

cloudera[®]

Cloudera Data Management

Important Notice

© 2010-2021 Cloudera, Inc. All rights reserved.

Cloudera, the Cloudera logo, 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. If this documentation includes code, including but not limited to, code examples, Cloudera makes this available to you under the terms of the Apache License, Version 2.0, including any required notices. A copy of the Apache License Version 2.0, including any notices, is included herein. A copy of the Apache License Version 2.0 can also be found here: <https://opensource.org/licenses/Apache-2.0>

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.

**395 Page Mill Road
Palo Alto, CA 94306
info@cloudera.com
US: 1-888-789-1488
Intl: 1-650-362-0488
www.cloudera.com**

Release Information

Version: Cloudera Navigator Navigator 2.5.x
Date: February 3, 2021

Table of Contents

About Cloudera Data Management.....	5
Cloudera Navigator Metadata Architecture.....	6
Metadata Extraction and Indexing.....	7
Metadata Search Syntax and Properties.....	8
<i>Search Syntax.....</i>	<i>8</i>
<i>Search Properties.....</i>	<i>9</i>
Accessing Metadata.....	13
<i>Navigator Metadata UI.....</i>	<i>13</i>
<i>Navigator Metadata API.....</i>	<i>16</i>
Modifying Custom Metadata.....	16
Extending Metadata Types.....	22
Performing Actions on Entities.....	22
Cloudera Navigator Auditing Architecture.....	24
Service Access Audit Log Properties.....	24
Service Access Auditing Properties.....	26
<i>Auditing Impala Operations.....</i>	<i>29</i>
Cloudera Navigator Auditing.....	31
<i>Viewing Audit Events.....</i>	<i>31</i>
<i>Filtering Audit Events.....</i>	<i>31</i>
<i>Service Audit Event Fields.....</i>	<i>32</i>
Cloudera Navigator Audit Event Reports.....	35
<i>Creating Audit Event Reports.....</i>	<i>35</i>
<i>Editing Audit Event Reports.....</i>	<i>35</i>
<i>Downloading Audit Event Reports.....</i>	<i>36</i>
Downloading HDFS Directory Access Permission Reports.....	37
Cloudera Navigator Analytics.....	38
Metadata Policies.....	41
Metadata Policy Expressions.....	43
Cloudera Navigator Lineage Diagrams.....	51

Displaying a Template Lineage Diagram.....54

Displaying an Instance Lineage Diagram.....56

Displaying the Template Lineage Diagram for an Instance Lineage Diagram.....56

Impala Lineage Properties.....56

Schema.....57

Displaying Hive, Impala, and Sqoop Table Schema.....57

Displaying Pig Table Schema.....57

Displaying HDFS Dataset Schema.....58

Appendix: Apache License, Version 2.0.....61

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 system for the Hadoop platform. Cloudera Navigator features address the needs of a broad range of stakeholders interacting with data at scale:

- Compliance groups must track and protect access to sensitive data. Their concerns focus on being prepared for an audit, tracking who is accessing what data and what are they doing with it, and ensuring that sensitive data is governed and protected.
- Hadoop administrators and DBAs are responsible for boosting user productivity and cluster performance. These users are concerned with how is data being used and how it can be optimized for future workloads.
- Data stewards and curators manage and organize data assets at Hadoop scale. Their tasks involve managing the data lifecycle efficiently, from ingest to purge.
- Data scientists and BI users need to find the data that matters most. They want to be able explore data, trust what they find, and be able to visualize relationships between data sets.

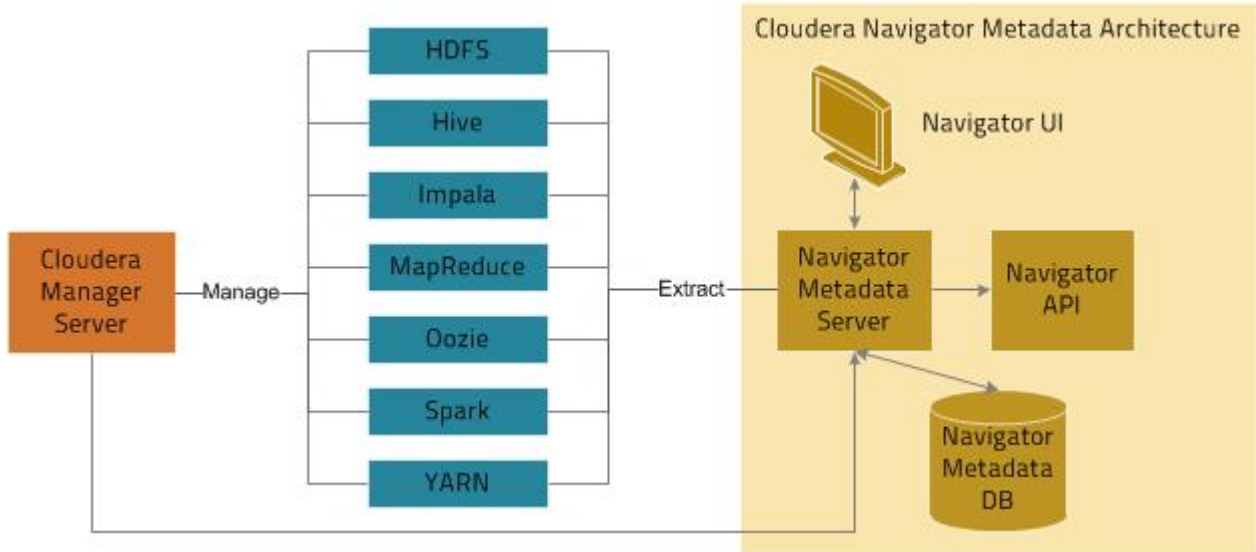
To address the requirements of all these users, Cloudera Navigator provides the following categories of functionality:

- **Data Management** - Data management provides visibility into and control over the data residing in Hadoop datastores and the computations performed on that data. The Cloudera Navigator features that address the data management needs of Hadoop administrators, data stewards, and data scientists are:
 - 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. Cloudera Navigator [auditing features](#) add secured, real-time audit components to key data and access frameworks. Cloudera Navigator allows compliance groups to configure, collect, and view audit events, and to understand who accessed what data and how.
 - Searching metadata and visualizing lineage - Cloudera Navigator [metadata management features](#) allow DBAs, data stewards, business analysts, and data scientists to define, search for, amend the properties of, and tag data entities and view relationships between datasets.
 - Policies - Cloudera Navigator [policy features](#) enable data stewards to specify automated actions based on data access or on a schedule to add metadata, create alerts, and move or purge data.
 - Analytics - Cloudera Navigator [analytics features](#) enable Hadoop administrators to examine data usage patterns and create policies based on those patterns.
- **Data Encryption** - 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 groups to manage encryption:
 - [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.
 - [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).

Cloudera Navigator data management and data encryption components can be installed independently.

Cloudera Navigator Metadata Architecture

Cloudera Navigator metadata features provide 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 CDH services from the Cloudera Manager Server
- Extracts metadata for the entities managed by those services at periodic intervals
- Manages and applies metadata extraction policies during metadata extraction
- Indexes and stores entity metadata
- Manages authorization data for Navigator users
- Manages audit report metadata
- Generates metadata and audit analytics
- Implements the Navigator UI and API

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

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:

The screenshot shows the Cloudera Navigator Metadata Architecture interface for a file named 'sample_07.csv'. The interface includes a top navigation bar with a file icon, the file name, a menu icon, and two tabs: 'Details' (selected) and 'Lineage'. Below the navigation bar, there are two main panels:

- Technical Metadata:**
 - Source Type: HDFS
 - Type: File
 - Path: /user/hive/warehouse/sample_...
 - Owner: sample
 - Group: supergroup
 - Permissions: rwxrwxrwt
 - Size: 44.98KiB
 - Block Size: 128.00MiB
 - Replication Count: 3
 - Last Accessed: Sep 15 2015 7:41 AM
 - Last Modified: Sep 15 2015 7:41 AM
 - Created: Sep 15 2015 7:41 AM
 - Source: HDFS-2
- Custom Metadata:**
 - Description: Occupational categories: salary and number of employees.
 - Tags: occupations, salaries
 - year: 2015

For all entities, as shown in the Details tab, 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 and after* entities are extracted.

In addition, for Hive entities, there are extended attributes.

Metadata Extraction and Indexing

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. See [Enabling Hive Metadata Extraction in a Secure Cluster](#).
- **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 YARN logs. (Unsupported and disabled by default. To enable, see [Enabling Spark Metadata Extraction](#).)
- **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.



Important: Tables created by Impala queries and Sqoop jobs are represented as Hive entities.

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 relationships between entities.

You can [search](#) entity metadata using the Navigator UI and API. Relationship metadata is implicitly visible in [lineage diagrams](#) and explicitly available by downloading the lineage using the [Cloudera Navigator Data Management API](#).

Metadata Search Syntax and Properties

In Cloudera Navigator, metadata 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](#) and the following three types of key-value pairs using the given syntax:

- **Technical metadata key-value pairs** - `key:value`, where
 - `key` is one of the properties listed in [Search Properties](#) on page 9.
 - `value` is a single value or range of values specified as `[value1 TO value2]`. In a value, `*` is a wildcard. In property values you must escape special characters `:`, `-`, `/`, and `*` with the backslash character `\` or enclose the property value in quotes. For example, `filePath:/tmp/hbase\ -staging`.

These key-value pairs are read-only and cannot be modified.

- **Custom metadata key-value pairs** - `up_key:value`, where
 - `key` is a user-defined property defined on an entity after extraction.
 - `value` is a single value or range of values specified as `[value1 TO value2]`. In a value, `*` is a wildcard. In property values you must escape special characters `:`, `-`, `/`, and `*` with the backslash character `\` or enclose the property value in quotes. For example, `filePath:/tmp/hbase\ -staging`.

Custom metadata key-value pairs can be modified.

- **Hive extended attribute key-value pairs** - `tp_key:value`, where
 - `key` is an extended attribute defined on a Hive entity before extraction. The syntax of the attribute is specific to Hive.
 - `value` is a single value supported by the entity type.

These key-value pairs are read-only and cannot be modified.

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`
- Technical key-value - In Hive you can specify table properties like this:

```
ALTER TABLE table_name SET TBLPROPERTIES ('key1'='value1');
```

To query for this property, specify `tp_key1:value1`.



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 specifying a property value: `type`, `filePath`, `inputs`, `jobId`, `mapper`, `mimeType`, `name`, `originalName`, `outputs`, `owner`, `principal`, `reducer`, and `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>ngrammedText</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>ngrammedText</code>	The type of an operation: <ul style="list-style-type: none"> • Pig - SCRIPT • Sqoop - Table Export, Query Import
<code>originalName</code>	<code>ngrammedText</code>	The name of the entity when it was extracted.
<code>originalDescription</code>	<code>text</code>	The description of the entity when it was extracted.
<code>owner</code>	<code>caseInsensitiveText</code>	The owner of the entity.
<code>principal</code>	<code>caseInsensitiveText</code>	For entities with type <code>OPERATION_EXECUTION</code> , the initiator of the entity.
<code>properties</code>	<code>string</code>	A set of key-value pairs that describe the entity.
<code>tags</code>	<code>ngrammedText</code>	A set of tags that describe the entity.
<code>type</code>	<code>tokenizedCaseInsensitiveText</code>	The type of the entity. The available types depend on the entity's source type: <ul style="list-style-type: none"> • <code>hdfs</code> - DIRECTORY, FILE, DATASET, FIELD

Name	Type	Description
		<ul style="list-style-type: none"> hive - DATABASE, TABLE, FIELD, OPERATION, OPERATION_EXECUTION, SUB_OPERATION, PARTITION, RESOURCE, VIEW impala - OPERATION, OPERATION_EXECUTION, SUB_OPERATION mapreduce - OPERATION, OPERATION_EXECUTION oozie - OPERATION, OPERATION_EXECUTION pig - OPERATION, OPERATION_EXECUTION spark - OPERATION, OPERATION_EXECUTION sqoop - OPERATION, OPERATION_EXECUTION, SUB_OPERATION yarn - OPERATION, OPERATION_EXECUTION, SUB_OPERATION
userEntity	Boolean	Indicates whether an entity was added using the Cloudera Navigator SDK .
Query		
queryText	string	The text of a Hive, Impala, or Sqoop query.
Source		
clusterName	string	The name of the cluster in which the source is managed.
sourceId	string	The ID of the source type.
sourceType	caseInsensitiveText	The source type of the entity: hdfs, hive, impala, mapreduce, oozie, pig, spark, sqoop, or yarn.
sourceUrl	string	The URL of web application for a resource.
Timestamps		
<p>The available timestamp fields vary by the source type:</p> <ul style="list-style-type: none"> hdfs - created, lastAccessed, lastModified hive - created, lastModified impala, mapreduce, pig, spark, 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
blockSize	long	The block size of an HDFS file.
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.
filePath	path	The path to the entity.
mimeType	ngrammedText	The MIME type of an HDFS file.

Name	Type	Description
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.
replication	int	The number of copies of HDFS file blocks.
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.

Dataset Properties

Name	Type	Description
compressionType	tokenizedCaseInsensitiveText	The type of compression of a dataset file.
dataType	string	The data type: record.
datasetType	tokenizedCaseInsensitiveText	The type of the dataset: Kite.
fileFormat	tokenizedCaseInsensitiveText	The format of a dataset file: Avro or Parquet.
fullDataType	string	The full data type: record.
partitionType	string	The type of the partition.
schemaName	string	The name of the dataset schema.
schemaNamespace	string	The namespace of the dataset schema.

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		

Name	Type	Description
inputs	string	The name of the entity input to an operation execution. For entities of resource type <code>mapreduce</code> , <code>yarn</code> , and <code>spark</code> , it is usually a directory. For entities of resource type <code>hive</code> , it is usually a table.
outputs	string	The name of the entity output from an operation execution. For entities of resource type <code>mapreduce</code> , <code>yarn</code> , and <code>spark</code> , it is usually a directory. For entities of resource type <code>hive</code> , it is usually a table.
engineType	string	The type of the engine used for an operation: MR or Spark.

Hive Properties

Name	Type	Description
Field		
dataType	ngramedText	The type of data stored in a field (column).
Table		
compressed	Boolean	Indicates whether a 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.
technical_properties	string	Hive extended attributes.
clusteredByColNames	string	The column names that identify how table content is divided into buckets.
sortByColNames	string	The column names that identify how table content is sorted within a bucket.

Oozie Properties

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

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.

Name	Type	Description
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:

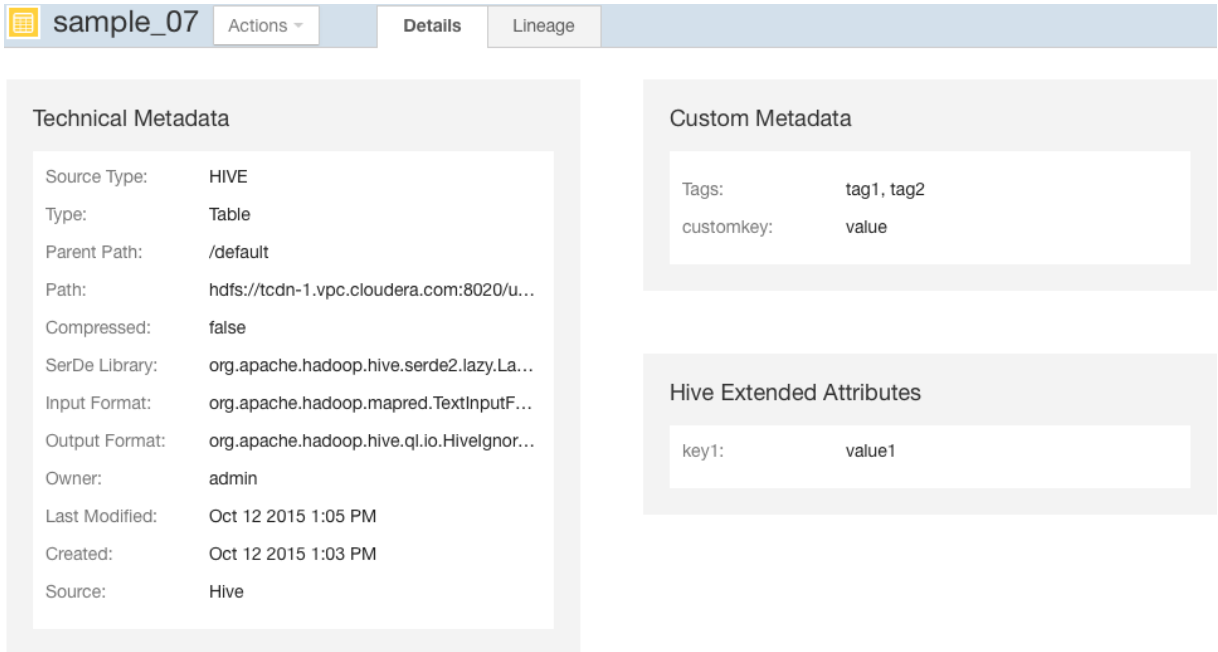
- Source type
- Name - the name is a link to a page that displays the entity details and [lineage diagram](#)
- Properties
- If Hue is running, a link at the far right labeled **View in Hue** that opens the Hue browser for the entity:
 - HDFS directories and files - File Browser
 - Hive database and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

For example:


Hive sample_07
 Type: Table Parent Path: /default Path: hdfs://tcdn-1.vpc.cloudera.com:8020/user/hive/warehouse/sample_07 Owner: admin [View in Hue](#)
 Last Modified: Oct 12 2015 1:05 PM Created: Oct 12 2015 1:03 PM Source: Hive

Displaying Entity Details

1. Perform a search.
2. In the search results, click an entity name link. For example, if you click the Hive table `sample_07` link in the search result displayed in the preceding section, you could see the following details:



The screenshot shows the 'sample_07' entity details page. It features a navigation bar with tabs for 'sample_07', 'Actions', 'Details', and 'Lineage'. The 'Details' tab is selected, displaying two main sections: 'Technical Metadata' and 'Custom Metadata'.

Technical Metadata:

Source Type:	HIVE
Type:	Table
Parent Path:	/default
Path:	hdfs://tcdn-1.vpc.cloudera.com:8020/u...
Compressed:	false
SerDe Library:	org.apache.hadoop.hive.serde2.lazy.La...
Input Format:	org.apache.hadoop.mapred.TextInputF...
Output Format:	org.apache.hadoop.hive ql.io.HiveIgnor...
Owner:	admin
Last Modified:	Oct 12 2015 1:05 PM
Created:	Oct 12 2015 1:03 PM
Source:	Hive

Custom Metadata:

Tags:	tag1, tag2
customkey:	value

Hive Extended Attributes:

key1:	value1
-------	--------

In addition to the technical metadata, this Hive table has custom metadata consisting of tags `tag1` and `tag2` and a key-value pair `customkey-value`, and an extended attribute key-value pair `key1-value1`.

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.

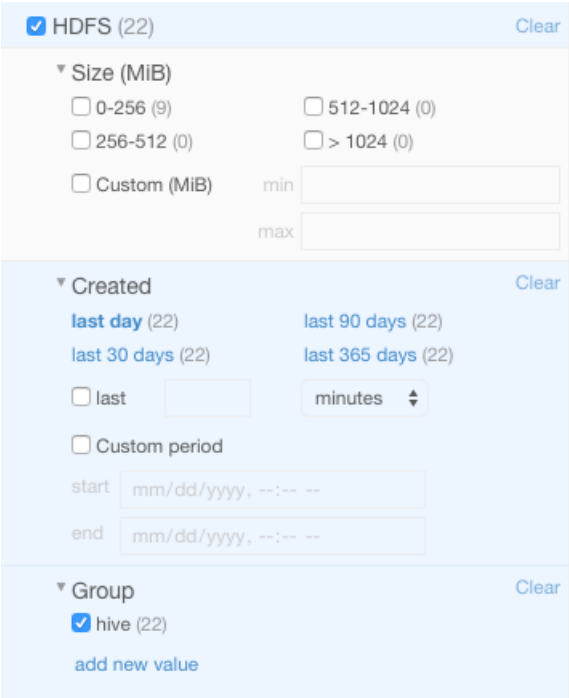
Specify a property value as follows:

- **Boolean** - Select the option 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.

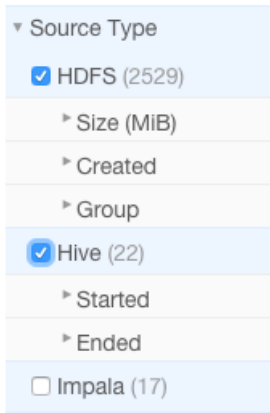
- **Timestamp** - Timestamps are used for started, ended, created, last accessed, and last modified properties. The server stores the timestamp in UTC and the UI displays the timestamp converted to the local timezone. Select one of the timestamp options:
 - A **Last XXX day(s)** link.
 - The **Last** 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 by clicking fields or by using the right and left arrow keys.

To remove filter values, click the  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 number in parentheses (facet count) after a property value is the number of extracted entities that have that property value:



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**, **Actions > Save Search_name**, or **Actions > Save As...**
3. If you have not previously saved the search, specify a name and click **Save**.

Reusing a Saved Search

1. Select **Actions > View saved searches....** A label with the saved search name is added under the Search box.
2. Click the saved search name. The breadcrumbs and full query if displayed are updated to reflect the saved search and the search results are refreshed immediately.

Navigator Metadata API

The Navigator API allows you to search entity metadata using a REST API. For information about the API, see [Cloudera Navigator Data Management 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, HDFS 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 entity Details tab displays.
- 3. To the left of the Details tab, click **Actions > Edit Metadata....** The Edit Custom Metadata dialog box drops down.
- 4. Edit any of the fields. Press **Enter** or **Tab** to create new tag entries. In the following screenshot, a description, the tags `occupations` and `salaries`, and property year with value 2015 have been added to the file `sample_07.csv`:

Edit Custom Metadata [Close]

Name: sample_07.csv

Description: Occupational categories: salary and number of employees.

Tags: occupations x salaries x

Key-Value Pairs: year : 2015 [Minus] [Plus]

[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 Custom Metadata pane:

The screenshot shows the Cloudera Navigator interface for a file named `sample_07.csv`. The interface has a top navigation bar with a file icon, the filename, a menu icon, and two tabs: `Details` (selected) and `Lineage`. Below the navigation bar, there are two main panels: `Technical Metadata` and `Custom Metadata`.

Technical Metadata:

Source Type:	HDFS
Type:	File
Path:	/user/hive/warehouse/sample_...
Owner:	sample
Group:	supergroup
Permissions:	rwxrwxrwt
Size:	44.98KIB
Block Size:	128.00MIB
Replication Count:	3
Last Accessed:	Sep 15 2015 7:41 AM
Last Modified:	Sep 15 2015 7:41 AM
Created:	Sep 15 2015 7:41 AM
Source:	HDFS-2

Custom Metadata:

Description:	Occupational categories: salary and number of employees.
Tags:	occupations, salaries
year:	2015

Modifying MapReduce Custom Metadata

You can associate custom metadata with arbitrary configuration parameters to MapReduce jobs and job executions. The specific configuration parameters to be extracted by Navigator can be specified statically or dynamically.

To specify configuration parameters statically for all MapReduce jobs and job executions, do the following:

1. Do one of the following:
 - Select **Clusters > Cloudera Management Service > Cloudera Management Service**.
 - On the Status tab of the **Home > Status** tab, 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 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. The property `nav.tags` can point to multiple property names that serve as tags, but each of those property names can only specify a *single* tag.
7. Click **Save Changes** to commit the changes.
8. Click the **Instances** tab.
9. Restart the role.
10. In the MapReduce job configuration, set the value of the property names you specified in step 6.

To specify configuration parameters dynamically:

1. Specify one or more of the following properties in a job configuration:
 - `job properties (type: OPERATION)`

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

The properties `nav.job.tags` and `nav.jobexec.tags` can point to multiple property names that serve as tags, but each of those property names can only specify a *single* tag.

2. In the MapReduce job configuration, set the value of the property names you specified in step 1.

Example: Setting Properties Dynamically

Add the tags `onetag` and `twotag` to a job:

1. Dynamically add the `job_tag1` and `job_tag2` properties:

```
conf.set("nav.job.tags", "job_tag1, job_tag2");
```

2. Set the `job_tag1` property to `onetag`:

```
conf.set("job_tag1", "onetag");
```

3. Set the `job_tag2` property to `twotag`:

```
conf.set("job_tag2", "twotag");
```

Add the tag `atag` to a job execution:

1. Dynamically add the `job_tag` property:

```
conf.set("nav.jobexec.tags", "job_exec_tag");
```

2. Set the `job_exec_tag` property to `atag`:

```
conf.set("job_exec_tag", "atag");
```

Add the user-defined property `foo` with the value `bar`:

1. Dynamically add the user-defined property `bar`:

```
conf.set("nav.job.user_defined_properties", "bar");
```

2. Set the value of the user-defined property `foo` to `bar`:

```
conf.set("foo", "bar")
```

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 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/v8/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/v8/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/v8/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/v8/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/v8/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/v8/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/v8/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"}}'
```

Extending Metadata Types

In addition to the metadata and lineage features provided by the Cloudera Navigator data management component, you can also define and add new types of metadata using the Navigator SDK. For information, see the [Navigator SDK documentation](#).

Performing Actions on Entities

Moving and Moving an HDFS Entity to Trash

Required Role:

Policy Administrator

Full Administrator

You can move an HDFS entity to another location and move an entity to [HDFS trash](#). In order to perform such actions you must be a member of a user group that has the appropriate access to HDFS files.

You can also schedule a move or move to trash in a [policy](#).

1. [Start and log into the Cloudera Navigator data management component UI](#).
2. Run a [search](#) in the Navigator UI.
3. Click an HDFS entity link returned in the search. The entity Details tab displays.
4. To the left of the Details tab, select **Actions > Move...** or **Actions > Move to Trash...**
5. For a move, specify the target path.
6. Click **Run Action**. When you delete a file, after a short delay the file displays a Deleted badge.

Viewing Command Action Status

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

2. In the top right, select **username** **Actions**. The Command Actions status page displays with a list of actions performed and the policy that caused the action if applicable.
3. If an action failed, a View Log button displays, which you can click to view the error message associated with the failure.

Viewing an Entity in Hue

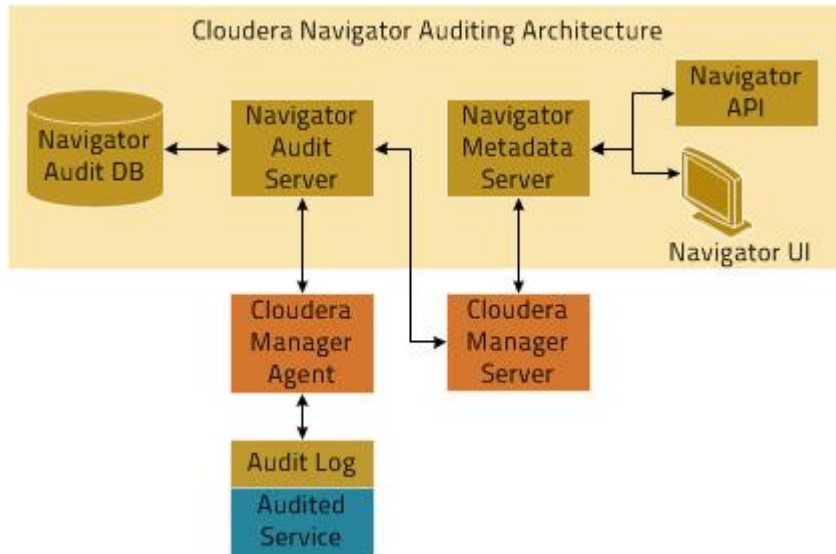
If you are running [Hue](#) you can view certain types of entities in the following Hue applications:

- HDFS directories and files - File Browser
 - Hive database and tables - Metastore Manager
 - MapReduce, YARN, Pig - Job Browser
1. [Start and log into the Cloudera Navigator data management component UI](#).
 2. Run a [search](#) in the Navigator UI.
 3. Do one of the following:
 - Search results
 1. Click the **View in Hue** link in a search result entry.
 - Entity details
 1. Click an entity link returned in the search. The entity Details tab displays.
 2. To the left of the Details tab, select **Actions** > **View in Hue**.

The entity displays in the supported Hue application.

Cloudera Navigator Auditing Architecture

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 service access events are added to the HDFS, HBase, and Hive (that is, the HiveServer2 and Beeswax servers) services. The plug-ins write the events to an audit log on the local filesystem. The existence of the plug-ins places [requirements](#) on these services when Cloudera Navigator is upgraded. Cloudera Impala, Sentry, and the Cloudera Navigator Metadata Server collect and filter access events and write them to an audit log file.

The Cloudera Manager Agent monitors the audit log files and sends the 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 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 event offset in the audit log that it has successfully transmitted, so on any crash/restart it picks up the 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 an event to the 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

Service Access Audit Log Properties

A service or role **Enable Audit Collection** property controls whether the Cloudera Manager Agent tracks a service or role's audit log file.

The following properties apply to an 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.

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.

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, Hue, 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 > Status** tab, 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 > Status** tab, 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 Access Auditing Properties

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

- **Enable Audit Collection** - See [Service Access Audit Log Properties](#) on page 24.
- **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. 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, Solr, Solr, Navigator Metadata Server**- 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 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 Hue or the 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:

The screenshot shows the rule editor for 'HDFS-1 (Service-Wide)'. It includes a 'View as JSON' link. There are three rules listed:

- Rule 1: Action: discard. Fields: username: (?:(cloudera-scm|hbase|mapred|hive|dr.who|solr|impala|spark)(?:/.+)?
- Rule 2: Action: discard. Fields: username: (?:(hdfs)(?:/.+)?, operation: (?:listStatus|listCachePools|listCacheDirectives|getFileinfo)
- Rule 3: Action: discard. Fields: src: /tmp(?:/.+)?

At the bottom, the 'Default action' is set to '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),",
    "is' 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

Minimum Required Role: [Navigator Administrator](#) (also provided by **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 28, [Enabling Solr Auditing](#) on page 28.

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

Minimum Required Role: [Configurator](#) (also provided by **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

Minimum Required Role: [Configurator](#) (also provided by **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 [Service Access Audit Log Properties](#) on page 24.
9. Click **Save Changes** to commit the changes.
10. Restart the service.

Enabling and Disabling Navigator Metadata Server Auditing

Minimum Required Role: [Navigator Administrator](#) (also provided by **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 > Status** tab, 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.

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 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 [Cloudera Navigator Auditing](#) on page 31.

Cloudera Navigator Auditing

[Required Role:](#)

Auditing Viewer

Full Administrator

An **audit event** is an event that describes an action that has been taken for a cluster, host, license, parcel, role, service or user.

Cloudera Manager records cluster, host, license, parcel, role, and service **lifecycle events** (activate, create, delete, deploy, download, install, start, stop, update, upgrade, and so on), user **security-related events** (add and delete user, login failed and succeeded), and provides an audit UI and API to view, filter, and export such events. For information on Cloudera Manager auditing features, see [Lifecycle and Security Auditing](#).

The Cloudera Navigator Audit Server records **service access events** and the Cloudera Navigator Metadata Server provides an audit UI and API to view, filter, and export both service access events and the lifecycle and security events retrieved from Cloudera Manager.




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

You filter audit events by specifying a time range or adding one or more filters containing an audit event field, operator, and value.

Specifying a Time Range

1. Click the date-time range at the top right of the Audits tab.
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 by clicking fields or by 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 **Select Property...** 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. 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.

Service Audit Event Fields

The following fields can appear in a service audit event:

Display Name	Field	Description
Additional Info	additional_info	JSON text that contains more details about an 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 the 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 of a Navigator Metadata Server entity. The ID can be retrieved using the Navigator Metadata Server API.
Event Time	timestamp	Date and time an action was performed. The Navigator Audit Server stores the timestamp in the timezone of the Navigator Audit Server. The Navigator UI displays the timestamp converted to the local timezone. Exported audit events contain the stored timestamp.
Family	family	HBase column family.

Display Name	Field	Description
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 an 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 • HiveServer2 - EXPLAIN, LOAD, EXPORT, IMPORT, CREATEDATABASE, DROPDATABASE, SWITCHDATABASE, DROPTABLE, DESC TABLE, DESC FUNCTION, MSCK, ALTABLE_ADDCOLS, ALTABLE_REPLACECOLS, ALTABLE_RENAMECOL, ALTABLE_RENAMEPART, ALTABLE_RENAME, 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 • Hue - USER_LOGIN, USER_LOGOUT, EDIT_USER, ADD_LDAP_USERS, ADD_LDAP_GROUPS, SYNC_LDAP_USERS_GROUPS, EDIT_GROUP, EDIT_PERMISSION, CREATE_USER, CREATE_GROUP, DELETE_USER, DELETE_GROUP

Display Name	Field	Description
		<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, DML (Data Manipulation Language statements) 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. For Hue, the user or group that was added, edited, or deleted.
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)
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, createAuditReport, deleteAuditReport, updateAuditReport authorization - searchGroup, deleteGroup, fetchGroup, fetchRoles, updateRoles metadata - updateMetadata, fetchMetadata, fetchAllMetadata policy - fetchAllPolicies, createPolicy, deletePolicy, updatePolicy, fetchPolicySchedule, updatePolicySchedule, deletePolicySchedule

Display Name	Field	Description
		<ul style="list-style-type: none"> 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.

**Note:**

Cloudera Navigator does not capture audit events for queries that are run on HiveServer1/Hive CLI. If you want to use Cloudera Navigator to capture auditing for Hive operations, upgrade to HiveServer2 if you have not done so already.

Cloudera Navigator Audit Event Reports

[Required Role:](#)

Auditing Viewer

Full Administrator

An **audit report** is a collection of [audit events](#) that satisfy a set of filters. Audit report metadata is recorded by the [Cloudera Navigator Metadata Server](#).

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 in CSV and JSON formats. 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 [Service Audit Event Fields](#) on page 32.

In addition to downloading audit events, you can configure the Navigator Audit Server to publish audit events to a Kafka topic or syslog. See [Publishing Audit Events](#).

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 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/v8/audits/?query=service%3Dhive&startTime=1431025200000&endTime=1431032400000\
&limit=5&offset=0&format=JSON&attachment=false -X GET -u username:password
```

The `startTime` and `endTime` parameters are required 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" : ""
  }
}
}
]

```

Downloading HDFS Directory Access Permission Reports

Minimum Required Role: [Cluster Administrator](#) (also provided by **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.

Cloudera Navigator Analytics

Cloudera Navigator allows you to view metadata and audit analytics for HDFS entities. On the analytics pages, you can view which HDFS entities satisfy the following property values:

- Metadata - the number of files by creation and access times, size, block size, and replication count. After selecting a property value range for one of these properties, you can filter the matching files by directory, owner, and tag.
- Audit
 - Activity tab - by directory which files have been accessed using the `open` operation and how many times they have been accessed. Activity analytics are based on summarized data computed once a day and will not match the number of events viewed in the [Audits](#) tab at all times.
 - Top Users tab - the top- n commands and the top- n users and top n commands those users performed during various time windows (1 min–1 day), where n is 1, 5, 10, 20, 50, or 100.

Viewing Metadata Analytics

[Required Role:](#)

Lineage Viewer

Policy Administrator

Full Administrator

1. [Start and log into the Cloudera Navigator data management component UI.](#)
2. Click the **Analytics** tab. The Metadata analytics tab displays.
3. Choose an HDFS service instance from the *service_name* Analytics drop-down list.
4. The Metadata tab displays a set of bar graphs that list the number of files that satisfy groups of values for last access time, created time, size, block size, and replication count.
 - To display the files at the right, click a bar. This draws a blue selection outline around the bar and selects the property checkbox.
 - To select more than one value, grab a bar edge and brush a range of values.
 - To change a range, click a bar, drag to a different range of values, and drop.
 - To reduce a range, grab a bar edge and contract the range.
 - To deselect a property, deselect the checkbox. The previous selection is indicated with a gray outline.
 - When you select a previously selected property, the previous selection is reused. For example, if you had previously selected one and three for replication count, and you reselect the replication count checkbox, the values one and three are reselected.
 - To clear all selections, present and previous, click **Clear all selections**.
5. In the listing on the right, select an option to display the number of files by directory, owner, or tag. In the listing:
 - Filter the selections by typing strings in the search box and pressing **Enter** or **Return**.
 - Add categories (directory, owner, or tag) to a search query and display the Search tab by doing one of the following:
 - Clicking a directory, owner, or tag name link.
 - Selecting **Actions > Show in search**. To further refine the query, select one or more checkboxes, and select **Actions > Show selection in search**.
 - **Policy Administrator**

Add categories to the search query of a new policy and display the Policies tab by selecting **Actions > Create**

a policy. To further refine the query, select one or more checkboxes, and select **Actions > Create a policy from selection.**

Viewing Audit Analytics

Required Role:

Auditing Viewer

Full Administrator

1. [Start and log into the Cloudera Navigator data management component UI.](#)
2. Click the **Analytics** tab. If the logged-in user has a role that permits access to metadata analytics, the Metadata analytics tab displays.
3. Choose an HDFS service instance from the *service_name* Analytics drop-down list.
4. If not already displayed, click the **Audits** tab. The Activity tab displays a bar graph that lists the number of files that have been read the number of times listed in the x-axis.
 - To display at the right the directories containing the files that have been read, click an activity bar. This draws a blue selection outline around the bar and selects the Activity checkbox.
 - To select more than one value, grab a bar edge and brush a range of values.
 - To change a range, click a bar, drag to a different range of values, and drop.
 - To reduce a range, grab a bar edge and contract the range.
 - To deselect Activity, deselect the checkbox. The previous selection is indicated with a gray outline.
 - When you select Activity and the graph had a previous selection, the previous selection is reused. For example, if you had previously selected values spanning six through nine for the number of times files have been read, and you select the checkbox, six through nine will be reselected.
5. In the directory listing on the right:
 - Filter the directories by typing directory strings in the search box and pressing **Enter** or **Return**.
 - **Metadata Administrator**

Add selected directories to a search query and display the Search tab by doing one of the following:

 - Clicking a directory name link.
 - Selecting one or more directory checkboxes and selecting **Actions > Show selection in search.**
 - **Policy Administrator**

Full Administrator

Add selected directories to the search query of a new policy and display the Policies tab by selecting one or more directory checkboxes and selecting **Actions > Create a policy from selection.**

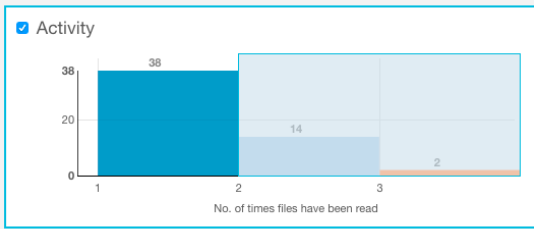
For example, the following screenshot shows files that have been accessed two and three times, match the string `sample`, and are in the `/usr/hive/warehouse/sample*` directories. Each directory has one file that has been accessed.

HDFS-1 Analytics ▾

Metadata

Audit

Activity **Top Users**



HDFS Files

clusterName = Cluster 1 sourceType = hdfs type = file

sample

Actions ▾

Directory	Number of Files
<input checked="" type="checkbox"/> /user/hive/warehouse/sample_08	1
<input type="checkbox"/> /user/hive/warehouse/sample_07	1

Display 15 Entries 1 to 2 << < > >>

Metadata Policies

A **metadata policy** defines a set of actions performed by the Cloudera Navigator Metadata Server on a class of entities. The following actions are supported:

- Adding custom metadata such as tags and properties.
- Performing an command such as moving an HDFS entity to another location or moving an HDFS entity to [HDFS trash](#).

If a policy creator configures a command action to move a directory and the creator doesn't have access to the directory, the action will fail. Similarly, if a policy creator doesn't have access to a file within the directory, the action will also fail. To ensure that command actions don't fail, policies containing command actions should be created by data stewards, who are members of a user group that has the appropriate access to HDFS files.

- 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"}
```

In order to send a message to a JMS message queue you must configure the [JMS server](#) properties.

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

A policy is run as the user who created the policy in the home directory of the user who created the policy. If you want to change who a policy runs as, log into Navigator as the new user you want to run the policy as, clone the policy as the new user, then delete or disable the old policy.

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.

Viewing a Policy

1. [Start and log into the Cloudera Navigator data management component UI](#).
2. Click the **Policies** tab.
3. In a policy row, click a policy name link or select **Actions > View**.

Enabling and Disabling Policies

1. [Start and log into the Cloudera Navigator data management component UI](#).
2. Click the **Policies** tab.
3. In a policy row, click a policy name link or select **Actions > Enable** or **Actions > Disable**.

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 New Policy.
Search results page	<ol style="list-style-type: none"> 1. Select Actions > Create a policy.

3. In the Status field, check the **Enable** checkbox.
4. Enter a name for the policy.
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 you choose to use policy expressions in properties that support expressions, specify required imports in the **Import Statements** field. See [Metadata Policy Expression Examples](#) on page 44.
8. Choose the schedule for applying the policy:
 - On Change - when the entities matching the search string change.
 - Immediate - when the policy is created.
 - Once - at the time specified in the Start Time field.
 - Recurring - at recurring times specified by the Start and End Time fields at the interval specified in the Interval field.

For the Once and Recurring fields, specify dates and times as follows:

- 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 by clicking fields or by using the right and left arrow keys.

9. Follow the appropriate procedure for the actions performed by the policy:

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.
Configure Command Actions	<ol style="list-style-type: none"> 1. Select Add Action > Move to Trash and/or Add Action > Move. For a move, specify the location to move the entity to in the Target Path field. If you specify multiple actions, they are run in the order in which they are specified. <p>Command actions are supported only for HDFS entities. If you configure a command action for unsupported entities, a runtime error will be logged when the policy runs.</p> <p>See Viewing Command Action Status on page 22.</p>

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**.

Copying 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 a policy row, select **Actions > Copy** or **Actions > Edit**.
4. Edit the policy name, search query, or policy actions.
5. 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 a policy row, select **Actions > Delete** and **OK** to confirm.

Metadata Policy Expressions

A **metadata policy expression** allows you to specify certain [metadata extraction policy](#) properties using Java expressions instead of string literals. The supported properties are: entity name and description, key-value pairs, and JMS notification message.

You must declare classes accessed in the expression in the policy's **Import Statements** field. A metadata policy expression must evaluate to a string.

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

Including 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 44.

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()`.

Metadata Policy 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;
```

- Add the entity's creation date to the entity 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;
```

- Set the key-value pair: `retain_util-seven years from today's local time`:

```
new DateTime().plusYears(7).toLocalDateTime().toString("MMM dd yyyy", Locale.US)
```

Import statements:

```
import org.joda.time.DateTime; import java.util.Locale;
```

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  
}
```

Cloudera Navigator Lineage Diagrams

[Required Role:](#)

Lineage Viewer

Metadata Administrator

Full Administrator

Cloudera Navigator provides an automatic collection and easy visualization of upstream and downstream data lineage to verify reliability. For each data source, it shows, down to the column-level within that data source, what the precise upstream data sources were, the transforms performed to produce it, and the impact that data has on downstream artifacts.















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



There are two types of lineage diagrams:

- **Template** - represents a diagram that is a model for other diagram
- **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 Metadata Extraction.)	
<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"> •  •  		
---	--	--	--

Important: Tables created by Impala queries and Sqoop jobs are represented as Hive entities.

In the following circumstances the entity type icon will appear as



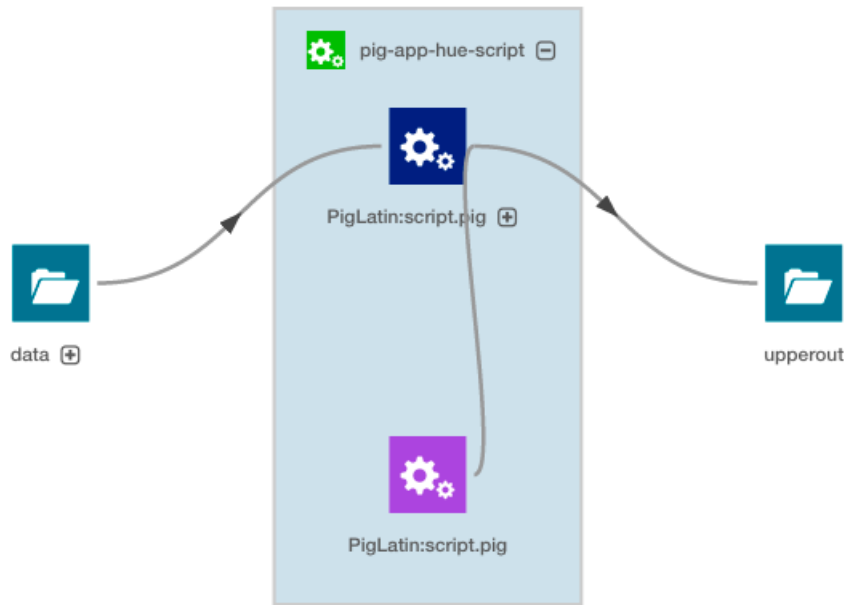
- The entity has not yet been extracted. In this case



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 7.

- A Hive entity has been deleted from the system before it could be extracted.

Parent entities are represented by a blue 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 the source file in the `data` folder and destination folder `upperout`:



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.

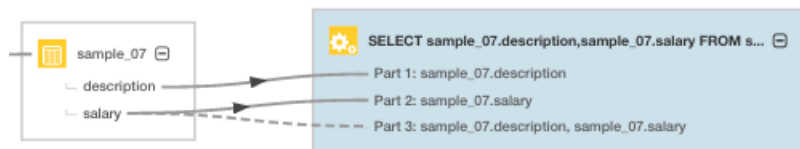
Lineage diagrams display the following line types:

- A solid line (—————) represents a "data flow" relationship, indicating that the columns will appear (possibly transformed) in the output. For example, a solid line will appear between the columns used in a `select` clause.
- A dashed line (- - - - -) represents a "control flow" relationship, indicating that the columns determine which rows will flow to the output. For example, a dashed line will appear between the columns used in an `insert` or `select` clause and the `where` clause of a Hive query.

The following query:

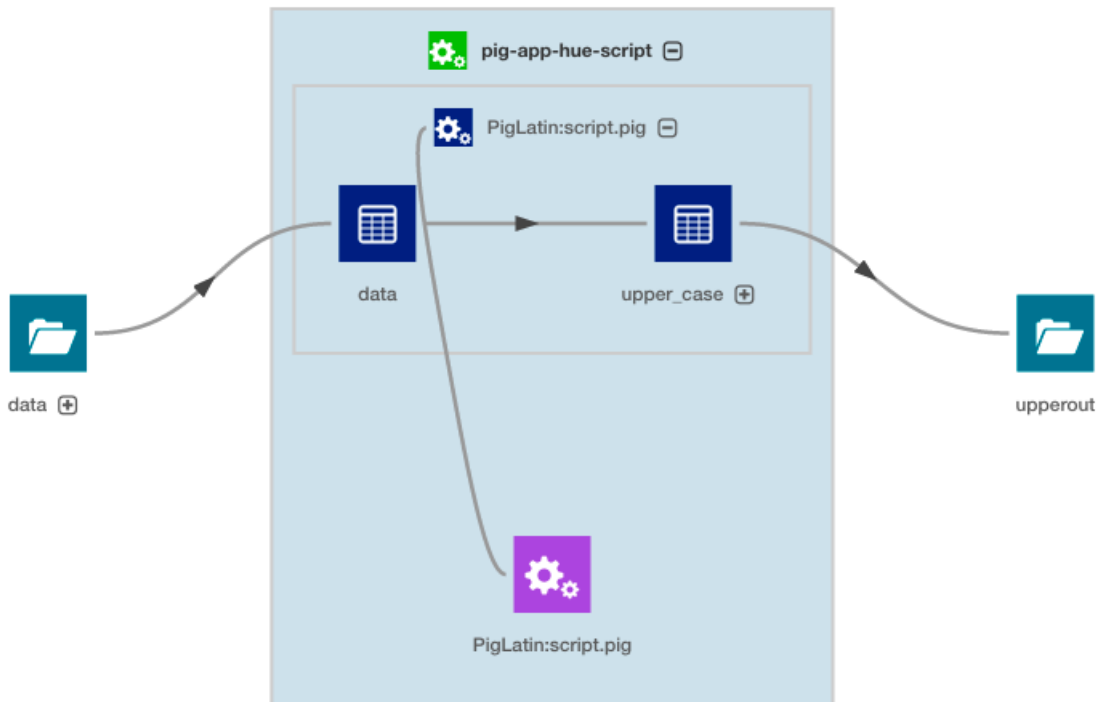
```
SELECT sample_07.description,sample_07.salary FROM sample_07
WHERE ( sample_07.salary > 100000)
ORDER BY sample_07.salary DESC LIMIT 1000
```


has solid lines between the columns in the `select` clause and a dashed line between the columns in the `where` clause:



Manipulating Lineage Diagrams

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



- To improve the layout of a lineage diagram you can drag and drop entities (in this case `data` and `upperout`) located outside a parent box.
- You can use the mouse scroll wheel or the  control to zoom the lineage diagram in and out.
- You can move an the entire 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.
3. Click the **Lineage** tab. 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 Details page that displays the entity property sheet:

sample_09 Details Lineage View in Hue

Technical Metadata

Source Type: HIVE
Type: Table
Parent Path: /default
File System Path: hdfs://tcdn1-1.vpc.cloudera.co...
Compressed: false
SerDe Library: org.apache.hadoop.hive.serde2...
Input Format: org.apache.hadoop.mapred.Tex...
Output Format: org.apache.hadoop.hive.qi.io.Hi...
Owner: admin
Last Accessed: Dec 31 1969 4:00 PM
Created: Aug 27 2015 12:07 PM
Source: HIVE-1

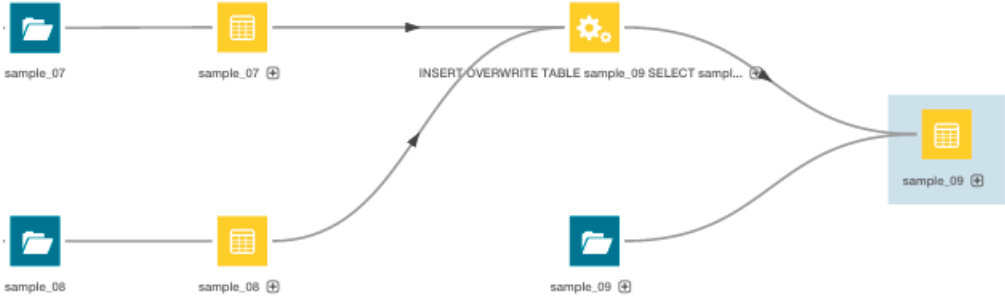
Custom Metadata Edit

No metadata available

Schema

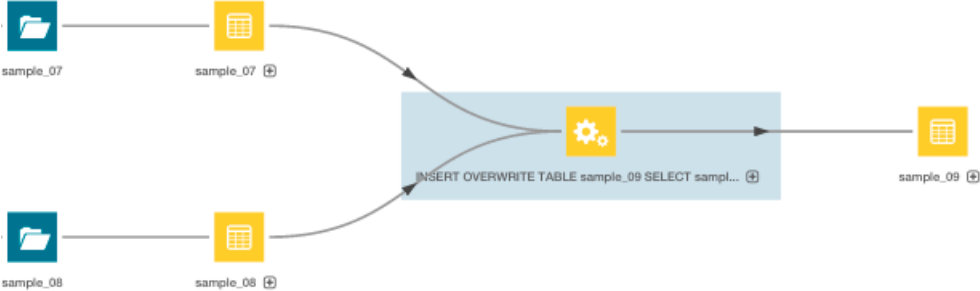
- code string
- description string

After you click the **Lineage** tab, the lineage diagram displays:

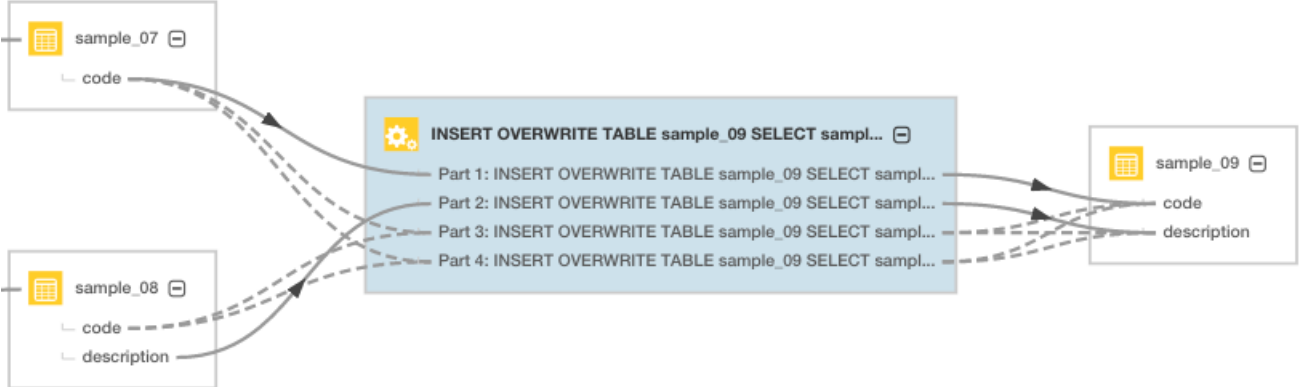


The selected entity `sample_09` appears with a white box as a background.

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



When you click the **+** icon, columns and lines connecting the source and destination columns display:



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:

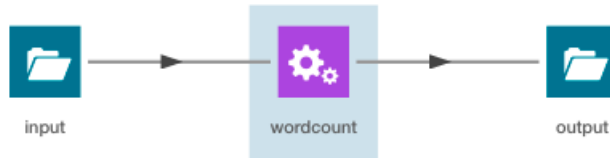
1. Perform a search and click a link of type Operation.
2. Click a link in the **Instances** box.
3. Click the **Lineage** tab:



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 **Details** tab.
3. Click the value of the **Template** property to go to the instance's template.



Impala Lineage Properties

Minimum Required Role: [Configurator](#) (also provided by **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:

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

Minimum Required Role: [Configurator](#) (also provided by **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.

A Kite dataset ingested into HDFS contains information about the names and types of the fields in an HDFS Avro or Parquet file used to create the dataset.

Displaying Hive, Impala, and Sqoop Table Schema

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

Displaying Pig Table Schema

1. Perform a metadata [search](#) for entities of source type **Pig**.
2. In the list of results, click a result entry of type **Table**. The table schema displays in the Details tab.

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. The dataset schema displays in the Details tab.

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:

Home / user / hdfs / Stocks

<input type="checkbox"/>		Name
<input type="checkbox"/>		↑
<input type="checkbox"/>	<input type="checkbox"/>	.
<input type="checkbox"/>	<input type="checkbox"/>	.metadata

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



3. Click the **Stocks** link. The schema displays at the right of the Details tab.

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

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

Technical Metadata	
Source Type:	HDFS
Table:	Stocks
Type:	Field
Data Type:	UNION
Parent Path:	/Stocks
Source:	HDFS-1

4. Then use the `kite-dataset csv-import` command to import structured data:

```
kite-dataset csv-import ./Stocks.csv dataset:hdfs:/user/hdfs/Stocks --no-header
```

where `Stocks.csv` is:

```
AAPL,20150206,120.02,120.25,118.45,118.93,43372000
AAPL,20150205,120.02,120.23,119.25,119.94,42246200
GOOG,20150304,571.87,577.11,568.01,573.37,1713800
GOOG,20150303,570.45,575.39,566.52,573.64,1694300
GOOG,20150302,560.53,572.15,558.75,571.34,2118400
GOOG,20150209,528,532,526.02,527.83,1264300
GOOG,20150206,527.64,537.2,526.41,531,1744600
GOOG,20150205,523.79,528.5,522.09,527.58,1844700
FB,20150304,79.3,81.15,78.85,80.9,28014500
```

```
FB,20150303,79.61,79.7,78.52,79.6,18567300
FB,20150302,79,79.86,78.52,79.75,21604400
FB,20150227,80.68,81.23,78.62,78.97,30635700
FB,20150226,79.88,81.37,79.72,80.41,31111900
TWTR,20150211,46.27,47.78,46.11,47.5,24747000
TWTR,20150210,47.35,47.39,45.57,46.26,32287800
TWTR,20150209,46.73,47.69,46.5,47.32,36177900
TWTR,20150206,46.12,48.5,45.8,48.01,102669800
TWTR,20150205,42.04,42.47,40.91,41.26,61997300
MSFT,20150304,43.01,43.21,42.88,43.06,25705800
MSFT,20150303,43.56,43.83,43.09,43.28,31748600
MSFT,20150302,43.67,44.19,43.55,43.88,31924000
MSFT,20150227,44.13,44.2,43.66,43.85,33807700
MSFT,20150226,43.99,44.23,43.89,44.06,28957300
ORCL,20150304,43.2,43.66,42.82,43.61,14663900
ORCL,20150303,43.83,43.88,43.17,43.38,10058700
ORCL,20150302,43.81,44.04,43.48,44.03,11091000
ORCL,20150227,43.77,44.11,43.68,43.82,9549500
ORCL,20150226,43.8,44.15,43.71,43.89,8519300
ORCL,20150225,43.83,44.09,43.38,43.73,11785400
```

Appendix: Apache License, Version 2.0

SPDX short identifier: Apache-2.0

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims

licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution.

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

1. You must give any other recipients of the Work or Derivative Works a copy of this License; and
2. You must cause any modified files to carry prominent notices stating that You changed the files; and
3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
4. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions.

Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks.

This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty.

Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability.

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

```
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
```