

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 2.10.x
Date: February 3, 2021

Table of Contents

About Cloudera Data Management.....	5
--	----------

Cloudera Navigator Metadata Architecture.....	7
--	----------

Data Stewardship Dashboard.....	8
<i>Dashboard.....</i>	<i>9</i>
<i>Data Explorer.....</i>	<i>13</i>
Cloudera Navigator and S3.....	13
<i>S3 Metadata Entities in Navigator.....</i>	<i>14</i>
<i>Enabling Cloudera Navigator Access to Amazon S3.....</i>	<i>14</i>
<i>Eventual Consistency.....</i>	<i>16</i>
<i>S3 Event Notification.....</i>	<i>16</i>
<i>Setting API Limits.....</i>	<i>16</i>
<i>Limitations of Navigator for S3.....</i>	<i>16</i>
<i>S3 Data Extraction for Navigator.....</i>	<i>17</i>
Metadata Extraction and Indexing.....	22
Metadata Search Syntax and Properties.....	23
<i>Search Syntax.....</i>	<i>23</i>
<i>Search Properties.....</i>	<i>24</i>
Accessing Metadata.....	29
<i>Navigator Metadata UI.....</i>	<i>29</i>
<i>Navigator Metadata API.....</i>	<i>34</i>
Defining Managed Metadata.....	34
<i>Creating Custom Properties Using the Metadata UI.....</i>	<i>34</i>
<i>Managing Custom Properties Using the Metadata UI.....</i>	<i>36</i>
<i>Navigator Built-in Classes.....</i>	<i>38</i>
<i>Defining Metadata with the Navigator API and Navigator SDK.....</i>	<i>39</i>
Adding and Editing Metadata.....	39
Performing Actions on Entities.....	46
Cloudera Navigator Provenance Use Case.....	47

Cloudera Navigator Auditing Architecture.....	51
--	-----------

Cloudera Navigator Auditing.....	51
<i>Viewing Audit Events.....</i>	<i>52</i>
<i>Filtering Audit Events.....</i>	<i>52</i>
<i>Verifying that Auditing is Running.....</i>	<i>53</i>
<i>Service Audit Event Fields.....</i>	<i>54</i>

Cloudera Navigator Audit Event Reports.....57
Creating Audit Event Reports.....57
Editing Audit Event Reports.....57
Downloading Audit Event Reports.....58
Downloading HDFS Directory Access Permission Reports.....59
Cloudera Navigator Auditing Use Cases.....59

Cloudera Navigator Analytics.....64

Metadata Policies.....66

Metadata Policy Expressions.....68

Cloudera Navigator Lineage Diagrams.....76

Manipulating Lineage Diagrams.....79
Displaying a Template Lineage Diagram.....83
Displaying an Instance Lineage Diagram.....85
Displaying the Template Lineage Diagram for an Instance Lineage Diagram.....85
Schema.....85
Displaying Hive, Impala, and Sqoop Table Schema.....85
Displaying Pig Table Schema.....86
Displaying HDFS Dataset Schema.....86

Troubleshooting Navigator Data Management.....89

Navigator Audit Server.....89
Navigator Metadata Server.....90

Appendix: Apache License, Version 2.0.....93

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 enables you to work effectively with data at scale and helps various stakeholders answer the following questions:

- Compliance groups
 - Who accessed the data, and what did they do with it?
 - Are we prepared for an audit?
 - Is our sensitive data protected?
- Hadoop administrators and DBAs
 - How can we boost productivity and cluster performance?
 - How is data being used?
 - How can data be optimized for future workloads?
- Data stewards and curators
 - How can data assets be managed and organized?
 - What is the lifecycle of the data?
 - How can I get "at-a-glance" information about overall cluster activity for a specific time period?
- Data scientists and Business Intelligence users
 - Where is the most important data?
 - Is this data trustworthy?
 - What is the relationship between data sets?

Cloudera Navigator provides the following components to help you answer these questions and meet data-management and security requirements.

- **Data Management** - Provides visibility into and control over the data in Hadoop datastores, and the computations performed on that data. Hadoop administrators, data stewards, and data scientists can use Cloudera Navigator to:
 - Get an "at-a-glance" overview of cluster activity - Use the [Data Stewardship dashboard](#) to get a high-level view of the state of the data and data usage on the cluster. The dashboard displays information about table, file, and database activity, table and file creation and modification trends, operations, and other information captured by the [Navigator Metadata Server](#).
 - Audit data access and verify access privileges - The goal of auditing is to capture a complete and immutable record of all activity within a system. Cloudera Navigator [auditing](#) adds secure, real-time audit components to key data and access frameworks. Compliance groups can use Cloudera Navigator to configure, collect, and view audit events that show who accessed data, and how.
 - Search metadata and visualize lineage - Cloudera Navigator [metadata management](#) allows 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.

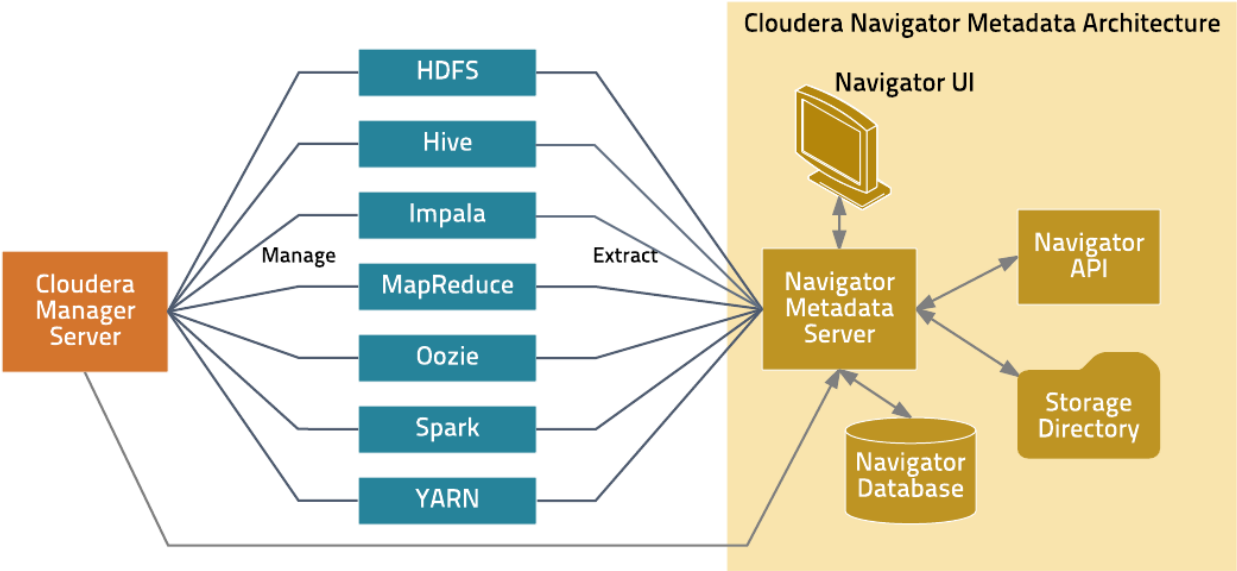
About Cloudera Data Management

- Policies - Data stewards can use Cloudera Navigator [policies](#) to define automated actions, based on data access or on a schedule, to add metadata, create alerts, and move or purge data.
- Analytics - Hadoop administrators can use Cloudera Navigator [analytics](#) 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 datacenter. Encryption and key management are also requirements 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).

You can install Cloudera Navigator data management and data encryption components independently.

Cloudera Navigator Metadata Architecture

Cloudera Navigator metadata provides data discovery and data lineage functions. The following figure depicts the Cloudera Navigator metadata architecture.



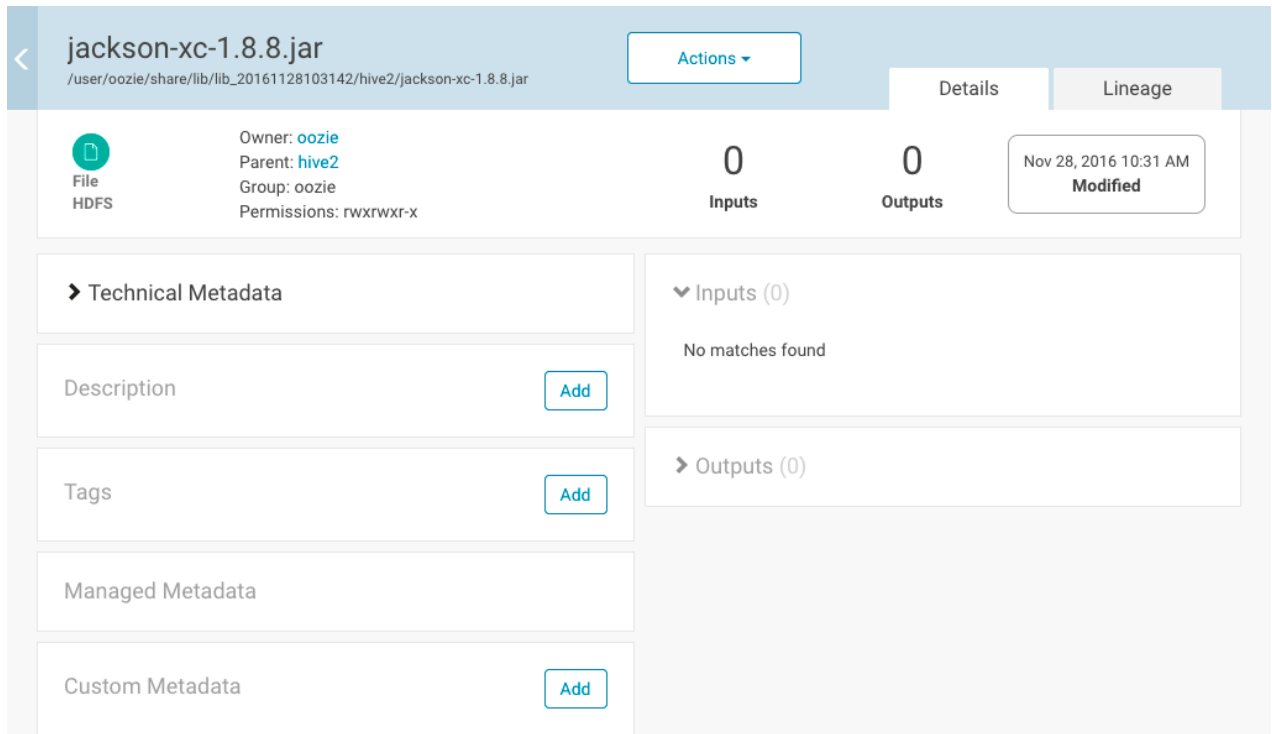
The Navigator Metadata Server performs the following functions:

- Obtains connection information about CDH services from the Cloudera Manager Server
- At periodic intervals, extracts metadata for the entities managed by those services
- Manages and applies metadata extraction policies during metadata extraction
- Indexes and stores entity metadata
- Manages authorization data for Cloudera Navigator users
- Manages audit report metadata
- Generates metadata and audit analytics
- Implements the Navigator UI and API

The Navigator database stores policies, user authorization and audit report metadata, and analytic data. The storage directory stores the extraction state and extracted metadata.

The Cloudera Navigator Metadata Server manages metadata about the entities in a CDH cluster and relations between the entities. The metadata schema defines the types of metadata that are available for each entity type it supports.

For example, the following figure shows the entity details of a file entity:



Three classes of metadata are defined for all entities:

- **Technical Metadata** - Metadata defined when entities are extracted. Such metadata includes:
 - Name of an entity
 - Service that manages or uses the entity
 - Type
 - Path to the entity
 - Date and time of creation
 - Access permissions
 - Modification, size, owner, purpose, and relations—parent-child, data flow, and instance of—between entities

You cannot modify technical metadata.

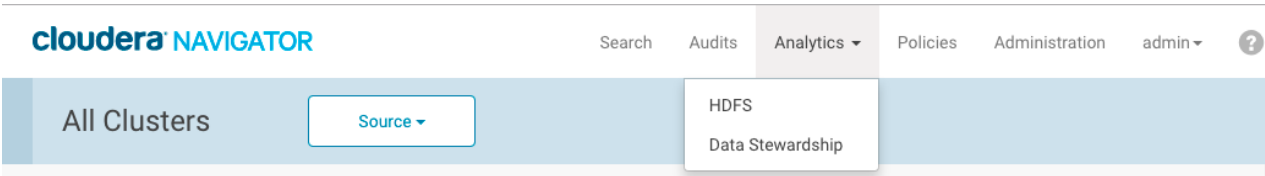
- **Custom Metadata** - Key-value pairs that can be added to entities. You can add and modify custom metadata *before and after* entities are extracted.
- **Managed Metadata** - Descriptions, key-value pairs, and tags that can be added to entities. Managed metadata key-value pairs are similar to custom metadata key-value pairs, but can also define the keys within a namespace and enforce conformance to value constraints (for example, require the value to be a date). You can add and modify managed metadata *after* entities are extracted.

In addition, for Hive entities, Cloudera Navigator supports extended attributes, which are added by Hive clients before entities are extracted.

Data Stewardship Dashboard

The Cloudera Navigator Data Stewardship dashboard captures a variety of information about data, metadata, and user jobs that process the data. The Data Stewardship dashboard provides information and metrics to help you understand the state of the data and data usage, and allows you to visualize trends and averages for a variety of data sources and actions.

Access the dashboard by clicking **Analytics** and then choosing **Data Stewardship** on the navigation bar. Specify the source clusters by clicking **Source** and clicking a cluster name or **All Clusters**.



On the Data Stewardship dashboard, select a tab for the information you want to view:

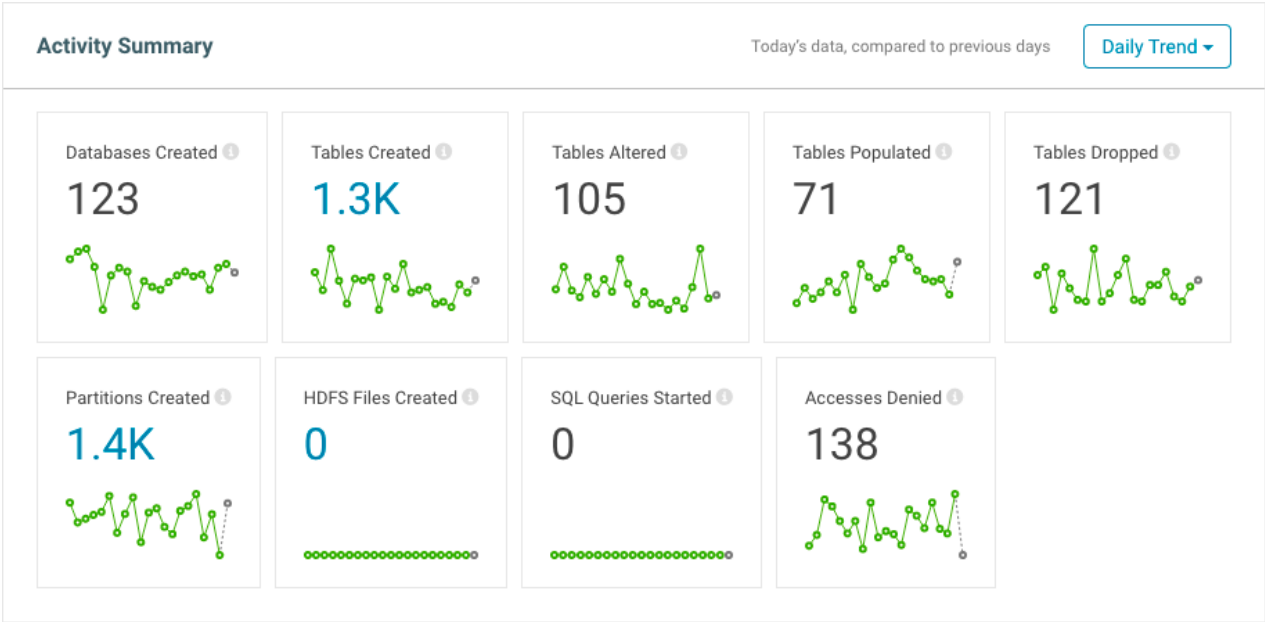
- Dashboard. Provides "at-a-glance" information about databases, tables, operations, and files and directories
- Data Explorer. Allows you to select various cluster actions to view and compare, for a specified period of time, as well as chart averages and trendlines.

The dashboard is divided into the following major information areas:

- [Activity Summary](#)
- [Databases](#)
- [Hive Tables](#)
- [Files and Directories](#)
- [Operations and Operation Executions](#)

Dashboard

Activity Summary



Each tile in the Activity Summary section provides summary information for actions on a particular entity type and includes the following:

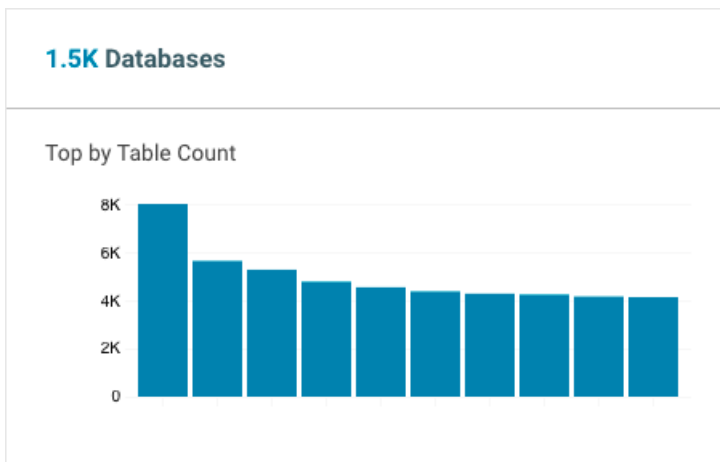
- The name of the activity
- The number of occurrences for that activity for a time period that you select (daily, weekly, monthly, quarterly, all time)
- A line graph showing activity trends based on the time period that you select

A graphical representation of the time-lapse summary for each activity tile is located at the bottom of the tile. Hovering over a point displays the value for that entity on a particular date. For example, if you select Daily Trend, the number in the graph shows number of occurrences for the day so far (since midnight), and hovering over a graph point shows the number of occurrences for that full day as well as the average for the 20-day period represented by the graph.

The Activity Summary area includes the following information:

Databases Created	Number of new databases that were added to the cluster.
Tables Created	Number of new tables that were added to the cluster. Click the value to link to the Search page that shows the search results of the query defined. You can apply filters to narrow the search results and perform any other search actions.
Tables Altered	Number of tables that were changed.
Tables Populated	Number of tables that were populated with data.
Tables Dropped	Number of tables that were deleted.
Partitions Created	Number of partitions added. You can apply filters to narrow the search results and perform any other search actions.
HDFS Files Created	Files that were created. Click the new files value to link the Search page that shows the results of the query defined. You can apply filters to narrow the search results and perform any other search actions.
SQL Queries Started	Number of SQL queries that were run.
Accesses Denied	Number of access attempts by users that were denied.

Databases



The Databases area of the Dashboard shows the total number of databases in the source clusters. The top 10 databases, by table count, are displayed in the bar graph.

Click the value next to the Database heading (in this case, 1.5K) to open a Search page showing results from the query "Deleted=Do not show deleted" "Type=Database". Apply or remove filters to refine or broaden the results.

Search Actions ▾

Filters [Add Filters](#) [Clear All Filters](#)

1,525 results [Show full query](#)

Deleted = Do not show Deleted × Type = Database ×

- Show Deleted only
- Do not show Deleted
- Show when there is a value for Deleted

▼ SOURCE TYPE

Hive 1,525

- Hive [marketsriskcalc_work](#)
Type Database Path hdfs://Enchilada/data/marketsriskcalc/work/hive Source hive1
- Hive [catssolprn_work](#)
Type Database Path hdfs://Enchilada/data/catssolprn/work/hive Source hive1
- Hive [amlimp_raw](#)
Type Database Path hdfs://Enchilada/data/amlimp/raw/hive Source hive1

Hover over the bar in the graph to see information about that database, and click the bar to open the Details page for that database. The following figure shows the Details page for the database **nav_policy_db**.

nav_policy_db Actions ▾
hdfs://eng-nav-1.gce.cloudera.com:8020/user/hive/warehouse/nav_policy_db.db

Database Hive **2** Tables **1** Views

▼ Technical Metadata

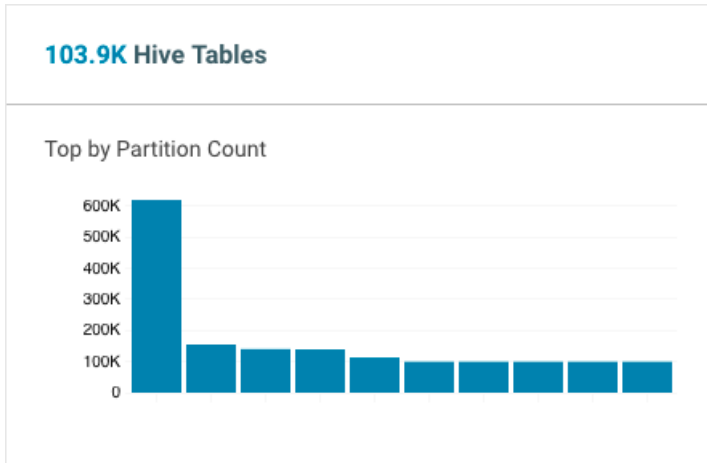
Source Type	Hive
Type	Database
Parent	(no parent)
Path	hdfs://eng-nav-1.gce.cloudera.com:8020/...
Source	HIVE-1
Classname	Hive Database
Package Name	nav

▼ Tables (2) Q

- [policy1](#)
- [policy_employees](#)

► Views (1)

Hive Tables

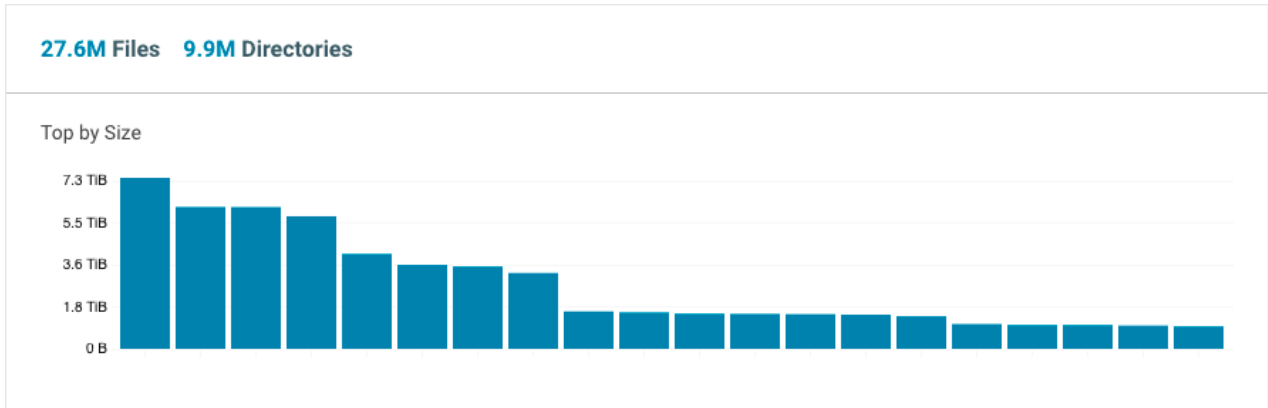


The Tables area of the dashboard shows the total number of Hive tables in the cluster. The top 10 tables, by partition count, are displayed in the bar graph.

Click the value next to the Hive Tables heading (in this case, 103.9K) to view matching tables in Search.

Click the bar to open the Details page for that table.

Files and Directories

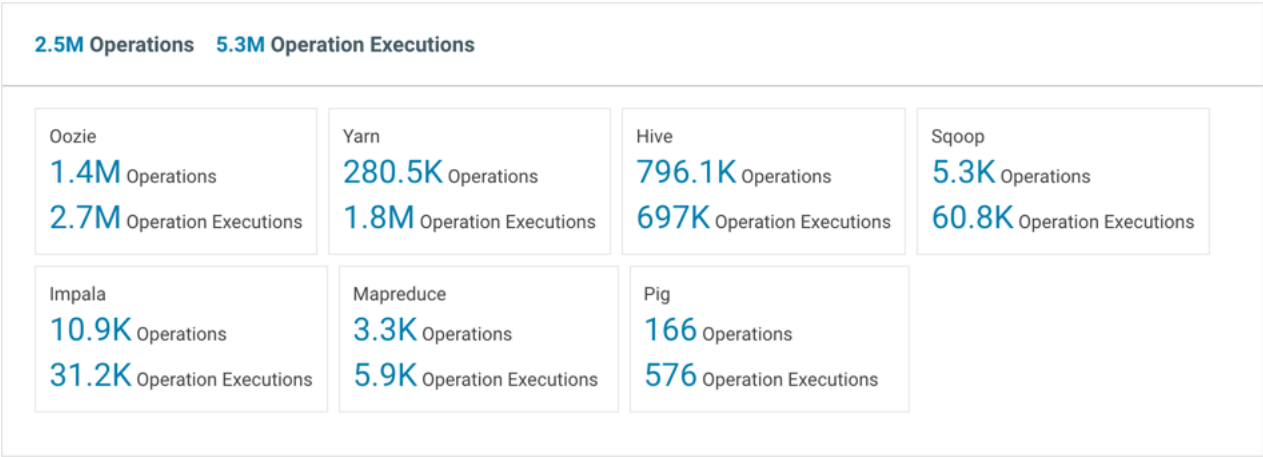


The Files and Directories area of the Dashboard shows the total number of files and directories in the cluster.

Clicking the value next to the Files or Directory heading (in this case, 64.6M or 16.2M, respectively) to show matching files or directories in Search.

The bar graph displays the top 20 files, based on size. Hover over the bar in the graph to see information about that file. Click the bar to open the Details page for that file.

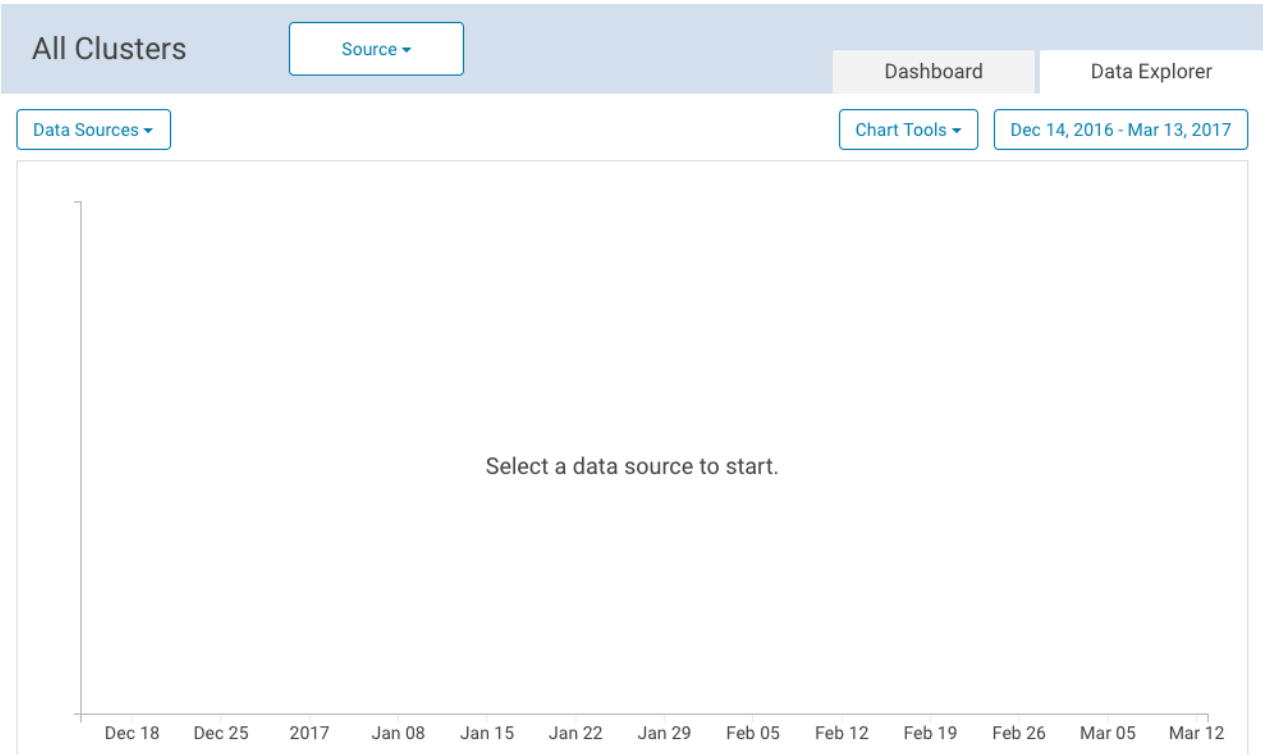
Operations and Operation Executions



The Operations and Operation Executions area of the Dashboard shows the total number of operations and operation executions that occurred in the cluster for the specified period of time.

Click the value next to the Operations or Operations Executions for a service to view matching operations or operation executions in Search.

Data Explorer



Cloudera Navigator and S3

As cloud storage becomes increasingly prevalent, many clusters use Amazon Simple Storage Service (S3) for long-running, persistent storage.

Starting with Cloudera Navigator release 2.9.0, you can use Navigator to view technical metadata, assign business metadata, and view lineage for S3 objects on your cluster. This topic provides an overview of S3 metadata entities in

Navigator, describes how to configure Navigator to use S3 data, and describes unique aspects and limitations of working with S3 data in Navigator.

S3 Metadata Entities in Navigator

Amazon S3 has a flat structure, without the hierarchy found in typical filesystems. S3 entities include *buckets* and *objects*. The bucket is the container for the object.

In Cloudera Navigator, S3 entities include the following:

- S3 Bucket
- Directory - Although S3 entities are limited to buckets and objects in those buckets, S3 supports the concept of a **folder** that can be used to organize objects. Folders in S3 are extracted as directories in Navigator.
- File

Implicit Folders

Navigator creates implicit S3 folders to mimic the behavior of a file system. For example, for an object with key **real_estate/sales/pending**, Navigator creates a file with the path **real_estate/sales/pending**, and also creates two directories: **real_estate** and **real_estate/sales**.



Important: When working with S3 entities in Cloudera Navigator, keep in mind the following:

- Currently, Navigator does not mark implicit folders as deleted if children under it are deleted.
- Navigator does not mark objects and folders as deleted if they are deleted in S3.
- You can search for implicit folders in Navigator Search by adding `implicit:true` in the Search bar; use `implicit:false` to show regular (explicit) folders only

For example, in the S3 bucket **implicit-folder-test**, if you create the folder structure **/implicit/implicit2** and add the file **explicit** to the **implicit2** folder, Cloudera Navigator shows the following for this object when extracted:

▼ Technical Metadata

Source Type	S3
Type	File
Bucket	implicit-folder-test
Path	implicit/implicit2/explicit

The directory/file combination is labeled as a Path.

You work with S3 entities in Navigator much as you would with entities for HDFS. For information on S3 entity properties, see [S3 Properties](#) on page 28.

For more information about Amazon S3, see the [Amazon S3 documentation](#).

Enabling Cloudera Navigator Access to Amazon S3

To configure Navigator for S3, you must configure AWS credentials for Cloudera Manager and enable Cloudera Navigator to access data written to S3 buckets.

You configure **AWS Credentials** to specify the Access Key Authentication type.

This type of authentication requires an AWS Access Key and an AWS Secret key that you obtain from Amazon. For more information about setting up keys in AWS, see [Creating an IAM User in Your AWS Account](#) in the AWS Identity and Access Management documentation.

**Important:**

- Navigator supports only a single key for authentication; only one AWS credential can be used at a time.
- A Navigator instance must extract from only one AWS account, and one AWS account can be extracted by only one Navigator instance. Multiple Navigator instances extracting from one AWS account can cause unpredictable behavior and is not supported.
- The key provided to Navigator requires permissions for SQS and S3. For more information, see [S3 Data Extraction for Navigator](#) on page 17.

Cloudera Manager stores these values securely and does not store them in world-readable locations. The credentials are masked in the Cloudera Manager Admin console, encrypted in the configurations passed to processes managed by Cloudera Manager, and [redacted](#) from the logs.

Minimum Required Role: [User Administrator](#) (also provided by **Full Administrator**)

To enable Cloudera Navigator access to Amazon S3, you must add **AWS Credentials** for Amazon S3 and then enable Navigator access. If you have already added AWS credentials, skip to step 5:

1. Open the Cloudera Manager Admin Console.
2. Click **Administration > AWS Credentials**.
3. Click **Add** and select **Access Key Authentication**. This authentication mechanism requires you to obtain AWS credentials from Amazon.
 - a. Enter a **Name** for this account. The name can contain alphanumeric characters, hyphens, underscores, and spaces.
 - b. Enter the **AWS Access Key ID**.
 - c. Enter the **AWS Secret Key**.



Important: Although AWS offers two types of authentication—IAM Role-based Authentication and Access Key Authentication—you must specify Access Key Authentication for Cloudera Navigator. IAM Role-based Authentication is not supported.

4. Click **Add**. The **Connect to Amazon Web Services** screen displays.
5. Click the **Enable for Cloudera Navigator** link.

✕

Connect to Amazon Web Services

The AWS Credential **test** has been added. You now can:

<p>Cloud Backup and Restore</p> <p>Back up and restore data in S3 by creating a replication schedule and choose the above name as the destination or the source.</p> <p>Replication Schedules</p>	<p>Cluster Access to S3</p> <p>Enable Impala users to query S3-backed tables without directly providing AWS credentials, subject to having the proper permissions defined via Sentry.</p> <p>Enable Hue administrators to browse the S3 filesystem and define Impala tables backed by S3 data authorized to that AWS identity.</p> <p>Enable for Cluster 1</p>	<p>Cloudera Navigator Access to S3</p> <p>Enable Cloudera Navigator to extract metadata and lineage for data that is written to S3 buckets in this account.</p> <p>Enable for Cloudera Navigator</p> <ul style="list-style-type: none"> • Only one AWS Credential may be used at a time. Enabling this credential will disable any others previously configured. • Accessing the same S3 buckets across multiple Cloudera Navigator instances is not supported.
---	--	---

[Close](#)

6. [Restart the Cloudera Navigator Metadata server](#) to enable access.

Extracted S3 information should now be available in Cloudera Navigator.

Eventual Consistency

Amazon S3 uses an [eventual consistency](#) model. To provide high availability, consistency is informally guaranteed: Eventually, an item returns the last updated value to all accesses of that item, assuming no new updates for a period of time.

Because S3 uses eventual consistency, it might take some time for the S3 object to appear in Navigator. In addition, you might notice discrepancies between the object in Navigator and the object in S3. In Navigator, if you do not immediately see an S3 object that you created or do not see modifications that you made, that does not mean the object does not exist or was not successfully edited. In most cases, the lag time associated with eventual consistency is causing the object to not appear in Navigator or to not match the most recent version in S3.

S3 Event Notification

In Amazon S3, you [enable a bucket to send notification messages](#) whenever certain events occur. Cloudera Navigator uses Amazon Simple Queue Service (Amazon SQS) to extract S3 information.

Amazon SQS is a distributed, highly scalable hosted queue for storing messages. Navigator pulls data from the SQS queue. For more information about Amazon SQS, see [Getting Started with Amazon SQS](#).

By default, Navigator sets up these queues and configures S3 event notification for each bucket for you. Navigator does not overwrite existing S3 event notifications. However, if any buckets have existing S3 event notifications, you must use "[bring your own queue](#)" and use an [Amazon SNS "fanout"](#). In a "fanout" scenario, an Amazon SNS message is sent to a topic and then replicated and pushed to multiple Amazon SQS queues, HTTP endpoints, or email addresses.

For more information about configuring Navigator data extraction, including "bring your own queue", see [S3 Data Extraction for Navigator](#) on page 17.

Setting API Limits

You can set an API limit for Amazon S3 API and SQS API. You are billed on a monthly basis, depending on your usage; the billing cycle resets each month. By setting these API limits, you can manage the monthly cost of using the APIs.

To set API limits, add the following in the **Navigator Metadata Server Advanced Configuration Snippet (Safety Valve) for cloudera-navigator.properties** in Cloudera Manager:

```
nav.aws.api.limit=any_int
```

Once your API limit is reached, Navigator suspends extraction until the next 30-day interval begins. Then at that point, Navigator extracts any data that was not extracted during the time activity was suspended.

Cloudera Navigator does not indicate if your use of the API exceeds the monthly limit; monitor your monthly use of the APIs to manage your costs.

Limitations of Navigator for S3

S3 entities in Cloudera Navigator work in much the same way that HDFS entities do. However, the current release of Navigator has some limitations related to S3:

- Only one instance of Navigator can be configured per S3 account, and Navigator can use only one AWS credential.
- IAM role-based authentication is not supported.
- Extraction limitations:
 - Navigator extracts only [user-defined metadata](#) in S3. System-defined metadata types are not extracted.
 - Navigator does not extract tags for S3 buckets and objects.
 - Navigator extracts only the latest versions in S3; it does not extract historical versions.
 - AWS supports unnamed directories, but Navigator does not extract them.
- Auditing is not available for S3.

- Lineage is supported for Hive, Impala, and MapReduce on S3. Other types are not supported.
- MapReduce glob paths are not supported.
- S3 object removal with [Object Lifecycle Management](#) is not supported.

For example, if you set a lifecycle rule to automatically remove any objects older than 10 days, that delete event is not be tracked by SQS and therefore not tracked by Navigator.

To use lifecycle rules with Navigator extraction, you must use only bulk extraction and not incremental extraction.

S3 Data Extraction for Navigator

You can extract S3 data for Navigator using different methods, depending on your requirements. This topic discusses the different ways that you can extract S3 data for Cloudera Navigator, and describes extraction settings you can set using Advanced Configuration Snippets in Cloudera Manager. It also shows the IAM policy documents that you need to create and attach for the Navigator user in AWS.

Creating IAM Policy Documents

To enable S3 data extraction for Cloudera Navigator, you must create a policy document in AWS and attach that policy document to the AWS user associated with the Cloudera Navigator instance. Each of the extraction methods described include a policy document that you create to enable extraction.

Amazon documentation includes extensive information about creating IAM policies and managing access:

- [Access Policy Language Overview](#) describes the basic elements used in policies.
- [Creating a New Policy](#) describes how you can create an access policy document. Use the instructions in [Edit a policy using the policy editor](#) to create a policy by copying the policy text defined in each extraction type section and pasting it in the policy editor.
- For additional information and [walkthroughs](#), see the various topics in [Managing Access Permissions to Your Amazon S3 Resources](#).

Bulk and Incremental Extraction

By default, Navigator uses combined bulk and incremental extraction. The first extraction is a bulk extraction; all subsequent extractions are incremental.



Important: Use combined bulk and incremental extraction if you do not have existing S3 event notification configured for any S3 buckets.

Bulk and incremental extraction is recommended for production environments.

AWS Resources Created by Cloudera Navigator:

- SQS queue in each region in which you have buckets
- S3 event notification for each bucket

Advantages:

- Highest performance of all extraction types.
- Least expensive in terms of API cost.
- Easiest to use; no additional Navigator or Cloudera Manager setup is required.

Disadvantages:

- You cannot use this method for any buckets that have existing S3 event notification.
- You must obtain additional permissions on AWS.
- Navigator changes your AWS environment. SQS queues are created in all regions that have buckets, and event notification is updated on all buckets.

Policy Document - Bulk and Incremental Extraction

To enable bulk and incremental extraction, you create the following policy document by copying the policy text and pasting it in the [policy editor](#), and then attaching it to the Navigator user in AWS.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1481678612000",
      "Effect": "Allow",
      "Action": [
        "sqs:CreateQueue",
        "sqs:DeleteMessage",
        "sqs:DeleteMessageBatch",
        "sqs:GetQueueAttributes",
        "sqs:GetQueueUrl",
        "sqs:ReceiveMessage",
        "sqs:SetQueueAttributes"
      ],
      "Resource": "*"
    },
    {
      "Sid": "Stmt1481678744000",
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "s3:GetObject",
        "s3:GetObjectAcl",
        "s3:GetBucketNotification",
        "s3:PutBucketNotification"
      ],
      "Resource": [
        "arn:aws:s3::*:*"
      ]
    }
  ]
}
```

Bulk Extraction Only

Use bulk extraction only for proof-of-concept or demonstration environments.



Note: Using only bulk extraction is not recommended for production environments.

Advantages:

- No additional AWS resources are required.
- Minimum permissions are required to read from S3.

Disadvantages:

Because all S3 data is re-extracted each time it is run, bulk extraction is:

- Slow; it can take 10 times as long as extraction takes with [Bulk and Incremental Extraction](#) on page 17.
- Expensive in terms of API cost.

Setup

To set up bulk extraction only, add the following to the **Navigator Metadata Server Advanced Configuration Snippet (Safety Valve) for cloudera-navigator.properties** in Cloudera Manager, and then restart the metadata server:

```
nav.s3.extractor.incremental.enable=false
```

Policy Document - Bulk Extraction

To enable bulk extraction only, create the following policy document by copying the policy text and pasting it in the [policy editor](#), and then attaching it to the Navigator user in AWS.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1481676614000",
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "s3:GetObject",
        "s3:GetObjectAcl"
      ],
      "Resource": [
        "arn:aws:s3::*:*"
      ]
    }
  ]
}
```

Event Notification for an External Queue

If you have existing S3 event notification configured for any S3 buckets, you must use configure extraction to use that external queue. This requires you to set up the queues and configure event notification. "Bring your own queue" extraction is recommended for production environments.

AWS Resources Created by the User:

- SQS queue for each region in which you have buckets.
- S3 event notification for each bucket to send change events to the Navigator queue.

Advantages:

- Full control over your AWS environment.
- Performance level is high.

Disadvantages:

- Requires significant manual setup and configuration.

Setup

To set up event notification for an external queue, you must create a queue from which Navigator can extract, and then configure S3 event notifications to send to Navigator.



Important: For new buckets, make sure to configure s3 event notification before you add any data in the bucket.

If you create that queue instead of having Navigator set it up for you, do the following:

1. Stop Cloudera Navigator.
2. Open the Amazon SQS console and create a queue with the following settings:
 - Default Visibility Timeout: 10 minutes
 - Message Retention Period: 14 days
 - Delivery Delay: 0 seconds
 - Receive Message Wait Time: 0 seconds
3. Select the queue you created, click the **Permissions** tab, click **Add a Permission**, and configure the following in the **Add a Permission to...** dialog box:

- Effect: Allow
- Principal: Everybody
- Actions: SendMessage

In the Conditions (optional) area, set the following values:

- Qualifier: None
- Condition: ArnLike
- Key: aws:SourceArn
- Value: arn:aws:s3::*.*

When finished, click **Add Condition**, and then click **Add Condition**.

4. Set up a queue in each region in which you have buckets.
5. Configure [event notification](#) for every bucket:
 - Name: nav-send-metadata-on-change
 - Events: ObjectCreated(All) and ObjectRemoved(All)
 - Send to: SQS queue
 - SQS queue: The name of your queue
6. Configure SNS fanout if you have existing S3 event notification. For more information about SNS fanout, see [Common SNS Scenarios](#).
7. In Cloudera Manager, add the following to the **Navigator Metadata Server Advanced Configuration Snippet (Safety Valve) for cloudera-navigator.properties** in Cloudera Manager:

```
nav.s3.extractor.incremental.enable=true
nav.s3.extractor.incremental.auto_setup.enable=false
nav.s3.extractor.incremental.queues=queue_json
```

`queue_json` must have the following JSON format, with no spaces:

```
[{"region":"us-west-1", "queueUrl":"https://sqs.us-west-1.amazonaws.com/account_num/queue_name"}, {"queue_2"}, ... {"queue_n"}]
```



Important:

- In JSON, you must use two backslash (\\) characters to escape commas (,).
- Navigator extracts only one queue per region.

8. Restart Navigator.

Policy Document - External Queue

To enable event notification for an external queue, create the following policy document by copying the policy text and pasting it in the [policy editor](#), and then attaching it to the Navigator user in AWS.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1481678612000",
      "Effect": "Allow",
      "Action": [
        "sqs:DeleteMessage",
        "sqs:DeleteMessageBatch",
        "sqs:GetQueueAttributes",
        "sqs:ReceiveMessage"
      ],
      "Resource": "*"
    },
    {
      "Sid": "Stmt1481678744000",
```

```

    "Effect": "Allow",
    "Action": [
      "s3:GetBucketLocation",
      "s3:ListAllMyBuckets",
      "s3:ListBucket",
      "s3:GetObject",
      "s3:GetObjectAcl",
      "s3:GetBucketNotification",
      "s3:PutBucketNotification"
    ],
    "Resource": [
      "arn:aws:s3::*"
    ]
  }
]
}
}

```

Navigator S3 Extraction Options

You can use the following S3 extraction options to configure extraction from S3. The options are available in **Navigator Metadata Server Advanced Configuration Snippet (Safety Valve) for cloudera-navigator.properties** in Cloudera Manager.

Table 1: S3 Extraction Options in Cloudera Navigator

Option	Description	Default Value
<code>nav.s3.extractor.max_threads</code>	Number of extractions workers to run in parallel.	3
<code>nav.s3.extractor.enable</code>	Enable or disable S3 extraction.	true (if a key is provided to Navigator through Cloudera Manager)
<code>nav.s3.extractor.incremental.enable</code>	Enable or disable incremental extraction. If set to false, bulk extraction is run. You can enable bulk extraction by setting this value to to false and restarting Navigator.	true
<code>nav.s3.extractor.incremental.batch_size</code>	Number of messages to keep in memory at a time.	1000
<code>nav.s3.extractor.incremental.auto_setup.enable</code>	Autoconfigure queues and configure S3 event notification. Set to false to use "bring your own queue".	true
nav.s3.extractor.incremental.queues	List of queues to use in external queue use case. <div style="border: 1px solid orange; padding: 10px; margin-top: 10px;"> <p>Important:</p> <ul style="list-style-type: none"> In JSON, you must use a double backslash (\\) character to escape commas (,). Navigator extracts only one queue per region. </div>	N/A

Option	Description	Default Value
<code>nav.aws.api.limit</code>	Maximum number of API calls that Navigator can make per month.	5,000,000,000
<code>nav.sqs.max_receive_count</code>	Number of retries for SQS messages that might be inconsistent due to eventual consistency.	10
<code>nav.s3.implicit.batch_size</code>	Number of Solr documents to keep in memory when updating the state of implicit folders.	1000
<code>nav.s3.home_region</code>	The closest AWS region. Using the closest region can reduce API request time.	us-west-1

Metadata Extraction and Indexing

Metadata Extraction

The [Navigator Metadata Server](#) extracts metadata for the following resource types.

Table 2: Resource Metadata Extraction

Resource Type	Metadata Extracted
HDFS	HDFS metadata at the next scheduled extraction run after an HDFS checkpoint. If high availability is enabled, metadata is extracted as soon as it is written to the JournalNodes.
Hive	Database, table, and query metadata from Hive lineage logs. See Managing Hive and Impala Lineage Properties .
Impala	Database, table, and query metadata from the Impala Daemon lineage logs. See Managing Hive and Impala Lineage Properties .
MapReduce	Job metadata from the JobTracker. The default setting in Cloudera Manager retains a maximum of five jobs; if you run more than five jobs between Navigator extractions, the Navigator Metadata Server extracts the five most recent jobs.
Oozie	Oozie workflows from the Oozie Server.
Pig	Pig script runs from the JobTracker or Job History Server.
S3	Bucket and object metadata.
Spark	Spark job metadata from YARN logs. (Unsupported and disabled by default. To enable, see Enabling Spark Metadata Extraction .)
Sqoop 1	Database and table metadata from Hive lineage logs; job runs from the JobTracker or Job History Server.
YARN	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 is extracted and linked in Navigator after the extraction poll period (10 minutes by default) plus a service-specific interval, as follows:

- **HDFS:** $t_0 + (\text{extraction poll period}) + (\text{HDFS checkpoint interval (1 hour by default)})$
- **HDFS + HA:** $t_0 + (\text{extraction poll period})$
- **Hive:** $t_0 + (\text{extraction poll period}) + (\text{Hive maximum wait time (60 minutes by default)})$
- **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 four types of key-value pairs, using the indicated syntax:

- **Technical metadata key-value pairs** - *key:value*
 - *key* is one of the properties listed in [Search Properties](#) on page 24.
 - *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.

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

- **Custom metadata key-value pairs** - *up_key:value*
 - *key* is a user-defined property.
 - *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.

Custom metadata key-value pairs can be modified.

- **Hive extended attribute key-value pairs** - *tp_key:value*
 - *key* is an extended attribute set on a Hive entity. The syntax of the attribute is specific to Hive.
 - *value* is a single value supported by the entity type.

Hive extended attribute key-value pairs are read-only and cannot be modified.

- **Managed metadata key-value pairs** - *namespace.key:value*
 - *namespace* is the namespace containing the property. See [Defining Managed Metadata](#) on page 34.
 - *key* is the name of a managed metadata property.
 - *value* is a single value, a range of values specified as [*value1* TO *value2*], or a set of values separated by spaces. 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.

Only the values of managed metadata key-value pairs can be modified.

- **S3 key-value pairs** - `tp_key:value`
 - `key` is the name of [user-defined metadata](#).
 - `value` is a single value.
 - Only file metadata is extracted; bucket and folder metadata is not extracted.

Constructing Compound Search Strings

To construct compound search strings, you can join multiple property-value pairs using the [Lucene Query Parser Boolean operators](#):

- `, +, -`
- OR, AND, NOT

In both syntaxes, you use `()` to group multiple clauses into a single field and to form subqueries. When you [filter results](#) in the Navigator Metadata UI, the constructed search strings use the `, +, -` syntax.

Example Search Strings

- Entities in the path `/user/hive` that have not been deleted - `+("/user/hive") +(-deleted:true)`
- Descriptions that start with the string "Banking" - `description:Banking*`
- Entities of type MapReduce or entities of type Hive - `sourceType:mapreduce sourceType:hive OR sourceType:mapreduce OR sourceType:hive`
- Entities of type HDFS with size equal to or greater than 1024 MiB or entities of type Impala - `(+sourceType:hdfs +size:[1073741824 TO *]) sourceType:impala`
- Directories owned by `hdfs` in the path `/user/hdfs/input` - `+owner:hdfs +type:directory +filePath:"/user/hdfs/input" OR 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]`
- Custom key-value - `project-customer1 -up_project:customer1`
- Technical key-value - In Hive, specify table properties like this:

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

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

- Managed key-value with multivalued property - `MailAnnotation.emailTo:"dana@example.com" MailAnnotation.emailTo:"lee@example.com"`



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

Search Properties

The following reference describes 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
description	text	Description of the entity.
group	caseInsensitiveText	The group to which the owner of the entity belongs.
name	ngrammedText	The overridden name of the entity. If the name has not been overridden, this value is empty. Names cannot contain spaces.
operationType	ngrammedText	The type of an operation: <ul style="list-style-type: none"> • Pig - SCRIPT • Sqoop - Table Export, Query Import
originalName	ngrammedText	The name of the entity when it was extracted.
originalDescription	text	The description of the entity when it was extracted.
owner	caseInsensitiveText	The owner of the entity.
principal	caseInsensitiveText	For entities with type OPERATION_EXECUTION, the initiator of the entity.
properties	string	A set of key-value pairs that describe the entity.
tags	ngrammedText	A set of tags that describe the entity.
type	tokenizedCaseInsensitiveText	The type of the entity. The available types depend on the entity's source type: <ul style="list-style-type: none"> • hdfs - DIRECTORY, FILE, DATASET, FIELD • 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		

Name	Type	Description
The available timestamp fields vary by the source type: <ul style="list-style-type: none"> hdfs - created, lastAccessed, lastModified hive - created, lastModified impala, mapreduce, pig, spark, sqoop, and yarn - started, ended 	date	Timestamps in the Solr Date Format . For example: <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]

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.

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	long	The time the entity was moved to the Trash folder.
filePath	path	The path to the entity.
mimeType	ngamedText	The MIME type of an HDFS file.
parentPath	string	The path to the parent entity of a child entity. For example: parent path: /default/sample_07 for the table sample_07 from the Hive database default.
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: size: [1000 TO *], size: [* TO 2000], and size: [* TO *] to find all fields with a size value.

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.

MapReduce and YARN Properties

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

Operation Properties

Name	Type	Description
Operation		
inputFormat	string	The fully qualified name of the class of the input format.
outputFormat	string	The fully qualified name of the class of the output format.
Operation Execution		
inputs	string	The name of the entity input to an operation execution. For entities of resource type <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.

Name	Type	Description
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.

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.

S3 Properties

Name	Type	Description
Object Properties		
region	string	The geographic region in which the bucket is stored
bucketName	string	The name of the bucket in which the object is stored
filePath	path	The key of the S3 object.
size	long	Object size in bytes.
lastModified	date	Object creation date or the last modified date, whichever is the latest.
etag	string	A hash of the object. The ETag reflects changes only to the contents of an object, not its metadata. The ETag may or may not be an MD5 digest of the object data.
storageClass	string	Storage class used for storing the object.
owner	string	Owner of the object.
sequencer	string	Latest S3 event notification sequencer. Used to order events.
parentPath	string	Parent of the S3 object.
technicalProperties	key-value pairs	Custom metadata for each S3 object.
Bucket Properties		
region	string	Region for the bucket.
created	date	Date the bucket was created.
owner	string	Owner of the bucket.

Sqoop Properties

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

Accessing Metadata

Minimum Required Role: [Metadata Viewer](#) (also provided by **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 in to the Cloudera Navigator data management component UI.](#)
2. To display all entities, click **Clear all filters**.
3. Filter the search results by specifying filters or typing search strings in the Search box.

Filter Example

The following filter example demonstrates how to narrow search results by selecting a built-in Source Type filter set to **HDFS** and the managed property **BLOCK SIZE (MIB)** with the value **>=1024**. This example also shows the full query, which you can see by clicking **Show full query** in the results pane.

Search [Search Box] Actions ▾

Filters [Add Filters](#) [Clear All Filters](#)

Add Filter...

▼ **BLOCK SIZE (MIB)** ×

- 0 - < 256
- 256 - < 512
- 512 - < 1024
- >= 1024
- Custom

▼ **SOURCE TYPE**

- HDFS** 3
- > SIZE (MIB)
- > CREATED
- > GROUP

Full Query ×

+blockSize:[1073741824 TO *)+sourceType:hdfs

3 results

Block Size >= 1024 MIB **Source Type = HDFS**

HDFS 5179ab77-80c2-4513-be69-3b7aa4f83d04

Type File Path /accumulo/wal/msd-22.gce.cloudera.com+10011/5179ab77-80c2-4513-be69-3b7aa4f83d04
 Owner accumulo Group accumulo Permissions rwxr-xr-x Size 1.75 KiB Block Size 1.10 GiB [View in Hue](#)
 Last Accessed Dec 8, 2016 5:10 PM Last Modified Dec 8, 2016 5:10 PM Created Dec 8, 2016 5:10 PM
 Source HDFS-1

HDFS dba466fa-fbea-4e80-b958-85dc8822c0bb

Type File Path /accumulo/wal/msd-22.gce.cloudera.com+10011/dba466fa-fbea-4e80-b958-85dc8822c0bb
 Owner accumulo Group accumulo Permissions rwxr-xr-x Size 212 B Block Size 1.10 GiB [View in Hue](#)
 Last Accessed Dec 8, 2016 5:19 PM Last Modified Dec 8, 2016 5:19 PM Created Dec 8, 2016 5:19 PM
 Source HDFS-1

HDFS fc96462e-c480-49f5-8fdd-8637caf1351f

Type File Path /accumulo/wal/msd-24+10011/fc96462e-c480-49f5-8fdd-8637caf1351f Owner accumulo
 Group accumulo Permissions rwxr-xr-x Size 0 B Block Size 1.10 GiB [View in Hue](#)

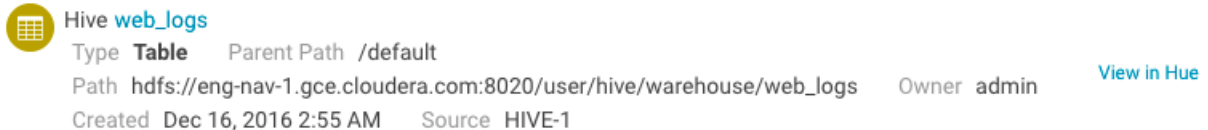
Search Results

The Search Results pane displays the number of matching entries in pages listing 25 entities per page. You can view the pages using the page control at the bottom of each page.

Each entry in the result list contains:

- Source type
- Name - 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:



The screenshot shows a search result for a Hive table. On the left is a yellow icon with a grid pattern. To its right is the text 'Hive web_logs'. Below this, the word 'Type' is followed by 'Table'. To the right of 'Table' is 'Parent Path /default'. Below that is the 'Path' field with the value 'hdfs://eng-nav-1.gce.cloudera.com:8020/user/hive/warehouse/web_logs'. To the right of the path is 'Owner admin'. Below the path is 'Created Dec 16, 2016 2:55 AM' and 'Source HIVE-1'. On the far right, there is a blue link labeled 'View in Hue'.

Displaying Entity Details

The entity Details page displays all three types of metadata for an entity—technical, managed, and custom—and entity type-specific information:

- HDFS directories - Directory contents
- HDFS datasets and fields - Schema
- Hive and Impala databases - Tables and views
- Hive tables - Extended attributes, table schema, partitions
- Impala tables - Table schema
- MapReduce, YARN, and Spark operations - Instances
- Pig operation executions - Tables
- S3 buckets and folders - Folder contents

All entity types can display inputs and outputs. See [Configuring Display of Inputs and Outputs](#).

If managed properties have been defined for a particular entity type, the **Show All** checkbox in the Managed Metadata pane displays all properties that can be assigned values for the selected entity. To display only those properties that have values, clear the checkbox. If all properties have values, the checkbox has no effect.

To display entity details:

1. Perform a search.
2. In the search results, click an entity name link. The Details tab displays.

Hive Table Entity Details

For example, if you click the Hive table `weblogs` link in the search result displayed in preceding example, you see the following details:

The screenshot shows the 'web_logs' table details in Cloudera Navigator. At the top, there's a header with the table name and an 'Actions' dropdown. Below that, a summary row displays: Owner: admin, Parent: default, 29 Columns, 4 Partitions, 0 Inputs, and 0 Outputs. A 'Created' timestamp of Dec 16, 2016 2:55 AM is shown in a box. The main content area is divided into two columns. The left column contains 'Technical Metadata' (expanded), 'Description' (with an 'Add' button), 'Tags' (with an 'Add' button), and 'Custom Metadata' (with an 'Add' button'). The right column contains 'Columns (29)', 'Inputs (0)', and 'Outputs (0)', each with a caret icon to expand.

The caret > indicates fields with content; click to expand them. In addition to the technical metadata, you can view the names of the columns and the inputs and outputs.

Managed Metadata Example

The following account table entity has two managed properties in the realestate namespace: Department and AccessAttempts.

The screenshot shows the 'account' table details. The header includes the table name and an 'Actions' dropdown. The summary row shows: Owner: gfcharan, Parent: I4_amlharan, 43 Columns, 21 Partitions, 16 Inputs, and 0 Outputs. The main content area is split into two columns. The left column shows 'Managed Metadata' (expanded) with an 'Edit' button. Under the 'realestate' namespace, there are two properties: 'Department' with value 'Finance' and 'AccessAttempts' with value '10'. The right column shows 'Columns (43)' with a search icon and a scrollable list of columns: accommodation_account_flag (string), account_booking_country (string), account_closing_date (string), and account_currency (string).


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, and tags) and property values (also called facets). You can add a filter by clicking in Add Filters... and scrolling or by typing in the filter combo box to search for it. To remove nondefault filter properties, click the x 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 or specify the value using the spinner control  and select the unit minutes, hours, or days.
 - 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 press the up and down arrow keys.
 - Time - Click the hour, minute, and AM/PM fields and click the spinner arrows or press the 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, clear 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:

▼ SOURCE TYPE

● HDFS 415,271

▼ SIZE (MIB)

0 - < 256 412,676 512 - < 1024 0

256 - < 512 0 >= 1024 0

Custom

▼ CREATED

[Last 30 days](#) 264,154 [Last 90 days](#) 415,270

[Last day](#) 9,710 [Last 365 days](#) 415,270

Last

Custom period

▼ GROUP

spark 407,266

hadoop 3,159

cmjobuser 2,131

hbase 1,150

oozie 712

[Add New Value](#)

The number to the right a property value is the number of extracted entities that have that property value:

SOURCE TYPE	
<input checked="" type="checkbox"/> ● HDFS	415,301
<ul style="list-style-type: none"> > SIZE (MIB) > CREATED > GROUP 	
<input checked="" type="checkbox"/> ● YARN	1,084
<ul style="list-style-type: none"> > STARTED > ENDED 	
<input checked="" type="checkbox"/> ● Hive	239
<ul style="list-style-type: none"> > STARTED > ENDED 	
<input type="checkbox"/> ● Pig	18
<input type="checkbox"/> ● Impala	5

Facet values with the count 0 are not displayed.

When you type values, the value is enclosed in quotes; the value inside the quotes must match the metadata exactly. 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, use parentheses () to specify multiple values of a property. Add multiple properties by using the + operator. For example, entities of type HDFS or Hive that are of type file or directory:

```
+(sourceType:hdfs sourceType:hive) +(type:file type:directory)
```

and:

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

Saving Searches

1. Specify a search string or set of filters.
2. To the right of the Search box, 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. To the right of the Search box, 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](#).

Defining Managed Metadata

Minimum Required Role: [Metadata Administrator](#) (also provided by **Full Administrator**)

You can use managed metadata to add typed metadata to classes of entities. You can add namespaces and properties.

A **namespace** is a container for properties. Four namespaces are reserved:

- `nav` for Navigator [metadata classes](#) (for example, `fsElement` and user-defined custom fields)
- `up` ([custom metadata](#))
- `tp` (technical properties)
- `xt` (partner applications)

The combination of namespace and property name must be unique.



Note: You cannot remove a namespace by using through the Navigator user interface. You can delete empty namespaces by using the Cloudera Navigator API.

A property can be one of the following types:


- Boolean
- date
- integer
- long
- float
- double
- text (with optional maximum length and regular expression validation criteria)
- enum (of string)


A property can be single-valued or assume multiple values.

Once you have created properties and assigned property values to specific entities, you can create [search filters](#) for property values.

Creating Custom Properties Using the Metadata UI

To create custom properties in Cloudera Navigator:

1. [Start and log into the Cloudera Navigator data management component UI](#) with the credentials of a user having one of the following user roles:
 - Cloudera Manager Full Administrator
 - Cloudera Manager Navigator Administrator
 - Cloudera Navigator Full Administrator
 - Cloudera Navigator Metadata Administrator
2. Click the **Administration** link in the upper right. The **Managed Metadata** tab displays the list of namespaces and the properties defined in the namespaces.
3. Click the **New Property...** button.
4. In the Classes field, click  and type the beginning of a [Navigator entity classname](#).

- 5. Select the class of entities to which the property applies. To clear the field, hover over the field and click the  that displays at the right of the field.
- 6. Click the Namespace field and select a namespace. If the Namespace drop-down list is empty, click **Create Namespace....**
 - a. Specify a namespace name and optional description.
 - b. Click **Continue**.

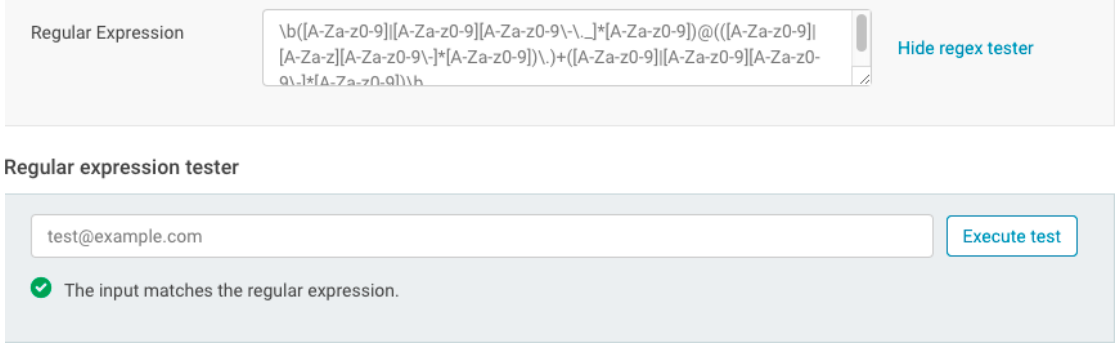
7. Add the name for the property.

The name can contain letters, numbers, underscores, and hyphens and can be up to 50 characters long.

- 8. Specify an optional description.
- 9. Select the **Multivalued Enable** checkbox if the property can have more than one value. For example, an `emailFrom` property should accept only one value, but an `emailTo` property could accept more than one value.

10 In the **Type** drop-down list, select the property type and specify constraints on the value.

- **Boolean** - Boolean: true or false.
- **Date** - Date and time.
- **Enumeration** - A set of values. In the **Enumeration** field, type valid enumeration values and press **Enter** or **Tab**.
- **Number** - A number. In the **Number Type** field, select the type of the number: **Integer**, **Long**, **Float**, **Double**.
- **Text** - A string.
 - **Maximum Length** - The maximum length of the string.
 - **Regular Expression** - A regular expression that determines whether a string is a valid value. You can test the expression by clicking **Show regex tester**, entering input that you expect to match the expression, and clicking **Execute test**. In the following example, the expression tester indicates that `test@example.com` matches the defined expression.



- 11 Click **Continue to Review**. The Review screen displays.
- 12 Click **Create** to create the property, **Cancel** to return to the Properties page, or **Back to Edit Property** to continue editing the property.

Example Properties

The following figure shows two properties in the namespace `MailAnnotation` that apply to entities of the `HDFS Entity` class (HDFS files and directories). The `emailFrom` property is of type `TEXT` and can be assigned a single value. The `MailTo` property is also of type `TEXT` but can have multiple values.

Properties

See the [documentation](#) before creating or modifying properties.

Name	Status	Type	Classes	Multivalued	Actions
MailAnnotation					
EmailFrom	Active	Text	HDFS Entity	No	Actions ▾
EmailTo	Active	Text	HDFS Entity	Yes	Actions ▾



Managing Custom Properties Using the Metadata UI

You can view managed metadata property summary details by clicking property name in the Properties table, or by clicking the **Actions** box in the property row and then clicking **View** in the dropdown.

You can also edit some aspects of a property, delete and restore a property, and purge a deleted property.

Editing a Property

After a property is created, you can edit property data in the following ways:

- Add [classes](#) to which the property applies
 - Add and remove enumeration values
 - Change the description
 - Change the maximum length
 - Change the regex pattern
1. [Start and log into the Cloudera Navigator data management component UI](#) with the credentials of a user having one of the following roles:
 - Cloudera Manager Full Administrator
 - Cloudera Manager Navigator Administrator
 - Cloudera Navigator Full Administrator
 - Cloudera Navigator Metadata Administrator
 2. Click the **Administration** link in the upper right. The **Managed Metadata** tab displays the list of namespaces and the properties defined in the namespaces.
 3. Open the property Edit page by clicking the **Actions** box in the property row and then clicking **Edit** in the dropdown.
 4. In the Additional Class field, click  or type the beginning of a [Navigator entity classname](#).
 5. Select the class of entities to which the property applies. To clear the field, hover over the field and click the  that displays at the right of the field.
 6. In the Description field, add a description or edit an existing description.
 7. If the property is of the Enumeration type, you can add or remove values in the Enumeration field.



Note: If you delete an enumeration value, it cannot be used in new Managed Metadata assignments, but it continues to exist in entities or policies that are already using them.

8. For Text properties:

- In the Maximum Length field, add or change the value for the maximum length.

- In the Regular Expression field, edit the expression. Click [Show regex tester](#) to test input against any changes you make.

9. Click **Continue to Review**. The Review screen displays.

10. Click **Update** to commit the change or **Back to Edit Property** to continue editing the property, or **Cancel** to return to the Properties page.

Deleting, Restoring, and Purging Managed Metadata Properties

After a property is deleted, it cannot be used in a filter or in value assignments for entities. The deleted property is visible only from the Admin interface; it is labeled as Deleted in the Status column of the Properties table. In the following example, the `EmailFrom` property has been deleted.

[Purge Deleted Properties](#)
[New Property](#)

Name	Status	Type	Classes	Multivalued	Actions
MailAnnotation					
EmailFrom	Deleted	Text	HDFS Entity	No	Actions ▾
EmailTo	Active	Text	HDFS Entity	Yes	Actions ▾

For non-Admin users, a deleted property is hidden. However, the property still exists, and the values assigned to entities using this property are not affected until the deleted property is **purged**. Purging permanently removes the properties and any values from all entities. Any policies using these purged properties for metadata assignments are not modified by the purge and will fail the next time they run.

Because properties are not completely removed until purge, you cannot create a new property with the same name until the deleted property has been purged.

Deleting a Property

1. In the Properties table, for the property that you are deleting, click the **Actions** button, and then click **Delete** in the drop-down menu.
2. In the Delete Property dialog box, review the property deletion information. If any entities are affected, you see a **View affected entities** link; click to see all entities that use the property. Click **Confirm Delete** to delete the property, or click **Cancel**.

Restoring a Property

If you have not yet purged a deleted property, you can restore it.

- In the Properties table, for the property that you are restoring, click the **Actions** button, and then click **Restore** in the drop-down menu.

Purging a Property

You can permanently remove deleted properties by purging them. All values assigned to the deleted properties are lost; however, the affected entities are not deleted. Purging permanently removes all properties marked as Deleted in the Status column.



Note: Purging is a long-running task and may take some time. Navigator is unavailable to all users until the purge is completed.

1. In the Properties table, click **Purge Deleted Properties**. The Purge all Deleted Properties dialog box opens, describing the effects of the purge and reporting the number of entities that use the property.
2. In the Purge all Deleted Properties dialog box, click **Confirm Purge** to permanently remove all deleted properties, or click **Cancel** to return to the Properties page.

Navigator Built-in Classes

Class	Description
HDFS Dataset	Logical dataset backed by a path in HDFS.
HDFS Dataset Field	Field in an HDFS dataset.
HDFS Entity	HDFS file or directory.
Hive Column	Column in a Hive table.
Hive Database	Hive database.
Hive Partition	Partition of a Hive table.
Hive Query	Hive query template.
Hive Query Execution	Instance of a Hive query.
Hive Query Part	Component of a Hive query that maps specific input columns to output columns.
Hive Table	A Hive table.
Hive View	View on one or more Hive tables.
Impala Query	Impala query template.
Impala Query Execution	Instance of an Impala query.
Impala Query Part	Component of an Impala query that maps specific input columns to output columns.
Job Instance	Instance of a MapReduce, YARN, or Spark job.
Job Template	Template for a MapReduce, YARN, or Spark job.
Oozie Workflow	Template for an Oozie workflow.
Oozie Workflow Instance	Instance of an Oozie workflow.
Pig Field	Field for a relation in Pig; similar to a column in a Hive table.
Pig Operation	Template for a Pig transformation.
Pig Operation Execution	Instance of a Pig transformation.
Pig Relation	Pig relation; similar to a Hive table.
S3 Bucket	A bucket in S3.
S3 Object	A file or directory in an S3 bucket.
Sqoop Export Sub-operation	Sqoop export component that connects specific columns.
Sqoop Import Query	Sqoop import job with query options.
Sqoop Import Sub-operation	Sqoop import component that connects specific columns.
Sqoop Operation Execution	Instance of a Sqoop job.
Sqoop Table Export	Sqoop table export operation template.
Sqoop Table Import	Sqoop table import operation template.
User Sub-operation	User-specified sub-operation of a MapReduce or YARN job; used for specifying custom column-level lineage.

Defining Metadata with the Navigator API and Navigator SDK

In addition to defining metadata using features provided by the Navigator Metadata UI, you can also define metadata using the Navigator API and Navigator SDK.

For information on the Navigator API, see [Cloudera Navigator Data Management API](#).

For information on the SDK, see the [Navigator SDK documentation](#).

Adding and Editing Metadata

Minimum Required Role: [Metadata Administrator](#) (also provided by **Full Administrator**)

Cloudera Navigator supports adding metadata to extracted entities. You can add and edit two types of metadata:

- Custom metadata - Display name, description, tags, and key-value pairs. You can add and edit custom metadata using the Navigator UI, MapReduce service and job properties, HDFS metadata files, and the [Cloudera Navigator Data Management API](#). Custom metadata is typically implemented by end users who want to be able to classify and organize information for their own uses or to collaborate with other users.
- [Managed metadata](#). You can add and edit managed metadata using the Navigator UI and the API. Managed metadata is typically implemented for centralized curation of data sets.

Adding and Editing Metadata Using the Navigator UI

1. Run a [search](#) in the Navigator UI.
2. Click an entity link returned in the search. The Details tab displays.
3. To the left of the Details tab, click **Actions > Edit Metadata....** The Edit Metadata dialog box drops down.
4. Add metadata fields:

- In the Name field, type a new display name.
- In the Description field, type a description.
- **Managed Metadata**

1. Click the



and select a property.

2. Click the value field after the **:** to display type-specific selection controls such as integer spinners and date selection controls. Either type the value or use the controls to select a value.



to add another managed property key-value pair or another value for a given key.

- **Custom Metadata:** In the Tags field, type a tag and press **Enter** or **Tab** to create new tag entries.
- **Key-Value Pairs**

1. Click



to add a key-value pair.

2. Type a key and a value. You can specify special characters (for example, ".", " ") in the name, but it makes searching for the entity more difficult because some characters collide with special characters in the [search syntax](#).



Note: You cannot assign managed metadata in the Key-Value Pairs field because you cannot specify the namespace. All properties specified in the Key-Value Pairs field are treated as custom metadata.

5. Click **Save**. The new metadata appears in the Managed Metadata or Custom Metadata pane.

Custom Metadata Example

In the following example, the tag `archive_personal` and the property `year` with value `2015` have been added to the file `2015_11_20`:

Custom Metadata

Tags

archive_personal ×

Key-Value Pairs

year : 2015 - +

Cancel Save

After you save, the metadata appears in the Tags and Custom Metadata panes:

2015_11_20
/user/admin/2015_11_20

Actions ▾

Directory HDFS
Owner: admin
Parent: admin
Group: admin
Permissions: rwxrwxrwx

▼ Tags (1) Edit

archive_personal

▼ Custom Metadata Edit

year 2015

Managed Metadata Example

The following example shows the `Department` and `RetainUntil` managed properties for the `customers` entity:

Edit Metadata ✕

Name

Description

Managed Metadata

Department

:

Finance

RetainUntil

:

12/31/2017, 12:00 AM

After you specify the values and save, the properties display in the Managed Metadata pane:

account
Actions ▾

hdfs://Enchilada/data/gfcharan/work/hive/l4_amlharan/acco...
Details
Lineage

Table
Hive

Owner:
gfcharan

Parent:
l4_amlharan

43

Columns

21

Partitions

16

Inputs

0

Outputs

▼ Managed Metadata Edit

realestate

Department	Finance
AccessAttempts	10

▼ Columns (43) 🔍

- ▶ accommodation_account_flag string
- ▶ account_booking_country string
- ▶ account_closing_date string
- ▶ account_currency string

Editing MapReduce Custom Metadata

You can associate custom metadata with arbitrary configuration parameters to MapReduce jobs and job executions. The 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**.
 - On 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` - A comma-separated list of user-defined property names.

- `nav.tags` - A 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` - A comma-separated list of user-defined property names
 - `nav.job.tags` - A comma-separated list of property names that serve as tags
- Job execution properties (`type:OPERATION_EXECUTION`)
 - `nav.jobexec.user_defined_properties` - A comma-separated list of user-defined property names
 - `nav.jobexec.tags` - A 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 key `key` with the value `value`:

1. Dynamically add the user-defined key `key`:

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

2. Set the value of the user-defined key `key` to `value`:

```
conf.set("key", "value")
```

Editing HDFS Custom Metadata Using Metadata Files

You can add tags and properties to HDFS entities using metadata files. With metadata files, you can assign metadata to entities in bulk and create metadata before it 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.

Editing HDFS and Hive Metadata Using the Navigator Metadata API

You can use the [Cloudera Navigator Data Management API](#) to modify the custom metadata of HDFS or Hive entities, whether the entities have been extracted or not. If an entity has been extracted when the API is called, the metadata is 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 custom metadata is stored with the entity's:

- Identity
- Source ID
- Metadata fields (name, description, tags, properties)
- Fields relevant to the identifier

The rest of the entity fields (such as type) are not present. To view all stored metadata, use the API to search for entities without an internal type:

```
curl http://Navigator_Metadata_Server_host:port/api/v10/entities/?query=-internalType:*
-u username:password -X GET
```

Custom metadata provided through the API overwrites existing metadata. For example, if 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 omit the tags or properties fields, the existing values remain unchanged.

Modifying custom 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 work 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. Some inconsistency can occur if a metadata file is merged when 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. Use the following syntax:

- PUT

```
curl http://Navigator_Metadata_Server_host:port/api/v10/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 - Custom metadata properties. The format is {key: value}.
- customProperties - Managed metadata properties. The format is {namespace: {key: value}}. If a property is assigned a value that does not conform to type constraints, an error is returned.

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

- POST

```
curl http://Navigator_Metadata_Server_host:port/api/v10/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/v10/entities/?query=type:SOURCE -u
username:password -X GET
```

For example:

```
[ ...
{ {
  "identity": "61cfefd303d4284b7f5014b701f2c76d",
  "originalName": "source.listing",
  "originalDescription": null,
  "sourceId": "012437f9eeb3c23dc69e679ac94a7fa2",
  "firstClassParentId": null,
  "parentPath": "/user/hdfs/.cm/distcp/2016-02-03_487",
  ...
  "properties": {
    "__cloudera_internal_hueLink":
    "http://vc0142.halxg.cloudera.com:8888/filebrowser/#/user/hdfs/.cm/distcp/2016-02-03_487/source.listing"
  },
  "technicalProperties": null,
  "filePath": "/user/hdfs/.cm/distcp/2016-02-03_487/source.listing",
  "type": "FILE",
  "size": 92682,
  "created": "2016-02-03T21:12:16.587Z",
  "lastModified": "2016-02-03T21:12:16.587Z",
  "lastAccessed": "2016-02-03T21:12:16.587Z",
  "permissions": "rw-r--r--",
  "owner": "hdfs",
  "group": "supergroup",
  "blockSize": 134217728,
  "mimeType": "application/octet-stream",
  "replication": 3,
  "userEntity": false,
  "deleted": false,
  "sourceType": "HDFS",
  "metaClassName": "fselement",
  "packageName": "nav",
  "internalType": "fselement"
}, ...
]
```

If you have multiple services of a given type, you must specify the source ID that contains the entity you expect 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 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 or folders in that directory.
 - Hive database - If you are updating database metadata, 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` - Custom metadata properties. The format is `{key: value}`.

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

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

```
curl
http://Navigator_Metadata_Server_host:port/api/v10/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 Custom Metadata Example for `/user/admin/input_dir` Directory

```
curl http://Navigator_Metadata_Server_host:port/api/v10/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 Custom Metadata Example for `total_emp` Column

```
curl http://Navigator_Metadata_Server_host:port/api/v10/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"}}'
```

HDFS PUT Managed Metadata Example

The following example demonstrates how to set two properties in the MailAnnotation namespace: a multivalued property emailTo and a single-valued property emailFrom:

```
curl
http://Navigator_Metadata_Server_host:port/api/v10/entities/87afcb92d5de856c7e8292e2e12cf1ea
-u admin:admin -X PUT -H "Content-Type: application/json"
-d
'{"customProperties":{"MailAnnotation":{"emailTo":["lee@example.com","dana@example.com"],"emailFrom":"terry@email.com"}}}'
```

The response is:

```
{
  "identity" : "87afcb92d5de856c7e8292e2e12cf1ea",
  "originalName" : "years",
  "originalDescription" : null,
  "sourceId" : "012437f9eeb3c23dc69e679ac94a7fa2",
  "firstClassParentId" : null,
  "parentPath" : "/user/admin",
  "extractorRunId" : "012437f9eeb3c23dc69e679ac94a7fa2##1",
  "customProperties" : {
    "MailAnnotation" : {
      "emailTo" : [ "lee@example.com", "dana@example.com" ],
      "emailFrom" : "terry@email.com"
    }
  },
  "name" : null,
  "description" : null,
  "tags" : null,
  "properties" : {
    "__cloudera_internal_hueLink" : "Hue_Server_host:8888/filebrowser/#/user/admin/years"
  },
  "technicalProperties" : null,
  "filePath" : "/user/admin/years",
  "type" : "DIRECTORY",
  "size" : null,
  "created" : "2016-03-22T17:55:31.902Z",
  "lastModified" : "2016-03-22T17:59:14.065Z",
  "lastAccessed" : null,
  "permissions" : "rwxr-xr-x",
  "owner" : "hdfs",
  "group" : "admin",
  "blockSize" : null,
  "mimeType" : null,
  "replication" : null,
  "sourceType" : "HDFS",
  "metaClassName" : "fselement",
  "userEntity" : false,
  "deleted" : false,
  "packageName" : "nav",
  "internalType" : "fselement"
}
```

Accessing and Editing Metadata with the Navigator SDK

In addition to the metadata features provided by the Cloudera Navigator data management component UI, you can also access and edit metadata using the Navigator SDK. For information on the SDK, see the [Navigator SDK documentation](#).

Performing Actions on Entities

Minimum Required Role: [Policy Administrator](#) (also provided by [Full Administrator](#))

Moving an HDFS Entity and Moving an HDFS Entity to Trash

You can move an HDFS entity to another location, and to [HDFS trash](#). 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 in to 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 in to the Cloudera Navigator data management component UI](#).
2. In the top right, click **username > Command 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](#) component is running on the cluster view some entities using Hue and related tools, as follows:

File Browser	HDFS directories and files
Hive Metastore Manager (HMS)	Hive database and tables
Job Browser	YARN, Oozie

1. Access the Cloudera Navigator console using the correct host name and port for your instance:

```
http://fqdn-1.example.com:7187/login.html
```

The login prompt displays.

2. Log in to the Cloudera Navigator console using the [credentials](#) assigned by your administrator.
3. Run a [search](#) in the Navigator UI.
4. 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 Provenance Use Case

A number of business decisions and transactions rely on the verifiability of the data used in those decisions and transactions. Data-verification questions might include:

- How was this mortgage credit score computed?
- How can I prove that this number on a sales report is correct?
- What data sources were used in this calculation?

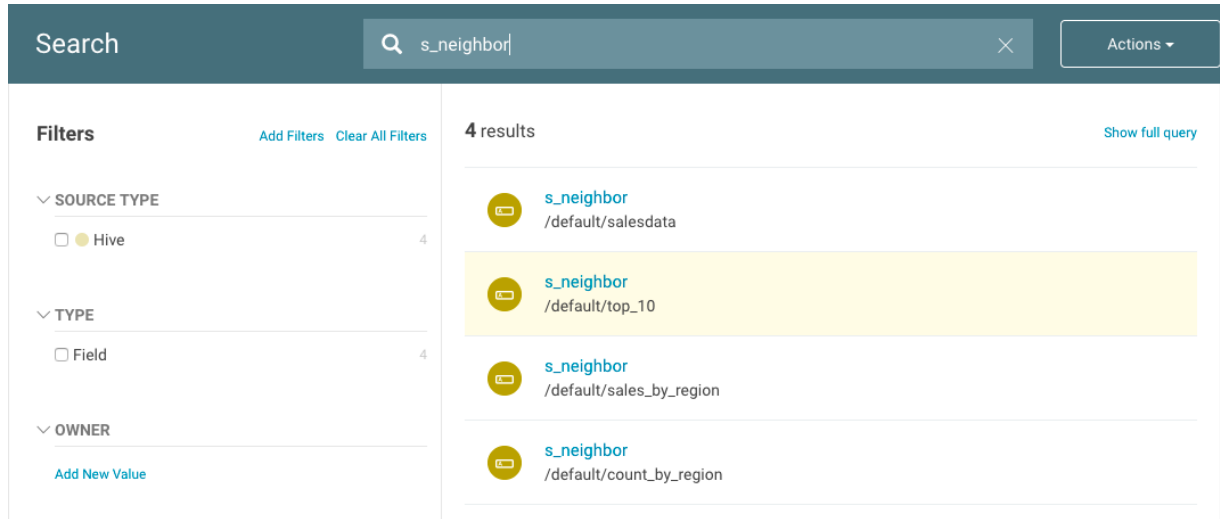
You can use Cloudera Navigator to answer these and other questions about your data. Using metadata and lineage, you can get track the life of the data to verify its *provenance*—that is, determine its origin.

How Can I Verify a Value in a Table?

A number of business transactions require you to verify that information is correct and that it is derived from a reliable source. For example, if you work in a sales organization, you might verify that information in sales reports is accurate, that you can trust the contents, and that you can identify the origin of the information.

The following example shows how you can verify information in a field named **s_neighbor** by tracing it to its source. You will replace the fields and other information in this example with the actual information that you want to verify.

1. [Log into the Cloudera Navigator data management UI](#) and click the **Search** tab.
2. Type **s_neighbor** in the search box.



You see four instances of the **s_neighbor** field.

3. View details of the field in the **top_10** table by clicking **s_neighbor** in the entry with the Parent Path **/default/top10**.

s_neighbor

Field
Hive

Parent: top_10
Data Type: string

1 Inputs
0 Outputs

Tags (1) Edit

- sensitive

Technical Metadata

Source Type	Hive
Type	Field
Parent	top_10
Data Type	string
Parent Path	/default/top_10
Source	HIVE-1
Classname	Hive Column
Package Name	nav

Inputs (1) Search

- salesdata

Outputs (0)

You see that the parent table is **top_10**, and the input or upstream source of the data is the **salesdata** database.

Where did **salesdata** come from originally? It was imported using sqoop, with syntax similar to the following; actual arguments vary:

```
> sqoop import-all-tables \
  -m {{cluster_data.worker_node_hostname.length}} \
  --connect jdbc:mysql://{{cluster_data.manager_node_hostname}}:3306/retail_db \
  --username=admin \
  --password=password \
  --compression-codec=snappy \
  --as-parquetfile \
  --warehouse-dir=/user/hive/warehouse \
  --hive-import
```

4. To see a graphical representation of the relationships among the entities:

- a. Click the **Lineage** tab.
- b. In Lineage Options, select **Operations** and clear any other check boxes.

s_neighbor

Details Lineage

Lineage Options

- Operations
- Control Flow Relations
- Only Upstream Downstream
- Latest Partition and Operation Execution
- Deleted Entities

Search

Click an entity or link to view more information.

Lineage Diagram:

```
graph LR; salesdata[salesdata] --> create_table[create table top_10 as select s_neighbor, avg(s_pri...)]; create_table --> top_10[top_10]; top_10 --> s_neighbor[s_neighbor]
```

See that **s_neighbor** can be traced back to the original table **salesdata**.

5. Click the operation entity in the center of the lineage diagram, and see details about it on the lower right side of the lineage window.

SELECTED ENTITY
**create table top_10 as select s_neighbor,
avg(s_pri...**

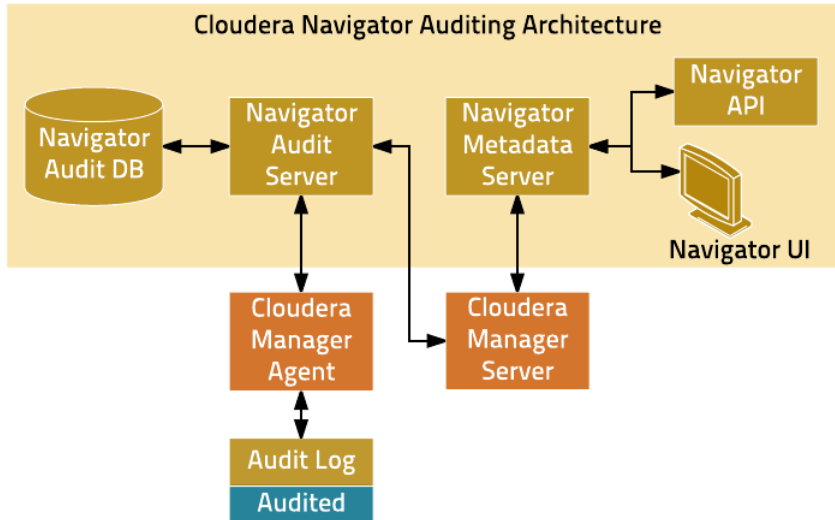
[View Details](#) [View Lineage](#)

Source Type	Impala
Type	Operation
Query Text	create table top_10 ... ⓘ
Source	IMPALA-1
Classname	Impala Query
Package Name	nav

Information about the selected entity indicates that the operation is an Impala query. Click the information icon on the Query Text line to see the entire query. This query was used to derive **top_10** from the original table.

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

Cloudera Navigator Auditing

Minimum Required Role: [Auditing Viewer](#) (also provided by **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 in to 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 press the up and down arrow keys.
 - Time - Click the hour, minute, and AM/PM fields and click the spinner arrows or press the 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.

Verifying that Auditing is Running

Navigator auditing captures a complete and immutable record of all activity within a system. Depending on how you use this information, you may want to know as soon as possible if auditing is not running. To monitor auditing and ensure that it has not failed without generating a warning message, you can configure an auditing pipeline health check to verify that auditing is working and is not silently down. The health check uses the following metrics to determine if auditing is running:

- Number of bytes of audits processed.
- Number of bytes of audits remaining.
- Number of errors when sending audits from the Cloudera Manager Agent to the Cloudera Manager server.

The audit generates a warning in Cloudera Manager if:

- The number of bytes of audits processed is zero.
- The number of bytes of audits remaining is not zero.
- Errors occur when audits are sent.

The health check is run for each role that can generate audits.

Configuring the Audit Pipeline Health Check

Minimum Required Role: [Navigator Administrator](#) (also provided by **Full Administrator**)

In Cloudera Manager, you configure the audit pipeline health check as follows:

1. Go to the service you want to configure.
2. Click the **Configuration** tab.
3. Search for **mgmt.navigator**.
4. Edit the following configuration items:
 - **Navigator Audit Pipeline Health Check** - Select or deselect the check box to enable the audit health check. You can enable the health check for specific groups. By default, all groups are selected.
 - **Monitoring Period for Audit Failures** - Set the period of time that elapses before a failure warning is sent. The default time period is 20 minutes.
 - **Navigator Audit Failure Thresholds** - Set the size of the audit failure, in bytes, that triggers a Warning or Critical error message. The value that you specify for this threshold is the number of bytes of audit data that have not been sent to audit server. You can specify different thresholds for Warning and Critical errors. By default, Critical errors are sent for failures of any size.
5. Click **Save Changes**.

For example, in the following graphic, the pipeline health check is enabled for all groups in the service. The failure period is set to 15 minutes, and the health check sends a warning for failures of any size and a critical error when more than 2 KiB of audit events have not been sent.

Show All Descriptions

Navigator Audit Pipeline Health Check DataNode Default Group ...and 6 others

mgmt.navigator.status.check.enabled

[Edit Individual Values](#)

Monitoring Period For Audit Failures DataNode Default Group ...and 6 others C

mgmt.navigator.failure.window

[Edit Individual Values](#)

Navigator Audit Failure Thresholds DataNode Default Group ...and 6 others C

mgmt.navigator.failure.thresholds

Warning:

Critical:

[Edit Individual Values](#)

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.
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.

Display Name	Field	Description
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, DESCTABLE, DESCFUNCTION, 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, NAVIGATOR_ADD_TAG, NAVIGATOR_DELETE_TAG • 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.

Display Name	Field	Description
		<ul style="list-style-type: none"> • 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	<p>Subtype of operation performed in Navigator Metadata Server. Valid values are:</p> <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 • 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

Minimum Required Role: [Auditing Viewer](#) (also provided by **Full Administrator**)

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

Creating Audit Event Reports

1. [Start and log in to 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 by clicking **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 in to 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 Navigator UI or by 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 54.

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 in to 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 use the API to download the audits events for a service named `hive`, issue the request

```
curl
http://Navigator_Metadata_Server_host:port/api/v10/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** > **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 Auditing Use Cases

The Navigator Audit Server tracks the actions performed on the data in a Hadoop cluster. By applying filters on these actions, you can use Cloudera Navigator auditing to view specific information and answer a variety of questions about data and user actions; for example:

- What was a specific user doing on a specific day?
- Who deleted a particular directory?
- What happened to data in a production database, and why is it no longer available?

To answer these questions using Navigator auditing, you begin by [logging into the Cloudera Navigator data management UI](#) and clicking the **Audits** tab. Cloudera Navigator displays a list of all audit events for the last hour. The following use cases describe how Navigator can answer some specific questions about data and users.

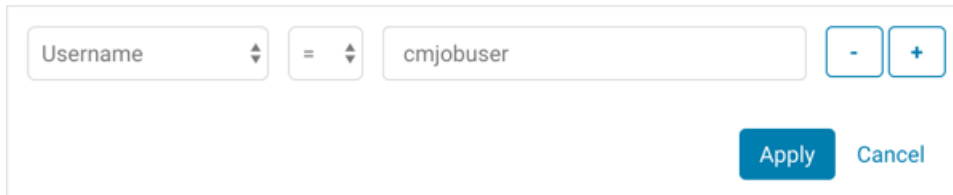
What Did a User Do on a Specific Day?

In some cases, you may want to identify actions that a specific user performed during a period of time. To determine a user's actions for a time period, you use filters to first specify the user and then define the time period.

The following example identifies the actions of the user named **navigator_user** on June 9, 2016:

1. Filter the list of events for a specific user:

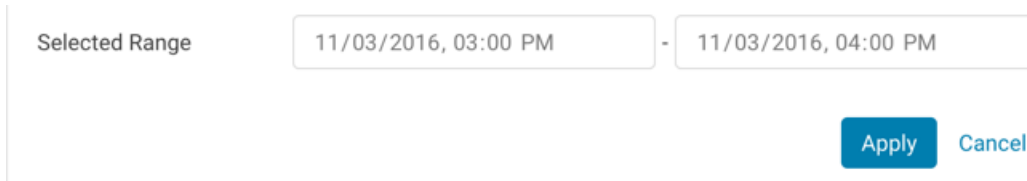
- a. Click **Filters**.
- b. Select **Select Property... > Username**.
- c. In the field to the right of =, type the username and click **Apply**. The username filter is added to the list of filters, and the list of events is filtered and reloaded. This filter specifies the user **cmjobuser**.



The screenshot shows a filter configuration interface. On the left, there is a dropdown menu with 'Username' selected. To its right is an equals sign (=) followed by another dropdown menu. The text 'cmjobuser' is entered into the input field to the right of the equals sign. To the right of the input field are two small buttons: a minus sign (-) and a plus sign (+). Below the input field are two buttons: 'Apply' (highlighted in blue) and 'Cancel'.

2. Filter the list of events for a specific date and time:

- a. Click the date-time field at the top right of the Audit Events page. A set of links display with relative time periods (**Last hour**, **Last 2 hours**, and so on) and a **Custom Range** link that you can use to specify an absolute time range. The Selected Range field displays the currently selected range, which by default is the last hour of the current day.
- b. To choose a specific day, click **Custom Range**. The Selected Range field is enabled for input.
- c. Use the field controls to choose specific dates and times. The following figure shows the selections for November 3, 2016, 3:00 PM to November 3, 2016, 4:00 PM.



The screenshot shows a 'Selected Range' filter configuration. The label 'Selected Range' is on the left. To its right are two input fields separated by a minus sign (-). The first input field contains the text '11/03/2016, 03:00 PM' and the second contains '11/03/2016, 04:00 PM'. Below the input fields are two buttons: 'Apply' (highlighted in blue) and 'Cancel'.

- d. Click **Apply**.

The following figure shows the first page of the filter results: audit events for the user **cmjobuser** during the 24 hour period from June 9, 2016 12:00 a.m. to June 10, 2016 12:00 a.m.

Audit Events

Actions ▾

> FILTERS

> NOV 3, 2016 3:00 PM - NOV 3, 2016 4:00 PM

> Timestamp	Username	IP Address	Service N...	Operation	Resource
> Nov 3, 2016 3:06:54.595 PM	cmjobuser	172.31.8.56	HDFS-1	delete	/user/cmjobuser/.staging/job_1474...
> Nov 3, 2016 3:06:53.565 PM	cmjobuser	172.31.8.56	HDFS-1	rename	/user/history/done_intermediate/c...
> Nov 3, 2016 3:06:53.563 PM	cmjobuser	172.31.8.56	HDFS-1	rename	/user/history/done_intermediate/c...
> Nov 3, 2016 3:06:53.561 PM	cmjobuser	172.31.8.56	HDFS-1	rename	/user/history/done_intermediate/c...
> Nov 3, 2016 3:06:53.559 PM	cmjobuser	172.31.8.56	HDFS-1	setPermi..	/user/history/done_intermediate/c...
> Nov 3, 2016 3:06:53.539 PM	cmjobuser	172.31.8.56	HDFS-1	create	/user/history/done_intermediate/c...
> Nov 3, 2016 3:06:53.536 PM	cmjobuser	172.31.8.56	HDFS-1	open	/user/cmjobuser/.staging/job_1474...
> Nov 3, 2016 3:06:53.534 PM	cmjobuser	172.31.8.56	HDFS-1	getFileinfo	/user/history/done_intermediate/c...
> Nov 3, 2016 3:06:53.532 PM	cmjobuser	172.31.8.56	HDFS-1	getFileinfo	/user/cmjobuser/.staging/job_1474...

Who Deleted Files from Hive Warehouse Directory?

The Hive warehouse directory is usually set to `/user/hive/warehouse`. In this example, files have been deleted from the directory and you want to identify who removed them.

To determine who deleted files from this directory, use filters in Cloudera Navigator to do the following:

1. Filter the list of events for the source `/user/hive/warehouse`:
 - a. Click **Filters**.
 - b. Select **Select Property...** > **Source**.
 - c. In the operator field, select **like**.
 - d. In the empty field to the right of **like**, type `/user/hive/warehouse` and click **Apply**. The source filter is added to the list of filters and the list of events is filtered and reloaded.
2. Filter the list of events for the delete operation:
 - a. Click **Add New Filter**.
 - b. Select **Select Property...** > **Operation**.
 - c. In the operator field, select **=**.
 - d. In the empty field to the right of **=**, type `delete` and click **Apply**. The operation filter is added to the list of filters and the list of events is filtered and reloaded.

The following figure shows the resulting filters.

Filters ▾

The screenshot shows the filter configuration interface with two rules:

- Rule 1: Source **like** `/user/hive/warehouse`
- Rule 2: Operation **=** `delete`

The following figure shows the results of the filters: **navigator_user** deleted or attempted to delete (indicated by the red text) the displayed resources from the Hive warehouse directory during the 30-day period from May 28, 2016 to June 27, 2016.

Audit Events						Save As Report
Filters ▾		Source like /user/hive/warehouse x		Operation = delete x		May 28 2016 12:18 PM - Jun 27 2016 12:18 PM ▾
Export ▾						< 1 - 5 >
Timestamp	Username	IP Address	Service Name	Operation	Resource	
Jun 9 2016 7:46 PM	navigator_user	10.17.207.26	hdfs	delete	/user/hive/warehouse/sample_09/hive-staging_hive_2016-06-09	
Jun 9 2016 1:39 PM	navigator_user	10.17.207.26	hdfs	delete	/user/hive/warehouse/sample_09/hive-staging_hive_2016-06-09	
Jun 9 2016 1:39 PM	navigator_user	10.17.207.26	hdfs	delete	/user/hive/warehouse/sample_09/hive-staging_hive_2016-06-09	
Jun 9 2016 1:19 PM	navigator_user	10.17.207.26	hdfs	delete	/user/hive/warehouse/sample_09/000000_0	
Jun 9 2016 1:19 PM	navigator_user ▾	10.17.207.26 ▾	hdfs ▾	delete ▾	/user/hive/warehouse/sample_09/hive-staging_hive_2016-06-09	

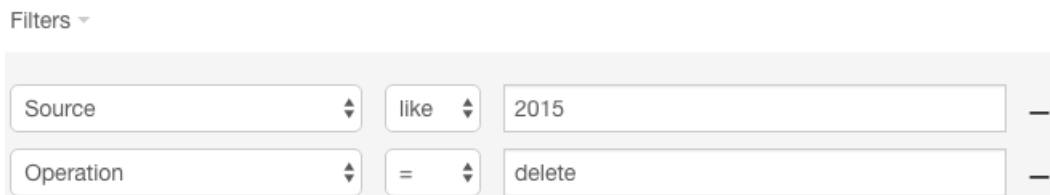
What Happened to Data in the Database?

Typically, data in the database is partitioned into folders or files labeled by date. In this example, data from 2015 is missing from the production database, and you want to find out what happened to it. You can use Cloudera Navigator to determine what happened to data that was created during this period of time.

Data created in 2015 has the string "2015" in the filename. To determine what happened to the data stored in folders and files in the year 2015, do the following:

1. Filter the list of events for sources containing the string "2015":
 - a. Click **Filters**.
 - b. Select **Select Property...** > **Source** to specify the path of an HDFS file or directory.
 - c. In the operator field, select **like**.
 - d. In the empty field to the right of **like**, type **2015** and click **Apply**. The source filter is added to the list of filters, and the list of events is filtered and reloaded.
2. Filter the list of events for the delete operation:
 - a. Click **Add New Filter**.
 - b. Select **Select Property...** > **Operation**.
 - c. In the operator field, select **=**.
 - d. In the empty field to the right of **=**, type **delete** and click **Apply**. The operation filter is added to the list of filters and the list of events is filtered and reloaded.
3. Set the date range to one year:
 - a. Click the date-time field at the top right of the Audit Events page.
 - b. To set the range to be the last year, click **Custom Range**. The Selected Range field is enabled for input.
 - c. In the left date field, use the field controls to specify a date one year ago.
 - d. Click **Apply**.

The following figure shows the resulting filters.



The following figure shows the results of the filter application. During the last year, the user **hdfs** deleted the directories with names that contain "2015":

Audit Events

[Save As Report](#)

Filters ▾ Source like 2015 ✕ Operation = delete ✕

Jun 13 2015 12:36 PM - Jun 13 2016 12:36 PM ▾

⚙ Export ▾ < 1 - 4 >

▶ Timestamp	Username	IP Address	Service Name	Operation	Resource
▶ Jun 13 2016 11:45 AM	hdfs	10.17.207.26	hdfs	delete	/user/navigator_user/2015_11_21
▶ Jun 13 2016 11:45 AM	hdfs	10.17.207.26	hdfs	delete	/user/navigator_user/2015_11_20
▶ Jun 13 2016 11:45 AM	hdfs	10.17.207.26	hdfs	delete	/user/navigator_user/2015_11_19
▶ Jun 13 2016 11:45 AM	hdfs ▾	10.17.207.26 ▾	hdfs ▾	delete ▾	/user/navigator_user/2015_11_18

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

Minimum Required Role: [Lineage Viewer](#) and [Policy Administrator](#) (also provided by **Full Administrator**)

1. [Start and log in to the Cloudera Navigator data management component UI.](#)
2. Click the **Analytics** tab. The Metadata analytics tab displays.
3. Click the `Source` button and select an HDFS service instance from the 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 clear a property, clear 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**.
 - **Minimum Required Role:** [Policy Administrator](#) (also provided by **Full 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

Minimum Required Role: [Auditing Viewer](#) (also provided by **Full Administrator**)

1. [Start and log in to 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. Click the **Source** button and select an HDFS service instance from the drop-down list.
4. If not already displayed, click the **Audit** 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 clear Activity, clear 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**.
 - **Minimum Required Role:** [Lineage Administrator](#) (also provided by **Metadata Administrator**, **Full 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**.
 - **Minimum Required Role:** [Metadata Viewer](#) (also provided by **Metadata Administrator**, **Full Administrator**)
 - Minimum Required Role:** [Lineage Administrator](#) (also provided by **Metadata 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 once and match the string `staging`. Each directory has six files that has been accessed.

The screenshot displays the Cloudera Navigator Analytics interface for HDFS-1. The top navigation bar includes 'HDFS-1 Analytics', a 'Source' dropdown, and tabs for 'Metadata' and 'Audit'. Below this, there are buttons for 'Activity' and 'Top Users'. The main content area is divided into two sections:

Activity Chart: A bar chart titled 'Activity' with a checked checkbox. The y-axis represents the number of files read, ranging from 0 to 1,008. The x-axis is labeled 'No. of times files have been read' with a value of 1. A single blue bar is shown at x=1 with a height of 1,008. A blue selection outline is drawn around the bar.

HDFS Files Table: A table titled 'HDFS Files' with a search box containing 'staging' and an 'Actions' dropdown. The table lists four directories, each with a checkbox and a 'Number of Files' column showing 6 files.

Directory	Number of Files
<input type="checkbox"/> /user/cmjobuser/.staging/job_1474062788287_1053	6
<input type="checkbox"/> /user/cmjobuser/.staging/job_1474062788287_1048	6
<input type="checkbox"/> /user/cmjobuser/.staging/job_1474062788287_1087	6
<input type="checkbox"/> /user/cmjobuser/.staging/job_1474062788287_1049	6

Below the activity chart, a note states: "The data in the Activity chart is summarized and computed once a day."

Metadata Policies

A **metadata policy** defines a set of actions performed by the Cloudera Navigator Metadata Server on a class of entities. You can perform the following actions:

- Add multi-valued managed metadata of all types (Boolean, date, enumeration, number, and text), and custom metadata such as tags and key-value pairs.
- Run a 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 does not have access to the directory, the action fails. Similarly, if a policy creator does not have access to a file in the directory, the action fails. To ensure that command actions do not 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.

- Move an entity to a target path or to trash.
- Send 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"}
```

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

For some actions, you can specify 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. 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, and then delete or disable the old policy.

Creating Policies

Minimum Required Role: [Policy Administrator](#) (also provided by **Full Administrator**)

1. [Start and log in to 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 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 **Test Query** link.
6. Specify an optional description for the policy.
7. If you use policy expressions in properties that support expressions, specify required imports in the **Import Statements** field. See [Metadata Policy Expression Examples](#) on page 69.
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:

- **Date** - Click the down arrow to display a calendar and select a date, or click a field and click the spinner arrows or press the up and down arrow keys.
- **Time** - Click the hour, minute, and AM/PM fields and click the spinner arrows or press the 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:

- **Metadata Assignments:** Specify the custom [metadata or managed metadata](#) to be assigned. Optionally, you can specify a Java [policy expression](#) for fields that support expressions by checking the **Expression** checkbox. The following fields support expressions:

- Name
- Description
- Managed Metadata
- Key-Value Pairs

- **Command Actions:** Select **Add Action > Move to Trash** 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.

Command actions are supported only for HDFS entities. If you configure a command action for unsupported entities, a runtime error is logged when the policy runs.

See [Viewing Command Action Status](#) on page 47.

- **JMS Notifications:** If not already configured, [configure a JMS server and queue](#). Specify the queue name and message. Optionally, check the **Expression** checkbox and specify a policy expression for the message.

10. Click **Save**.

Viewing Policies

Minimum Required Role: [Policy Viewer](#) (also provided by [Policy Administrator](#), [Full Administrator](#))

1. [Start and log in to 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**. The policy detail page is displayed.

You can also [edit](#), [copy](#), or [delete](#) a policy from the policy details page by clicking the **Actions** button.

The screenshot displays the 'Policies' section of the Cloudera Navigator UI. A specific policy, 'hdfsImmediatePolicy', is selected. The policy details are as follows:

- Status:** Enabled (indicated by a green checkmark)
- Search Query:** fileSystemPath:"/tmp/policy_hdfs_data/testfile1" AND sourceType:hdfs AND deleted:false
- Policy Description:** (The description text is not visible in the image)
- Last Run On:** Friday, December 9th 2016, 6:47 am

An 'Actions' dropdown menu is open, showing three options: Edit, Copy, and Delete.

Enabling and Disabling Policies

As a policy administrator, you can manage access to policies by enabling and disabling them.

Minimum Required Role: [Policy Administrator](#) (also provided by [Full Administrator](#))

Metadata Policies

1. [Start and log in to 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**.

Copying and Editing a Policy

If you have an existing policy that you want to use as a template for another similar property, you can copy it and then make any required adjustments. You can also edit existing policies if you need to make changes to it.

Minimum Required Role: [Policy Administrator](#) (also provided by **Full Administrator**)

1. [Start and log in to the Cloudera Navigator data management component UI.](#)
2. Click the **Policies** tab.
3. In a policy row, select **Actions > Copy** or **Actions > Edit**. You can also click the policy row and then on the policy details page, select **Actions > Copy** or **Actions > Edit**.
4. Edit the policy name, search query, or policy actions.
5. Click **Save**.

Deleting Policies

Minimum Required Role: [Policy Administrator](#) (also provided by **Full Administrator**)

1. [Start and log in to 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, managed metadata, 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.

In the Navigator UI, you see an Expression check box under or next to elements for which you can define an expression, as well as a pop-up that you can open to see an expression example:



You can define expressions for the following when you create a policy:

- Metadata Assignments
 - Name
 - Description
 - Managed Metadata
 - Key-Value Pairs
- JMS Notification Messages

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 69.

If you do not 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


















Minimum Required Role: [Lineage Administrator](#) (also provided by **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 400 entities. Once that limit is reached, certain entities display as a "hidden" icon.

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 field • Pig operation, operation execution 	<ul style="list-style-type: none"> •  •  • 
Hive and Impala		Spark (Supported in CDH 5.11 and higher.) Spark Lineage information is produced only for data that is read/written and processed using the Dataframe and SparkSQL APIs. Lineage is not available for data that is read/written or processed using Spark's RDD APIs. To turn metadata extraction off or on, see Enabling and Disabling Metadata Extraction.)	
<ul style="list-style-type: none"> • Table • Field • Operation, suboperation, execution • Impala operation, suboperation, execution 	<ul style="list-style-type: none"> •  •  •  •  	<ul style="list-style-type: none"> • Operation, operation execution. (Spark RDDs and aggregation operations are not included in the diagrams.) 	<ul style="list-style-type: none"> • 
MapReduce and YARN		Sqoop	
<ul style="list-style-type: none"> • MapReduce operation and operation execution • YARN operation and operation execution 	<ul style="list-style-type: none"> •  •  	<ul style="list-style-type: none"> • Operation, suboperation, execution 	<ul style="list-style-type: none"> • 
Oozie		S3	
<ul style="list-style-type: none"> • Operation, operation execution 	<ul style="list-style-type: none"> •  	<ul style="list-style-type: none"> • Directory • File • S3 Bucket 	<ul style="list-style-type: none"> •  •  • 

Hidden			
<ul style="list-style-type: none">  	See Viewing the Lineage of Hidden Entities on page 79.		

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

In the following circumstances, the entity type icon appears as



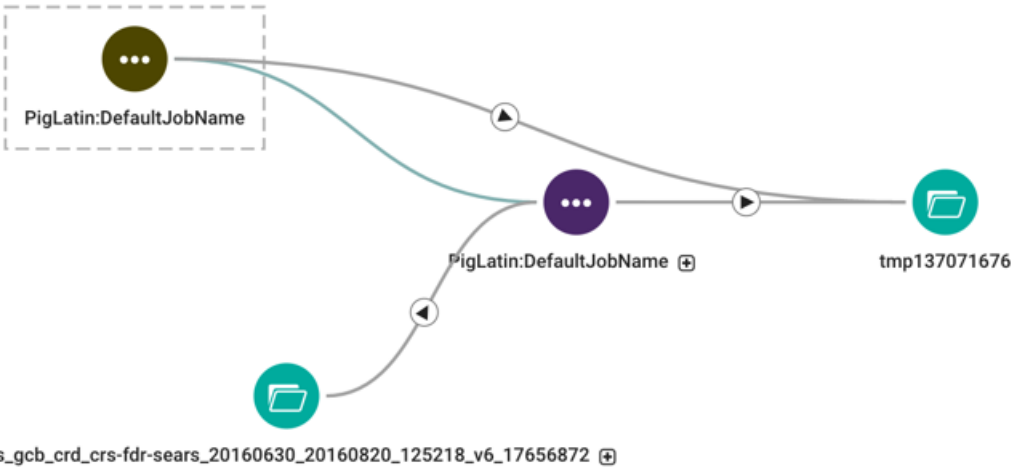
- The entity has not yet been extracted. In this case,



is eventually replaced with the correct entity icon after the entity is extracted and linked in Navigator. For information on how long it takes for newly created entities to be extracted, see [Metadata Extraction](#) on page 22.

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

The following lineage diagram illustrates the relations between the YARN operation `DefaultJobName` and Pig script `DefaultJobName` and the source file in the `ord_us_gcb_crd_crs-fdr-sears` folder and destination folder `tmp137071676`:







Relations

Relations between the entities are represented graphically by lines, with arrows indicating the direction of the data flow. Navigator supports 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.
Parent-child	Describes a parent-child relation. For example, between a directory and a file.

Relation Type	Description
Logical-physical	Describes the relation between a logical entity and its physical entity. For example, between a Hive query and a MapReduce job.
Instance of	Describes the relation between a template and its instance. For example, an operation execution is an instance of operation. Instance of relations are never visualized in the lineage, however you can navigate between template and instance lineage diagrams. See Displaying an Instance Lineage Diagram on page 85 and Displaying the Template Lineage Diagram for an Instance Lineage Diagram on page 85.
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 contain the following line types:


- Solid () represents a "data flow" relationship, indicating that the columns appear (possibly transformed) in the output (when directional with arrow) and "logical- physical" (when no arrow). For example, a solid line appears between the columns used in a `select` clause.
- Dashed () represents a "control flow" relationship, indicating that the columns determine which rows flow to the output. For example, a dashed line appears between the columns used in an `insert` or `select` clause and the `where` clause of a Hive query. Control flow lines are hidden by default. See [Filtering Lineage Diagrams](#) on page 80.
- Blue () represents a selected link.
- Green () represents a summary link that contains operations. When you click the link, the link turns blue (for selected) and the nested operations display in the selected link summary:

SELECTED LINK
Data Flow Summary

 sample_07

Operations:

 INSERT OVERWRITE TABLE sample_09
 SELECT sample_07.co...

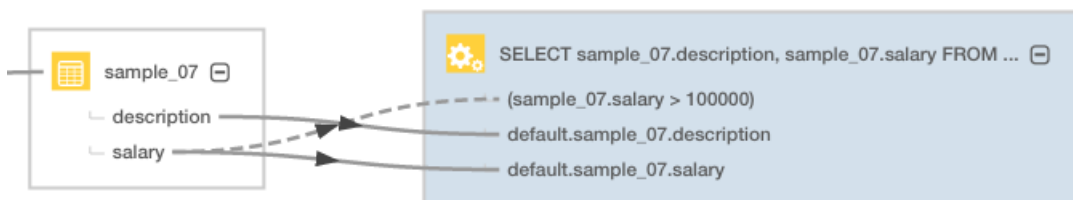
 sample_09

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
```

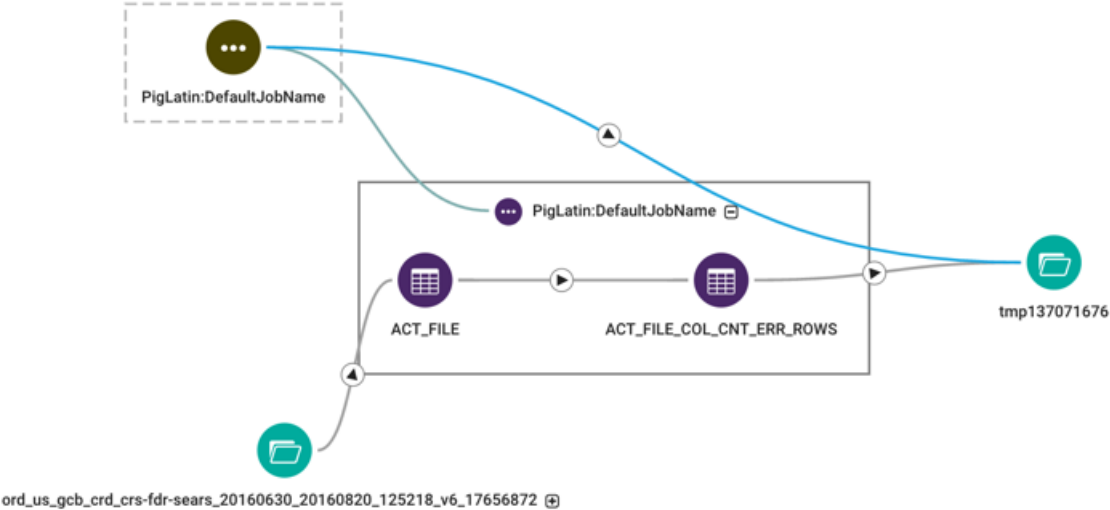
has solid, directed lines between the columns in the `select` clause and a dashed line between the columns in the `where` clause:



Manipulating Lineage Diagrams

Expanding Entities

You can click a  icon in a parent entity to display its child entities. For example, you can click an Oozie job to display its child Pig script and the Pig script to display its child tables:



Modifying Lineage Layout

- To improve the layout of a lineage diagram, you can drag entities (like tmp137071676) located outside a parent box.
- Use the mouse scroll wheel or the



control to zoom the lineage diagram in and out.

- You can move an entire lineage diagram in the lineage pane by pressing the mouse button and dragging it.

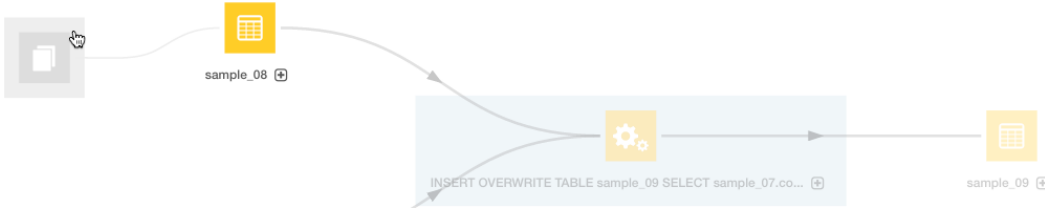
Viewing the Lineage of Hidden Entities

Lineage that is not fully traversed (that is, you do not see a subset of the actual lineage) is illustrated by the

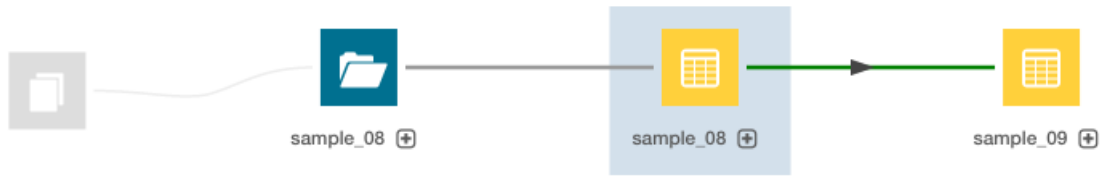


icon. This icon displays when the lineage diagram has more than 400 entities. For example:

One or more links from sample_08 were not included in this lineage. To explore these links further, [view the lineage of sample_08](#)



To view the lineage of hidden entities, select the hidden entity and click **view the lineage** in the box on the right to display a new lineage centered around that entity. After clicking the link, you would see the following:



Filtering Lineage Diagrams

To reduce the time and resources required to render large lineage diagrams, you can filter out classes of entities and links by selecting checkboxes in the **Lineage Options** box on the right of the diagram. The following are the default selections:

▼ **Lineage Options**

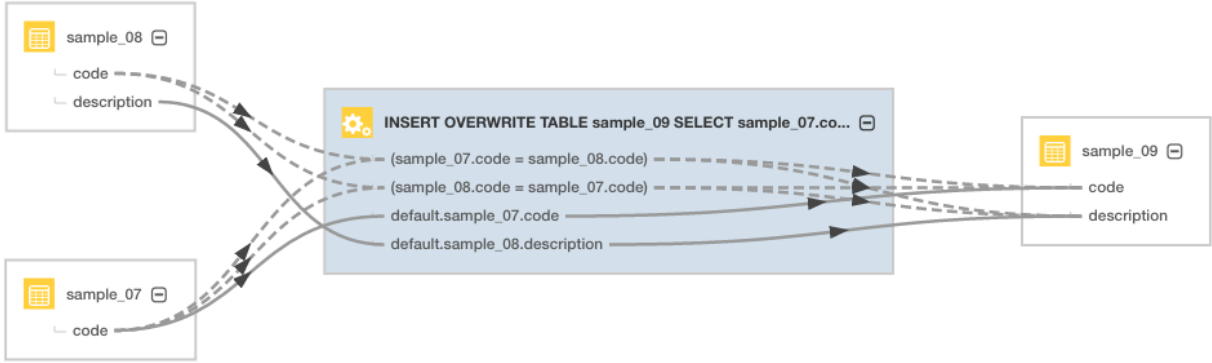
- Operations
- Control Flow Relations
- Only Upstream Downstream
- Deleted Entities
- Latest Partition and Operation

The **Only Upstream/Downstream** filter allows you to filter out entities and links that are input (upstream) to and output (downstream) from another entity.

Use the **Latest Partition and Operation** filter to reduce rendering time when you have similar partitions created and operations performed periodically. For example, if Hive partitions are created daily, the filter allows you to display only the latest partition.

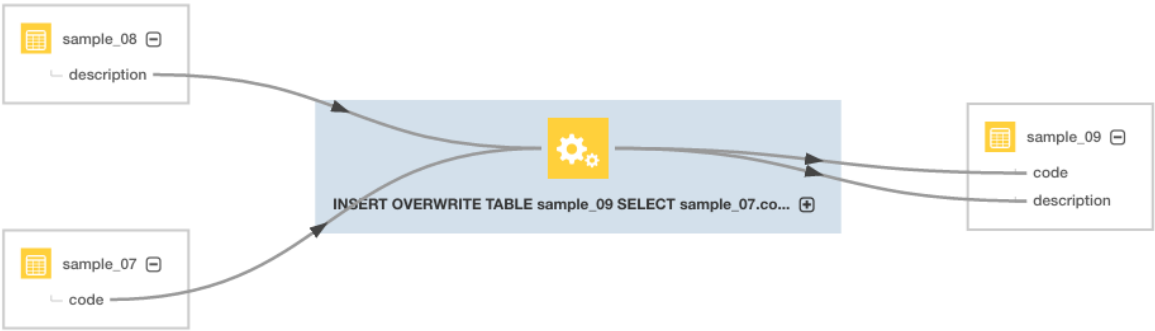
Filter Example

If you display the lineage of the `sample_09` table with no filtering options selected (other than hiding deleted items), the lineage appears as follows.

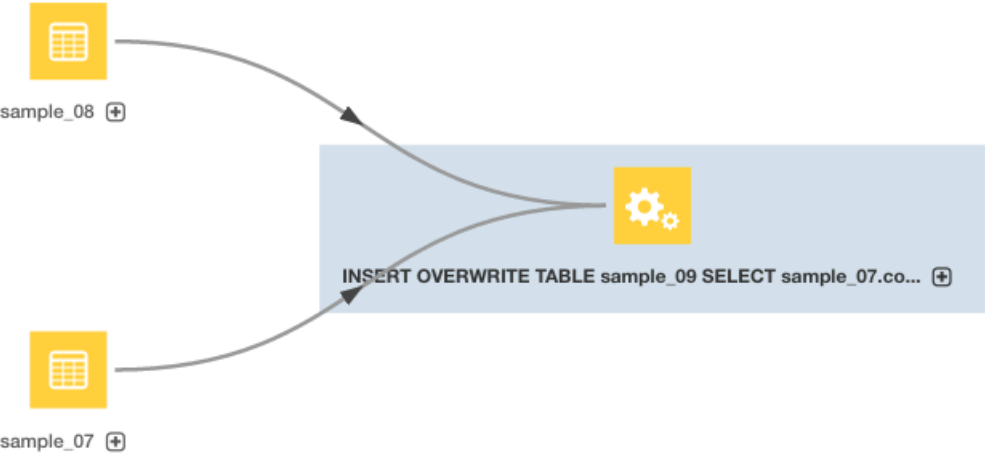


Subsequent diagrams show the result of using each supported filter type:

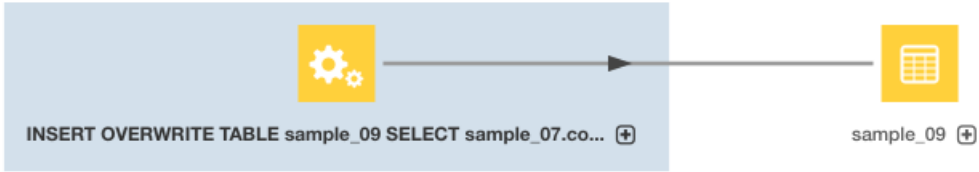
- **Control Flow Relations** - The operation is collapsed and control flow links are hidden.



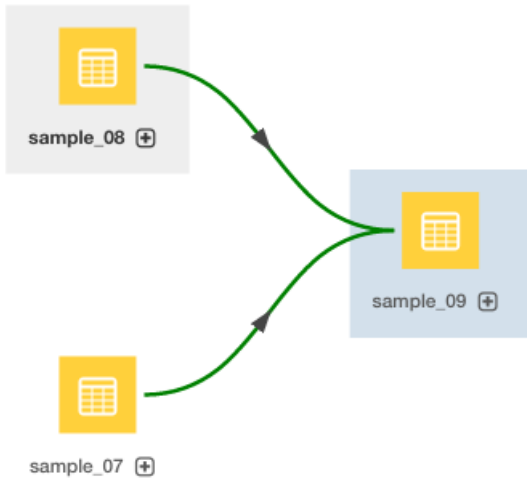
- **Show Upstream** and **Show Downstream** - The operation is collapsed and only upstream entities and links are shown. The output table is hidden.



Here, the operation is collapsed and only downstream entities and links are shown. The input tables are hidden.

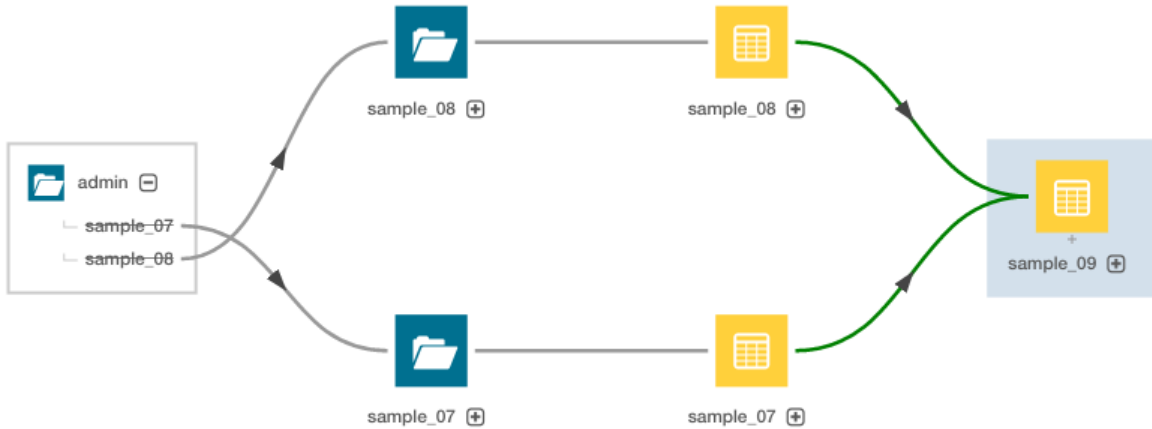


- **Operations** - In the diagram, the operation is hidden.



The green links indicate that one or more operations are collapsed into the links.

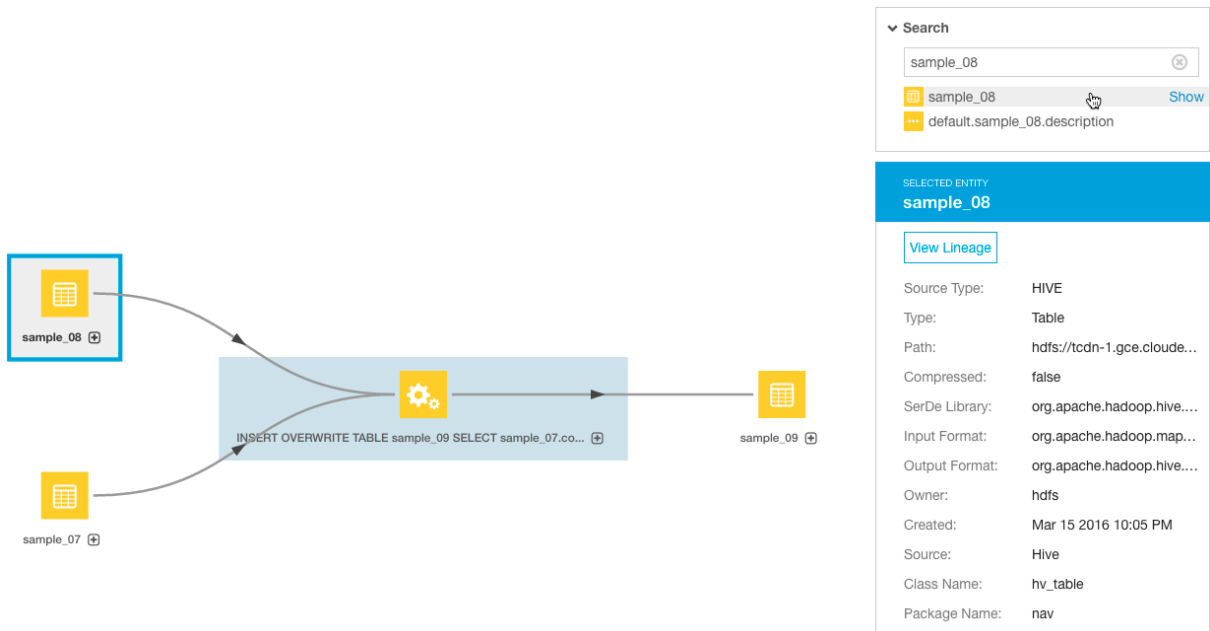
- **Deleted Entities** - Here, the operation is hidden but deleted entities are displayed.



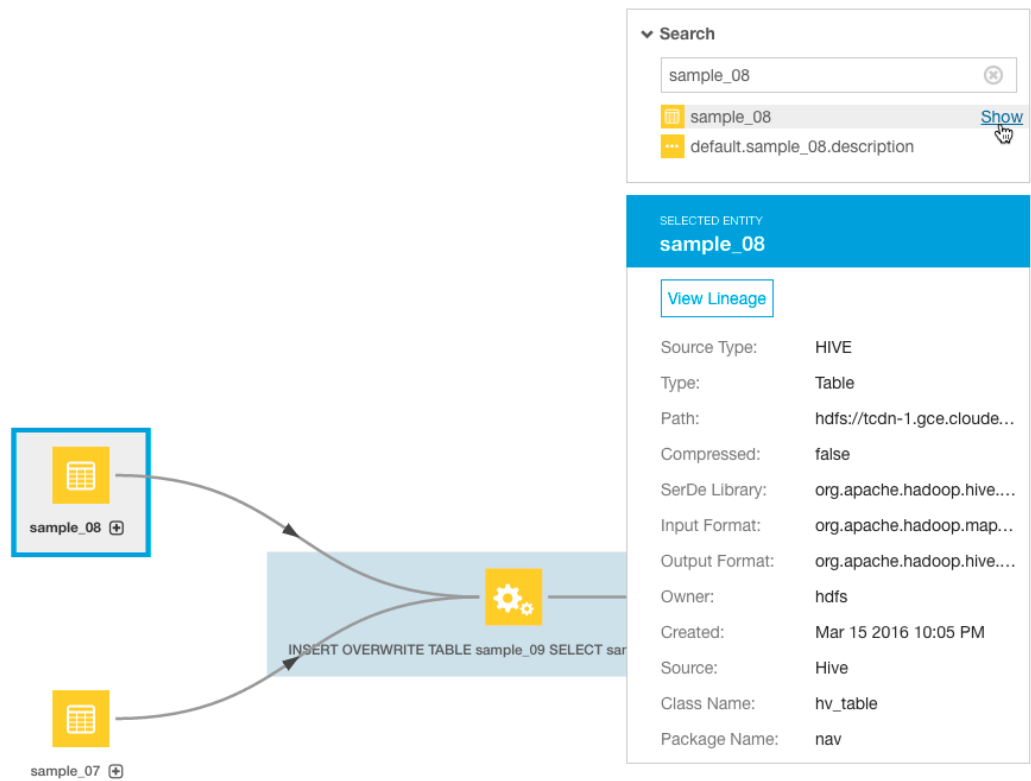
Searching a Diagram

You can search a lineage diagram for an entity by doing the following:

1. In the Search box at the right of the diagram, type an entity name. A list of matching entities displays below the box.
2. Click an entity in the list. A blue box is drawn around the entity and the entity details display in a box below the Search box.



3. Click the **Show** link next to the entity. The selected entity moves to the center of the diagram.



4. Optionally, click the **View Lineage** link in the entity details box to view the lineage of the selected entity.

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 entity. The entity Details page displays. 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` Actions ▾ Details Lineage

Technical Metadata

Source Type: HIVE
 Type: Table
 Parent Path: /default
 Path: hdfs://nightly57-1.gce.cloudera.com:80...
 Compressed: false
 SerDe Library: org.apache.hadoop.hive.serde2.lazy.La...
 Input Format: org.apache.hadoop.mapred.TextInputF...
 Output Format: org.apache.hadoop.hive.qi.io.HiveIgnor...
 Owner: admin
 Created: Mar 18 2016 10:15 AM
 Source: HIVE-1
 Class Name: hv_table
 Package Name: nav

Managed Metadata

No metadata available

Schema 🔍

- code string
- description string

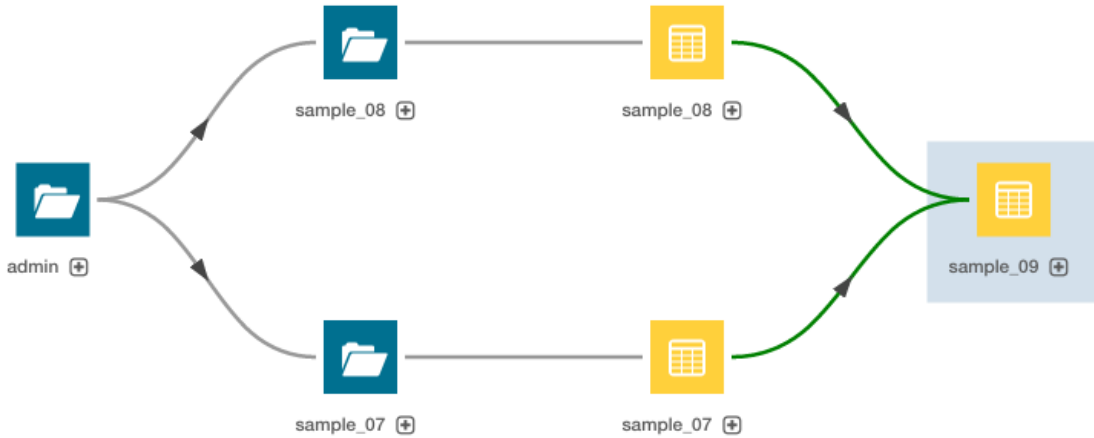
Custom Metadata

No metadata available

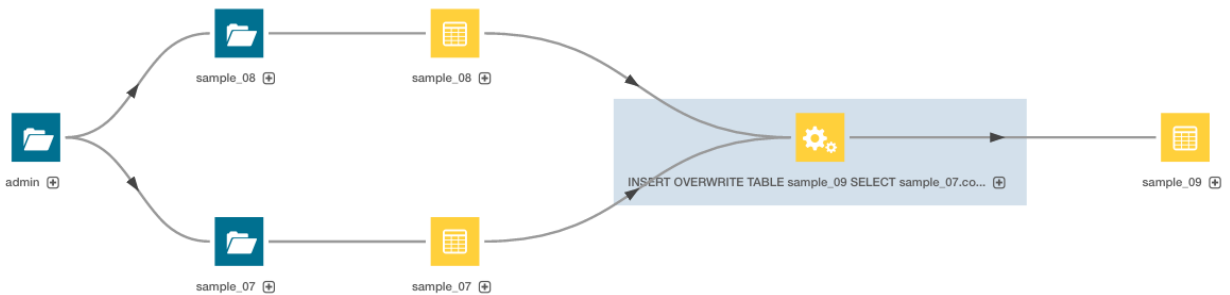
Inputs 🔍


- sample_07
- sample_08

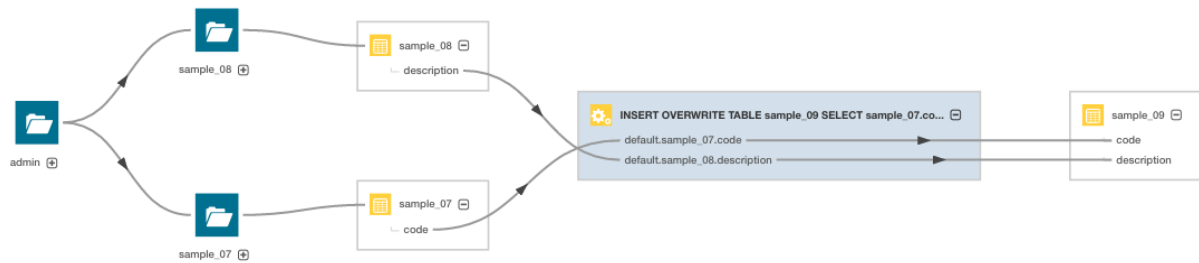
3. Click the **Lineage** tab. For example, clicking the Lineage tab for the `sample_09` table displays the following lineage diagram:



This example shows 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

To browse 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.



Schema

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

Name
.
.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
```


Troubleshooting Navigator Data Management

This page contains troubleshooting tips and workarounds for various issues that can arise.

Navigator Audit Server

"No serializer found" error

Symptom: Selecting an Audits page results in error message.

Error message:

```
No serializer found
```

Possible cause: Communication failure somewhere between Navigator Metadata Server and Cloudera Manager Server and its communication to Navigator Audit Server. This is an existing Known Issue that needs to be resolved so that the error message can be properly mapped.

Workaround: This workaround does not resolve the issue, but provides additional information as to the actual source of the underlying error.

1. Log in to the Cloudera Navigator console.
2. Enable Navigator API logging.
3. Perform your action on the Audits page that raised the error message and see if the underlying issue is caught in the API log. If the Navigator API logging reveals no additional information, turn on API logging for Cloudera Manager:
 - Log in to the Cloudera Manager Admin Console.
 - Select **Administrator > Settings**
 - Select **Advanced** for the **Category** filter.
 - Click the **Enable Debugging of API** check box to select it. The server log will contain all API calls. Try your request again and see if the source error message displays in the response.

Processing a backlog of audit logs

Problem: You have logs that include audit events that were not processed by the Cloudera Navigator Audit Server, such as logs for events that took place before audit server was online or during a period when audit server was offline.

Solution: A backlog of logs can be processed by audit server as follows:

1. Backup audit files for all roles on all hosts.

Do this right away as there are retention periods configured for these files and they will gradually be deleted.

2. Determine what days were not processed. The audit log files have the UNIX epoch appended to their name:

```
hdfs-NAMENODE-341ad9b94435839ce45b8b22e7c805b3-HdfsAuditLogger-1499844328581
```

You want the dates from the oldest and newest files. You can do this by sorting the list and identifying the first and last files. For example, for HDFS audit files:

```
The oldest: $ ls ./hdfs* | head -1
```

```
The newest: $ ls ./hdfs* | tail -1
```

Depending on your shell, you can convert the epoch value to a date using `date -r epoch`.

3. Create partitions in the Navigator Audit Server database for days that have missing audits.

Create a partition from an existing template table using SQL commands. Use `SHOW TABLES` to list the template tables. For example, the template table for HDFS is `HDFS_AUDIT_EVENTS`.

Using MySQL as an example, if you have partitions up to March 31, 2017 and your audit logs include HDFS data for the first week of April, you would run the following SQL commands:

```
CREATE TABLE HDFS_AUDIT_EVENTS__2017_04_01 LIKE HDFS_AUDIT_EVENTS;  
CREATE TABLE HDFS_AUDIT_EVENTS__2017_04_02 LIKE HDFS_AUDIT_EVENTS;  
CREATE TABLE HDFS_AUDIT_EVENTS__2017_04_03 LIKE HDFS_AUDIT_EVENTS;  
CREATE TABLE HDFS_AUDIT_EVENTS__2017_04_04 LIKE HDFS_AUDIT_EVENTS;  
CREATE TABLE HDFS_AUDIT_EVENTS__2017_04_05 LIKE HDFS_AUDIT_EVENTS;  
CREATE TABLE HDFS_AUDIT_EVENTS__2017_04_06 LIKE HDFS_AUDIT_EVENTS;  
CREATE TABLE HDFS_AUDIT_EVENTS__2017_04_07 LIKE HDFS_AUDIT_EVENTS;
```

4. In the `PARTITION_INFO` table, add new rows for each of the tables you created in the previous step, including the "END_TS" in Unix epoch time, 24 hours after the previous entry.

The following example command inserts a row for Feb 14, 2019 into `PARTITION_INFO` table with correct `END_TS` value for multiple partition tables:

```
insert into PARTITION_INFO values (PARTITION_INFO_SEQUENCE.nextval,  
'HDFS_AUDIT_EVENTS_2019_02_03', 'HDFS_AUDIT_EVENTS', 1550145600000, null, 0);
```

5. Change the Navigator Audit expiration period to be longer than the time period of the logs you want to process.

Set the expiration period in Cloudera Manager. Search for the property "Navigator Audit Server Data Expiration Period".

6. Stop the Cloudera Manager agent on the host running the role for the service whose logs will be restored.

For example, if the logs are for HDFS, stop the agent on the namenode. For HiveServer2, stop the Cloudera Manager agent on the node where HiveServer2 is running. See [Starting, Stopping, and Restarting Cloudera Manager Agents](#).

If you aren't sure of which host is involved, the log files contain the UUID of the role which generated them. Go to the host defined for that role. For high-availability clusters, make sure that you stop the role on the active host.

7. On that host, copy the backed up audit logs to the audit directory.

The location is typically `/var/log/service/audit`. For example, `/var/log/hadoop-hdfs/audit`. The specific locations are configured in the Cloudera Manager `audit_event_log_dir` properties for each service.

8. Move the WAL file out of this directory. (Back it up in a safe location.)

9. Restart the Cloudera Manager agent.

On restart, Cloudera Manager agent will not find the WAL file and will create a new one. It will process the older audit logs one by one.

10. Verify that the audits appear in Navigator.

You can also check the contents of the WAL file to see which logs were processed.

Navigator Metadata Server

"Missing required field: id" (Solr exception)

Symptom: After removing the Navigator Metadata Server storage directory (the `datadir`), an error message for the `id` field displays.

Error message:

```
[qtp1676605578-63]: org.apache.solr.common.SolrException:
[doc=0002ac6b75cb17721ad8324700f473cb] missing required field: id
```

Possible cause: This error indicates that the Navigator Metadata Server database and storage directory are out-of-sync because of a change that was made in the Cloudera Navigator 2.9 (Cloudera Manager 5.10) release. When the Navigator Metadata Server role starts, it compares state information in the database with the same information in the storage directory. This error can appear if the storage directory has been deleted but the database has not.

Steps to resolve: If this error message occurs and the storage directory was recently deleted, you can remove the stale state information from the database. To do this:

1. Take note of the Navigator Metadata Server Storage Dir directory.
Log into Cloudera Manager and browse to **Cloudera Management Services > Configuration > Navigator Metadata Server > Navigator Metadata Server Storage Dir**. Note the storage directory location.
2. Stop the Navigator Metadata Server role by navigating to **Cloudera Management Service Instances > Navigator Metadata Server > Actions > Stop**.
3. Log in to the database instance configured for use with Navigator Metadata Server.
4. Run the following SQL commands:

```
delete from NAV_UPGRADE_ORDINAL;
insert into nav_upgrade_ordinal values (-1, -1);
```

5. Start Navigator Metadata Server role using **Cloudera Management Service Instances > Navigator Metadata Server > Actions > Start**.

The Navigator Metadata Server role should restart successfully.

Repairing metadata in the storage directory after upgrading

If you ran Cloudera Navigator on Cloudera Manager versions 5.10.0, 5.10.1, or 5.11.0, you should upgrade to a later release immediately to avoid a known software problem as reported in the knowledge base entry "Upgrade needed for Cloudera Navigator included with Cloudera Manager releases 5.10.0, 5.10.1, and 5.11.0 due to high risk of failure through exhausting data directory resources." This document describes some additional steps to perform after upgrade to mitigate for duplicate and potentially broken relation metadata added to the Navigator storage directory because of the broken versions.

To fully recover the Navigator storage directory:

1. Upgrade Cloudera Manager to 5.10.2, 5.11.1, or later releases.

See [Upgrading Cloudera Manager](#).

2. Set properties to enable background repair tasks.

From Cloudera Manager, go to **Cloudera Management Service > Navigator Metadata Server > Advanced**.

Enter the following configurations to the "Navigator Metadata Server Advanced Configuration Snippet (Safety Valve) for cloudera-navigator.properties" property.

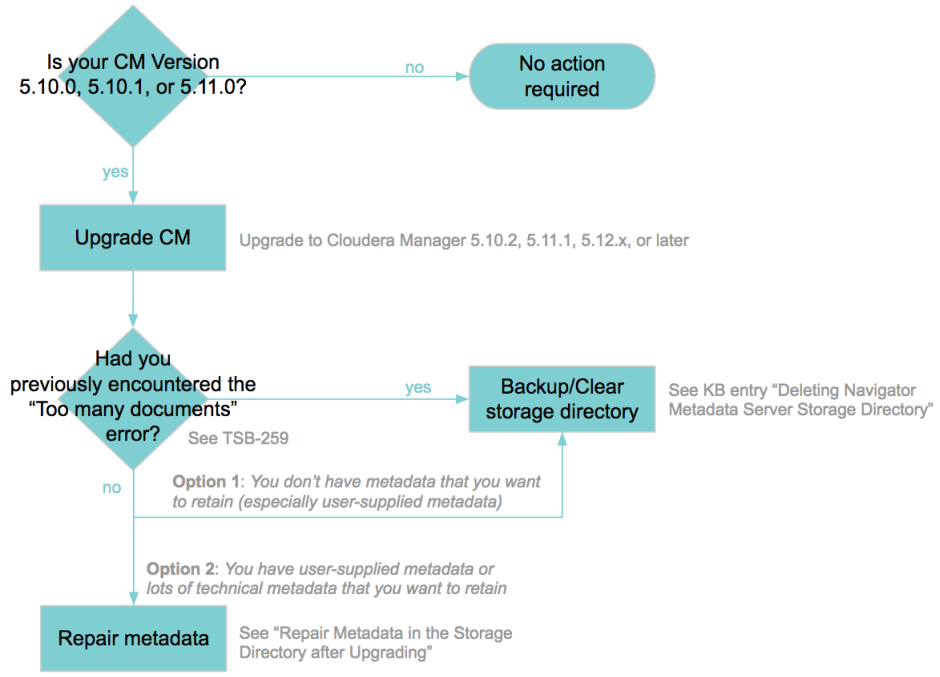
```
nav.backend.task.TRIGGER_HIVE_TABLES_EXTRACTION.enabled=true
nav.backend.task.HDFS_PC_RELATIONS_BUILDER.enabled=true
nav.backend.task.HDFS_MR_RELATIONS_BUILDER.enabled=true
nav.backend.task.PIG_HDFS_RELATIONS_BUILDER.enabled=true
```

3. Restart the Navigator Metadata Server role.

The full cycle of cleaning up the relations may take days or weeks depending on the number of relations affected.

These instructions correspond to "Repair metadata" in the following decision tree:

Troubleshooting Navigator Data Management



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.
```