true,
  "serviceValues" : {
    "object_type" : "TABLE",
    "database_name" : "default",
    "operation_text" : "INSERT OVERWRITE \n  TABLE sample_09 \nSELECT \n
sample_07.code,sample_08.description \n  FROM sample_07 \n  JOIN sample_08 \n  WHERE
sample_08.code = sample_07.code",
    "resource_path" : "/user/hive/warehouse/sample_08",
    "table_name" : "sample_08"
  }
}, {
  "timestamp" : "2015-05-07T20:33:50.287Z",
  "service" : "hive",
  "username" : "hdfs",
  "ipAddress" : "12.20.199.170",
  "command" : "SWITCHDATABASE",
  "resource" : "default:",
  "operationText" : "USE default",
  "allowed" : true,
  "serviceValues" : {
    "object_type" : "DATABASE",
    "database_name" : "default",
    "operation_text" : "USE default",
    "resource_path" : "/user/hive/warehouse",
    "table_name" : ""
  }
}, {
  "timestamp" : "2015-05-07T20:33:23.792Z",
  "service" : "hive",
  "username" : "hdfs",
  "ipAddress" : "12.20.199.170",
  "command" : "CREATETABLE",
  "resource" : "default:",
  "operationText" : "CREATE TABLE sample_09 (code string,description string) ROW FORMAT
DELIMITED FIELDS TERMINATED BY '\\t' STORED AS TextFile",
  "allowed" : true,
  "serviceValues" : {
    "object_type" : "DATABASE",
    "database_name" : "default",
    "operation_text" : "CREATE TABLE sample_09 (code string,description string) ROW
FORMAT DELIMITED FIELDS TERMINATED BY '\\t' STORED AS TextFile",
    "resource_path" : "/user/hive/warehouse",
    "table_name" : ""
  }
} ]
```

Audit Event Fields

The following fields can appear in an audit event:

Display Name	Field	Description
Additional Info	additional_info	JSON text that contains more details about operation performed on entities in Navigator Metadata Server.
Allowed	allowed	Indicates whether the request to perform an operation failed or succeeded. A failure occurs if the user is not authorized to perform the action.
Collection Name	collection_name	The name of affected Solr collection.
Database Name	database_name	For Sentry, Hive, and Impala, the name of the database on which the operation was performed.
Delegation Token ID	delegation_token_id	Delegation token identifier generated by HDFS NameNode that is then used by clients when submitting a job to JobTracker.
Destination	dest	Path of the final location of an HDFS file in a rename or move operation.
Entity ID	entity_id	Identifier representing a Navigator Metadata Server entity. The identity of an entity can be retrieved using the Navigator Metadata Server API.
Event Time	timestamp	Date and time the action was performed. The server stores the timestamp in the timezone of the Navigator Audit Server and the Navigator UI displays the timestamp converted to the local timezone.
Family	family	HBase column family.
Impersonator	impersonator	<p>If an action was requested by another service, the name of the user that invoked the action on behalf of the user.</p> <ul style="list-style-type: none"> When Sentry is enabled, the Impersonator field displays for services other than Hive. When Sentry is not enabled, the Impersonator field always displays.
IP Address	ipAddress	The IP address of the host where the action occurred.
Object Type	object_type	For Sentry, Hive, and Impala, the type of the object (TABLE, VIEW, DATABASE) on which operation was performed.
Operation	command	<p>The action performed.</p> <ul style="list-style-type: none"> HBase - createTable, deleteTable, modifyTable, addColumn, modifyColumn, deleteColumn, enableTable, disableTable, move, assign, unassign, balance, balanceSwitch, shutdown, stopMaster, flush, split, compact, compactSelection, getClosestRowBefore, get, exists, put, delete, checkAndPut, checkAndDelete, incrementColumnValue, append, increment, scannerOpen, grant, revoke HDFS - setPermission, setOwner, open, concat, setTimes, createSymlink, setReplication, create, append, rename, delete, getFileinfo, mkdirs, listStatus, fsck, listSnapshottableDirectory, setPermission, setReplication Hive - EXPLAIN, LOAD, EXPORT, IMPORT, CREATEDATABASE, DROPDATABASE, SWITCHDATABASE, DROPTABLE, DESC TABLE, DESC FUNCTION, MSCK, ALTERNATE_ADDCOLS, ALTERNATE_REPLACE_COLS, ALTERNATE_RENAME_COL, ALTERNATE_RENAME_PART, ALTERNATE_RENAME,

Display Name	Field	Description
		<p> ALTABLE_DROPPARTS, ALTABLE_ADDPARTS, ALTABLE_TOUCH, ALTABLE_ARCHIVE, ALTABLE_UNARCHIVE, ALTABLE_PROPERTIES, ALTABLE_SERIALIZER, ALTERPARTITION_SERIALIZER, ALTABLE_SERDEPROPERTIES, ALTERPARTITION_SERDEPROPERTIES, ALTABLE_CLUSTER_SORT, SHOWDATABASES, SHOWTABLES, SHOW_TABLESTATUS, SHOW_TBLPROPERTIES, SHOWFUNCTIONS, SHOWINDEXES, SHOWPARTITIONS, SHOWLOCKS, CREATEFUNCTION, DROPFUNCTION, CREATEVIEW, DROPVIEW, CREATEINDEX, DROPINDEX, ALTERINDEX_REBUILD, ALTERVIEW_PROPERTIES, LOCKTABLE, UNLOCKTABLE, ALTABLE_PROTECTMODE, ALTERPARTITION_PROTECTMODE, ALTABLE_FILEFORMAT, ALTERPARTITION_FILEFORMAT, ALTABLE_LOCATION, ALTERPARTITION_LOCATION, CREATETABLE, CREATETABLE_AS_SELECT, QUERY, ALTERINDEX_PROPS, ALTERDATABASE, DESCDATABASE, ALTER_TABLE_MERGE, ALTER_PARTITION_MERGE, GRANT_PRIVILEGE, REVOKE_PRIVILEGE, SHOW_GRANT, GRANT_ROLE, REVOKE_ROLE, SHOW_ROLE_GRANT, CREATEROLE, DROPROLE </p> <ul style="list-style-type: none"> ▪ Impala - Query, Insert, Update, Delete, GRANT_PRIVILEGE, REVOKE_PRIVILEGE, SHOW_GRANT, GRANT_ROLE, REVOKE_ROLE, SHOW_ROLE_GRANT, CREATEROLE, DROPROLE ▪ Navigator Metadata Server - auditReport, authorization, metadata, policy, search, savedSearch. For the operation subtypes, see Sub Operation. ▪ Sentry - GRANT_PRIVILEGE, REVOKE_PRIVILEGE, ADD_ROLE_TO_GROUP, DELETE_ROLE_FROM_GROUP, CREATE_ROLE, DROP_ROLE ▪ Solr - add, commit, deleteById, deleteByQuery, finish, query, rollback, CREATE, CREATEALIAS, CREATESHARD, DELETE, DELETEALIAS, DELETESHARD, LIST, LOAD, LOAD_ON_STARTUP, MERGEINDEXES, PERSIST, PREPRECOVERY, RELOAD, RENAME, REQUESTAPPLYUPDATES, REQUESTRECOVERY, REQUESTSYNCSHARD, SPLIT, SPLITSHARD, STATUS, SWAP, SYNCSHARD, TRANSIENT, UNLOAD
Operation Params	operation_params	Solr query or update parameters used when performing the action.
Operation Text	operation_text	For Sentry, Hive, and Impala, the SQL query that was executed by user.
Permissions	permissions	HDFS permission of the file or directory on which the HDFS operation was performed.
Privilege	privilege	Privilege needed to perform an Impala operation.
Qualifier	qualifier	HBase column qualifier.
Query ID	query_id	The query ID for an Impala operation.
Resource	resource	A service-dependent combination of multiple fields generated during fetch. This field is not supported for filtering as it is not persisted.
Resource Path	resource_path	HDFS URL of Hive objects (TABLE, VIEW, DATABASE, and so on)

Display Name	Field	Description
Service Name	service	The name of the service that performed the action.
Session ID	session_id	Impala session ID.
Solr Version	solr_version	Solr version number.
Source	src	Path of the HDFS file or directory present in an HDFS operation.
Status	status	Status of an Impala operation providing more information on success or failure.
Stored Object Name	stored_object_name	Name of a policy, saved search, or audit report in Navigator Metadata Server.
Sub Operation	sub_operation	Subtype of operation performed in Navigator Metadata Server. Valid values are: <ul style="list-style-type: none"> auditReport - fetchAllReports, fetchAuditReport, createAuditReport, deleteAuditReport, updateAuditReport authorization - searchGroup, deleteGroup, fetchGroup, fetchRoles, updateRoles metadata - updateMetadata, fetchMetadata, fetchAllMetadata policy - fetchAllPolicies, createPolicy, deletePolicy, updatePolicy, fetchPolicySchedule, updatePolicySchedule, deletePolicySchedule savedSearch - fetchAllSavedSearches, fetchSavedSearch, createSavedSearch, deleteSavedSearch, updateSavedSearch
Table Name	table_name	For Sentry, HBase, Hive and Impala, the name of the table on which action was performed.
Username	username	The name of the user that performed the action.

Downloading HDFS Directory Access Permission Reports

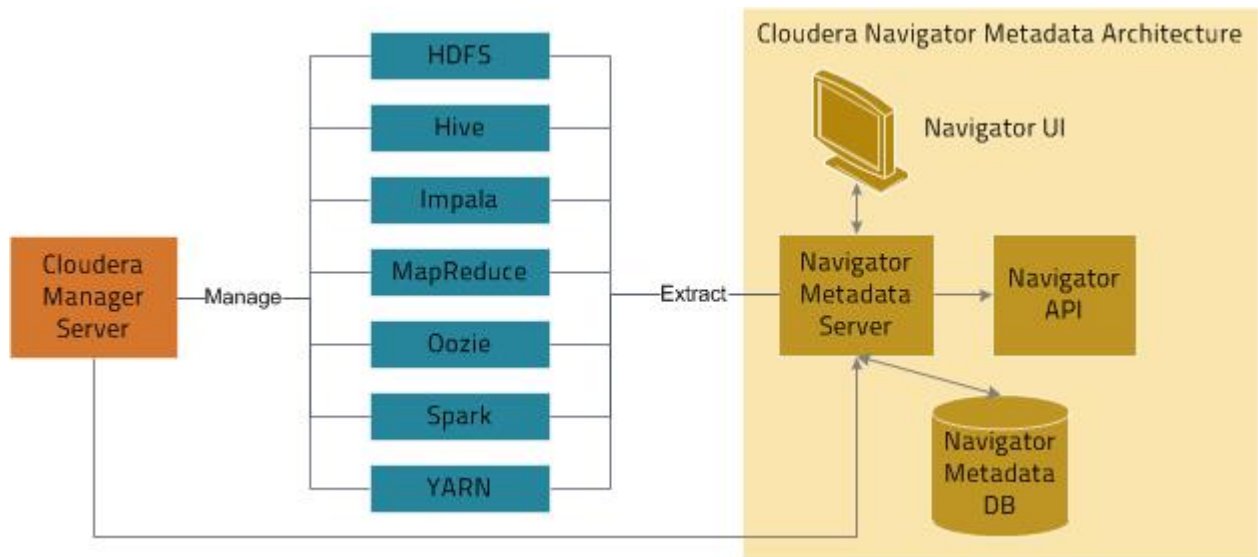
Required Role: **Cluster Administrator** **Full Administrator**

For each HDFS service you can download a report that details the HDFS directories a group has permission to access.

1. In the Cloudera Manager Admin Console, click **Clusters** > **ClusterName** > **General** > **Reports**.
2. In the Directory Access by Group row, click **CSV** or **XLS**. The Download User Access Report pop-up displays.
 - a. In the pop-up, type a group and directory.
 - b. Click **Download**. A report of the selected type will be generated containing the following information – path, owner, permissions, and size – for each directory contained in the specified directory that the specified group has access to.

Metadata

Cloudera Navigator metadata features provides data discovery and data lineage functions. The Cloudera Navigator metadata architecture is illustrated below.



The Navigator Metadata Server performs the following functions:

- Obtains connection information about the services whose data it manages from the Cloudera Manager Server
- Extracts entity metadata from the services at periodic intervals
- Manages and applies metadata extraction policies
- Indexes and stores entity metadata
- Manages user authorization data
- Manages audit report metadata
- Implements the Navigator UI and API

The Navigator Metadata database stores entity metadata, policies, and user authorization and audit report metadata.

The Cloudera Navigator Metadata Server manages metadata about the entities in a CDH cluster and relations between the entities. The metadata schema defines the types of metadata that are available for each entity type it supports. The types of metadata defined by the Navigator Metadata component include: the name of an entity, the service that manages or uses the entity, type, path to the entity, date and time of creation, access, and modification, size, owner, purpose, and relations—parent-child, data flow, and instance of—between entities. For example, the following shows the property sheet of a file entity:

sample_07.csv

[Edit](#)

Tags:

Source Type: HDFS

Type: File

Path: /user/hive/warehouse/sample_07/sample_07.csv

Owner: sample

Group: supergroup

Permissions: rwxrwxrwt

Size: 44.98KiB

Last Accessed: Apr 8 2015 8:25 AM

Last Modified: Apr 8 2015 8:25 AM

Created: Apr 8 2015 8:25 AM

Source: HDFS-2

There are two classes of metadata:

- **technical metadata** - metadata defined *when* entities are extracted. You cannot modify technical metadata.
- **custom metadata** - metadata [added](#) to extracted entities. You can add and modify custom metadata *before* or *after* entities are extracted.

Metadata Extraction

The [Navigator Metadata Server](#) extracts metadata for the following resource types from the listed servers:

- **HDFS** - Extracts HDFS metadata at the next scheduled extraction run after an HDFS checkpoint. However, if you have high availability enabled, metadata is extracted as soon as it is written to the JournalNodes.
- **Hive** - Extracts database and table metadata from the Hive Metastore Server.
- **Impala** - Extracts database and table metadata from the Hive Metastore Server. Extracts query metadata from the Impala Daemon lineage logs.
- **MapReduce** - Extracts job metadata from the JobTracker. The default setting in Cloudera Manager retains a maximum of five jobs, which means if you run more than five jobs between Navigator extractions, the Navigator Metadata Server would extract the five most recent jobs.
- **Oozie** - Extracts Oozie workflows from the Oozie Server.
- **Pig** - Extracts Pig script runs from the JobTracker or Job History Server.
- **Spark** - Extracts Spark job metadata from the YARN logs.
- **Sqoop 1** - Extracts database and table metadata from the Hive Metastore Server. Extracts job runs from the JobTracker or Job History Server.
- **YARN** - Extracts job metadata from the ResourceManager.

If an entity is created at time t_0 in the system, that entity will be extracted and linked in Navigator after the extraction poll period (default 10 minutes) plus a service-specific interval as follows:

- **HDFS**: $t_0 + \text{extraction poll period} + \text{HDFS checkpoint interval}$ (default 1 hour)
- **HDFS + HA**: $t_0 + \text{extraction poll period}$
- **Hive**: $t_0 + \text{extraction poll period} + \text{Hive maximum wait time}$ (default 60 minutes)
- **Impala**: $t_0 + \text{extraction poll period}$

Metadata Indexing

After metadata is extracted it is indexed and made available for [searching](#) by an embedded [Solr](#) engine. The Solr schema indexes two types of metadata: entity properties and relationship between entities.

You can search entity metadata using the Navigator UI. Relationship metadata is implicitly visible in [lineage diagrams](#) and explicitly available in a [lineage file](#).

Metadata Search

Search is implemented by an embedded Solr engine that supports the syntax described in [LuceneQParserPlugin](#).

Search Syntax

You construct search strings by specifying the value of a [default property](#), property name-value pairs, or user-defined name-value pairs using the syntax:

- **Property name-value pairs** - `propertyName:value`, where
 - `propertyName` is one of the properties listed in [Search Properties](#) on page 23.
 - `value` is a single value or range of values specified as `[value1 TO value2]`. In a value, `*` is a wildcard. In property name-value pairs you must escape special characters `:`, `-`, and `*` with the backslash character `\`. For example, `filePath:/tmp/hbase\-staging`.
- **User-defined name-value pairs** - `up_propertyName:value`.

To construct complex strings, join multiple property-value pairs using the `or` and `and` operators.

Example Search Strings

- Filesystem path `/user/admin` - `filePath:/user/admin`
- Descriptions that start with the string "Banking" - `description:Banking*`
- Sources of type MapReduce or Hive - `sourceType:mapreduce` or `sourceType:hive`
- Directories owned by `hdfs` in the path `/user/hdfs/input` - `owner:hdfs` and `type:directory` and `filePath:/user/hdfs/input`
- Job started between 20:00 to 21:00 UTC - `started:[2013-10-21T20:00:00.000Z TO 2013-10-21T21:00:00.000Z]`
- User-defined key-value `project-customer1` - `up_project:customer1`

- **Note:** When viewing MapReduce jobs in the Cloudera Manager Activities page, the string that appear in a job's Name column equates to the `originalName` property. Therefore, to specify a MapReduce job's name in a search, use the following string: `(sourceType:mapreduce) and (originalName:jobName)`, where `jobName` is the value in the job's Name column.

Search Properties

A reference for the search schema properties.

Default Properties

The following properties can be searched by simply specifying a property value: `type`, `filePath`, `inputs`, `jobId`, `mapper`, `mimeType`, `name`, `originalName`, `outputs`, `owner`, `principal`, `reducer`, `tags`.

Common Properties

Name	Type	Description
<code>description</code>	<code>text</code>	Description of the entity.
<code>group</code>	<code>caseInsensitiveText</code>	The group to which the owner of the entity belongs.
<code>name</code>	<code>ngramedText</code>	The overridden name of the entity. If the name has not been overridden, this value is empty. Names cannot contain spaces.
<code>operationType</code>	<code>ngramedText</code>	The type of an operation: <ul style="list-style-type: none"> ▪ Pig - <code>SCRIPT</code>

Name	Type	Description
		<ul style="list-style-type: none"> Sqoop - Table Export, Query Import
originalName	ngramedText	The name of the entity when it was extracted.
originalDescription	text	The description of the entity when it was extracted.
owner	caseInsensitiveText	The owner of the entity.
principal	caseInsensitiveText	For entities with type OPERATION_EXECUTION, the initiator of the entity.
tags	ngramedText	A set of tags that describe the entity.
type	ngramedText	<p>The type of the entity. The available types depend on the entity's source type:</p> <ul style="list-style-type: none"> hdfs - DIRECTORY, FILE hive - DATABASE, TABLE, FIELD, OPERATION, OPERATION_EXECUTION, SUB_OPERATION, PARTITION, RESOURCE, UNKNOWN, VIEW mapreduce - OPERATION, OPERATION_EXECUTION oozie - OPERATION, OPERATION_EXECUTION pig - OPERATION, OPERATION_EXECUTION sqoop - OPERATION, OPERATION_EXECUTION, SUB_OPERATION yarn - OPERATION, OPERATION_EXECUTION
Query		
queryText	string	The text of a Hive or Sqoop query.
Source		
clusterName	string	The name of the cluster in which the entity is stored.
sourceId	string	The ID of the source type.
sourceType	caseInsensitiveText	The source type of the entity: hdfs, hive, impala, mapreduce, oozie, pig, sqoop, yarn.
Timestamps		
<p>The available timestamp fields vary by the source type:</p> <ul style="list-style-type: none"> hdfs - lastModified, lastAccessed hive - created, lastAccessed impala, mapreduce, pig, sqoop, and yarn - started, ended 	date	<p>Timestamps in the Solr Date Format. For example:</p> <ul style="list-style-type: none"> lastAccessed: [* TO NOW] created: [1976-03-06T23:59:59.999Z TO *] started: [1995-12-31T23:59:59.999Z TO 2007-03-06T00:00:00Z] ended: [NOW-1YEAR/DAY TO NOW/DAY+1DAY] created: [1976-03-06T23:59:59.999Z TO 1976-03-06T23:59:59.999Z+1YEAR] lastAccessed: [1976-03-06T23:59:59.999Z/YEAR TO 1976-03-06T23:59:59.999Z]

HDFS Properties

Name	Type	Description
filePath	path	The path to the entity.
compressed	Boolean	Indicates whether the entity is compressed.
deleted	Boolean	Indicates whether the entity has been moved to the Trash folder.
deleteTime	date	The time the entity was moved to the Trash folder.
mimeType	ngramedText	The MIME type of the entity.
parentPath	string	The path to the parent entity of a child entity. For example: <code>parent path: /default/sample_07</code> for the table <code>sample_07</code> from the Hive database <code>default</code> .
permissions	string	The UNIX access permissions of the entity.
size	long	The exact size of the entity in bytes or a range of sizes. Range examples: <code>size:[1000 TO *]</code> , <code>size: [* TO 2000]</code> , and <code>size:[* TO *]</code> to find all fields with a size value.

MapReduce and YARN Properties

Name	Type	Description
inputRecursive	Boolean	Indicates whether files are searched recursively under the input directories, or just files directly under the input directories are considered.
jobId	ngramedText	The ID of the job. For a job spawned by Oozie, the workflow ID.
mapper	string	The fully-qualified name of the mapper class.
outputKey	string	The fully-qualified name of the class of the output key.
outputValue	string	The fully-qualified name of the class of the output value.
reducer	string	The fully-qualified name of the reducer class.

Operation Properties

Name	Type	Description
Operation		
inputFormat	string	The fully-qualified name of the class of the input format.
outputFormat	string	The fully-qualified name of the class of the output format.
Operation Execution		
inputs	string	The name of the entity input to an operation execution. For entities of resource type MAPREDUCE, it is usually a directory. For entities of resource type Hive, it is usually a table.
outputs	string	The name of the entity output from an operation execution. For entities of resource type MAPREDUCE, it is usually a directory. For entities of resource type Hive, it is usually a table.

Metadata

Hive Properties

Name	Type	Description
Field		
dataType	ngramedText	The type of data stored in a field (column).
Table		
compressed	Boolean	Indicates whether a Hive table is compressed.
serDeLibName	string	The name of the library containing the SerDe class.
serDeName	string	The fully-qualified name of the SerDe class.
Partition		
partitionColNames	string	The table columns that define the partition.
partitionColValues	string	The table column values that define the partition.

Oozie Properties

Name	Type	Description
status	string	The status of the Oozie workflow: RUNNING, SUCCEEDED, or FAILED.

Pig Properties

Name	Type	Description
scriptId	string	The ID of the Pig script.

Sqoop Properties

Name	Type	Description
dbURL	string	The URL of the database from or to which the data was imported or exported.
dbTable	string	The table from or to which the data was imported or exported.
dbUser	string	The database user.
dbWhere	string	A where clause that identifies which rows were imported.
dbColumnExpression	string	An expression that identifies which columns were imported.

Accessing Metadata

Required Role: **Lineage Viewer** **Policy Administrator** **Metadata Administrator** **Full Administrator**

You can access metadata through the Navigator UI or through the Navigator API.

Navigator Metadata UI

Searching Metadata

1. [Start and log into the Cloudera Navigator data management component UI.](#)

2. Do one of the following:

- Type a search string into the **Search** box that conforms to the [search syntax](#) and press **Return** or **Enter**.
- Click the **Click here** link.

The Search page displays.

The Search page has a Search box and two panes: the Filters pane and the Search Results pane.

To display all entities, click **Clear all filters** or type * in the Search box and press **Return** or **Enter**. You filter the search results by specifying filters or typing search strings in the Search box.

Search Results

The Search Results pane displays the number of matching entries **1 to 25 of 83 results** in pages listing 25 entities per page. You can view the pages using the page control « **1** 2 3 4 » at the bottom of each page.

Each entry in the result list contains:

- Source type
- Name - the name is a link to a page that displays the entity [property editor](#) and [lineage diagram](#)
- Properties
- If Hue is running, a link at the far right labeled **View in Hue** that opens the Hue browser for the entity:
 - HDFS directories and files - File Browser
 - Hive database and tables - Metastore Manager
 - MapReduce, YARN, Pig - Job Browser

For example:



The screenshot shows a search result for 'Hive sample_07'. It includes a small icon of a table, the name 'Hive sample_07', and several metadata fields: 'Type: Table', 'Parent Path: /default', 'Path: hdfs://nightly53-5.ent.cloudera.com:8020/user/hive/warehouse/sample_07', 'Owner: sample', 'Created: Dec 9 2014 7:16 AM', and 'Source: HIVE-2'. A blue link labeled 'View in Hue' is visible on the right.

Filtering Search Results

To filter search results, specify filters in the Filters pane or type [search strings](#) in the **Search** box.

The Filters pane contains a set of default properties (source type, type, owner, cluster, tags) and property values (also referred to as facets). You can add a filter by clicking **Add another filter....**





As you add filters, filter breadcrumbs are added between Search box and search results, and search results are refreshed immediately. Multiple filters composed with the AND operator are separated with the | character.

Source Type = Hive ✕ | Type = Table ✕

To remove non-default filter properties, click the ✕ in the filter.

Specify a property value as follows:

- **Boolean** - Click the radio button to respectively not display, or display only those entries, with the value set to true: **Do not show XXX** (the default) or **Show XXX only**, where XXX is the Boolean property.
- **Enumerated or freeform string**
 - Select the checkbox next to a value or click a value link.

- If a property has no values, click **add a new value**, click the text box and select from the populated values in the drop-down list or type a value.
- **Timestamp** - Timestamps are used for started, ended, created, last accessed, and last modified properties. The server stores the timestamp in UTC and the UI displays the timestamp converted to the local timezone. Select one of the timestamp options:
 - A **Last XXX day(s)** link.
 - The **Last** checkbox, type a value, and select minutes, hours, or days using the spinner control .
 - The **Custom period** checkbox and specify the start and end date.
 - **Date** - Click the down arrow  to display a calendar and select a date, or click a field and click the spinner arrows  or up and down arrow keys.
 - **Time** - Click the hour, minute, and AM/PM fields and click the spinner arrows  or up and down arrow keys to specify the value.
 - Move between fields using the right and left arrow keys.

To remove filter values, click the **x** in the breadcrumb or deselect the checkbox.

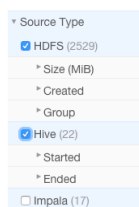
When you select a specific source type value, additional properties that apply to that source type display. For example, HDFS has size, created, and group properties:



The screenshot shows a filter panel for HDFS (22) with a 'Clear' button. It contains three sections:

- * Size (MiB)**: Radio buttons for 0-256 (0), 256-512 (0), 512-1024 (0), and > 1024 (0). There is also a 'Custom (MiB)' option with 'min' and 'max' input fields.
- * Created**: Links for 'last day (22)', 'last 30 days (22)', 'last 90 days (22)', and 'last 365 days (22)'. There is a 'last' checkbox with a text input and a 'minutes' spinner. A 'Custom period' checkbox is also present with 'start' and 'end' date inputs.
- * Group**: A checked 'hive (22)' checkbox and an 'add new value' link.

The number in parentheses (facet count) after a property value is the number of extracted entities that have that property value:



The screenshot shows a 'Source Type' filter panel with a 'Clear' button. It has two main sections:

- * Source Type**: A checked 'HDFS (2529)' checkbox.
- * Hive (22)**: A checked checkbox with sub-sections for '* Started' and '* Ended'.
- * Impala (17)**: An unchecked checkbox.

Facet values with the count of 0 are not displayed.

When you type values, the value is enclosed in quotes; the value inside the quotes must exactly match the metadata. For example, typing "sample_*" in the `originalName` property returns only entities whose names match that exact string. To perform a wildcard search, type the wildcard string in the Search box. For example, typing the string "sample_*" in the Search box returns all entities with "sample_" at the beginning of their original name.

When you construct search strings with filters, multiple values of a given property are added with the `OR` operator. Multiple properties are added with the `AND` operator. For example:

```
(sourceType:hive OR sourceType:hdfs) AND (type:table OR type:directory)
```

and:


```
((sourceType:hdfs AND created:[NOW/DAY-30DAYS TO NOW/DAY+1DAY])
```

To specify different operators, for example to `OR` properties, explicitly type the search string containing `OR`'d properties in the Search box.

Saving Searches

1. Specify a search string or set of filters.
2. Select **Actions** > **Save As...**
3. Specify a name and click **OK**.

Reusing a Saved Search

1. Click the down arrow to the right of  in the Search box and select a saved search name. A label with the saved search name is added over the Search box and search results are refreshed immediately.

Navigator API


The Navigator API allows you to search entity metadata using a REST API. For information about the API, see [Cloudera Navigator Data Management Component API](#).

Modifying Custom Metadata

You can add and modify the following custom metadata associated with entities: display name, description, tags, and user-defined name-value pairs using the Navigator Metadata UI, MapReduce service and job properties, Navigator metadata files, and the Navigator Metadata API.

Required Role: Policy Administrator Metadata Administrator Full Administrator

Modifying Custom Metadata Using the Navigator UI

1. Run a [search](#) in the Navigator UI.
2. Click an entity link returned in the search. The metadata pane displays on the left and the lineage page displays on the right.
3. In the top-right of the metadata pane, click . The Editing *entity* dialog drops down.
4. Edit any of the fields as instructed. Press **Enter** or **Tab** to create new tag entries. For example, a description, the tags `occupations` and `salaries`, and property `year` with value `2012` have been added to the file `sample_07.csv`:

Editing sample_07.csv

×

Provide a name to use when displaying the element:

sample_07.csv

Describe the element:

Occupational categories: salary and number of employees.

Provide a list of tags that relate to this element:

occupations ×

salaries ×

Examples: user, metadata or logs

Provide any named values that are relevant:

year

:

2012

+

-

Save

Cancel

You can specify special characters (for example, ".", " ") in the name, but it will make searching for the entity more difficult as some characters collide with special characters in the [search syntax](#).

5. Click **Save**. The new metadata appears in the metadata pane:

sample_07.csv

Occupational categories: salary and number of employees.

tags: occupations salaries

source type: HDFS

category: FILE

path: /user/hdfs/sample_07.csv

owner: hdfs

group: supergroup

size: 44.98KiB

last accessed: Oct 8 2013 1:33 PM

last modified: Oct 8 2013 1:33 PM

year: 2012

Modifying MapReduce Custom Metadata

You can set MapReduce job metadata statically for all jobs, or dynamically for a specific job or job instance.

To statically set metadata for all MapReduce jobs, do the following:

1. Do one of the following:
 - Select **Clusters > Cloudera Management Service > Cloudera Management Service**.
 - On the Status tab of the Home page, in **Cloudera Management Service** table, click the **Cloudera Management Service** link.
2. Click the **Configuration** tab.
3. Select **Scope > Navigator Metadata Server**.
4. Select **Category > Advanced**.
5. Click **Navigator Metadata Server Advanced Configuration Snippet for cloudera-navigator.properties**.
6. Specify values for one or more of the following properties:
 - `nav.user_defined_properties` = comma-separated list of user-defined property names
 - `nav.tags` = comma-separated list of property names that serve as tags
7. Click **Save Changes**.
8. Click the **Instances** tab.
9. Restart the role.

To modify parameters dynamically, specify one or more of the following properties in a job configuration:

- `nav.job.user_defined_properties` = comma-separated list of user-defined property names for a job (`type:OPERATION`)
- `nav.job.tags` = comma-separated list of property names that serve as tags for a job
- `nav.jobexec.user_defined_properties` = comma-separated list of user-defined property names for a job execution (`type:OPERATION_EXECUTION`)
- `nav.jobexec.tags` = comma-separated list of property names that serve as tags for a job execution

For example, to capture `hive.mapred.partitioner` in the MapReduce jobs that are launched through Hive, set `nav.user_defined_properties=hive.mapred.partitioner`.

Modifying HDFS Custom Metadata Using Metadata Files

You can add tags and properties to HDFS entities using metadata files. The reasons to use metadata files are to assign metadata to entities in bulk and to create metadata before the metadata is extracted. A metadata file is a JSON file with the following structure:

```
{
  "name" : "aName",
  "description" : "a description",
  "properties" : {
    "prop1" : "value1", "prop2" : "value2"
  },
  "tags" : [ "tag1" ]
}
```

To add metadata files to files and directories, create a metadata file with the extension `.navigator`, naming the files as follows:

- **File** - The path of the metadata file must be `.filename.navigator`. For example, to apply properties to the file `/user/test/file1.txt`, the metadata file path is `/user/test/.file1.txt.navigator`.
- **Directory** - The path of the metadata file must be `dirpath/.navigator`. For example, to apply properties to the directory `/user`, the metadata path must be `/user/.navigator`.

The metadata file is applied to the entity metadata when the extractor runs.

Modifying HDFS and Hive Custom Metadata Using the Navigator API

You can use the [Cloudera Navigator Data Management Component API](#) to modify the metadata of HDFS or Hive entities whether or not the entities have been extracted. If an entity has been extracted at the time the API is called, the metadata will be applied immediately. If the entity has not been extracted, you can preregister metadata which is then applied once the entity is extracted. Metadata is saved regardless of whether or not a matching entity is extracted, and Navigator does not perform any cleanup of unused metadata.

If you call the API before the entity is extracted, the metadata is stored with the entity's identity, source ID, metadata fields (name, description, tags, properties), and the fields relevant to the identifier. The rest of the entity fields (such as type) will not be present. To view all stored metadata, you can use the API to search for entities without an internal type:

```
curl http://Navigator_Metadata_Server_host:port/api/v5/entities/?query=-internalType:*
-u username:password -X GET
```

The metadata provided via the API overwrites existing metadata. If, for example, you call the API with an empty name and description, empty array for tags, and empty dictionary for properties, the call removes this metadata. If you leave out the tags or properties fields, the existing values remain unchanged.

Modifying metadata using HDFS metadata files and the metadata API at the same time *is not* supported. You must use one or the other, because the two methods behave slightly differently. Metadata specified in files is merged with existing metadata whereas the API overwrites metadata. Also, the updates provided by metadata files wait in a queue before being merged, but API changes are committed immediately. This means there may be some inconsistency if a metadata file is being merged around the same time the API is in use.

You modify metadata using either the `PUT` or `POST` method. Use the `PUT` method if the entity has been extracted and the `POST` method to preregister metadata. The syntax of the methods are:

- `PUT`

```
curl http://Navigator_Metadata_Server_host:port/api/v5/entities/identity -u
username:password -X PUT -H\
"Content-Type: application/json" -d '{properties}'
```

where *identity* is an entity ID and *properties* are:

- name: name metadata
- description: description metadata
- tags: tag metadata
- properties: property metadata

All existing naming rules apply, and if any value is invalid, the entire request will be denied.

- `POST`

```
curl http://Navigator_Metadata_Server_host:port/api/v5/entities/ -u
username:password -X POST -H\
"Content-Type: application/json" -d '{properties}'
```

where *properties* are:

- [sourceId](#) (required): An existing source ID. After the first extraction, you can retrieve source IDs using the call:

```
curl http://Navigator_Metadata_Server_host:port/api/v5/entities/?query=type:SOURCE
-u username:password -X GET
```

For example:

```
{
  ...
  {
    "identity" : "a09b0233cc58ff7d601eaa68673a20c6",
```



```

    "originalName" : "HDFS-1",
    "sourceId" : null,
    "firstClassParentId" : null,
    "parentPath" : null,
    "extractorRunId" : null,
    "name" : "HDFS-1",
    "description" : null,
    "tags" : null,
    "properties" : null,
    "clusterName" : "Cluster 1",
    "sourceUrl" : "hdfs://hostname:8020",
    "sourceType" : "HDFS",
    "sourceExtractIteration" : 4935,
    "type" : "SOURCE",
    "internalType" : "source"
  }, ...

```

If you have multiple services of a given type, you must specify the source ID that contains the entity you're expecting it to match.

- `parentPath`: The path of the parent entity, defined as:
 - HDFS file or directory: `filePath` of the parent directory (do not provide this field if the entity being affected is the root directory). Example `parentPath` for `/user/admin/input_dir`: `/user/admin`. If you add metadata to a directory, the metadata does not propagate to any files and folders in that directory.
 - Hive database: If you are updating database metadata, you do not specify this field.
 - Hive table or view: The name of database containing the table or view. Example for a table in the default database: `default`.
 - Hive column: *database name/ table name/ view name*. Example for a column in the `sample_07` table: `default/sample_07`.
- `originalName` (required): The name as defined by the source system.
 - HDFS file or directory: name of file or directory (`ROOT` if the entity is the root directory). Example `originalName` for `/user/admin/input_dir`: `input_dir`.
 - Hive database, table, view, or column: the name of the database, table, view, or column.
 - Example for default database: `default`
 - Example for `sample_07` table: `sample_07`
- `name`: name metadata
- `description`: description metadata
- `tags`: tag metadata
- `properties`: property metadata

All existing naming rules apply, and if any value is invalid, the entire request will be denied.

HDFS PUT Example for `/user/admin/input_dir` Directory

```

curl http://Navigator_Metadata_Server_host:port/api/v5/entities/e461de8de38511a3ac6740dd7d51b8d0 \
-u username:password -X PUT -H "Content-Type: application/json" \
-d '{"name":"my_name","description":"My description", \
"tags":["tag1","tag2"],"properties":{"property1":"value1","property2":"value2"}}'

```

HDFS POST Example for `/user/admin/input_dir` Directory

```

curl http://Navigator_Metadata_Server_host:port/api/v5/entities/ -u username:password \
-X POST -H "Content-Type: application/json" \
-d '{"sourceId":"a09b0233cc58ff7d601eaa68673a20c6", \
"parentPath":"/user/admin","originalName":"input_dir", "name":"my_name","description":"My \
description", \
"tags":["tag1","tag2"],"properties":{"property1":"value1","property2":"value2"}}'

```

Hive POST Example for total_emp Column

```
curl http://Navigator_Metadata_Server_host:port/api/v5/entities/ -u username:password \
-X POST -H "Content-Type: application/json" \
-d '{"sourceId":"4fbdadc6899638782fc8cb626176dc7b",\
"parentPath":"default/sample_07","originalName":"total_emp",\
"name":"my_name","description":"My description",\
"tags":["tag1","tag2"],"properties":{"property1":"value1","property2":"value2"}}'
```

Policies

A policy defines a set of actions performed when a class of entities is extracted. The following actions are supported:

- Adding custom metadata such as tags and properties.
- Sending a message to a JMS message queue. The JSON format message contains the metadata of the entity to which the policy applies and the message text specified in the policy:

```
{"entity":entity_properties, "userMessage":"some message text"}
```

For each action, certain properties support specifying a value using a [policy expression](#).

Viewing Policies

Required Role: **Policy Viewer** **Policy Administrator** **Full Administrator**

1. [Start and log into the Cloudera Navigator data management component UI](#).
2. Click the **Policies** tab.
3. In the left pane, click a policy.

Creating Policies

Required Role: **Policy Administrator** **Full Administrator**

1. [Start and log into the Cloudera Navigator data management component UI](#).
2. Depending on the starting point, do one of the following:

Action	Procedure
Policies page	<ol style="list-style-type: none"> 1. Click the Policies tab. 2. Click Create a New Policy.
Search results page	<ol style="list-style-type: none"> 1. Select Actions > Create a policy.

3. Enter a name for the policy.
4. Check the **Enable Policy** checkbox.
5. Specify the [search query](#) that defines the class of entities to which the policy applies. If you arrive at the Policies page by clicking a search result, the query property is populated with the query that generated the result. To display a list of entities that satisfy a search query, click the **Search Results** link.
6. Specify an optional description for the policy.
7. If [policy expressions](#) are enabled and you choose to use policy expressions in properties that support expressions, specify required imports in the **Import Statements** field. For example, if your policy expression uses [policy enums](#), for example: `entity.get(FSEntityProperties.ORIGINAL_NAME, Object.class)` your import would be: `import com.cloudera.nav.hdfs.model.FSEntityProperties`. If your expression uses another library, such as Date, add the required import for that library.
8. Choose the schedule for applying the policy: On Data Change, Immediate, Once, or Recurring.
9. Check the checkboxes next to the desired actions and follow the appropriate procedure:

Action	Procedure
Assign Metadata	<ol style="list-style-type: none"> 1. Specify the custom metadata. Optionally check the Expression checkbox and specify a policy expression for the indicated fields.

Action	Procedure
Send Notification to JMS	<ol style="list-style-type: none"> 1. If not already configured, configure a JMS server and queue. 2. Specify the queue name and message. Optionally check the Expression checkbox and specify a policy expression for the message.

10. Click **Save**.

Cloning and Editing a Policy

Required Role: **Policy Administrator** **Full Administrator**

1. [Start and log into the Cloudera Navigator data management component UI](#).
2. Click the **Policies** tab.
3. In the left pane, click a policy.
4. Click **Clone Policy** or **Edit Policy**.
5. Edit the policy name, search query, or policy actions.
6. Click **Save**.

Deleting Policies

Required Role: **Policy Administrator** **Full Administrator**

1. [Start and log into the Cloudera Navigator data management component UI](#).
2. Click the **Policies** tab.
3. In the left pane, click a policy.
4. Click **Delete** and click **OK** to confirm.

Policy Expressions

Policy expressions allow certain policy properties to be specified programmatically using Java expressions instead of string literals.

- **Note:** A policy expression must evaluate to a string.

Policy expressions are not enabled by default. To enable policy expressions, follow the procedure in [Enabling and Disabling Policy Expression Input](#).

The supported policy properties are entity name and description, key-value pairs, and JMS notification message.

Entity Properties in Policy Expressions

To include entity properties in property expressions, use the `entity.get` method, which takes a property and a return type:

```
entity.get(XXProperties.Property, return_type)
```

`XXProperties.Property` is the Java enumerated value representing an entity property, where

- `XX` is [FSEntity](#), [HiveColumn](#), [HiveDatabase](#), [HivePartition](#), [HiveQueryExecution](#), [HiveQueryPart](#), [HiveQuery](#), [HiveTable](#), [HiveView](#), [JobExecution](#), [Job](#), [WorkflowInstance](#), [Workflow](#), [PigField](#), [PigOperationExecution](#), [PigOperation](#), [PigRelation](#), [SqoopExportSubOperation](#), [SqoopImportSubOperation](#), [SqoopOperationExecution](#), [SqoopQueryOperation](#), [SqoopTableExportOperation](#), or [SqoopTableImportOperation](#).
- `Property` is one of the properties listed in [Entity Property Enum Reference](#) on page 37.

If you don't need to specify a return type, use `Object.class` as the return type. However, if you want to do type-specific operations with the result, set the return type to the type in the comment in the enum property reference. For example, in `FSEntityProperties`, the return type of the `ORIGINAL_NAME` property is `java.lang.String`. If you use `String.class` as the return type, you can use the `String` method `toLowerCase()` to modify the returned value: `entity.get(FSEntityProperties.ORIGINAL_NAME, String.class).toLowerCase()`.

Expression Examples

- Set a filesystem entity name to the original name concatenated with the entity type:

```
entity.get(FSEntityProperties.ORIGINAL_NAME, Object.class) + " " +
entity.get(FSEntityProperties.TYPE, Object.class)
```

Import Statements:

```
import com.cloudera.nav.hdfs.model.FSEntityProperties;
```

- Expression to add the `CREATED` date to the name:

```
entity.get(FSEntityProperties.ORIGINAL_NAME, Object.class) + " - "
+ new SimpleDateFormat("yyyy-MM-dd").format(entity.get(FSEntityProperties.CREATED,
Instant.class).toDate())
```

Import Statements:

```
import com.cloudera.nav.hdfs.model.FSEntityProperties; import
java.text.SimpleDateFormat; import org.joda.time.Instant;
```

Entity Property Enum Reference

The following reference lists the Java enumerated values for retrieving properties of each entity type.

```
com.cloudera.nav.hdfs.model.FSEntityProperties
public enum FSEntityProperties implements PropertyEnum {
    PERMISSIONS, // Return type: java.lang.String
    TYPE, // Return type: java.lang.String
    SIZE, // Return type: java.lang.Long
    OWNER, // Return type: java.lang.String
    LAST_MODIFIED, // Return type: org.joda.time.Instant
    SOURCE_TYPE, // Return type: java.lang.String
    DELETED, // Return type: java.lang.Boolean
    FILE_SYSTEM_PATH, // Return type: java.lang.String
    CREATED, // Return type: org.joda.time.Instant
    LAST_ACCESSED, // Return type: org.joda.time.Instant
    GROUP, // Return type: java.lang.String
    MIME_TYPE, // Return type: java.lang.String
    DELETE_TIME, // Return type: java.lang.Long
    NAME, // Return type: java.lang.String
    ORIGINAL_NAME, // Return type: java.lang.String
    USER_ENTITY, // Return type: boolean
    SOURCE_ID, // Return type: java.lang.String
    EXTRACTOR_RUN_ID, // Return type: java.lang.String
    PARENT_PATH, // Return type: java.lang.String
}
```

```
com.cloudera.nav.hive.model.HiveColumnProperties
public enum HiveColumnProperties implements PropertyEnum {
    TYPE, // Return type: java.lang.String
    SOURCE_TYPE, // Return type: java.lang.String
    DELETED, // Return type: java.lang.Boolean
    DATA_TYPE, // Return type: java.lang.String
    ORIGINAL_DESCRIPTION, // Return type: java.lang.String
    NAME, // Return type: java.lang.String
    ORIGINAL_NAME, // Return type: java.lang.String
    USER_ENTITY, // Return type: boolean
}
```

```

SOURCE_ID, // Return type: java.lang.String
EXTRACTOR_RUN_ID, // Return type: java.lang.String
PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.hive.model.HiveDatabaseProperties
public enum HiveDatabaseProperties implements PropertyEnum {
    TYPE, // Return type: java.lang.String
    ORIGINAL_DESCRIPTION, // Return type: java.lang.String
    SOURCE_TYPE, // Return type: java.lang.String
    DELETED, // Return type: java.lang.Boolean
    FILE_SYSTEM_PATH, // Return type: java.lang.String
    NAME, // Return type: java.lang.String
    ORIGINAL_NAME, // Return type: java.lang.String
    USER_ENTITY, // Return type: boolean
    SOURCE_ID, // Return type: java.lang.String
    EXTRACTOR_RUN_ID, // Return type: java.lang.String
    PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.hive.model.HivePartitionProperties
public enum HivePartitionProperties implements PropertyEnum {
    TYPE, // Return type: java.lang.String
    SOURCE_TYPE, // Return type: java.lang.String
    DELETED, // Return type: java.lang.Boolean
    FILE_SYSTEM_PATH, // Return type: java.lang.String
    CREATED, // Return type: org.joda.time.Instant
    LAST_ACCESSED, // Return type: org.joda.time.Instant
    COL_VALUES, // Return type: java.util.List
    NAME, // Return type: java.lang.String
    ORIGINAL_NAME, // Return type: java.lang.String
    USER_ENTITY, // Return type: boolean
    SOURCE_ID, // Return type: java.lang.String
    EXTRACTOR_RUN_ID, // Return type: java.lang.String
    PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.hive.model.HiveQueryExecutionProperties
public enum HiveQueryExecutionProperties implements PropertyEnum {
    SOURCE_TYPE, // Return type: java.lang.String
    TYPE, // Return type: java.lang.String
    ENDED, // Return type: org.joda.time.Instant
    INPUTS, // Return type: java.util.Collection
    OUTPUTS, // Return type: java.util.Collection
    STARTED, // Return type: org.joda.time.Instant
    PRINCIPAL, // Return type: java.lang.String
    WF_INST_ID, // Return type: java.lang.String
    NAME, // Return type: java.lang.String
    ORIGINAL_NAME, // Return type: java.lang.String
    USER_ENTITY, // Return type: boolean
    SOURCE_ID, // Return type: java.lang.String
    EXTRACTOR_RUN_ID, // Return type: java.lang.String
    PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.hive.model.HiveQueryPartProperties
public enum HiveQueryPartProperties implements PropertyEnum {
    TYPE, // Return type: java.lang.String
    SOURCE_TYPE, // Return type: java.lang.String
    NAME, // Return type: java.lang.String
    ORIGINAL_NAME, // Return type: java.lang.String
    USER_ENTITY, // Return type: boolean
    SOURCE_ID, // Return type: java.lang.String
    EXTRACTOR_RUN_ID, // Return type: java.lang.String
}

```

```
PARENT_PATH; // Return type: java.lang.String
}
```

```
com.cloudera.nav.hive.model.HiveQueryProperties
public enum HiveQueryProperties implements PropertyEnum {
    SOURCE_TYPE, // Return type: java.lang.String
    INPUTS, // Return type: java.util.Collection
    OUTPUTS, // Return type: java.util.Collection
    QUERY_TEXT, // Return type: java.lang.String
    TYPE, // Return type: java.lang.String
    WF_IDS, // Return type: java.util.Collection
    NAME, // Return type: java.lang.String
    ORIGINAL_NAME, // Return type: java.lang.String
    USER_ENTITY, // Return type: boolean
    SOURCE_ID, // Return type: java.lang.String
    EXTRACTOR_RUN_ID, // Return type: java.lang.String
    PARENT_PATH; // Return type: java.lang.String
}
```

```
com.cloudera.nav.hive.model.HiveTableProperties
public enum HiveTableProperties implements PropertyEnum {
    OWNER, // Return type: java.lang.String
    INPUT_FORMAT, // Return type: java.lang.String
    OUTPUT_FORMAT, // Return type: java.lang.String
    DELETED, // Return type: java.lang.Boolean
    FILE_SYSTEM_PATH, // Return type: java.lang.String
    COMPRESSED, // Return type: java.lang.Boolean
    PARTITION_COL_NAMES, // Return type: java.util.List
    CLUSTERED_BY_COL_NAMES, // Return type: java.util.List
    SORT_BY_COL_NAMES, // Return type: java.util.List
    SER_DE_NAME, // Return type: java.lang.String
    SER_DE_LIB_NAME, // Return type: java.lang.String
    TYPE, // Return type: java.lang.String
    SOURCE_TYPE, // Return type: java.lang.String
    CREATED, // Return type: org.joda.time.Instant
    LAST_ACCESSED, // Return type: org.joda.time.Instant
    NAME, // Return type: java.lang.String
    ORIGINAL_NAME, // Return type: java.lang.String
    USER_ENTITY, // Return type: boolean
    SOURCE_ID, // Return type: java.lang.String
    EXTRACTOR_RUN_ID, // Return type: java.lang.String
    PARENT_PATH; // Return type: java.lang.String
}
```

```
com.cloudera.nav.hive.model.HiveViewProperties
public enum HiveViewProperties implements PropertyEnum {
    DELETED, // Return type: java.lang.Boolean
    QUERY_TEXT, // Return type: java.lang.String
    TYPE, // Return type: java.lang.String
    SOURCE_TYPE, // Return type: java.lang.String
    CREATED, // Return type: org.joda.time.Instant
    LAST_ACCESSED, // Return type: org.joda.time.Instant
    NAME, // Return type: java.lang.String
    ORIGINAL_NAME, // Return type: java.lang.String
    USER_ENTITY, // Return type: boolean
    SOURCE_ID, // Return type: java.lang.String
    EXTRACTOR_RUN_ID, // Return type: java.lang.String
    PARENT_PATH; // Return type: java.lang.String
}
```

```
com.cloudera.nav.mapreduce.model.JobExecutionProperties
public enum JobExecutionProperties implements PropertyEnum {
    SOURCE_TYPE, // Return type: java.lang.String
    JOB_ID, // Return type: java.lang.String
    ENDED, // Return type: org.joda.time.Instant
    INPUT_RECURSIVE, // Return type: boolean
    TYPE, // Return type: java.lang.String
    INPUTS, // Return type: java.util.Collection
    OUTPUTS, // Return type: java.util.Collection
}
```

```

STARTED, // Return type: org.joda.time.Instant
PRINCIPAL, // Return type: java.lang.String
WF_INST_ID, // Return type: java.lang.String
NAME, // Return type: java.lang.String
ORIGINAL_NAME, // Return type: java.lang.String
USER_ENTITY, // Return type: boolean
SOURCE_ID, // Return type: java.lang.String
EXTRACTOR_RUN_ID, // Return type: java.lang.String
PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.mapreduce.model.JobProperties
public enum JobProperties implements PropertyEnum {
    ORIGINAL_NAME, // Return type: java.lang.String
    INPUT_FORMAT, // Return type: java.lang.String
    OUTPUT_FORMAT, // Return type: java.lang.String
    OUTPUT_KEY, // Return type: java.lang.String
    OUTPUT_VALUE, // Return type: java.lang.String
    MAPPER, // Return type: java.lang.String
    REDUCER, // Return type: java.lang.String
    SOURCE_TYPE, // Return type: java.lang.String
    TYPE, // Return type: java.lang.String
    WF_IDS, // Return type: java.util.Collection
    NAME, // Return type: java.lang.String
    USER_ENTITY, // Return type: boolean
    SOURCE_ID, // Return type: java.lang.String
    EXTRACTOR_RUN_ID, // Return type: java.lang.String
    PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.oozie.model.WorkflowInstanceProperties
public enum WorkflowInstanceProperties implements PropertyEnum {
    TYPE, // Return type: java.lang.String
    SOURCE_TYPE, // Return type: java.lang.String
    CREATED, // Return type: org.joda.time.Instant
    JOB_ID, // Return type: java.lang.String
    STATUS, // Return type: java.lang.String
    ENDED, // Return type: org.joda.time.Instant
    INPUTS, // Return type: java.util.Collection
    OUTPUTS, // Return type: java.util.Collection
    STARTED, // Return type: org.joda.time.Instant
    PRINCIPAL, // Return type: java.lang.String
    WF_INST_ID, // Return type: java.lang.String
    NAME, // Return type: java.lang.String
    ORIGINAL_NAME, // Return type: java.lang.String
    USER_ENTITY, // Return type: boolean
    SOURCE_ID, // Return type: java.lang.String
    EXTRACTOR_RUN_ID, // Return type: java.lang.String
    PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.oozie.model.WorkflowProperties
public enum WorkflowProperties implements PropertyEnum {
    TYPE, // Return type: java.lang.String
    SOURCE_TYPE, // Return type: java.lang.String
    WF_IDS, // Return type: java.util.Collection
    NAME, // Return type: java.lang.String
    ORIGINAL_NAME, // Return type: java.lang.String
    USER_ENTITY, // Return type: boolean
    SOURCE_ID, // Return type: java.lang.String
    EXTRACTOR_RUN_ID, // Return type: java.lang.String
    PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.pig.model.PigFieldProperties
public enum PigFieldProperties implements PropertyEnum {
    TYPE, // Return type: java.lang.String
    INDEX, // Return type: int
    SOURCE_TYPE, // Return type: java.lang.String
}

```



```

DATA_TYPE, // Return type: java.lang.String
NAME, // Return type: java.lang.String
ORIGINAL_NAME, // Return type: java.lang.String
USER_ENTITY, // Return type: boolean
SOURCE_ID, // Return type: java.lang.String
EXTRACTOR_RUN_ID, // Return type: java.lang.String
PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.pig.model.PigOperationExecutionProperties
public enum PigOperationExecutionProperties implements PropertyEnum {
    SOURCE_TYPE, // Return type: java.lang.String
    TYPE, // Return type: java.lang.String
    ENDED, // Return type: org.joda.time.Instant
    INPUTS, // Return type: java.util.Collection
    OUTPUTS, // Return type: java.util.Collection
    STARTED, // Return type: org.joda.time.Instant
    PRINCIPAL, // Return type: java.lang.String
    WF_INST_ID, // Return type: java.lang.String
    NAME, // Return type: java.lang.String
    ORIGINAL_NAME, // Return type: java.lang.String
    USER_ENTITY, // Return type: boolean
    SOURCE_ID, // Return type: java.lang.String
    EXTRACTOR_RUN_ID, // Return type: java.lang.String
    PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.pig.model.PigOperationProperties
public enum PigOperationProperties implements PropertyEnum {
    SOURCE_TYPE, // Return type: java.lang.String
    OPERATION_TYPE, // Return type: java.lang.String
    SCRIPT_ID, // Return type: java.lang.String
    TYPE, // Return type: java.lang.String
    WF_IDS, // Return type: java.util.Collection
    NAME, // Return type: java.lang.String
    ORIGINAL_NAME, // Return type: java.lang.String
    USER_ENTITY, // Return type: boolean
    SOURCE_ID, // Return type: java.lang.String
    EXTRACTOR_RUN_ID, // Return type: java.lang.String
    PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.pig.model.PigRelationProperties
public enum PigRelationProperties implements PropertyEnum {
    TYPE, // Return type: java.lang.String
    SOURCE_TYPE, // Return type: java.lang.String
    FILE_SYSTEM_PATH, // Return type: java.lang.String
    SCRIPT_ID, // Return type: java.lang.String
    NAME, // Return type: java.lang.String
    ORIGINAL_NAME, // Return type: java.lang.String
    USER_ENTITY, // Return type: boolean
    SOURCE_ID, // Return type: java.lang.String
    EXTRACTOR_RUN_ID, // Return type: java.lang.String
    PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.sqoop.model.SqoopExportSubOperationProperties
public enum SqoopExportSubOperationProperties implements PropertyEnum {
    TYPE, // Return type: java.lang.String
    SOURCE_TYPE, // Return type: java.lang.String
    INPUTS, // Return type: java.util.Collection
    FIELD_INDEX, // Return type: int
    NAME, // Return type: java.lang.String
    ORIGINAL_NAME, // Return type: java.lang.String
    USER_ENTITY, // Return type: boolean
    SOURCE_ID, // Return type: java.lang.String
    EXTRACTOR_RUN_ID, // Return type: java.lang.String
}

```

```

    PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.sqoop.model.SqoopImportSubOperationProperties
public enum SqoopImportSubOperationProperties implements PropertyEnum {
    DB_COLUMN_EXPRESSION, // Return type: java.lang.String
    TYPE, // Return type: java.lang.String
    SOURCE_TYPE, // Return type: java.lang.String
    INPUTS, // Return type: java.util.Collection
    FIELD_INDEX, // Return type: int
    NAME, // Return type: java.lang.String
    ORIGINAL_NAME, // Return type: java.lang.String
    USER_ENTITY, // Return type: boolean
    SOURCE_ID, // Return type: java.lang.String
    EXTRACTOR_RUN_ID, // Return type: java.lang.String
    PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.sqoop.model.SqoopOperationExecutionProperties
public enum SqoopOperationExecutionProperties implements PropertyEnum {
    SOURCE_TYPE, // Return type: java.lang.String
    TYPE, // Return type: java.lang.String
    ENDED, // Return type: org.joda.time.Instant
    INPUTS, // Return type: java.util.Collection
    OUTPUTS, // Return type: java.util.Collection
    STARTED, // Return type: org.joda.time.Instant
    PRINCIPAL, // Return type: java.lang.String
    WF_INST_ID, // Return type: java.lang.String
    NAME, // Return type: java.lang.String
    ORIGINAL_NAME, // Return type: java.lang.String
    USER_ENTITY, // Return type: boolean
    SOURCE_ID, // Return type: java.lang.String
    EXTRACTOR_RUN_ID, // Return type: java.lang.String
    PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.sqoop.model.SqoopQueryOperationProperties
public enum SqoopQueryOperationProperties implements PropertyEnum {
    SOURCE_TYPE, // Return type: java.lang.String
    INPUTS, // Return type: java.util.Collection
    QUERY_TEXT, // Return type: java.lang.String
    DB_USER, // Return type: java.lang.String
    DB_URL, // Return type: java.lang.String
    OPERATION_TYPE, // Return type: java.lang.String
    TYPE, // Return type: java.lang.String
    WF_IDS, // Return type: java.util.Collection
    NAME, // Return type: java.lang.String
    ORIGINAL_NAME, // Return type: java.lang.String
    USER_ENTITY, // Return type: boolean
    SOURCE_ID, // Return type: java.lang.String
    EXTRACTOR_RUN_ID, // Return type: java.lang.String
    PARENT_PATH; // Return type: java.lang.String
}

```

```

com.cloudera.nav.sqoop.model.SqoopTableExportOperationProperties
public enum SqoopTableExportOperationProperties implements PropertyEnum {
    DB_TABLE, // Return type: java.lang.String
    SOURCE_TYPE, // Return type: java.lang.String
    DB_USER, // Return type: java.lang.String
    DB_URL, // Return type: java.lang.String
    OPERATION_TYPE, // Return type: java.lang.String
    TYPE, // Return type: java.lang.String
    WF_IDS, // Return type: java.util.Collection
    NAME, // Return type: java.lang.String
    ORIGINAL_NAME, // Return type: java.lang.String
    USER_ENTITY, // Return type: boolean
    SOURCE_ID, // Return type: java.lang.String
    EXTRACTOR_RUN_ID, // Return type: java.lang.String
}

```

```
    PARENT_PATH; // Return type: java.lang.String  
}
```

```
com.cloudera.nav.sqoop.model.SqoopTableImportOperationProperties  
public enum SqoopTableImportOperationProperties implements PropertyEnum {
```

```
    DB_TABLE, // Return type: java.lang.String  
    DB_WHERE, // Return type: java.lang.String  
    SOURCE_TYPE, // Return type: java.lang.String  
    DB_USER, // Return type: java.lang.String  
    DB_URL, // Return type: java.lang.String  
    OPERATION_TYPE, // Return type: java.lang.String  
    TYPE, // Return type: java.lang.String  
    WF_IDS, // Return type: java.util.Collection  
    NAME, // Return type: java.lang.String  
    ORIGINAL_NAME, // Return type: java.lang.String  
    USER_ENTITY, // Return type: boolean  
    SOURCE_ID, // Return type: java.lang.String  
    EXTRACTOR_RUN_ID, // Return type: java.lang.String  
    PARENT_PATH; // Return type: java.lang.String  
}
```

Lineage Diagrams

Required Role: Lineage Viewer Metadata Administrator Full Administrator

















A **lineage diagram** is a directed graph that depicts an entity and its relations with other entities. A lineage diagram is limited to 3000 entities.

There are two types of lineage diagrams:

- **Template** - represents an entity that is a model for other entities
- **Instance** - represents an instance or execution of a template

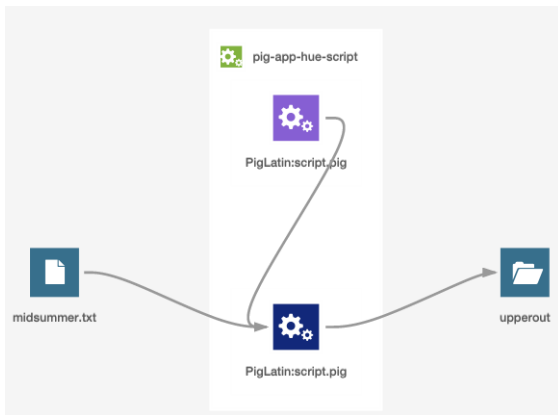
Entities


In a lineage diagram, entity types are represented by icons:


HDFS		Pig	
<ul style="list-style-type: none"> ▪ File ▪ Directory 	<ul style="list-style-type: none"> ▪  ▪  	<ul style="list-style-type: none"> ▪ Table ▪ Pig script ▪ Pig script execution 	<ul style="list-style-type: none"> ▪  ▪  ▪ 
Hive and Impala		Spark (Unsupported and disabled by default. To enable, see Enabling Spark Lineage .)	
<ul style="list-style-type: none"> ▪ Table ▪ Query template ▪ Query execution 	<ul style="list-style-type: none"> ▪  ▪  ▪  	<ul style="list-style-type: none"> ▪ Job template ▪ Job execution 	<ul style="list-style-type: none"> ▪  ▪ 
MapReduce and YARN		Sqoop	
<ul style="list-style-type: none"> ▪ Job template ▪ Job execution 	<ul style="list-style-type: none"> ▪  ▪  	<ul style="list-style-type: none"> ▪ Job template ▪ Job execution 	<ul style="list-style-type: none"> ▪  ▪ 
Oozie			
<ul style="list-style-type: none"> ▪ Job template ▪ Job execution 	<ul style="list-style-type: none"> ▪  ▪  		

- **Note:** Tables created by Impala queries and Sqoop jobs are represented as Hive entities.

Parent entities are represented by a white box enclosing other entities. The following lineage diagram illustrates the relations between the YARN job `script.pig` and Pig script `script.pig` invoked by the parent Oozie workflow `pig-app-hue-script` and its source file `midsummer.txt` and destination folder `upperout`:



Note: In the following circumstances the entity type icon will appear as :

- Entities are not yet extracted. In this case  will eventually be replaced with the correct entity icon after the entity is extracted and linked in Navigator. For information on how long it takes for newly created entities to be extracted, see [Metadata Extraction](#) on page 22.
- Hive entities have been deleted from the system before they could be extracted by Navigator.

Relations

Relations between the entities are represented graphically by gray lines, with arrows indicating the direction of the data flow. There are the following types of relations:

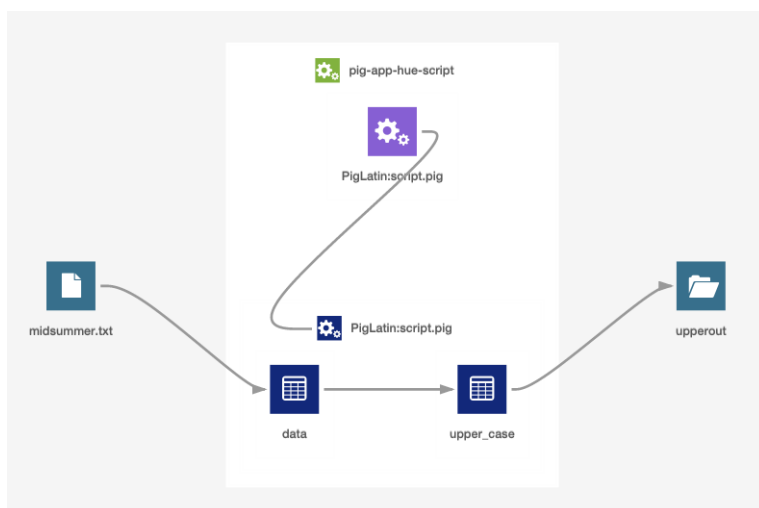
Relation Type	Description
Data flow	Describes a relation between data and a processing activity. For example, between a file and a MapReduce job or vice versa.
Alias	Describes an alias relation. For example, from a table to a synonym.
Parent-child	Describes a parent-child relation. For example, between a directory and a file.
Logical-physical	Describes the relation between a logical entity and its physical entity. For example, between a Hive query and a MapReduce job.
Conjoint	Describes a non-directional relation. For example, between a table and an index.
Instance of	Describes the relation between a template and its instance. For example, an operation execution is an instance of operation.
Control flow	Describes a relation where the source entity controls the data flow of the target entity. For example, between the columns used in an <code>insert</code> clause and the <code>where</code> clause of a Hive query.

For lines connecting database columns, a dashed line indicates that the column is in the `where` clause; a solid line indicates that the column is in the `select` clause.

Manipulating Lineage Diagrams

You can click a parent entity to display its child entities. For example, you can click the Pig script to display its child tables:

Lineage Diagrams



- To improve the layout of a lineage diagram you can drag and drop entities (in this case `midsummer.txt` and `upperout`) located outside a parent box.
- You can use the mouse scroll wheel to zoom the lineage diagram in and out.
- You can move the lineage diagram in the lineage pane by pressing the mouse button and dragging it.

Displaying a Template Lineage Diagram

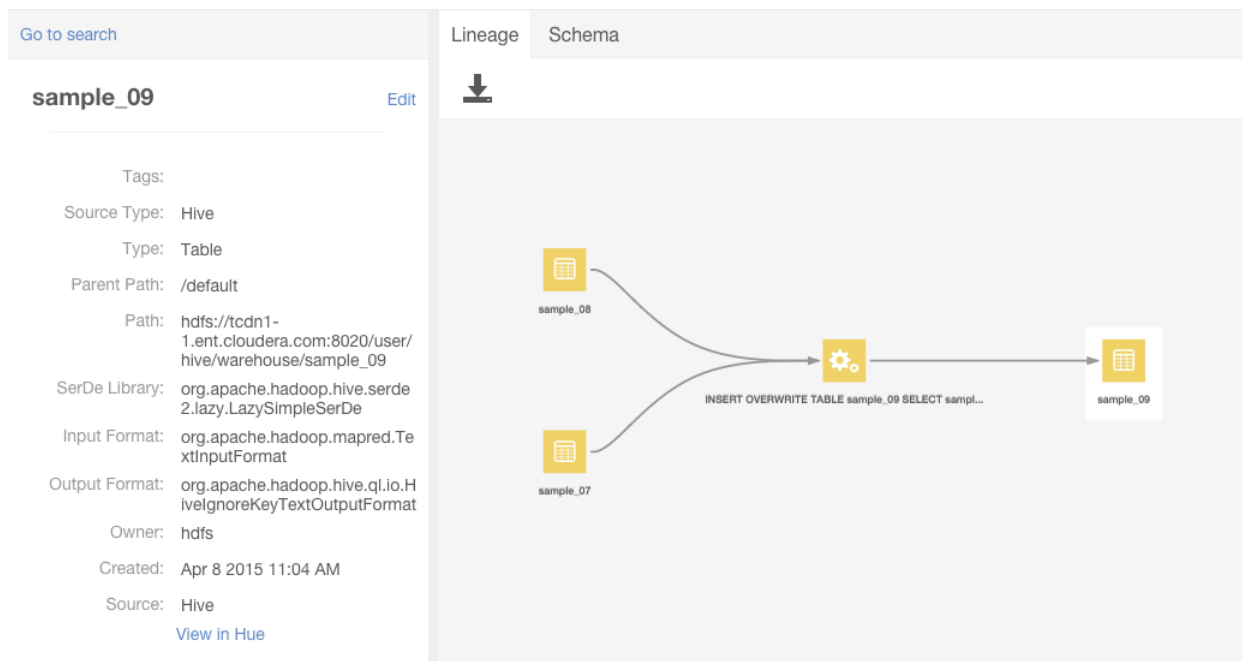
A **template lineage diagram** contains template entities, such as jobs and queries, that can be instantiated, and the input and output entities to which they are related.

To display a template lineage diagram:

1. Perform a metadata [search](#).
2. In the list of results, click an Operation or Query result entry. For example, when you click the `sample_09` result entry:

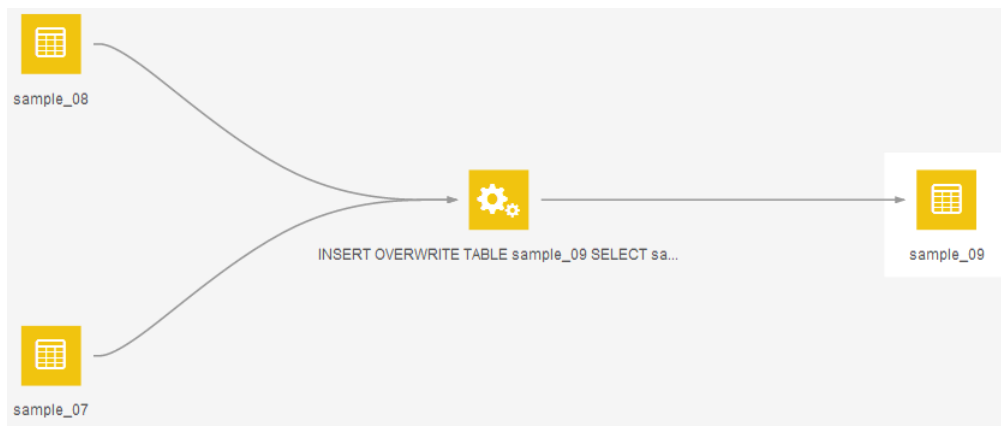
	Hive sample_09		
Type: Table	Parent Path: /default	Path: hdfs://tcdn1-1.ent.cloudera.com:8020/user/hive/warehouse/sample_09	Owner: hdfs
Created: Apr 8 2015 11:04 AM	Source: Hive		

the Search screen is replaced with a page that displays the entity property sheet on the left and lineage diagram on the right:



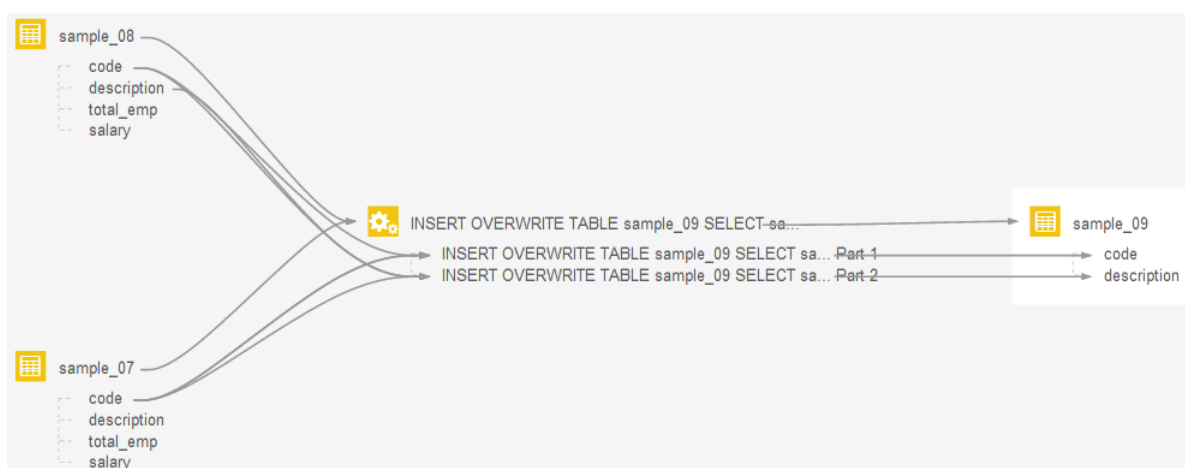
The selected entity `sample_09` appears with a white box as a background.

This example lineage diagram illustrates the relations between a Hive query execution entity and its source and destination tables:



When you click each entity icon, columns and lines connecting the source and destination columns display:

Lineage Diagrams



If you hover over a part, the source and destination columns are highlighted:



Displaying an Instance Lineage Diagram

An **instance lineage diagram** displays instance entities, such as job and query executions, and the input and output entities to which they are related.

To display an instance lineage diagram:

1. Display a template lineage diagram. For example:



2. Click the **Instances** tab, which contains a list of links to instances of the template.
3. Click a link to display an instance lineage diagram. For the preceding template diagram, the job instance `job_1426651548889_0004` replaces the `word count` job template.



Displaying the Template Lineage Diagram for an Instance Lineage Diagram


You can navigate from an instance diagram to its template.

1. Display an instance lineage diagram.
2. Click the value of the **Template** property to navigate to the instance's template.

Downloading a Lineage File

Lineage is externalized in a lineage file in JSON format.

1. Display a template or instance lineage diagram.

2. Click the  icon at the top left of the diagram.

A lineage file named `lineage.json` is downloaded. For example, the lineage file representing `job_1426651548889_0004` from the preceding section is: Tracing through the relations shows that `job_1426651548889_0004`, which has the identity `69b79a8c0c7701f316dd86894b97fe58`, has the `INSTANCE_OF` relation with `word count` and the `DATA_FLOW` relation with `/user/hdfs/input` and `/user/hdfs/out1`.

```
{
  "entities": {
    "01043ab3a019a68f37f3d33efa122f0f": {
      "level": 1,
      "physical": [],
      "logical": [],
      "aliasOf": [],
      "aliases": [],
      "instances": [],
      "children": [],
      "workflows": [],
      "identity": "01043ab3a019a68f37f3d33efa122f0f",
      "originalName": "part-r-00001",
      "sourceId": "a09b0233cc58ff7d601eaa68673a20c6",
      "firstClassParentId": null,
      "parentPath": "/user/hdfs/out1",
      "extractorRunId": "a09b0233cc58ff7d601eaa68673a20c6##1372",
      "name": "part-r-00001",
      "description": null,
      "tags": null,
      "fileSystemPath": "/user/hdfs/out1/part-r-00001",
      "type": "FILE",
      "size": 8,
      "created": "2015-03-27T17:44:20.639Z",
      "lastModified": "2015-03-27T17:44:20.639Z",
      "lastAccessed": "2015-03-27T17:44:16.832Z",
      "permissions": "rw-r--r--",
      "owner": "hdfs",
      "group": "supergroup",
      "blockSize": null,
      "mimeType": "application/octet-stream",
      "replication": null,
      "userEntity": false,
      "deleted": false,
      "sourceType": "HDFS",
      "internalType": "fselement",
    }
  }
}
```

```

    "nameField": "originalName",
    "sourceName": "HDFS-1",
    "hueLink":
"http://tcdn2-1.ent.cloudera.com:8888/filebrowser/view/user/hdfs/out1/part-r-00001",

    "isScript": false,
    "hasUpstream": true,
    "parent": "89612c409b76f7bdf00036df9c3cb717",
    "activeChildren": []
  },
  "72c31f8dbe14a520bd46a747d1382d89": {
    "level": 1,
    "physical": [],
    "logical": [],
    "aliasOf": [],
    "aliases": [],
    "instances": [],
    "children": [
      "f2eca1680ecca38fa514dc191613c7b4",
      "f3929c0b9b2a16490ee57e0a498eee5e"
    ],
    "workflows": [],
    "identity": "72c31f8dbe14a520bd46a747d1382d89",
    "originalName": "input",
    "sourceId": "a09b0233cc58ff7d601eaa68673a20c6",
    "firstClassParentId": null,
    "parentPath": "/user/hdfs",
    "extractorRunId": "a09b0233cc58ff7d601eaa68673a20c6##1370",
    "name": "input",
    "description": null,
    "tags": null,
    "fileSystemPath": "/user/hdfs/input",
    "type": "DIRECTORY",
    "size": null,
    "created": "2015-03-27T17:40:43.665Z",
    "lastModified": "2015-03-27T17:41:06.825Z",
    "lastAccessed": null,
    "permissions": "rwxr-xr-x",
    "owner": "hdfs",
    "group": "supergroup",
    "blockSize": null,
    "mimeType": null,
    "replication": null,
    "userEntity": false,
    "deleted": false,
    "sourceType": "HDFS",
    "internalType": "fselement",
    "nameField": "originalName",
    "sourceName": "HDFS-1",
    "hueLink":
"http://tcdn2-1.ent.cloudera.com:8888/filebrowser/view/user/hdfs/input",
    "isScript": false,
    "hasDownstream": true,
    "column": -1,
    "renderOrdinal": 1,
    "activeChildren": [
      {
        "level": 1,
        "physical": [],
        "logical": [],
        "aliasOf": [],
        "aliases": [],
        "instances": [],
        "children": [],
        "workflows": [],
        "identity": "f3929c0b9b2a16490ee57e0a498eee5e",
        "originalName": "test.txt",
        "sourceId": "a09b0233cc58ff7d601eaa68673a20c6",
        "firstClassParentId": null,
        "parentPath": "/user/hdfs/input",
        "extractorRunId": "a09b0233cc58ff7d601eaa68673a20c6##1370",
        "name": "test.txt",
        "description": null,

```

```

      "tags": null,
      "filePath": "/user/hdfs/input/test.txt",
      "type": "FILE",
      "size": 6,
      "created": "2015-03-27T17:41:06.825Z",
      "lastModified": "2015-03-27T17:41:06.825Z",
      "lastAccessed": "2015-03-27T17:41:06.405Z",
      "permissions": "rw-r--r--",
      "owner": "hdfs",
      "group": "supergroup",
      "blockSize": null,
      "mimeType": "application/octet-stream",
      "replication": null,
      "userEntity": false,
      "deleted": false,
      "sourceType": "HDFS",
      "internalType": "fselement",
      "nameField": "originalName",
      "sourceName": "HDFS-1",
      "hueLink":
"http://tcdn2-1.ent.cloudera.com:8888/filebrowser/view/user/hdfs/input/test.txt",
      "isScript": false,
      "hasDownstream": true,
      "parent": "72c31f8dbel4a520bd46a747d1382d89",
      "activeChildren": []
    },
    {
      "x": -222.4375,
      "y": -52
    },
    {
      "f2ecal680ecca38fa514dc191613c7b4": {
        "level": 1,
        "physical": [],
        "logical": [],
        "aliasOf": [],
        "aliases": [],
        "instances": [],
        "children": [],
        "workflows": [],
        "identity": "f2ecal680ecca38fa514dc191613c7b4",
        "originalName": "test.txt._COPYING_",
        "sourceId": "a09b0233cc58ff7d601eaa68673a20c6",
        "firstClassParentId": null,
        "parentPath": "/user/hdfs/input",
        "extractorRunId": "a09b0233cc58ff7d601eaa68673a20c6##1370",
        "name": "test.txt._COPYING_",
        "description": null,
        "tags": null,
        "filePath": "/user/hdfs/input/test.txt._COPYING_",
        "type": "FILE",
        "size": 6,
        "created": "2015-03-27T17:41:06.405Z",
        "lastModified": "2015-03-27T17:41:06.405Z",
        "lastAccessed": "2015-03-27T17:41:06.405Z",
        "permissions": "rw-r--r--",
        "owner": "hdfs",
        "group": "supergroup",
        "blockSize": null,
        "mimeType": "application/octet-stream",
        "replication": null,
        "userEntity": false,
        "deleted": true,
        "sourceType": "HDFS",
        "internalType": "fselement",
        "nameField": "originalName",
        "sourceName": "HDFS-1",
        "hueLink":
"http://tcdn2-1.ent.cloudera.com:8888/filebrowser/view/user/hdfs/input/test.txt._COPYING_",
        "isScript": false,
        "parent": "72c31f8dbel4a520bd46a747d1382d89",
        "activeChildren": []
      }
    }
  ]
}

```

```

},
"16b093b257033463bab26bba4c707450": {
  "level": 1,
  "physical": [],
  "logical": [],
  "aliasOf": [],
  "aliases": [],
  "instances": [],
  "children": [],
  "workflows": [],
  "identity": "16b093b257033463bab26bba4c707450",
  "originalName": "_temporary",
  "sourceId": "a09b0233cc58ff7d601eaa68673a20c6",
  "firstClassParentId": null,
  "parentPath": "/user/hdfs/out1",
  "extractorRunId": "a09b0233cc58ff7d601eaa68673a20c6##1372",
  "name": "_temporary",
  "description": null,
  "tags": null,
  "fileSystemPath": "/user/hdfs/out1/_temporary",
  "type": "DIRECTORY",
  "size": null,
  "created": "2015-03-27T17:41:32.486Z",
  "lastModified": "2015-03-27T17:41:32.486Z",
  "lastAccessed": null,
  "permissions": "rwxr-xr-x",
  "owner": "hdfs",
  "group": "supergroup",
  "blockSize": null,
  "mimeType": null,
  "replication": null,
  "userEntity": false,
  "deleted": false,
  "sourceType": "HDFS",
  "internalType": "fselement",
  "nameField": "originalName",
  "sourceName": "HDFS-1",
  "hueLink":
"http://tc2dn-1.ent.cloudera.com:8888/filebrowser/view/user/hdfs/out1/_temporary",
  "isScript": false,
  "parent": "89612c409b76f7bdf00036df9c3cb717",
  "activeChildren": []
},
"89612c409b76f7bdf00036df9c3cb717": {
  "level": 1,
  "physical": [],
  "logical": [],
  "aliasOf": [],
  "aliases": [],
  "instances": [],
  "children": [
    "fcd80476d5a968e29e86411b4a67af87",
    "01043ab3a019a68f37f3d33efa122f0f",
    "16b093b257033463bab26bba4c707450",
    "75470b40586cde9e092a01d37798d921"
  ],
  "workflows": [],
  "identity": "89612c409b76f7bdf00036df9c3cb717",
  "originalName": "out1",
  "sourceId": "a09b0233cc58ff7d601eaa68673a20c6",
  "firstClassParentId": null,
  "parentPath": "/user/hdfs",
  "extractorRunId": "a09b0233cc58ff7d601eaa68673a20c6##1372",
  "name": "out1",
  "description": null,
  "tags": null,
  "fileSystemPath": "/user/hdfs/out1",
  "type": "DIRECTORY",
  "size": null,
  "created": "2015-03-27T17:41:32.486Z",
  "lastModified": "2015-03-27T17:44:20.848Z",
  "lastAccessed": null,

```

```

    "permissions": "rwxr-xr-x",
    "owner": "hdfs",
    "group": "supergroup",
    "blockSize": null,
    "mimeType": null,
    "replication": null,
    "userEntity": false,
    "deleted": false,
    "sourceType": "HDFS",
    "internalType": "fselement",
    "nameField": "originalName",
    "sourceName": "HDFS-1",
    "hueLink":
"http://tcdn2-1.ent.cloudera.com:8888/filebrowser/view/user/hdfs/out1",
    "isScript": false,
    "hasUpstream": true,
    "column": 1,
    "renderOrdinal": 2,
    "activeChildren": [
      {
        "level": 1,
        "physical": [],
        "logical": [],
        "aliasOf": [],
        "aliases": [],
        "instances": [],
        "children": [],
        "workflows": [],
        "identity": "fcd80476d5a968e29e86411b4a67af87",
        "originalName": "_SUCCESS",
        "sourceId": "a09b0233cc58ff7d601eaa68673a20c6",
        "firstClassParentId": null,
        "parentPath": "/user/hdfs/out1",
        "extractorRunId": "a09b0233cc58ff7d601eaa68673a20c6##1372",
        "name": "_SUCCESS",
        "description": null,
        "tags": null,
        "filePath": "/user/hdfs/out1/_SUCCESS",
        "type": "FILE",
        "size": 0,
        "created": "2015-03-27T17:44:20.848Z",
        "lastModified": "2015-03-27T17:44:20.848Z",
        "lastAccessed": "2015-03-27T17:44:20.848Z",
        "permissions": "rw-r--r--",
        "owner": "hdfs",
        "group": "supergroup",
        "blockSize": null,
        "mimeType": "application/octet-stream",
        "replication": null,
        "userEntity": false,
        "deleted": false,
        "sourceType": "HDFS",
        "internalType": "fselement",
        "nameField": "originalName",
        "sourceName": "HDFS-1",
        "hueLink":
"http://tcdn2-1.ent.cloudera.com:8888/filebrowser/view/user/hdfs/out1/_SUCCESS",
        "isScript": false,
        "parent": "89612c409b76f7bdf00036df9c3cb717",
        "hasUpstream": true,
        "activeChildren": []
      },
      {
        "level": 1,
        "physical": [],
        "logical": [],
        "aliasOf": [],
        "aliases": [],
        "instances": [],
        "children": [],
        "workflows": [],
        "identity": "75470b40586cde9e092a01d37798d921",
        "originalName": "part-r-00000",

```

```

"sourceId": "a09b0233cc58ff7d601eaa68673a20c6",
"firstClassParentId": null,
"parentPath": "/user/hdfs/out1",
"extractorRunId": "a09b0233cc58ff7d601eaa68673a20c6##1372",
"name": "part-r-00000",
"description": null,
"tags": null,
"filePath": "/user/hdfs/out1/part-r-00000",
"type": "FILE",
"size": 0,
"created": "2015-03-27T17:44:20.576Z",
"lastModified": "2015-03-27T17:44:20.576Z",
"lastAccessed": "2015-03-27T17:44:16.831Z",
"permissions": "rw-r--r--",
"owner": "hdfs",
"group": "supergroup",
"blockSize": null,
"mimeType": "application/octet-stream",
"replication": null,
"userEntity": false,
"deleted": false,
"sourceType": "HDFS",
"internalType": "fselement",
"nameField": "originalName",
"sourceName": "HDFS-1",
"href": "http://tcdn2-1.ent.cloudera.com:8888/filebrowser/view/user/hdfs/out1/part-r-00000",

"isActive": false,
"hasUpstream": true,
"parent": "89612c409b76f7bdf00036df9c3cb717",
"activeChildren": []
},
{
  "level": 1,
  "physical": [],
  "logical": [],
  "aliasOf": [],
  "aliases": [],
  "instances": [],
  "children": [],
  "workflows": [],
  "identity": "01043ab3a019a68f37f3d33efal22f0f",
  "originalName": "part-r-00001",
  "sourceId": "a09b0233cc58ff7d601eaa68673a20c6",
  "firstClassParentId": null,
  "parentPath": "/user/hdfs/out1",
  "extractorRunId": "a09b0233cc58ff7d601eaa68673a20c6##1372",
  "name": "part-r-00001",
  "description": null,
  "tags": null,
  "filePath": "/user/hdfs/out1/part-r-00001",
  "type": "FILE",
  "size": 8,
  "created": "2015-03-27T17:44:20.639Z",
  "lastModified": "2015-03-27T17:44:20.639Z",
  "lastAccessed": "2015-03-27T17:44:16.832Z",
  "permissions": "rw-r--r--",
  "owner": "hdfs",
  "group": "supergroup",
  "blockSize": null,
  "mimeType": "application/octet-stream",
  "replication": null,
  "userEntity": false,
  "deleted": false,
  "sourceType": "HDFS",
  "internalType": "fselement",
  "nameField": "originalName",
  "sourceName": "HDFS-1",
  "href": "http://tcdn2-1.ent.cloudera.com:8888/filebrowser/view/user/hdfs/out1/part-r-00001",

  "isActive": false,

```

```

        "hasUpstream": true,
        "parent": "89612c409b76f7bdf00036df9c3cb717",
        "activeChildren": []
    }
},
"x": 222.4375,
"y": -52
},
"a3ac8013155effa2f96e9de0f177eeb5": {
    "level": 1,
    "physical": [],
    "logical": [],
    "aliasOf": [],
    "aliases": [],
    "instances": [
        "69b79a8c0c7701f316dd86894b97fe58"
    ],
    "children": [],
    "workflows": [],
    "identity": "a3ac8013155effa2f96e9de0f177eeb5",
    "originalName": "word count",
    "sourceId": "a063e69e6c0660353dc378c836837935",
    "firstClassParentId": null,
    "parentPath": null,
    "extractorRunId": "a063e69e6c0660353dc378c836837935##1381",
    "name": "word count",
    "description": null,
    "tags": null,
    "wfIds": null,
    "inputFormat": null,
    "outputFormat": null,
    "outputKey": "org.apache.hadoop.io.Text",
    "outputValue": "org.apache.hadoop.io.IntWritable",
    "mapper": "org.apache.hadoop.examples.WordCount$TokenizerMapper",
    "reducer": "org.apache.hadoop.examples.WordCount$IntSumReducer",
    "sourceType": "YARN",
    "type": "OPERATION",
    "userEntity": false,
    "deleted": null,
    "internalType": "mrjobspec",
    "nameField": "name",
    "sourceName": "YARN-1",
    "isScript": false
},
"69b79a8c0c7701f316dd86894b97fe58": {
    "level": 1,
    "physical": [],
    "logical": [],
    "aliasOf": [],
    "aliases": [],
    "instances": [],
    "children": [],
    "workflows": [],
    "identity": "69b79a8c0c7701f316dd86894b97fe58",
    "originalName": "job_1426651548889_0004",
    "sourceId": "a063e69e6c0660353dc378c836837935",
    "firstClassParentId": null,
    "parentPath": null,
    "extractorRunId": "a063e69e6c0660353dc378c836837935##1381",
    "name": "job_1426651548889_0004",
    "description": null,
    "tags": null,
    "started": "2015-03-27T17:41:20.896Z",
    "ended": "2015-03-27T17:44:21.969Z",
    "principal": "hdfs",
    "inputs": [
        "hdfs://tcdn2-1.ent.cloudera.com:8020/user/hdfs/input"
    ],
    "outputs": [
        "hdfs://tcdn2-1.ent.cloudera.com:8020/user/hdfs/out1"
    ],
    "wfInstId": null,
    "jobID": "job_1426651548889_0004",

```

```

    "sourceType": "YARN",
    "inputRecursive": false,
    "type": "OPERATION_EXECUTION",
    "userEntity": false,
    "deleted": null,
    "internalType": "mrjobinstance",
    "nameField": "originalName",
    "sourceName": "YARN-1",
    "hueLink":
"http://tcdn2-1.ent.cloudera.com:8888/jobbrowser/jobs/application_1426651548889_0004",

    "isScript": false,
    "hasDownstream": true,
    "hasUpstream": true,
    "template": "a3ac8013155effa2f96e9de0f177eeb5",
    "active": true,
    "column": 0,
    "renderOrdinal": 0,
    "activeChildren": [],
    "x": 0,
    "y": -52
  },
  "75470b40586cde9e092a01d37798d921": {
    "level": 1,
    "physical": [],
    "logical": [],
    "aliasOf": [],
    "aliases": [],
    "instances": [],
    "children": [],
    "workflows": [],
    "identity": "75470b40586cde9e092a01d37798d921",
    "originalName": "part-r-00000",
    "sourceId": "a09b0233cc58ff7d601eaa68673a20c6",
    "firstClassParentId": null,
    "parentPath": "/user/hdfs/out1",
    "extractorRunId": "a09b0233cc58ff7d601eaa68673a20c6##1372",
    "name": "part-r-00000",
    "description": null,
    "tags": null,
    "filePath": "/user/hdfs/out1/part-r-00000",
    "type": "FILE",
    "size": 0,
    "created": "2015-03-27T17:44:20.576Z",
    "lastModified": "2015-03-27T17:44:20.576Z",
    "lastAccessed": "2015-03-27T17:44:16.831Z",
    "permissions": "rw-r--r--",
    "owner": "hdfs",
    "group": "supergroup",
    "blockSize": null,
    "mimeType": "application/octet-stream",
    "replication": null,
    "userEntity": false,
    "deleted": false,
    "sourceType": "HDFS",
    "internalType": "fselement",
    "nameField": "originalName",
    "sourceName": "HDFS-1",
    "hueLink":
"http://tcdn2-1.ent.cloudera.com:8888/filebrowser/view/user/hdfs/out1/part-r-00000",

    "isScript": false,
    "hasUpstream": true,
    "parent": "89612c409b76f7bdf00036df9c3cb717",
    "activeChildren": []
  },
  "fcd80476d5a968e29e86411b4a67af87": {
    "level": 1,
    "physical": [],
    "logical": [],
    "aliasOf": [],
    "aliases": [],
    "instances": [],

```



```

"children": [],
"workflows": [],
"identity": "fcd80476d5a968e29e86411b4a67af87",
"originalName": "_SUCCESS",
"sourceId": "a09b0233cc58ff7d601eaa68673a20c6",
"firstClassParentId": null,
"parentPath": "/user/hdfs/out1",
"extractorRunId": "a09b0233cc58ff7d601eaa68673a20c6##1372",
"name": "_SUCCESS",
"description": null,
"tags": null,
"filePath": "/user/hdfs/out1/_SUCCESS",
"type": "FILE",
"size": 0,
"created": "2015-03-27T17:44:20.848Z",
"lastModified": "2015-03-27T17:44:20.848Z",
"lastAccessed": "2015-03-27T17:44:20.848Z",
"permissions": "rw-r--r--",
"owner": "hdfs",
"group": "supergroup",
"blockSize": null,
"mimeType": "application/octet-stream",
"replication": null,
"userEntity": false,
"deleted": false,
"sourceType": "HDFS",
"internalType": "fselement",
"nameField": "originalName",
"sourceName": "HDFS-1",
"hueLink":
"http://tcn2-1.ent.cloudera.com:8888/filebrowser/view/user/hdfs/out1/_SUCCESS",
"isScript": false,
"parent": "89612c409b76f7bdf00036df9c3cb717",
"hasUpstream": true,
"activeChildren": []
},
"f3929c0b9b2a16490ee57e0a498eee5e": {
  "level": 1,
  "physical": [],
  "logical": [],
  "aliasOf": [],
  "aliases": [],
  "instances": [],
  "children": [],
  "workflows": [],
  "identity": "f3929c0b9b2a16490ee57e0a498eee5e",
  "originalName": "test.txt",
  "sourceId": "a09b0233cc58ff7d601eaa68673a20c6",
  "firstClassParentId": null,
  "parentPath": "/user/hdfs/input",
  "extractorRunId": "a09b0233cc58ff7d601eaa68673a20c6##1370",
  "name": "test.txt",
  "description": null,
  "tags": null,
  "filePath": "/user/hdfs/input/test.txt",
  "type": "FILE",
  "size": 6,
  "created": "2015-03-27T17:41:06.825Z",
  "lastModified": "2015-03-27T17:41:06.825Z",
  "lastAccessed": "2015-03-27T17:41:06.405Z",
  "permissions": "rw-r--r--",
  "owner": "hdfs",
  "group": "supergroup",
  "blockSize": null,
  "mimeType": "application/octet-stream",
  "replication": null,
  "userEntity": false,
  "deleted": false,
  "sourceType": "HDFS",
  "internalType": "fselement",
  "nameField": "originalName",
  "sourceName": "HDFS-1",
  "hueLink":

```

```
"http://tcdn2-1.ent.cloudera.com:8888/filebrowser/view/user/hdfs/input/test.txt",
  "isScript": false,
  "hasDownstream": true,
  "parent": "72c31f8dbel4a520bd46a747d1382d89",
  "activeChildren": []
},
"relations": {
  "bd3fe737364968a8fbc1831fc9915dca": {
    "identity": "bd3fe737364968a8fbc1831fc9915dca",
    "type": "DATA_FLOW",
    "propagatorId": "268fc2fbbba566558b83abd0f0fb680a1",
    "extractorRunId": "a09b0233cc58ff7d601eaa68673a20c6##1372",
    "sources": {
      "entityIds": [
        "69b79a8c0c7701f316dd86894b97fe58"
      ]
    },
    "targets": {
      "entityIds": [
        "01043ab3a019a68f37f3d33efa122f0f",
        "75470b40586cde9e092a01d37798d921"
      ]
    },
    "propagatable": false,
    "unlinked": false,
    "userSpecified": false
  },
  "33535116782b0baff207851f9e637cf2": {
    "identity": "33535116782b0baff207851f9e637cf2",
    "type": "DATA_FLOW",
    "propagatorId": "217788cald4de53a4071cf026299744f",
    "extractorRunId": "a09b0233cc58ff7d601eaa68673a20c6##1372",
    "sources": {
      "entityIds": [
        "f3929c0b9b2a16490ee57e0a498eee5e"
      ]
    },
    "targets": {
      "entityIds": [
        "69b79a8c0c7701f316dd86894b97fe58"
      ]
    },
    "propagatable": false,
    "unlinked": false,
    "userSpecified": false
  },
  "646e2547f1f1371e99259069f3bbd4db": {
    "identity": "646e2547f1f1371e99259069f3bbd4db",
    "type": "PARENT_CHILD",
    "propagatorId": null,
    "extractorRunId": "a09b0233cc58ff7d601eaa68673a20c6##1370",
    "children": {
      "entityIds": [
        "f2eca1680ecca38fa514dc191613c7b4"
      ]
    },
    "parent": {
      "entityId": "72c31f8dbel4a520bd46a747d1382d89"
    },
    "propagatable": false,
    "unlinked": false,
    "userSpecified": false
  },
  "da3e6b9ccbc9e39de59e85ea6d89fdd7": {
    "identity": "da3e6b9ccbc9e39de59e85ea6d89fdd7",
    "type": "PARENT_CHILD",
    "propagatorId": null,
    "extractorRunId": "a09b0233cc58ff7d601eaa68673a20c6##1372",
    "children": {
      "entityIds": [
        "fcd80476d5a968e29e86411b4a67af87"
      ]
    }
  }
}
```

```

    ],
    },
    "parent": {
      "entityId": "89612c409b76f7bdf00036df9c3cb717"
    },
    },
    "propagatable": false,
    "unlinked": false,
    "userSpecified": false
  },
  "15816a23933df14590026425fc0e8d85": {
    "identity": "15816a23933df14590026425fc0e8d85",
    "type": "PARENT_CHILD",
    "propagatorId": null,
    "extractorRunId": "a09b0233cc58ff7d601eaa68673a20c6##1372",
    "children": {
      "entityIds": [
        "01043ab3a019a68f37f3d33efa122f0f"
      ]
    },
    },
    "parent": {
      "entityId": "89612c409b76f7bdf00036df9c3cb717"
    },
    },
    "propagatable": false,
    "unlinked": false,
    "userSpecified": false
  },
  "f8f31d2c2638c22f17600a32631c5639": {
    "identity": "f8f31d2c2638c22f17600a32631c5639",
    "type": "PARENT_CHILD",
    "propagatorId": null,
    "extractorRunId": "a09b0233cc58ff7d601eaa68673a20c6##1370",
    "children": {
      "entityIds": [
        "16b093b257033463bab26bba4c707450"
      ]
    },
    },
    "parent": {
      "entityId": "89612c409b76f7bdf00036df9c3cb717"
    },
    },
    "propagatable": false,
    "unlinked": false,
    "userSpecified": false
  },
  "3dcd15d16d13786480052adbac5e7f7f": {
    "identity": "3dcd15d16d13786480052adbac5e7f7f",
    "type": "DATA_FLOW",
    "propagatorId": "268fc2fbba566558b83abd0f0fb680a1",
    "extractorRunId": "a09b0233cc58ff7d601eaa68673a20c6##1372",
    "sources": {
      "entityIds": [
        "69b79a8c0c7701f316dd86894b97fe58"
      ]
    },
    },
    "targets": {
      "entityIds": [
        "75470b40586cde9e092a01d37798d921",
        "01043ab3a019a68f37f3d33efa122f0f",
        "fcd80476d5a968e29e86411b4a67af87"
      ]
    },
    },
    "propagatable": false,
    "unlinked": false,
    "userSpecified": false
  },
  "268fc2fbba566558b83abd0f0fb680a1": {
    "identity": "268fc2fbba566558b83abd0f0fb680a1",
    "type": "DATA_FLOW",
    "propagatorId": null,
    "extractorRunId": "a09b0233cc58ff7d601eaa68673a20c6##1372",
    "sources": {
      "entityIds": [
        "69b79a8c0c7701f316dd86894b97fe58"
      ]
    }
  ]
}

```



```

    "template": {
      "entityId": "a3ac8013155effa2f96e9de0f177eeb5"
    },
    "propagatable": false,
    "unlinked": false,
    "userSpecified": false
  },
  "38358d90c0c9675c76626148732a63a4": {
    "identity": "38358d90c0c9675c76626148732a63a4",
    "type": "DATA_FLOW",
    "propagatorId": "268fc2fbbba566558b83abd0f0fb680a1",
    "extractorRunId": "a09b0233cc58ff7d601eaa68673a20c6##1372",
    "sources": {
      "entityIds": [
        "69b79a8c0c7701f316dd86894b97fe58"
      ]
    },
    "targets": {
      "entityIds": [
        "01043ab3a019a68f37f3d33efa122f0f",
        "75470b40586cde9e092a01d37798d921"
      ]
    },
    "propagatable": false,
    "unlinked": false,
    "userSpecified": false
  }
}

```

Impala Lineage Properties

Required Role: **Configurator** **Cluster Administrator** **Full Administrator**

The following property controls whether the Cloudera Manager Agent collects lineage entries:

- **Enable Impala Lineage Collection** - Indicates whether Impala lineage logs should be collected.

The following properties apply to the Impala lineage log file:

- **Enable Impala Lineage Generation** - Indicates whether Impala lineage logs should be generated.
- **Impala Daemon Lineage Log Directory** - The directory in which lineage log files are written.

- **Note:** If the value of this property is changed, and service is restarted, then the Cloudera Manager Agent will start monitoring the new log directory. In this case it is possible that not all events are published from the old directory. To avoid loss of lineage, when this property is changed, perform the following steps:

1. Stop the service.
2. Copy lineage log files and (for Impala only) the `impalad_lineage_wal` file from the old log directory to the new log directory. This needs to be done on all the hosts where Impala daemons are running.
3. Start the service.

- **Impala Daemon Maximum Lineage Log File Size** - The maximum size in number of queries of the lineage log file before a new file is created.

Managing Impala Lineage

Impala lineage is enabled by default. To control whether the Impala Daemon role logs to the lineage log and whether the Cloudera Manager Agent collects the lineage entries:

Lineage Diagrams

1. Go to the Impala service.
2. Click the **Configuration** tab.
3. Select **Scope > Impala Daemon**.
4. Select **Category > Logs**.
5. Select the **Enable Impala Lineage Generation** checkbox.
6. Select **Scope > All**.
7. Select **Category > Cloudera Navigator**.
8. Select the **Enable Lineage Collection** checkbox.
9. Click **Save Changes** to commit the changes.
10. Restart the service.

If you deselect *either* checkbox, Impala lineage is disabled.

Configuring Impala Daemon Lineage Logs

Required Role: **Configurator** **Cluster Administrator** **Full Administrator**

1. Go to the Impala service.
2. Click the **Configuration** tab.
3. Select **Scope > Impala Daemon**.
4. Type `lineage` in the Search box.
5. Edit the lineage log properties.
6. Click **Save Changes** to commit the changes.
7. Restart the service.

Schema

Required Role: **Lineage Viewer** **Metadata Administrator** **Full Administrator**

A table schema contains information about the names and types of the columns of a table.

HDFS schema contains information about the names and types of the fields in an HDFS Avro or Parquet file.

Displaying Hive, Sqoop, and Impala Table Schema

1. Perform a metadata [search](#) for an entities of source type **Hive** and type **Table**.
2. In the list of results, click a result entry.
3. Click the **Schema** tab. The table schema displays.

Lineage Schema

code	string
description	string

Displaying Pig Table Schema

1. Perform a metadata [search](#) for entities of source type **Pig**. Do one of the following:
 - In the list of results, click a result entry of type **Table**.
 - 1. In the list of results, click a result entry of type **Operation_Execution**.
 - 2. Click the **Tables** tab. A list of links to tables involved in the operation displays.

3. Click a table link.
2. Click the **Schema** tab. The table schema displays.

Displaying HDFS Dataset Schema

If you ingest a [Kite dataset](#) into HDFS you can view the schema of the dataset. The schema is represented as an entity of type Dataset and is implemented as an HDFS directory.

For Avro datasets, primitive types such as null, string, int, and so on, are not separate entities. For example, if you have a record type with a field A that's a record type and a field B that's a string, the subfields of A become entities themselves, but B has no children. Another example would be if you had a union of null, string, map, array, and record types; the union has 3 children - the map, array, and record subtypes.

To display an HDFS dataset schema:

1. Perform a metadata [search](#) for entities of type **Dataset**.
2. Click a result entry.
3. Click the **Schema** tab. The dataset schema displays.

Stocks Schema

1. Use the Stocks Avro schema file:

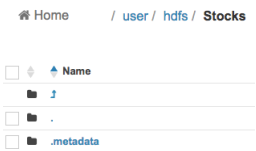
```
{
  "type" : "record",
  "name" : "Stocks",
  "namespace" : "com.example.stocks",
  "doc" : "Schema generated by Kite",
  "fields" : [ {
    "name" : "Symbol",
    "type" : [ "null", "string" ],
    "doc" : "Type inferred from 'AAIT'"
  }, {
    "name" : "Date",
    "type" : [ "null", "string" ],
    "doc" : "Type inferred from '28-Oct-2014'"
  }, {
    "name" : "Open",
    "type" : [ "null", "double" ],
    "doc" : "Type inferred from '33.1'"
  }, {
    "name" : "High",
    "type" : [ "null", "double" ],
    "doc" : "Type inferred from '33.13'"
  }, {
    "name" : "Low",
    "type" : [ "null", "double" ],
    "doc" : "Type inferred from '33.1'"
  }, {
    "name" : "Close",
    "type" : [ "null", "double" ],
    "doc" : "Type inferred from '33.13'"
  }, {
    "name" : "Volume",
    "type" : [ "null", "long" ],
    "doc" : "Type inferred from '400'"
  } ]
}
```

and the `kite-dataset` command to create a Stocks dataset:

```
kite-dataset create dataset:hdfs:/user/hdfs/Stocks -s Stocks.avsc
```

The following directory is created in HDFS:

Lineage Diagrams



2. In search results, the Stocks dataset appears as follows:



- 3. Click the **Stocks** link.
- 4. Click the **Schema** tab. The schema displays:

Lineage	Schema
Volume	union(null,long)
Date	union(null,string)
Symbol	union(null,string)
High	union(null,double)
Open	union(null,double)
Close	union(null,double)
Low	union(null,double)

Each subfield of the Stocks record is an entity of type field.

Volume

Type inferred from '400'

Tags:

Source Type: HDFS

Dataset: Stocks

Type: Field

Data Type: UNION

Parent Path: /Stocks

Source: HDFS-1