Cloudera Data Management
# Table of Contents

**About Cloudera Data Management** ......................................................................................................................... 4

**Cloudera Navigator Auditing Architecture** ........................................................................................................... 5
- Audit Log Properties .................................................................................................................................................. 5
- Service Auditing Properties .................................................................................................................................. 6
- *Auditing Impala Operations* ..................................................................................................................................... 8
- Audit Events and Audit Reports ............................................................................................................................. 10
- Viewing Audit Events ............................................................................................................................................... 11
- Filtering Audit Events ........................................................................................................................................... 11
- Creating Audit Reports ......................................................................................................................................... 12
- Editing Audit Reports ............................................................................................................................................ 12
- Downloading Audit Events ..................................................................................................................................... 12
- Downloading HDFS Directory Access Permission Reports ......................................................................................... 17

**Cloudera Navigator Metadata Component** ....................................................................................................... 18
- Metadata ............................................................................................................................................................ 18
- *Metadata Search Syntax and Properties* .................................................................................................................. 20
- Accessing Metadata Using Cloudera Navigator .................................................................................................... 23
- Modifying Custom Metadata .................................................................................................................................. 25
- Metadata Extraction Policies ................................................................................................................................. 31
- Policy Expressions ................................................................................................................................................ 33
- Introduction to Cloudera Navigator Lineage Diagrams ............................................................................................ 40
- Displaying a Template Lineage Diagram ................................................................................................................... 43
- Displaying an Instance Lineage Diagram .................................................................................................................. 44
- Displaying the Template Lineage Diagram for an Instance Lineage Diagram .............................................................. 45
- Downloading a Lineage File ..................................................................................................................................... 45
- Tables ................................................................................................................................................................. 64
- Displaying Table Schema ....................................................................................................................................... 64
- Displaying Pig Tables ............................................................................................................................................. 65

**Appendix: Apache License, Version 2.0** ............................................................................................................. 66
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.

For other licenses, the following applies:

- Cloudera Express - The feature is not available.
- Cloudera Enterprise Data Hub Edition Trial - The feature is available until you end the trial or the trial license expires.

To obtain a license for Cloudera Enterprise, fill in this form or call 866-843-7207. After you install a Cloudera Enterprise license, the feature will be available.

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

Cloudera Navigator provides two categories of functionality:

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

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

- HDFS data accessed through HDFS, Hive, HBase, Cloudera Impala services
- HBase and Impala operations
- Hive metadata
- Sentry access

- Searching metadata and visualizing lineage - Cloudera Navigator metadata management features allow DBAs, data modelers, business analysts, and data scientists to search for, amend the properties of, and tag data entities.

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

Audit Log Properties

Describes auditing log properties and how to configure the log properties.

The following properties apply to the audit log file:
- **Audit Log Directory** - The directory in which audit event log files are written. By default, this property is not set if Cloudera Navigator is not installed.
Note: If the value of this property is changed, and service is restarted, then the Cloudera Manager Agent will start monitoring the new log directory for audit events. In this case it is possible that not all events are published from the old audit log directory. To avoid loss of audit events, when this property is changed, perform the following steps:

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

- **Maximum Audit Log File Size** - The maximum size of the audit event log file before a new file is created. The unit of the file size is service dependent:
  - HDFS, HBase, Hive - MiB
  - Impala - lines (queries)

- **Number of Audit Logs to Retain** - Maximum number of rolled over audit logs to retain. The logs will not be deleted if they contain audit events that have not yet been propagated to Audit Server.

### Configuring Audit Logs

1. Click a supported service.
2. Click the Configuration tab.
3. Configure the log properties in the following categories:
   - Impala - Impala Daemon Default Group > Logs
   - HBase, HDFS, Hive, Sentry - Service-Wide > Logs
4. Edit the audit log properties.
5. Click Save Changes.
6. Restart the service.

### Service Auditing Properties

Describes service auditing properties and how to configure the properties.

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

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

- **Event Filter** - A set of rules that capture properties of auditable events and actions to be performed when an event matches those properties. This property is not supported for Sentry.

- **Event Tracker** - A set of rules for tracking and coalescing events. This feature is used to define equivalency between different audit events. When events match, according to a set of configurable parameters, only one entry in the audit list is generated for all the matching events.

- **Queue Policy** - The action to take when the audit event queue is full. The options are Drop or Shutdown. When a queue is full and the queue policy of the service is Shutdown, before shutting down the service, N audits will be discarded, where N is the size of the Cloudera Navigator Audit Server queue.
Note: If the queue policy is Shutdown, the Impala service is shut down only if Impala is unable to write to the audit log file. It is possible that an event may not appear in the audit event log due to an error in transfer to the Cloudera Manager Agent or database. In such cases Impala will not shut down and will keep writing to the log file. When the transfer problem is fixed the events will be transferred to the database.

The Event Filter and Event Tracker rules for filtering and coalescing events are expressed as JSON objects. For information on the structure of the objects, see the description on the configuration page within the Cloudera Manager Admin Console.

The default event filter discards events generated by the internal Cloudera and Hadoop users (cloudera-scm, hdfs, hbase, hive, mapred, solr, and dr.who) and that affect files in the /tmp directory.

Configuring Service Auditing Properties

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

1. Click a service that supports auditing.
2. Click the Configuration tab.
3. Click the Cloudera Navigator category. The Service-Wide category displays.
4. Edit the properties.
5. Click Save Changes.
6. Restart the service.

Configuring Impala Daemon Logging

**Minimum Required Role:** Configurator (also provided by Cluster Administrator, Full Administrator)

To control whether the Impala daemon logs to the audit log:

1. Click the Impala service.
2. Click the Configuration tab.
3. Expand the Impala Daemon Default Group > Logs category.
4. Edit the Enable Impala Audit Event Generation checkbox setting.
5. Click Save Changes.
6. Restart the service.

Audit Logging to Syslog

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

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

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

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

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

It is also possible to attach other appenders to the auditStream to provide other integration behaviors.

You can audit events to syslog in two formats: JSON and RSA EnVision. To configure audit logging to syslog, do the following:
1. Do one of the following:
   - Select Clusters > Cloudera Management Service > Cloudera Management Service.
   - On the Status tab of the Home page, in Cloudera Management Service table, click the Cloudera Management Service link.

2. Click the Configuration tab.


4. Click the Value field and depending on the format type, enter:

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

   To configure the specific stream type, enter:

<table>
<thead>
<tr>
<th>Format</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>JSON</td>
<td>log4j.additivity.auditStream = false</td>
</tr>
<tr>
<td>RSA EnVision</td>
<td>log4j.additivity.auditStreamEnVision = false</td>
</tr>
</tbody>
</table>

5. Click Save Changes to commit the changes.

Example Log Messages

<table>
<thead>
<tr>
<th>Format</th>
<th>Log Message Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>JSON</td>
<td>Jul 23 11:05:15 hostname local2: {&quot;type&quot;:&quot;HDFS&quot;,&quot;allowed&quot;:&quot;true&quot;,&quot;time&quot;:&quot;1374602714758&quot;, &quot;service&quot;:&quot;HDFS-1&quot;, &quot;user&quot;:&quot;root&quot;, &quot;ip&quot;:&quot;10.20.93.93&quot;,&quot;op&quot;:&quot;rmdir&quot;, &quot;src&quot;:&quot;/audit/root&quot;,&quot;perms&quot;:&quot;rwxr-xr-x&quot;}</td>
</tr>
<tr>
<td>RSA EnVision</td>
<td>Cloudera</td>
</tr>
</tbody>
</table>

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

Auditing Impala Operations

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

- On a system managed by Cloudera Manager, enable auditing by configuring Impala Daemon logging. On a system not managed by Cloudera Manager, include the option `--audit_event_log_dir=directory_path` in your `impalad` startup options. The log directory must be a local directory on the server, not an HDFS directory.

- Decide how many queries will be represented in each log file. By default, Impala starts a new log file every 5000 queries. To specify a different number, configure Impala Daemon logging in Cloudera Manager, or include the option `--max_audit_event_log_file_size=number_of_queries` in the `impalad` startup options for systems not managed by Cloudera Manager.

- Configure the Cloudera Navigator product to collect and consolidate the audit logs from all the nodes in the cluster.

- Use Cloudera Navigator or Cloudera Manager to filter, visualize, and produce reports based on the audit data. (The Impala auditing feature works with Cloudera Manager 4.7 to 5.1 and Cloudera Navigator 2.1 and higher.) Check the audit data to ensure that all activity is authorized and detect attempts at unauthorized access.

Durability and Performance Considerations for Impala Auditing

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

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

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

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

**Format of the Audit Log Files**

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

All the information about schema objects accessed by the query is encoded in a single nested record on the same line. For example, the audit log for an \texttt{INSERT ... SELECT} statement records that a select operation occurs on the source table and an insert operation occurs on the destination table. The audit log for a query against a view records the base table accessed by the view, or multiple base tables in the case of a view that includes a join query. Every Impala operation that corresponds to a SQL statement is recorded in the audit logs, whether the operation succeeds or fails. Impala records more information for a successful operation than for a failed one, because an unauthorized query is stopped immediately, before all the query planning is completed.

Impala records more information for a successful operation than for a failed one, because an unauthorized query is stopped immediately, before all the query planning is completed.

The information logged for each query includes:

- **Client session state:**
  - Session ID
  - User name
  - Network address of the client connection

- **SQL statement details:**
  - Query ID
  - Statement Type - DML, DDL, and so on
  - SQL statement text
  - Execution start time, in local time
  - Execution Status - Details on any errors that were encountered
  - Target Catalog Objects:
    - Object Type - Table, View, or Database
    - Fully qualified object name
    - Privilege - How the object is being used (\texttt{SELECT}, \texttt{INSERT}, \texttt{CREATE}, and so on)

**Which Operations Are Audited**

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

- Queries that are prevented due to lack of authorization.
- Queries that Impala can analyze and parse to determine that they are authorized. The audit data is recorded immediately after Impala finishes its analysis, before the query is actually executed.
The audit log does not contain entries for queries that could not be parsed and analyzed. For example, a query that fails due to a syntax error is not recorded in the audit log. The audit log also does not contain queries that fail due to a reference to a table that does not exist, if you would be authorized to access the table if it did exist.

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

### Reviewing the Audit Logs

You typically do not review the audit logs in raw form. The Cloudera Manager Agent periodically transfers the log information into a back-end database where it can be examined in consolidated form. For CDH 5, see [Audit Events and Audit Reports](#). For CDH 4, see the Cloudera Navigator documentation.

### Audit Events and Audit Reports

**Required Role:**

- **Auditing Viewer**
- **Full Administrator**

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

Audit events are recorded by the Cloudera Navigator Audit Server. Audit report metadata is recorded by the Cloudera Navigator Metadata Server.

The following properties can appear in an audit event entry:

- **Timestamp** - Date and time the action was performed. The server stores the timestamp in the timezone of the server and the UI displays the timestamp converted to the local timezone.
- **Operation** - The action performed.
  - **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`
  - **Hive** - `EXPLAIN`, `LOAD`, `EXPORT`, `IMPORT`, `CREATEDATABASE`, `DROPDATABASE`, `SWITCHDATABASE`, `DROPTABLE`, `DESCTABLE`, `DESCFUNCTION`, `MSCK`, `ALERTABLE_ADDCOLS`, `ALERTABLE_REPLACECOLS`, `ALERTABLE_RENAMECOL`, `ALERTABLE_RENAMEPART`, `ALERTABLE_RENAME`, `ALERTABLE_DROPPARTS`, `ALERTABLE_ADDPARTS`, `ALERTABLE_TOUCH`, `ALERTABLE_ARCHIVE`, `ALERTABLE_UNARCHIVE`, `ALERTABLE_PROPERTIES`, `ALERTABLE_SERIALIZER`, `ALERTPARTITION_SERIALIZER`, `ALERTTABLE_SERDEPROPERTIES`, `ALERTPARTITION_SERDEPROPERTIES`, `ALERTABLE_CLUSTER_SORT`, `SHOWDATABASES`, `SHOWTABLES`, `SHOW_TABLESTATUS`, `SHOW_TBLPROPERTIES`, `SHOWFUNCTIONS`, `SHOWINDEXES`, `SHOWPARTITIONS`, `SHOWLOCKS`, `CREATEFUNCTION`, `DROPFUNCTION`, `CREATEVIEW`, `DROPVIEW`, `CREATEINDEX`, `DROPINDEX`, `ALTERINDEX_REBUILD`, `ALTERVIEW_PROPERTIES`, `LOCKTABLE`, `UNLOCKTABLE`, `ALERTABLE_PROTECTMODE`, `ALERTPARTITION_PROTECTMODE`, `ALERTABLE_FILEFORMAT`, `ALTERPARTITION_FILEFORMAT`, `ALERTABLE_LOCATION`, `ALTERPARTITION_LOCATION`, `CREATETABLE`, `CREATETABLE_AS_SELECT`, `QUERY`, `ALTERINDEX_PROPS`, `ALERTERATABASE`, `DESCDATABASE`, `ALTER_TABLE_MERGE`, `ALERT_PARTITION_MERGE`, `GRANT_PRIVILEGE`, `REVOKE_PRIVILEGE`, `SHOW_GRANT`, `GRANT_ROLE`, `REVOKE_ROLE`, `SHOW_ROLE_GRANT`, `CREATEROLE`, `DROPROLE`
  - **Impala** - `Query`, `Insert`, `Update`, `Delete`, `GRANT_PRIVILEGE`, `REVOKE_PRIVILEGE`, `SHOW_GRANT`, `GRANT_ROLE`, `REVOKE_ROLE`, `SHOW_ROLE_GRANT`, `CREATEROLE`, `DROPROLE`
  - **Sentry** - `GRANT_PRIVILEGE`, `REVOKE_PRIVILEGE`, `ADD_ROLE_TO_GROUP`, `DELETE_ROLE_FROM_GROUP`, `CREATEROLE`, `DROPROLE`
- **Username** - The name of the user that performed the action.
• **Impersonator** - If the action was requested by another service, the name of the user that invoked the service action on behalf of the user.
  – When Sentry is not enabled, the Impersonator field always shows.
  – When Sentry is enabled, the Impersonator field shows for services other than Hive.

• **IP Address** - The IP address of the host where the service action occurred.

• **Service Name** - The name of the service that performed the service action.

### Viewing Audit Events
1. **Start and log into the Navigator UI.**
2. Click the **Audits** tab. The Audit Events report displays all audit events that occurred during the last hour.

### Filtering Audit Events

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

#### Adding a Filter

• Click the **icon** that displays next to a property when you hover in one of the event entries. A filter containing the property, operator, and its value is added to the list of filters at the top and Cloudera Navigator redispays all events that match the filter.

• Click the **Filters** link. The filters pane displays and a filter control with property, operation, and value fields is added to the list of filters.

1. Choose a property in the drop-down list. You can search by properties such as Username, Service Name, or Operation. The properties vary depending on the service or role.
2. Choose an operator in the operator drop-down list.
3. Type a property 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**.
4. Click **Apply**. The property, operation, and value display above the list of audit events and the list of events displays all events that match the filter criteria.
5. Click **Add New Filter** to add more filters and repeat steps 1 through 4.

#### Removing a Filter
1. Do one of the following:
Creating Audit Reports

1. Start and log into the Navigator 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:
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 Reports

1. Start and log into the Navigator 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.
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 Events

You can download audit events in the Audit UI or using the Audit API. An audit event contains the following fields: timestamp, service, username, ipAddress, command, resource, allowed, [operationText], serviceValues. The structure of the resource and serviceValues fields depends on the type of the service. Hive, Hue, Impala, and Sentry events have the operationText field, which contains the operation string.

Downloading Audit Events Using the Audit UI

1. Start and log into the Navigator 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. Select Export > format, where format is CSV or JSON.
HDFS Audit Log Example

```json
{
  "items": [
    {
      "timestamp": "2014-10-10T16:39:25.656Z",
      "service": "HDFS-1",
      "username": "admin",
      "ipAddress": "10.20.190.241",
      "command": "setPermission",
      "resource": "/user/hive/warehouse/sample_09/000000_0",
      "allowed": true,
      "serviceValues": {
        "dest": null,
        "delegation_token_id": null,
        "permissions": "rwxrwxrwt",
        "src": "10.20.190.241"
      }
    },
    {
      "timestamp": "2014-10-10T16:39:25.632Z",
      "service": "HDFS-1",
      "username": "admin",
      "ipAddress": "10.20.190.241",
      "command": "setPermission",
      "resource": "/user/hive/warehouse/sample_09",
      "allowed": true,
      "serviceValues": {
        "dest": null,
        "delegation_token_id": null,
        "permissions": "rwxrwxrwt",
        "src": "10.20.190.241"
      }
    },
    {
      "timestamp": "2014-10-10T16:39:25.606Z",
      "service": "HDFS-1",
      "username": "admin",
      "ipAddress": "10.20.190.241",
      "command": "setOwner",
      "resource": "/user/hive/warehouse/sample_09",
      "allowed": false,
      "serviceValues": {
        "dest": null,
        "delegation_token_id": null,
        "permissions": null,
        "src": "10.20.190.241"
      }
    },
    {
      "timestamp": "2014-10-10T16:39:25.590Z",
      "service": "HDFS-1",
      "username": "admin",
      "ipAddress": "10.20.190.241",
      "command": "delete",
      "resource": "/user/hive/warehouse/sample_09",
      "allowed": true,
      "serviceValues": {
        "dest": null,
        "delegation_token_id": null,
        "permissions": null,
        "src": "10.20.190.241"
      }
    },
    {
      "timestamp": "2014-10-10T16:39:25.581Z",
      "service": "HDFS-1",
      "username": "admin",
      "ipAddress": "10.20.190.241",
      "command": "getFileInfo",
      "resource": "/user/hive/warehouse",
      "allowed": true,
      "serviceValues": {
        "dest": null,
        "delegation_token_id": null,
        "permissions": null,
        "src": "10.20.190.241"
      }
    }
  ]
}
```
In this example, the first event access was denied, and therefore the `allowed` field has the value `false`.

**Hive Example - via downloaded JSON file**

The following records list Hive operations to create and load a table:

```json
[{
  "timestamp": "2014-10-10T16:39:26.184Z",
  "service": "HIVE-1",
  "username": "admin",
  "ipAddress": "10.20.190.241",
  "command": "QUERY",
  "resource": "default:sample_09",
  "operationText": "SELECT 
    sample_07.code,sample_08.description 
  FROM sample_07 
  JOIN sample_08 
  WHERE 
    sample_08.code = sample_07.code",
  "allowed": true,
  "serviceValues": {
    "object_type": "TABLE",
    "database_name": "default",
    "operation_text": "SELECT 
    sample_07.code,sample_08.description 
  FROM sample_07 
  JOIN sample_08 
  WHERE 
    sample_08.code = sample_07.code",
    "resource_path": "/user/hive/warehouse/sample_09",
    "table_name": "sample_09"
  }
}, {
  "timestamp": "2014-10-10T16:39:26.183Z",
  "service": "HIVE-1",
  "username": "admin",
  "ipAddress": "10.20.190.241",
  "command": "QUERY",
  "resource": "default:sample_07",
  "operationText": "SELECT 
    sample_07.code,sample_08.description 
  FROM sample_07 
  JOIN sample_08 
  WHERE 
    sample_08.code = sample_07.code",
  "allowed": true,
  "serviceValues": {
    "object_type": "TABLE",
    "database_name": "default",
    "operation_text": "SELECT 
    sample_07.code,sample_08.description 
  FROM sample_07 
  JOIN sample_08 
  WHERE 
    sample_08.code = sample_07.code",
    "resource_path": "/user/hive/warehouse/sample_09",
    "table_name": "sample_07"
  }
}, {
  "timestamp": "2014-10-10T16:39:26.182Z",
  "service": "HIVE-1",
  "username": "admin",
  "ipAddress": "10.20.190.241",
  "command": "QUERY",
  "resource": "default:sample_08",
  "operationText": "SELECT 
    sample_07.code,sample_08.description 
  FROM sample_07 
  JOIN sample_08 
  WHERE 
    sample_08.code = sample_07.code",
  "allowed": true,
  "serviceValues": {
    "object_type": "TABLE",
    "database_name": "default",
    "operation_text": "SELECT 
    sample_07.code,sample_08.description 
  FROM sample_07 
  JOIN sample_08 
  WHERE 
    sample_08.code = sample_07.code",
    "resource_path": "/user/hive/warehouse/sample_09",
    "table_name": "sample_08"
  }
}]
```
"allowed": true,
"serviceValues": {
  "object_type": "TABLE",
  "database_name": "default",
  "operation_text": "INSERT OVERWRITE 
  TABLE sample_09 
SELECT 
  sample_07.code,sample_08.description 
  FROM sample_07 
  JOIN sample_08 
WHERE 
  sample_08.code = sample_07.code",
  "resource_path": "/user/hive/warehouse/sample_08",
  "table_name": "sample_08"
}
},
"timestamp": "2014-10-10T16:38:18.604Z",
"service": "HIVE-1",
"username": "admin",
"ipAddress": "10.20.190.241",
"command": "CREATETABLE",
"resource": "default:sample_09",
"operationText": "CREATE TABLE sample_09 (code string,description string) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\\t' STORED AS TextFile",
"allowed": true,
"serviceValues": {
  "object_type": "TABLE",
  "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": ",",
  "table_name": "sample_09"
}
},
"timestamp": "2014-10-10T16:38:18.602Z",
"service": "HIVE-1",
"username": "admin",
"ipAddress": "10.20.190.241",
"command": "CREATETABLE",
"resource": "default:sample_09",
"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": ""
}
},
"timestamp": "2014-10-10T16:37:06.836Z",
"service": "HIVE-1",
"username": "admin",
"ipAddress": "10.20.190.241",
"command": "LOAD",
"resource": "\",
"operationText": "LOAD DATA INPATH
      '/user/admin/sample_08' OVERWRITE INTO
TABLE sample_08",
"allowed": true,
"serviceValues": {
  "object_type": "DFS_DIR",
  "database_name": "",
  "operation_text": "LOAD DATA INPATH
      '/user/admin/sample_08' OVERWRITE INTO
TABLE sample_08",
  "resource_path": "/user/admin/sample_08",
  "table_name": ""
}
},
"timestamp": "2014-10-10T16:37:06.836Z",
"service": "HIVE-1",
"username": "admin",
"ipAddress": "10.20.190.241",
"command": "LOAD",
"resource": "default:sample_08",
"operationText": "LOAD DATA INPATH
      '/user/admin/sample_08' OVERWRITE INTO
TABLE sample_08",
"allowed": true,
Downloading Audit Events Using the Audit API

You can filter and download audit events using the Cloudera Navigator API.

Hive Example - via audit API

To download the audits events using the API, issue the request http://host-1.ent.cloudera.com:7187/api/v3/audits?query=service==*HIVE*, which could return the following JSON items:

```json
{
  "items" : [ {
    "timestamp" : "2014-10-07T21:09:05.804Z",
    "service" : "HIVE-1",
    "username" : "test",
    "impersonator" : "",
    "ipAddress" : "20.10.191.128",
    "command" : "CREATEROLE",
    "resource" : "",
    "operationText" : "CREATE ROLE bad_role",
    "allowed" : false,
    "serviceValues" : { 
      "object_type" : "UNKNOWN",
      "database_name" : "",
      "operation_text" : "CREATE ROLE bad_role",
      "resource_path" : "",
      "table_name" : ""
    }
  }, {
    "timestamp" : "2014-10-07T21:08:52.036Z",
    "service" : "HIVE-1",
    "username" : "test",
    "ipAddress" : "20.10.191.128",
    "command" : "DROPTABLE",
    "resource" : "default:ratings_sum",
    "operationText" : "DROP TABLE ratings_sum",
    "allowed" : true,
    "serviceValues" : { 
      "object_type" : "TABLE",
      "database_name" : "default",
      "operation_text" : "DROP TABLE ratings_sum",
      "resource_path" : "user/hive/warehouse/ratings_sum",
      "table_name" : "ratings_sum"
    }
  }
}
```

Downloading HDFS Directory Access Permission Reports

**Minimum Required Role:** Cluster Administrator (also provided by Full Administrator)

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

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

The Cloudera Navigator metadata component provides data discovery and data lineage management functions. The architecture of the Cloudera Navigator metadata component is illustrated below.

The Navigator Metadata Server performs the following functions:

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

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

Metadata

The Cloudera Navigator Metadata component manages metadata about the entities in a CDH cluster and relations between the entities.

The Navigator metadata schema defines the types of metadata that are available for each entity type it supports. The types of metadata defined by the Navigator Metadata component include: the name of an entity, the service that manages or uses the entity, type, path to the entity, date and time of creation, access, and modification, size, owner, purpose, and relations—parent-child, data flow, and instance of—between entities.

For example, the following shows the property sheet of a file entity:
There are two classes of metadata:

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

### Metadata Extraction

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

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

If an entity is created at time t₀ in the system, that entity will be extracted and linked in Navigator after the extraction poll period (default 10 minutes) plus a service-specific interval as follows:

- **HDFS**: t₀ + extraction poll period + HDFS checkpoint interval (default 1 hour)
- **HDFS + HA**: t₀ + extraction poll period
- **Hive**: t₀ + extraction poll period + Hive maximum wait time (default 60 minutes)

### Metadata Indexing

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

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

Search in the Navigator Metadata component is implemented by an embedded Solr engine that supports the syntax described in LuceneQParserPlugin.

Search Syntax

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

- **Property name-value pairs** - `propertyName:value`, where
  - `propertyName` is one of the properties listed in Search Properties on page 20.
  - `value` is a single value or range of values specified as `[value1 TO value2]`. In a value, * is a wildcard. In property values you must escape special characters ,,-,/, and * with the backslash character \ or enclose the property value in quotes. For example, `fileSystemPath:/tmp/hbase-staging`.

- **User-defined name-value pairs** - `up_propertyName:value`.

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

Example Search Strings

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

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

Search Properties

A reference for the search schema properties.

Default Properties

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

Common Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>description</td>
<td>text</td>
<td>Description of the entity.</td>
</tr>
<tr>
<td>group</td>
<td>caseSensitiveText</td>
<td>The group to which the owner of the entity belongs.</td>
</tr>
<tr>
<td>name</td>
<td>ngramedText</td>
<td>The overridden name of the entity. If the name has not been overridden, this value is empty. Names cannot contain spaces.</td>
</tr>
</tbody>
</table>
| operationType | ngramedText | The type of an operation:  
  - Pig - SCRIPT
  - Sqoop - Table Export, Query Import |
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>originalName</td>
<td>ngramedText</td>
<td>The name of the entity when it was extracted.</td>
</tr>
<tr>
<td>originalDescription</td>
<td>text</td>
<td>The description of the entity when it was extracted.</td>
</tr>
<tr>
<td>owner</td>
<td>caseInsensitiveText</td>
<td>The owner of the entity.</td>
</tr>
<tr>
<td>principal</td>
<td>caseInsensitiveText</td>
<td>For entities with type OPERATION_EXECUTION, the initiator of the entity.</td>
</tr>
<tr>
<td>tags</td>
<td>ngramedText</td>
<td>A set of tags that describe the entity.</td>
</tr>
<tr>
<td>type</td>
<td>ngramedText</td>
<td>The type of the entity. The available types depend on the entity's source type:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• HDFS - DIRECTORY, FILE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• HIVE - DATABASE, TABLE, FIELD, OPERATION, OPERATION_EXECUTION, SUB_OPERATION, PARTITION, RESOURCE, UNKNOWN, VIEW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• MAPREDUCE - OPERATION, OPERATION_EXECUTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• OOZIE - OPERATION, OPERATION_EXECUTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• PIG - OPERATION, OPERATION_EXECUTION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• SQOOP - OPERATION, OPERATION_EXECUTION, SUB_OPERATION</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• YARN - OPERATION, OPERATION_EXECUTION</td>
</tr>
</tbody>
</table>

**Query**

| queryText            | string             | The text of a Hive or Sqoop query.                                         |

**Source**

| clusterName          | string             | The name of the cluster in which the entity is stored.                     |
| sourceId             | string             | The ID of the source type.                                                |
| sourceType           | caseInsensitiveText| The source type of the entity: HDFS, HIVE, MAPREDUCE, Oozie, PIG, SQoop, YARN. |
| sourceUrl            | string             | The URL of the source type.                                               |

**Timestamps**

The available timestamp fields vary by the source type:

- HDFS - lastModified, lastAccessed
- HIVE - created, lastAccessed
- MAPREDUCE, PIG, SQOOP, and YARN - started, ended

<table>
<thead>
<tr>
<th>date</th>
<th>Timestamps in the Solr Date Format. For example:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• lastAccessed: [* TO NOW]</td>
</tr>
<tr>
<td></td>
<td>• created: [1976-03-06T23:59:59.999Z TO *]</td>
</tr>
<tr>
<td></td>
<td>• started: [1995-12-31T23:59:59.999Z TO 2007-03-06T00:00:00Z]</td>
</tr>
<tr>
<td></td>
<td>• ended: [NOW-1YEAR/DAY TO NOW/DAY+1DAY]</td>
</tr>
<tr>
<td></td>
<td>• created: [1976-03-06T23:59:59.999Z TO 1976-03-06T23:59:59.999Z+1YEAR]</td>
</tr>
</tbody>
</table>
### HDFS Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fileSystemPath</td>
<td>path</td>
<td>The path to the entity.</td>
</tr>
<tr>
<td>compressed</td>
<td>Boolean</td>
<td>Indicates whether the entity is compressed.</td>
</tr>
<tr>
<td>deleted</td>
<td>Boolean</td>
<td>Indicates whether the entity has been moved to the Trash folder.</td>
</tr>
<tr>
<td>deleteTime</td>
<td>date</td>
<td>The time the entity was moved to the Trash folder.</td>
</tr>
<tr>
<td>mimeType</td>
<td>ngramedText</td>
<td>The MIME type of the entity.</td>
</tr>
<tr>
<td>parentPath</td>
<td>string</td>
<td>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.</td>
</tr>
<tr>
<td>permissions</td>
<td>string</td>
<td>The UNIX access permissions of the entity.</td>
</tr>
<tr>
<td>size</td>
<td>long</td>
<td>The exact size of the entity in bytes or a range of sizes. Range examples: size: [1000 TO <em>], size: [</em> TO 2000], and size: [* TO *] to find all fields with a size value.</td>
</tr>
</tbody>
</table>

### MAPREDUCE and YARN Properties

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

### OPERATION Properties

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

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

Oozie Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>status</td>
<td>string</td>
<td>The status of the Oozie workflow: RUNNING, SUCCEEDED, or FAILED.</td>
</tr>
</tbody>
</table>

PIG Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>scriptId</td>
<td>string</td>
<td>The ID of the Pig script.</td>
</tr>
</tbody>
</table>

SQOOP Properties

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

Accessing Metadata Using Cloudera Navigator

**Required Role:**

- Lineage Viewer
- Policy Administrator
- Metadata Administrator
- Full Administrator

You can access metadata through the Navigator UI or through the Navigator API.
You perform search in the Navigator UI by typing search strings or constructing search strings using UI controls.

1. Start and log into the Navigator UI.
2. Do one of the following:
   - Type a search string into the Search box that conforms to the search syntax. The Search Results page displays as soon as you start typing.
   - Click the Query Builder link. The Query Builder landing page displays with the result of the wildcard search (*). The Query Builder landing page displays Source Type and Type facets that match the search results with the number of results that match each value of those properties. You can filter the search results by clicking specific values for those properties or adding new properties.

The Full Query read-only box displays the search string constructed from the specified filters. Click Show n Results to display the Search Results page.

Search Results

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

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
- Entity name - the name is a link to a page that displays the entity property editor and lineage diagram.
- Entity 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:

```
Type: Table  Parent Path: /default  Path: hdfs://nightly55-5.ent.cloudera.com:8020/user/hive/warehouse/sample_07
Owner: sample  Created: Dec 9 2014 7:16 AM  Source: HIVE-2
```

View in Hue
Specifying Property Values in the Query Builder Pane

The Query Builder pane contains a Search box and a set of graphical controls that allow you to select property values to filter search results. You can filter using the Search box or the graphical controls.

In the Search box, type the values of default properties.

To filter on a property value for non-default properties, specify values as follows:

- **Boolean** - Check the checkbox.
- **Enumerated** - Start typing or click the field and then select from a drop-down list.
- **Timestamps** - Specified in the format mm/dd/yyyy hh:mm [AM|PM] in a date control. The server stores the timestamp in UTC and the UI displays the timestamp converted to the local timezone. In the date control:
  - **Date** - Click the down arrow to display a calendar and select a date, or click a subfield and click the spinner arrows or up and down arrow keys.
  - **Time** - Click the hour, minute, and AM or PM fields and click the spinner arrows or up and down arrow keys to specify the value.
  - Move between fields using the right and left arrow keys.

To add a property, click **Add another filter...** and select a property name.

Navigator API

The Navigator API allows you to search entity metadata using a REST API. For information about the API, see Cloudera Navigator API.

Modifying Custom Metadata

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

**Required Role:**

- **Policy Administrator**
- **Metadata Administrator**
- **Full Administrator**

Modifying Custom Metadata Using the Navigator UI

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

- **Name**: sample_07.csv
- **Description**: Occupational categories: salary and number of employees.
- **Tags**: occupations, salaries
- **Named Value**: year : 2012

**Note**: You can specify special characters (for example, ".", "") in the name, but it will make searching for the entity more difficult as some characters collide with special characters in the search syntax.
Modifying MapReduce Custom Metadata

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

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

1. Do one of the following:
   - Select Clusters > Cloudera Management Service > Cloudera Management Service.
   - On the Status tab of the Home page, in Cloudera Management Service table, click the Cloudera Management Service link.

2. Click the Configuration tab.


4. Select Category > Advanced.

5. Click Navigator Metadata Server Advanced Configuration Snippet for cloudera-navigator.properties.

6. Specify values for the following properties:
   - `nav.user_defined_properties = comma-separated list of user-defined property names`
   - `nav.tags = comma-separated list of property names that serve as tags. The property nav.tags can point to multiple property names that serve as tags, but each of those property names can only specify a single tag.`

7. Click Save Changes.

8. Click the Instances tab.
9. Restart the role.
10. In the MapReduce job configuration, set the value of the property names you specified in step 6.

To specify configuration parameters dynamically:

1. Specify one or more of the following properties in a job configuration:
   - **job properties (type: OPERATION)**
     - `nav.job.user_defined_properties` — comma-separated list of user-defined property names
     - `nav.job.tags` — comma-separated list of property names that serve as tags
   - **job execution properties (type: OPERATION_EXECUTION)**
     - `nav.jobexec.user_defined_properties` — comma-separated list of user-defined property names
     - `nav.jobexec.tags` — comma-separated list of property names that serve as tags

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

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

**Example: Setting Properties Dynamically**

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

1. Dynamically add the `job_tag1` and `job_tag2` properties:
   ```java
   conf.set("nav.job.tags", "job_tag1, job_tag2");
   ```

2. Set the `job_tag1` property to `onetag`:
   ```java
   conf.set("job_tag1", "onetag");
   ```

3. Set the `job_tag2` property to `twotag`:
   ```java
   conf.set("job_tag2", "twotag");
   ```

Add the tag `atag` to a job execution:

1. Dynamically add the `job_tag` property:
   ```java
   conf.set("nav.jobexec.tags", "job_exec_tag");
   ```

2. Set the `job_exec_tag` property to `atag`:
   ```java
   conf.set("job_exec_tag", "atag");
   ```

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

1. Dynamically add the user-defined property `bar`:
   ```java
   conf.set("nav.job.user_defined_properties", "bar");
   ```

2. Set the value of the user-defined property `foo` to `bar`:
   ```java
   conf.set("foo", "bar")
   ```
Modifying HDFS Custom Metadata Using Metadata Files

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

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

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

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

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

Modifying HDFS and Hive Custom Metadata Using the Navigator API

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

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

```
curl http://hostname:port/api/v2/entities/?query=-internalType:* -u username:password -X GET
```

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

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

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

- **PUT**

  ```
  ```

  where properties are:

  - name: name metadata
  - description: description metadata
Cloudera Navigator Metadata Component

- **tags**: tag metadata
- **properties**: property metadata

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

- **POST**

```bash
```

where **properties** are:

- **sourceId** (required): The source ID must match an existing source ID. After the first extraction, you can retrieve the source IDs using the call

```bash
```

For example:

```json
[...
  {
    "identity" : "a09b0233cc58ff7d601eaa68673a20c6",
    "originalName" : "HDFS-1",
    "sourceId" : null,
    "firstClassParentId" : null,
    "parentPath" : null,
    "extractorRunId" : null,
    "name" : "HDFS-1",
    "description" : null,
    "tags" : null,
    "properties" : null,
    "clusterName" : "Cluster 1",
    "sourceUrl" : "hdfs://hostname:8020",
    "sourceType" : "HDFS",
    "sourceExtractIteration" : 4935,
    "type" : "SOURCE",
    "internalType" : "source"
  }, ...
]
```

If you have HDFS-1 and HDFS-2, you must specify the source that contains the entity you’re expecting for it to match.

- **parentPath**: The path of the parent entity, as defined below:
  - HDFS file or directory: `fileSystemPath` of the parent directory (do not provide this field if the entity being affected is the root directory). Example: `parentPath` for `/user/admin/input_dir`/user/admin. If you apply metadata to a directory, the metadata does not propagate to any files and folders in that directory.
  - Hive database: If you are updating database metadata, you do not specify this field.
  - Hive table or view: Name of database containing this table or view. Example for table in default database: `default`.
  - Hive column: `database name/table/view name`. Example for column in 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: name of the database, table, view, or column.
    - Example for default database: `default`
    - Example for sample_07 table: `sample_07`

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

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

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

curl http://Navigator_Metadata_Server_host:port/api/v6/entities/e461de8de38511a3ac6740dd7d51b8d0 -u username:password -X PUT -H "Content-Type: application/json" -d '{"name":"my_name","description":"My description","tags":['tag1','tag2'],"properties":{"property1":"value1","property2":"value2"}}'

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

curl http://Navigator_Metadata_Server_host:port/api/v6/entities/ -u username:password -X POST -H "Content-Type: application/json" -d '{"sourceId":"a09b0233cc58ff7d601ea68673a20c6","parentPath":"/user/admin","originalName":"input_dir","name":"my_name","description":"My description","tags":['tag1','tag2'],"properties":{"property1":"value1","property2":"value2"}}'

**Hive POST Example for total_emp Column**

curl http://Navigator_Metadata_Server_host:port/api/v6/entities/ -u username:password -X POST -H "Content-Type: application/json" -d '{"sourceId":"4fbddadc6899638782fc8cb626176dc7b","parentPath":"default/sample_07","originalName":"total_emp","name":"my_name","description":"My description","tags":['tag1','tag2'],"properties":{"property1":"value1","property2":"value2"}}'

**Metadata Extraction Policies**

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

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

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

For each action, certain properties support specifying a value using a **policy expression**.

**Viewing Policies**

**Required Role:**

- Policy Viewer
- Policy Administrator
- Full Administrator

1. Start and log into the Navigator UI.
2. Click the Policies tab.
3. In the left pane, click a policy.
Creating Policies

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

1. **Start and log into the Navigation UI.**
2. Depending on the starting point, do one of the following:

<table>
<thead>
<tr>
<th>Action</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Policies page</td>
<td>1. Click the Policies tab.</td>
</tr>
<tr>
<td></td>
<td>2. Click Create a New Policy.</td>
</tr>
<tr>
<td>Search results page</td>
<td>1. In the Search results page, click Create a policy.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Action</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assign Metadata</td>
<td>1. Specify the custom metadata. Optionally check the Expression checkbox and specify a policy expression for the indicated fields.</td>
</tr>
<tr>
<td>Send Notification to JMS</td>
<td>1. If not already configured, configure a JMS server and queue.</td>
</tr>
<tr>
<td></td>
<td>2. Specify the queue name and message. Optionally check the Expression checkbox and specify a policy expression for the message.</td>
</tr>
</tbody>
</table>

10. Click **Save**.

Cloning and Editing a Policy

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

1. **Start and log into the Navigation UI.**
2. Click the Policies tab.
3. In the left pane, click a policy.
4. Click **Clone Policy** or **Edit Policy**.
5. Edit the policy name, search query, or policy actions.
6. Click **Save**.
Deleting Policies

Required Role:
- Policy Administrator
- Full Administrator

1. **Start and log into the Navigator UI.**
2. Click the **Policies** tab.
3. In the left pane, click a policy.
4. Click **Delete** and click **OK** to confirm.

**Policy Expressions**

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

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

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

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

**Entity Properties in Policy Expressions**

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

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

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

- **XX** is `FSEntity`, `HiveColumn`, `HiveDatabase`, `HivePartition`, `HiveQueryExecution`, `HiveQueryPart`, `HiveQuery`, `HiveTable`, `HiveView`, `JobExecution`, `Job`, `WorkflowInstance`, `Workflow`, `PigField`, `PigOperationExecution`, `PigOperation`, `PigRelation`, `SqoopExportSubOperation`, `SqoopImportSubOperation`, `SqoopOperationExecution`, `SqoopQueryOperation`, `SqoopTableExportOperation`, or `SqoopTableImportOperation`.
- **Property** is one of the properties listed in Entity Property Enum Reference on page 34.

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

```
entity.get(FSEntityProperties.ORIGINAL_NAME, String.class).toLowerCase()
```

**Expression Examples**

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

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

**Import Statements:**

```java
import com.cloudera.nav.hdfs.model.FSEntityProperties;
```
Expression to add the CREATED date to the name:

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

Import Statements:

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

**Entity Property Enum Reference**

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

```java
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
}
```

```java
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
}
```

```java
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
}
```
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_ACCESSSED, // 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
}

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
}

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
}

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
}

public enum HiveTableProperties 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.HiveViewProperties
public enum HiveViewProperties implements PropertyEnum {
    // Return type: java.lang.Boolean
    DELETED,
    QUERY_TEXT,
    TYPE,
    SOURCE_TYPE,
    CREATED,
    LAST_ACCESSSED,
    NAME,
    ORIGINAL_NAME,
    USER_ENTITY,
    SOURCE_ID,
    EXTRACTOR_RUN_ID,
    PARENT_PATH;
}

com.cloudera.nav.mapreduce.model.JobExecutionProperties
public enum JobExecutionProperties implements PropertyEnum {
    SOURCE_TYPE,
    JOB_ID,
    ENDED,
    INPUT_RECURSIVE,
    INPUTS,
    OUTPUTS,
    STARTED,
    PRINCIPAL,
    WF_INST_ID,
    ORIGINAL_NAME,
    USER_ENTITY,
    SOURCE_ID,
    EXTRACTOR_RUN_ID,
    PARENT_PATH;
}

com.cloudera.nav.mapreduce.model.JobProperties
public enum JobProperties implements PropertyEnum {
    ORIGINAL_NAME,
    INPUT_FORMAT,
    OUTPUT_FORMAT,
    OUTPUT_KEY,
    OUTPUT_VALUE,
    MAPPER,
    REDUCER,
    SOURCE_TYPE,
    TYPE,
    // Return type: java.lang.String
}
public enum WorkflowInstanceProperties implements PropertyEnum {
    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
}

public enum WorkflowProperties implements PropertyEnum {
    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
}

public enum PigFieldProperties implements PropertyEnum {
    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
}

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
}
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
}

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
}

public enum SqoopTableImportOperationProperties implements PropertyEnum {
    DB_TABLE, // Return type: java.lang.String
    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
}
**Introduction to Cloudera Navigator Lineage Diagrams**

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

A *lineage diagram* is a directed graph that depicts an entity and its relations with other entities. A lineage diagram is limited to 1000 entities and 2000 relations.

There are two types of lineage diagrams:
- **Template** - represents an entity that is a model for other entities
- **Instance** - represents an instance or execution of a template

**Entities**

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

<table>
<thead>
<tr>
<th>HDFS</th>
<th>Oozie</th>
</tr>
</thead>
<tbody>
<tr>
<td>• File</td>
<td>• Job template</td>
</tr>
<tr>
<td>• Directory</td>
<td>• Job execution</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Hive</td>
<td>Pig</td>
</tr>
<tr>
<td>• Table</td>
<td>• Table</td>
</tr>
<tr>
<td>• Query template</td>
<td>• Pig script</td>
</tr>
<tr>
<td>• Query execution</td>
<td>• Pig script execution</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>MapReduce</td>
<td>SQOOP</td>
</tr>
<tr>
<td>• Job template</td>
<td>• Job template</td>
</tr>
<tr>
<td>• Job execution</td>
<td>• Job execution</td>
</tr>
</tbody>
</table>
Parent entities are represented by a white box enclosing other entities. The following lineage diagram illustrates the relations between the YARN job `aggregate.pig` and Pig script `aggregate.pig` invoked by the parent Pig Oozie workflow and its source file and destination folder:

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

- Entities are not yet extracted. In this case will eventually be replaced with the correct entity icon after the entity is extracted and linked in Navigator. For information on how long it takes for newly created entities to be extracted, see Metadata Extraction on page 19.
- Hive entities have been deleted from the system before they could be extracted by Navigator.
Relations

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

<table>
<thead>
<tr>
<th>Relation Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data flow</td>
<td>Describes a relation between data and a processing activity. For example, between a file and a MapReduce job or vice versa.</td>
</tr>
<tr>
<td>Alias</td>
<td>Describes an alias relation. For example, from a table to a synonym.</td>
</tr>
<tr>
<td>Parent-child</td>
<td>Describes a parent-child relation. For example, between a directory and a file.</td>
</tr>
<tr>
<td>Logical-physical</td>
<td>Describes the relation between a logical entity and its physical entity. For example, between a Hive query and a MapReduce job.</td>
</tr>
<tr>
<td>Conjoint</td>
<td>Describes a non-directional relation. For example, between an table and an index.</td>
</tr>
<tr>
<td>Instance of</td>
<td>Describes the relation between a template and its instance. For example, an operation execution is an instance of operation.</td>
</tr>
<tr>
<td>Control flow</td>
<td>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.</td>
</tr>
</tbody>
</table>

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

Manipulating Lineage Diagrams

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

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

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

To display a template lineage diagram:

1. Perform a metadata search.
2. In the list of results, click a result entry. For example, when you click the `sample_09` result entry:

   ![Lineage Diagram Example](image)

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

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

   ![Lineage Diagram Example](image)

   When you click each entity 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. Display a template lineage diagram. For example:

2. Click the Instances tab, which contains a list of links to instances of the template.

3. Click a link to display an instance lineage diagram. The job instance `job_1396040412350_0003` replaces the wordcount job template.
Displaying the Template Lineage Diagram for an Instance Lineage Diagram

You can navigate from an instance diagram to its template.

1. Display a instance lineage diagram.
2. Click the value of the `template` property to navigate to the instance's template.

Downloading a Lineage File

Lineage is externalized in a lineage file in JSON format.

1. Display a template or instance lineage diagram.
2. Click the icon at the top left of the diagram.

A lineage file named `lineage.json` is downloaded. For example, the lineage file representing `job_1396040412350_0003` from the preceding section is:

```
{
    "entities": {
        "d212538318276a6ad8abdd308a4487cc": {
            "level": 1,
            "physical": [],
            "logical": [],
            "aliasOf": [],
            "aliases": [],
            "instances": [],
            "children": [],
            "workflows": [],
            "identity": "d212538318276a6ad8abdd308a4487cc",
            "originalName": "file0",
            "sourceId": "012437f9eeb3c23dc69e679ac94a7fa2",
            "firstClassParentId": null,
            "parentPath": "/user/hdfs/input",
            "extractorRunId": null,
            "name": "file0",
            "description": null,
            "tags": null,
            "fileSystemPath": "/user/hdfs/input/file0",
            "type": "FILE",
            "size": 22,
            "created": "2014-07-08T22:03:32.126Z",
            "lastModified": "2014-07-08T22:03:32.126Z",
            "permissions": 420,
            "owner": "hdfs",
            "group": "supergroup",
            "blockSize": null,
            "mimeType": "application/octet-stream",
            "deleted": false,
            "sourceType": "HDFS",
            "replication": null,
            "internalType": "fselement",
            "nameField": "originalName",
            "sourceName": "HDFS",
            "isScript": false,
            "hasDownstream": true,
            "parent": "e461de8de38511a3ac6740dd7d51b8d0",
            "activeChildren": []
        },
        "57c993fc305b553dae82210090d5da7a": {
            "level": 1,
            "physical": [],
            "logical": [],
            "aliasOf": [],
            "aliases": [],
            "instances": [
                "fc671272b2c7518ec738d6b17f08b7ed"
            ],
            "children": [],
            "workflows": [],
            "identity": "57c993fc305b553dae82210090d5da7a"
        }
    }
}
```
Cloudera Navigator Metadata Component

"originalName": "wordcount",
"sourceId": "262c2128425eabeb29bb1c96ee0eb35f",
"firstClassParentId": null,
"parentPath": null,
"extractorRunId": null,
"name": "wordcount",
"description": null,
"tags": null,
"wfIds": null,
"inputFormat": "org.apache.hadoop.mapred.TextInputFormat",
"outputFormat": "org.apache.hadoop.mapred.TextOutputFormat",
"outputKey": "org.apache.hadoop.io.Text",
"outputValue": "org.apache.hadoop.io.IntWritable",
"mapper": "org.myorg.WordCount$Map",
"reducer": "org.myorg.WordCount$Reduce",
"sourceType": "YARN",
"type": "OPERATION",
"internalType": "mrjobspec",
"nameField": "name",
"sourceName": "YARN (MR2 Included)",
"isScript": false,
"unorchestrated": true,
"active": true,
"column": 0,
"renderOrdinal": 0,
"activeChildren": []
"x": 0,
"y": -52.3046875
},
"2b5a90798459c168d9fb5ecc2fc46aa5": {
  "level": 1,
  "physical": [],
  "logical": [],
  "aliasOf": [],
  "aliases": [],
  "instances": [],
  "children": [],
  "workflows": [],
  "identity": "2b5a90798459c168d9fb5ecc2fc46aa5",
  "originalName": "part-00001",
  "sourceId": "012437f9eeb3c23dc69e679ac94a7fa2",
  "firstClassParentId": null,
  "parentPath": "/user/hdfs/output",
  "extractorRunId": null,
  "name": "part-00001",
  "description": null,
  "tags": null,
  "fileSystemPath": "/user/hdfs/output/part-00001",
  "type": "FILE",
  "size": 6,
  "created": "2014-07-08T22:08:53.602Z",
  "lastModified": "2014-07-08T22:08:53.602Z",
  "lastAccessed": "2014-07-08T22:08:52.923Z",
  "permissions": 420,
  "owner": "hdfs",
  "group": "supergroup",
  "blockSize": null,
  "mimeType": "application/octet-stream",
  "deleted": false,
  "sourceType": "HDFS",
  "replication": null,
  "internalType": "fselement",
  "nameField": "originalName",
  "sourceName": "HDFS",
  "isScript": false,
  "parent": "e9b3d1f7aee29134338df3de4cael100",
  "hasUpstream": true,
  "activeChildren": []
},
"fe5445fb6d070d97c418d96200a218cae": {
  "level": 1,
  "physical": [],
  "logical": []
}
"aliasOf": [],
"aliases": [],
"instances": [],
"children": [],
"workflows": [],
"identity": "fe5445fbd070d97c418d96200a218ca",
"originalName": "part-00002",
"sourceId": "012437f9eeb3c23dc69e679ac94a7fa2",
"firstClassParentId": null,
"parentPath": "/user/hdfs/output",
"extractorRunId": null,
"name": "part-00002",
"description": null,
"tags": null,
"fileSystemPath": "/user/hdfs/output/part-00002",
"type": "FILE",
"size": 9,
"created": "2014-07-08T22:08:53.619Z",
"lastModified": "2014-07-08T22:08:53.619Z",
"lastAccessed": "2014-07-08T22:08:52.464Z",
"permissions": 420,
"owner": "hdfs",
"group": "supergroup",
"blockSize": null,
"mimeType": "application/octet-stream",
"deleted": false,
"sourceType": "HDFS",
"replication": null,
"internalType": "fselement",
"nameField": "originalName",
"sourceName": "HDFS",
"isScript": false,
"parent": "e9b3df7ae29134338df3de4ca1e100",
"hasUpstream": true,
"activeChildren": []
},
"916b141fb9ce45094df8ef97ecdde41c": {
"level": 1,
"physical": [],
"logical": [],
"aliasOf": [],
"aliases": [],
"instances": [],
"children": [],
"workflows": [],
"identity": "916b141fb9ce45094df8ef97ecdde41c",
"originalName": "file1._COPYING",
"sourceId": "012437f9eeb3c23dc69e679ac94a7fa2",
"firstClassParentId": null,
"parentPath": "/user/hdfs/input",
"extractorRunId": null,
"name": "file1._COPYING",
"description": null,
"tags": null,
"fileSystemPath": "/user/hdfs/input/file1._COPYING",
"type": "FILE",
"size": 28,
"created": "2014-07-08T22:03:32.171Z",
"lastModified": "2014-07-08T22:03:32.244Z",
"lastAccessed": "2014-07-08T22:03:32.171Z",
"permissions": 420,
"owner": "hdfs",
"group": "supergroup",
"blockSize": null,
"mimeType": "application/octet-stream",
"deleted": true,
"sourceType": "HDFS",
"replication": null,
"internalType": "fselement",
"nameField": "originalName",
"sourceName": "HDFS",
"isScript": false,
"parent": "e461de8de38511a3ac6740dd75db8d0",}
"activeChildren": [],
"66bff6eeac3c17c5b7eb7c9035704eef": {
"level": 1,
"physical": [],
"logical": [],
"aliasOf": [],
"aliases": [],
"instances": [],
"children": [],
"workflows": [],
"identity": "66bff6eeac3c17c5b7eb7c9035704eef",
"originalName": "part-00005",
"sourceId": "012437f9eeb3c23dc69e679ac94a7fa2",
"firstClassParentId": null,
"parentPath": "/user/hdfs/output",
"extractorRunId": null,
"name": "part-00005",
"description": null,
"tags": null,
"fileSystemPath": "/user/hdfs/output/part-00005",
"type": "FILE",
"size": 8,
"created": "2014-07-08T22:08:53.668Z",
"lastModified": "2014-07-08T22:08:53.668Z",
"lastAccessed": "2014-07-08T22:08:52.419Z",
"permissions": 420,
"owner": "hdfs",
"group": "supergroup",
"blockSize": null,
"mimeType": "application/octet-stream",
"deleted": false,
"sourceType": "HDFS",
"replication": null,
"internalType": "fselement",
"nameField": "originalName",
"sourceName": "HDFS",
"isScript": false,
"hasUpstream": true,
"parent": "e9b3d1f7aee2913438df3de4ca12100",
"activeChildren": []
},
"51db076ee11470b7f968bd5f33429e6b": {
"level": 1,
"physical": [],
"logical": [],
"aliasOf": [],
"aliases": [],
"instances": [],
"children": [],
"workflows": [],
"identity": "51db076ee11470b7f968bd5f33429e6b",
"originalName": "file0._COPYING",
"sourceId": "012437f9eeb3c23dc69e679ac94a7fa2",
"firstClassParentId": null,
"parentPath": "/user/hdfs/input",
"extractorRunId": null,
"name": "file0._COPYING",
"description": null,
"tags": null,
"fileSystemPath": "/user/hdfs/input/file0._COPYING",
"type": "FILE",
"size": 22,
"created": "2014-07-08T22:03:31.612Z",
"lastModified": "2014-07-08T22:03:32.100Z",
"permissions": 420,
"owner": "hdfs",
"group": "supergroup",
"blockSize": null,
"mimeType": "application/octet-stream",
"deleted": true,
"sourceType": "HDFS"}
"replication": null,
"internalType": "fselement",
"nameField": "originalName",
"sourceName": "HDFS",
"isScript": false,
"parent": "e461de8de38511a3ac6740dd7d51b8d0",
"activeChildren": []
},
"c864e9c3b03f5f7a99315c56830e811d": {
"level": 1,
"physical": [],
"logical": [],
"aliasOf": [],
"aliases": [],
"instances": [],
"children": [],
"workflows": [],
"identity": "c864e9c3b03f5f7a99315c56830e811d",
"originalName": "_SUCCESS",
"sourceId": "012437f9ebeb3c23dc69e679ac94a7fa2",
"firstClassParentId": null,
"parentPath": "/user/hdfs/output",
"extractorRunId": null,
"name": "_SUCCESS",
"description": null,
"tags": null,
"fileSystemPath": "/user/hdfs/output/_SUCCESS",
"type": "FILE",
"size": 0,
"created": "2014-07-08T22:08:53.689Z",
"lastModified": "2014-07-08T22:08:53.694Z",
"lastAccessed": "2014-07-08T22:08:53.689Z",
"permissions": 420,
"owner": "hdfs",
"group": "supergroup",
"blockSize": null,
"mimeType": "application/octet-stream",
"deleted": false,
"sourceType": "HDFS",
"replication": null,
"internalType": "fselement",
"nameField": "originalName",
"sourceName": "HDFS",
"isScript": false,
"parent": "e9b3d1f7aee29134338df3de4ca100",
"hasUpstream": true,
"activeChildren": []
},
"9d7ca9e46a9e4624df0d5d10949a5fc6": {
"level": 1,
"physical": [],
"logical": [],
"aliasOf": [],
"aliases": [],
"instances": [],
"children": [],
"workflows": [],
"identity": "9d7ca9e46a9e4624df0d5d10949a5fc6",
"originalName": "part-00004",
"sourceId": "012437f9ebeb3c23dc69e679ac94a7fa2",
"firstClassParentId": null,
"parentPath": "/user/hdfs/output",
"extractorRunId": null,
"name": "part-00004",
"description": null,
"tags": null,
"fileSystemPath": "/user/hdfs/output/part-00004",
"type": "FILE",
"size": 10,
"created": "2014-07-08T22:08:53.651Z",
"lastModified": "2014-07-08T22:08:53.651Z",
"lastAccessed": "2014-07-08T22:08:52.856Z",
"permissions": 420,
"fileSystemPath": "/user/hdfs/output/part-00000",
"type": "FILE",
"size": 0,
"created": "2014-07-08T22:08:53.580Z",
"lastModified": "2014-07-08T22:08:53.580Z",
"lastAccessed": "2014-07-08T22:08:52.568Z",
"permissions": 420,
"owner": "hdfs",
"group": "supergroup",
"blockSize": null,
"mimeType": "application/octet-stream",
"deleted": false,
"sourceType": "HDFS",
"replication": null,
"internalType": "fselement",
"nameField": "originalName",
"sourceName": "HDFS",
"isScript": false,
"hasUpstream": true,
"parent": "e9b3df79a2913438df3de4ca1100",
"activeChildren": []
},
"e9b3df79a2913438df3de4ca1100": {
"level": 1,
"physical": [],
"logical": [],
"aliasesOf": [],
"aliases": [],
"instances": [],
"children": [
  "fe54545fbd070d97c418d96200a218c9e",
  "c864e9c3bc3f5f7a99315c56830e811d",
  "2b5a90798459c168d9fb5ecc2fc46a4a5",
  "4374529a34414bc8d30ee01bb3563",
  "2890183198a2d6f568188426ea9a82793",
  "666ff6e6ac3c175b7eb7c9035704eef",
  "9d7ca9e4da9e4624df0d510949a5fc6",
  "e586e89e89ecef7552e125ac9b1c2cf"
],
"workflows": [],
"identity": "e9b3df79a2913438df3de4ca1100",
"originalName": "output",
"sourceId": "012437f9eb323dc69e679ac94a7fa2",
"firstClassParentId": null,
"parentPath": "/user/hdfs",
"extractorRunId": null,
"name": "output",
"description": null,
"tags": null,
"fileSystemPath": "/user/hdfs/output",
"type": "DIRECTORY",
"size": null,
"created": "2014-07-08T22:08:31.569Z",
"lastModified": "2014-07-08T22:08:53.689Z",
"lastAccessed": null,
"permissions": 493,
"owner": "hdfs",
"group": "supergroup",
"blockSize": null,
"mimeType": null,
"deleted": false,
"sourceType": "HDFS",
"replication": null,
"internalType": "fselement",
"nameField": "originalName",
"sourceName": "HDFS",
"isScript": false,
"hasUpstream": true,
"column": 1,
"renderOrdinal": 2,
"activeChildren": []
}
Cloudera Navigator Metadata Component

"physical": [],
"logical": [],
"aliasOf": [],
"aliases": [],
"instances": [],
"children": [],
"workflows": [],
"identity": "c864e9c3bc3f5f7a99315c56830e811d",
"originalName": "_SUCCESS",
"sourceId": "012437f9eb3c23dc69e679ac94a7fa2",
"firstClassParentId": null,
"parentPath": "/user/hdfs/output",
"extractorRunId": null,
"name": "_SUCCESS",
"description": null,
"tags": null,
"fileSystemPath": "/user/hdfs/output/_SUCCESS",
"type": "FILE",
"size": 0,
"created": "2014-07-08T22:08:53.689Z",
"lastModified": "2014-07-08T22:08:53.694Z",
"lastAccessed": "2014-07-08T22:08:53.689Z",
"permissions": 420,
"owner": "hdfs",
"group": "supergroup",
"blockSize": null,
"mimeType": "application/octet-stream",
"deleted": false,
"sourceType": "HDFS",
"replication": null,
"internalType": "fselement",
"nameField": "originalName",
"sourceName": "HDFS",
"isScript": false,
"parent": "e9b3d1f7aee29134338df3de4cae1100",
"hasUpstream": true,
"activeChildren": []
},
{
"level": 1,
"physical": [],
"logical": [],
"aliasOf": [],
"aliases": [],
"instances": [],
"children": [],
"workflows": [],
"identity": "2690183198a2d6568188426ea9a82793",
"originalName": "part-00000",
"sourceId": "012437f9eb3c23dc69e679ac94a7fa2",
"firstClassParentId": null,
"parentPath": "/user/hdfs/output",
"extractorRunId": null,
"name": "part-00000",
"description": null,
"tags": null,
"fileSystemPath": "/user/hdfs/output/part-00000",
"type": "FILE",
"size": 0,
"created": "2014-07-08T22:08:53.580Z",
"lastModified": "2014-07-08T22:08:53.580Z",
"permissions": 420,
"owner": "hdfs",
"group": "supergroup",
"blockSize": null,
"mimeType": "application/octet-stream",
"deleted": false,
"sourceType": "HDFS",
"replication": null,
"internalType": "fselement",
"nameField": "originalName",
"sourceName": "HDFS"
"isScript": false,
"hasUpstream": true,
"parent": "e9b3df7ae29134338df3de4ca1100",
"activeChildren": []
},
{
"level": 1,
"physical": [],
"logical": [],
"aliasOf": [],
"aliases": [],
"instances": [],
"children": [],
"workflows": [],
"identity": "2b5a90798459c1e68d9fb5ecc2fc46aa5",
"originalName": "part-00001",
"sourceId": "012437f9ebeb3c23dc69e679ac94af2",
"firstClassParentId": null,
"parentPath": "/user/hdfs/output",
"extractorRunId": null,
"name": "part-00001",
"description": null,
"tags": null,
"fileSystemPath": "/user/hdfs/output/part-00001",
"type": "FILE",
"size": 6,
"created": "2014-07-08T22:08:53.602Z",
"lastModified": "2014-07-08T22:08:53.602Z",
"lastAccessed": "2014-07-08T22:08:52.923Z",
"permissions": 420,
"owner": "hdfs",
"group": "supergroup",
"blockSize": null,
"mimeType": "application/octet-stream",
"deleted": false,
"sourceType": "HDFS",
"replication": null,
"internalType": "fselement",
"nameField": "originalName",
"sourceName": "HDFS",
"isScript": false,
"parent": "e9b3df7ae29134338df3de4ca1100",
"hasUpstream": true,
"activeChildren": []
},
{
"level": 1,
"physical": [],
"logical": [],
"aliasOf": [],
"aliases": [],
"instances": [],
"children": [],
"workflows": [],
"identity": "fe5445fb00d95c418d96200a2182ae",
"originalName": "part-00002",
"sourceId": "012437f9ebeb3c23dc69e679ac94af2",
"firstClassParentId": null,
"parentPath": "/user/hdfs/output",
"extractorRunId": null,
"name": "part-00002",
"description": null,
"tags": null,
"fileSystemPath": "/user/hdfs/output/part-00002",
"type": "FILE",
"size": 9,
"created": "2014-07-08T22:08:53.619Z",
"lastModified": "2014-07-08T22:08:53.619Z",
"lastAccessed": "2014-07-08T22:08:52.923Z",
"permissions": 420,
"owner": "hdfs",
"group": "supergroup",
"blockSize": null,
"owner": "hdfs",
"group": "supergroup",
"blockSize": null,
"mimeType": "application/octet-stream",
"deleted": false,
"sourceType": "HDFS",
"replication": null,
"internalType": "fselement",
"nameField": "originalName",
"sourceName": "HDFS",
"isScript": false,
"hasDownstream": true,
"column": -1,
"renderOrdinal": 1,
"activeChildren": [ ]
}
{
  "level": 1,
  "physical": [],
  "logical": [],
  "aliasOf": [],
  "aliases": [],
  "instances": [],
  "children": [],
  "workflows": [],
  "identity": "51db076ee11470b7f968bd5f33429e6b",
  "originalName": "file0..COPYING",
  "sourceId": "012437f9e6b5bc3c23dc69e679ac94a7fa2",
  "firstClassParentId": null,
  "parentPath": "/user/hdfs/input",
  "extractorRunId": null,
  "name": "file0..COPYING",
  "description": null,
  "tags": null,
  "fileSystemPath": "/user/hdfs/input/file0",
  "type": "FILE",
  "size": 22,
  "created": "2014-07-08T22:03:32.126Z",
  "lastModified": "2014-07-08T22:03:32.126Z",
  "permissions": 420,
  "owner": "hdfs",
  "group": "supergroup",
  "blockSize": null,
  "mimeType": "application/octet-stream",
  "deleted": false,
  "sourceType": "HDFS",
  "replication": null,
  "internalType": "fselement",
  "nameField": "originalName",
  "sourceName": "HDFS",
  "isScript": false,
  "hasDownstream": true,
  "parent": "e461de8de38511a3ac6740dd7d518b0d",
  "activeChildren": [ ]
},
{
  "level": 1,
  "physical": [],
  "logical": [],
  "aliasOf": [],
  "aliases": [],
  "instances": [],
  "children": [],
  "workflows": [],
  "identity": "d212538318276a6a8abdd308a4487cc",
  "originalName": "file0",
  "sourceId": "012437f9e6b5bc3c23dc69e679ac94a7fa2",
  "firstClassParentId": null,
  "parentPath": "/user/hdfs/input",
  "extractorRunId": null,
  "name": "file0",
  "description": null,
  "tags": null,
  "fileSystemPath": "/user/hdfs/input/file0",
  "type": "FILE",
  "size": 22,
  "created": "2014-07-08T22:03:32.126Z",
  "lastModified": "2014-07-08T22:03:32.126Z",
  "permissions": 420,
  "owner": "hdfs",
  "group": "supergroup",
  "blockSize": null,
  "mimeType": "application/octet-stream",
  "deleted": false,
  "sourceType": "HDFS",
  "replication": null,
  "internalType": "fselement",
  "nameField": "originalName",
  "sourceName": "HDFS",
  "isScript": false,
  "hasDownstream": true,
  "parent": "e461de8de38511a3ac6740dd7d518b0d",
  "activeChildren": [ ]
}
Cloudera Navigator Metadata Component

"fileSystemPath": "/user/hdfs/input/file0._COPYING_
, "type": "FILE", "size": 22, "created": "2014-07-08T22:03:31.612Z", "lastModified": "2014-07-08T22:03:32.100Z", "lastAccessed": "2014-07-08T22:03:31.612Z", "permissions": 420, "owner": "hdfs", "group": "supergroup", "blockSize": null, "mimeType": "application/octet-stream", "deleted": true, "sourceType": "HDFS", "replication": null, "internalType": "fselement", "nameField": "originalName", "sourceName": "HDFS", "isScript": false, "parent": ":e461de8de38511a3ac6740dd7d51b8d0", "activeChildren": []
,
{
  "level": 1, "physical": [], "logical": [], "aliasOf": [], "aliases": [], "instances": [], "children": [], "workflows": [], "identity": "8c40fbf6b4584a0b213a257be2cae679", "originalName": "file1", "sourceId": "012437f9e9b3c23dc69e679ac94a7fa2", "firstClassParentId": null, "parentPath": "/user/hdfs/input", "extractorRunId": null, "name": "file1", "description": null, "tags": null, "fileSystemPath": "/user/hdfs/input/file1", "type": "FILE", "size": 28, "created": "2014-07-08T22:03:32.262Z", "lastModified": "2014-07-08T22:03:32.262Z", "lastAccessed": "2014-07-08T22:03:32.171Z", "permissions": 420, "owner": "hdfs", "group": "supergroup", "blockSize": null, "mimeType": "application/octet-stream", "deleted": false, "sourceType": "HDFS", "replication": null, "internalType": "fselement", "nameField": "originalName", "sourceName": "HDFS", "isScript": false, "hasDownstream": true, "parent": ":e461de8de38511a3ac6740dd7d51b8d0", "activeChildren": []
},
{
  "level": 1, "physical": [], "logical": [], "aliasOf": [], "aliases": [], "instances": [], "children": [], "workflows": [], "identity": "916b141fb9ce45094df8ef97ecdde41c", "originalName": "file1._COPYING_", "sourceId": "012437f9e9b3c23dc69e679ac94a7fa2", "activeChildren": []
}
null
"parentPath": "/user/hdfs/input",
"extractorRunId": null,
"name": "file1._COPYING_
"description": null,
"tags": null,
"fileSystemPath": "/user/hdfs/input/file1._COPYING_
"type": "FILE",
"size": 26,
"created": "2014-07-08T22:03:32.171Z",
"lastModified": "2014-07-08T22:03:32.244Z",
"lastAccessed": "2014-07-08T22:03:32.171Z",
"permissions": 420,
"owner": "hdfs",
"group": "supergroup",
"blockSize": null,
"mimeType": "application/octet-stream",
"deleted": true,
"sourceType": "HDFS",
"replication": null,
"internalType": "fselement",
"nameField": "originalName",
"sourceName": "HDFS",
"isScript": false,
"parent": "e461de8de38511a3ac6740dd7d51b8d0",
"activeChildren": []
},
"x": -182,
"y": -52.3046875
},
"fc671272b2c7518ec738d6b17f08b7ed": 
  "level": 1,
  "physical": [],
  "logical": [],
  "aliasOf": [],
  "aliases": [],
  "instances": [],
  "children": [],
  "workflows": [],
  "identity": "fc671272b2c7518ec738d6b17f08b7ed",
  "originalName": "job_1396040412350_0003",
  "sourceId": "262c2128425eabeb29bb1c96ee0eb35f",
  "firstClassParentId": null,
  "parentPath": null,
  "extractorRunId": null,
  "name": "job_1396040412350_0003",
  "description": null,
  "tags": null,
  "started": "2014-07-08T22:08:31.543Z",
  "ended": "2014-07-08T22:08:53.722Z",
  "principal": "hdfs",
  "inputs": [hdfs://tcdn501-1.ent.cloudera.com:8020/user/hdfs/input],
  "outputs": [hdfs://tcdn501-1.ent.cloudera.com:8020/user/hdfs/output],
  "wfInstId": null,
  "jobID": "job_1396040412350_0003",
  "sourceType": "YARN",
  "inputRecursive": false,
  "type": "OPERATION_EXECUTION",
  "internalType": "mrjobinstance",
  "nameField": "originalName",
  "sourceName": "YARN (MR2 Included)",
  "isScript": false,
  "hasUpstream": true,
  "template": "57c993fc305b553dae82210090d5da7a",
  "hasDownstream": true,
},
"e5586cdbc89edce7f552e25acb91c2cf": 
  "level": 1,
Cloudera Navigator Metadata Component

"type": "DATA_FLOW",
"propagatorId": "cf0c7b96382b834021f51638b135da9e",
"extractorRunId": "NOT_STORED",
"targets": {
  "entityIds": [
    "2690183198a2d6568188426ea9a82793",
    "2b5a90798459c168d9fb5ecc2fc46aa5",
    "fe5445fbd07d97c418d96200a218cae",
    "4374529a3a4441bc6db30ee01bba3563",
    "9d7ca9e46a9e4624df05d10949a5fc6",
    "66bff6e6eac3c17c5b7eb7c9035704eef",
    "c864e9c3bc3f5f7a99315c56830e811d"
  ]
},
"sources": {
  "entityIds": [
    "fc671272b2c7518ec738d6b17f08b7ed"
  ]
},
"unlinked": false,
"propagatable": false
}
,a2a35d9ba084495469290f8413e91ee": {
  "identity": "a2a35d9ba084495469290f8413e91ee",
  "type": "DATA_FLOW",
  "propagatorId": "cf0c7b96382b834021f51638b135da9e",
  "extractorRunId": "NOT_STORED",
  "targets": {
    "entityIds": [
      "fe5445fbd07d97c418d96200a218cae",
      "2b5a90798459c168d9fb5ecc2fc46aa5",
      "2690183198a2d6568188426ea9a82793",
      "4374529a3a4441bc6db30ee01bba3563",
      "66bff6e6eac3c17c5b7eb7c9035704eef",
      "9d7ca9e46a9e4624df05d10949a5fc6"
    ]
  },
  "sources": {
    "entityIds": [
      "fc671272b2c7518ec738d6b17f08b7ed"
    ]
  },
  "unlinked": false,
  "propagatable": false
}
,3a817d057f264d972c8c0e8dda4f9c3e": {
  "identity": "3a817d057f264d972c8c0e8dda4f9c3e",
  "type": "PARENT_CHILD",
  "propagatorId": "cf0c7b96382b834021f51638b135da9e",
  "extractorRunId": "NOT_STORED",
  "parent": {
    "entityId": "e9b3d1f7aee29134338df3de4cael100"
  },
  "children": {
    "entityIds": [
      "2690183198a2d6568188426ea9a82793"
    ]
  },
  "unlinked": false,
  "propagatable": false
}
,e0dfacd87d398fcd8ffbf91f30d675e43": {
  "identity": "e0dfacd87d398fcd8ffbf91f30d675e43",
  "type": "PARENT_CHILD",
  "propagatorId": "cf0c7b96382b834021f51638b135da9e",
  "extractorRunId": "NOT_STORED",
  "parent": {
    "entityId": "e9b3d1f7aee29134338df3de4cael100"
  },
  "children": {
    "entityIds": [
      "66bff6e6eac3c17c5b7eb7c9035704eef"
    ]
  }
}
Tracing through the relations shows that job_1396040412350_0003, which has the identity fc671272b2c7518ec738d6b17f08b7ed, has the INSTANCE_OF relation with wordcount and the DATA_FLOW relation with /user/hdfs/input and /user/hdfs/output.

### Tables

#### Displaying Table Schema

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

1. Perform a metadata search for an entity of source type Hive.
2. In the list of results, click a result entry of type Table. The Search screen is replaced with a page that displays the entity property sheet on the left and lineage diagram on the right.

3. Click the Schema tab. The table schema displays.

### Displaying Pig Tables

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

1. Perform a metadata search for an entity of source type Pig.
   - In the list of results, click a result entry of type Table.
   - 1. In the list of results, click a result entry of type Operation_Execution.
   2. Click the Tables tab. A list of links to tables involved in the operation displays.
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.

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.