Relativity Data Grid Guide

February 4, 2019 | Version 9.4
For the most recent version of this document, visit our documentation website.
Table of Contents

1 Relativity Data Grid ............................................................................................................. 8
  1.1 Data Grid compatibility matrix ....................................................................................... 8
  1.2 Data Grid components ...................................................................................................... 8
  1.3 Unsupported functionality .............................................................................................. 10

2 Data Grid system requirements ............................................................................................ 11
  2.1 Infrastructure considerations ......................................................................................... 11
  2.2 Data Grid cluster system requirements ......................................................................... 11
  2.3 Java version compatibility ............................................................................................... 12

3 Installing Data Grid ............................................................................................................... 13
  3.1 Creating master, client, and data nodes .......................................................................... 13
    3.1.1 Adding additional masters ....................................................................................... 19
    3.1.2 Java Virtual Machine (JVM) settings ...................................................................... 20
  3.2 Creating a cluster ............................................................................................................ 20
  3.3 Installing a monitoring cluster ........................................................................................ 21
  3.4 Installing the license ....................................................................................................... 23
    3.4.1 Setting up the Kibana server ...................................................................................... 24
    3.4.2 Installing Kibana ....................................................................................................... 25
    3.4.3 Installing Marvel and Sense ...................................................................................... 26
    3.4.4 Configuring Shield .................................................................................................... 26
    3.4.5 Updating your Data Grid cluster .............................................................................. 26
  3.5 Linking Relativity to the Data Grid cluster ................................................................. 27
  3.6 Configuring Shield authentication .................................................................................. 31
    3.6.1 Installing the Shield plugins ...................................................................................... 32
    3.6.2 Configuring custom authentication .......................................................................... 32
    3.6.3 Installing an X.509 certificate ................................................................................... 35
    3.6.4 Create a keystore and generate a keypair ............................................................... 35
    3.6.5 Create a certificate signing request (CSR) ............................................................. 36
    3.6.6 Send the certificate to your certificate authority (CA) for signing .......................... 36
    3.6.7 Install the signed certificate ....................................................................................... 36
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 Configuring Data Grid</td>
<td>39</td>
</tr>
<tr>
<td>4.1 Installing the Data Grid Core and Data Grid for Audit applications</td>
<td>39</td>
</tr>
<tr>
<td>4.2 Enabling your workspace and extracted text field for Data Grid</td>
<td>40</td>
</tr>
<tr>
<td>4.3 Enabling the Lucene Search search index</td>
<td>42</td>
</tr>
<tr>
<td>4.4 Data Grid agents</td>
<td>42</td>
</tr>
<tr>
<td>4.4.1 Data Grid Core agents</td>
<td>42</td>
</tr>
<tr>
<td>4.4.2 Data Grid for Audit agents</td>
<td>43</td>
</tr>
<tr>
<td>4.4.3 Adding a Data Grid agent</td>
<td>43</td>
</tr>
<tr>
<td>5 Monitoring Data Grid</td>
<td>47</td>
</tr>
<tr>
<td>5.1 Head</td>
<td>47</td>
</tr>
<tr>
<td>5.2 Marvel</td>
<td>47</td>
</tr>
<tr>
<td>5.3 Curator</td>
<td>49</td>
</tr>
<tr>
<td>5.3.1 Cluster metrics to monitor</td>
<td>49</td>
</tr>
<tr>
<td>5.3.2 Monitoring bulk rejections</td>
<td>50</td>
</tr>
<tr>
<td>5.3.3 Monitoring backup and recover times</td>
<td>50</td>
</tr>
<tr>
<td>5.3.4 Additional cluster monitoring information</td>
<td>50</td>
</tr>
<tr>
<td>5.4 Querying with Sense</td>
<td>51</td>
</tr>
<tr>
<td>5.4.1 Creating and deleting indexes</td>
<td>51</td>
</tr>
<tr>
<td>5.4.2 Retrieving documents and cluster settings</td>
<td>51</td>
</tr>
<tr>
<td>5.4.3 Shutting down a cluster</td>
<td>53</td>
</tr>
<tr>
<td>5.4.4 Backing up clusters</td>
<td>54</td>
</tr>
<tr>
<td>6 Backing up Relativity Data Grid</td>
<td>56</td>
</tr>
<tr>
<td>6.1 Creating a repository</td>
<td>56</td>
</tr>
<tr>
<td>6.2 Creating snapshots</td>
<td>58</td>
</tr>
<tr>
<td>6.2.1 Creating snapshots manually from within Sense</td>
<td>58</td>
</tr>
<tr>
<td>6.2.2 Scheduling a Windows task using Curator</td>
<td>59</td>
</tr>
<tr>
<td>6.3 Restoring a snapshot</td>
<td>65</td>
</tr>
<tr>
<td>6.3.1 Restoring snapshots from the Data Grid head console</td>
<td>65</td>
</tr>
<tr>
<td>6.3.2 Restoring snapshots with cURL</td>
<td>65</td>
</tr>
</tbody>
</table>
7 Upgrading from Data Grid 2.1.2 to 2.3.3.x ................................................................. 67
  7.1 Preparing the environment for upgrade ............................................................... 67
  7.2 Running the upgrade script ................................................................................. 68
  7.3 Verifying the upgrade ......................................................................................... 68
  7.4 Running a manual upgrade ................................................................................. 69
    7.4.1 Upgrading a node ......................................................................................... 69
    7.4.2 Verifying the upgrade ................................................................................... 72

8 Lucene Search ............................................................................................................ 74
  8.1 Using Lucene Search ........................................................................................... 74
  8.2 Lucene Search syntax considerations ................................................................. 78
    8.2.1 Phrases ......................................................................................................... 78
    8.2.2 Terms ............................................................................................................ 79
    8.2.3 Question mark single wildcard ..................................................................... 79
    8.2.4 Asterisk wildcard ......................................................................................... 80
    8.2.5 Fuzziness ...................................................................................................... 80
    8.2.6 Proximity ...................................................................................................... 81
    8.2.7 AND operator ............................................................................................... 82
    8.2.8 OR operator .................................................................................................. 83
    8.2.9 NOT operator ............................................................................................... 83
    8.2.10 Regular expressions (RegEx) ................................................................. 84
    8.2.11 Grouped queries ....................................................................................... 85
    8.2.12 Special Characters .................................................................................... 85
    8.2.13 Unsupported dtSearch syntax in Lucene Search ........................................ 86
  8.3 Lucene Search syntax help .................................................................................. 86
    8.3.1 Using syntax help for Lucene Search .......................................................... 87

9 Data Grid for Audit .................................................................................................... 89
  9.1 Home tab ............................................................................................................. 89
    9.1.1 Details filter ................................................................................................ 91
    9.1.2 ID filter ....................................................................................................... 92
    9.1.3 Timestamp filter ........................................................................................ 92
    9.1.4 Name filter .................................................................................................. 93
9.1.5 Action filter ................................................................. 94
9.1.6 Object Type filter .......................................................... 96
9.1.7 Execution Time filter ....................................................... 98
9.1.8 Artifact ID filter ............................................................ 98
9.1.9 User Name filter ............................................................ 99
9.1.10 Advanced filter ............................................................ 100
9.1.11 Saved Search filter ....................................................... 100
9.2 Saving and loading filter sets .............................................. 101
9.3 Adding widgets ............................................................... 103
  9.3.1 User Actions widget ..................................................... 104
  9.3.2 User Names widget ...................................................... 106
  9.3.3 Object Types widget ..................................................... 109
  9.3.4 Actions over Time widget ............................................... 111
  9.3.5 Results widget .......................................................... 113
9.4 Creating dashboards ........................................................ 117
9.5 Admin tab .................................................................. 118
  9.5.1 Migration Error Report .................................................. 118
  9.5.2 Migration Error Retry Script ........................................... 121
  9.5.3 Migration Status Report ............................................... 122
9.6 Audit Workspace Setting tab .............................................. 123

10 Searching documents over 100MB in a Data Grid enabled workspace ........................................... 125
  10.1 Importing documents using the Relativity Desktop Client .................................................... 125
    10.1.1 Ingesting long text fields into Data Grid .................................................. 125
    10.1.2 Identifying documents with long text errors ................................................... 125
    10.1.3 Storing long text fields with errors in SQL ................................................... 126
    10.1.4 Searching against your Data Grid and SQL Long Text fields ......................... 128
    10.2 Publishing a processing set ........................................................................ 128
      10.2.1 Ingesting long text fields into Data Grid .................................................. 128
      10.2.2 Identifying documents with long text errors .............................................. 129
      10.2.3 Storing long text fields with errors in SQL .............................................. 130
      10.2.4 Searching against your Data Grid and SQL Long Text fields ......................... 134
10.3 Migrating text using the Data Grid Text Migration application .................................................. 134
10.3.1 Ingesting long text fields into Data Grid .................................................................................134
10.3.2 Identifying documents with long text errors ..........................................................................134
10.3.3 Storing long text fields with errors in SQL ..............................................................................136
10.3.4 Searching against your Data Grid and SQL Long Text fields .............................................. 137
10.4 Running an OCR set ....................................................................................................................138
10.4.1 Ingesting long text fields into Data Grid .................................................................................138
10.4.2 Identifying documents with long text errors ..........................................................................138
10.4.3 Storing long text fields with errors in SQL ..............................................................................138
10.4.4 Searching against your Data Grid and SQL Long Text fields .............................................. 140
10.5 Searching against your Data Grid and SQL long text fields .................................................... 140
1 Relativity Data Grid

Relativity Data Grid is a NoSQL data store that allows you to horizontally distribute full text and audit data across any number of servers.

The benefits of using the Data Grid data store include a reduction in SQL Server database sizes; more performant database maintenance, backup, and upgrades; reduced memory requirements; automatic workspace distribution across available servers, increased visibility into Relativity audit data; and an increase in the natural limit of case sizes.

**Note:** Text or audit information stored in Relativity Data Grid is inaccessible for some third-party applications. It's recommended that you contact any vendors of third-party applications to confirm their compatibility with Relativity Data Grid.

The Data Grid data store relies on Elasticsearch as its underlying architecture. This page provides a brief description of Data Grid terminology, functionality, and important considerations to take into account before implementing the Data Grid data store for new workspaces.

**Note:** Data Grid supports Windows servers only.

1.1 Data Grid compatibility matrix

The following table lists the version of Data Grid compatible with each version of Relativity.

<table>
<thead>
<tr>
<th>Relativity 9.4</th>
<th>Relativity 9.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Grid 2.3.3</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

1.2 Data Grid components

The architecture on which Data Grid relies is made up of the following components:

- **Cluster** - a group of nodes with a common cluster name that share data. Data Grid automatically reorganizes data across nodes within a cluster as you add or remove nodes.

- **Node** - a running instance of Data Grid on a Java Virtual Machine (JVM). There are three node types:
  - **Master node** - the node within a cluster that manages changes across the entire cluster.
  - **Client node** - the node that serves as the gateway through which data enters a cluster. Relativity communicates with Data Grid through the client node.
  - **Data node** - a node that stores data within a cluster.

- **Index** - a store of related data that point to one or more shards.

- **Shard** - a container of data (multiple documents) on a node. There are two shard types:
  - **Primary shard** - a container of data on a node to which each document in the index belongs. You can have multiple primary shards. The number of primary shards determines the max-
The following diagram and accompanying steps depict how the different node types interact with each other within a cluster after Relativity sends data to Data Grid via import, processing, or audit migration. This diagram is meant for reference purposes only.

1. The master node tells the client node into which data node it should place the primary shard.
2. The client node places the primary shard into the data node specified by the master node. Data is written to this primary shard.
3. The master node tells the data node to replicate the primary shard.
4. The data node replicates the primary shard and sends the replica shard(s) to another data node based on a command from the master node.
5. Relativity continuously requests data from the endpoint of the client node.
6. The client node continuously supplies Relativity with data for Data Grid-enabled fields and searches.
## 1.3 Unsupported functionality

Once you enable a field’s access to the Data Grid data store, you can’t disable it, so it’s important to understand the benefits and limitations of allowing fields to access the Data Grid data store in new workspaces.

<table>
<thead>
<tr>
<th>Unsupported functionality</th>
<th>Possible workaround</th>
</tr>
</thead>
</table>
| You can’t use keyword search to search on Data Grid-enabled fields.                      | • Use Lucene Search which, like keyword search, is always up to date with no manual index builds. Lucene Search provides more search options than keyword search like proximity and fuzzy searching. For more information, see [Lucene Search on page 74](#).  
  • Build a dtSearch index from data in Data Grid.                                         |
| Analytics searches with default training set, default searchable set, or all documents in workspace selected as the data source don’t accurately incorporate fields with access to the Data Grid data store. | When building Analytics indexes, create an optimized saved search that contains fields with enabled access to the Data Grid data store enabled. |
| A Data Grid-enabled field can’t store more than 100mb of text, unlike a SQL field which can store up to 2GB. | To work around this issue, see [Searching documents over 100MB in a Data Grid enabled workspace on page 125](#). |
| The RAR Text Excerpt field can’t be made Data Grid-enabled.                              | The field is stored in SQL with no difference in behavior.                           |
| Transform Set Source and Destination fields do not support Data Grid-enabled long text fields | Where needed, store email fields in SQL. For example, “To”, “From”, “CC”.            |
| A document’s Data Grid-enabled field cannot be edited through a single edit, mass edit, or mass replace. | Keep any long text fields that require end-user updates in SQL, like “Attorney Notes”. If an edit must occur, a Desktop Client overlay can update text in a Data Grid-enabled field. |
| The “Export to File” mass operation will not export Data Grid-enabled field information. | Use the Desktop Client to perform an export of all information shown in a document list. |
| A Data Grid-enabled field cannot be made “Group By” or “Pivot” enabled.                  | If visualizing long text fields is required, store the long text field in SQL.        |
2 Data Grid system requirements

Depending on your infrastructure tier, you have different server specifications and recommendations for the Data Grid cluster available to you. Data Grid is built on a distributed architecture made up of many servers or nodes. A node is a running instance of Data Grid (a single instance of Data Grid running in the JVM). Every node in a Data Grid cluster can serve one of three roles.

- Master nodes are responsible for managing the cluster.
- Data nodes are responsible for indexing and searching of the stored data.
- Client nodes are load balancers that redirect operations to the node that holds the relevant data, while offloading other tasks.

Set up an entirely separate cluster to monitor Data Grid with one node that serves all three roles: master, data, and client. While this setup doesn’t take advantage of the distributed architecture, it acts as an isolated logging system that won’t affect the main cluster.

2.1 Infrastructure considerations

Consider the following factors when determining the infrastructure requirements for creating a Data Grid environment:

- **Infrastructure tier** - When you build out your initial Relativity environment, we use these measures to determine a tier level of 1, 2, or 3. This tier level takes into consideration the number of users, SQL sizes, and the amount of data and activity in your system.

- **Virtual versus physical servers** – We strongly recommend physical servers but do understand that they might not be a viable option for everyone.

- **Storage type** – Data Grid is a distributed system and you should run it on storage local to each server. For tier 1 systems, regular disks are acceptable, but as you move into tier 2 and 3 consider SSDs for their added performance benefits.

- **Network connectivity** – Because of the distributed architecture, network connectivity can impact performance, especially during peak activity. Consider 10 GBit as you move up to the higher tiers.

2.2 Data Grid cluster system requirements

The number of nodes required and the specifications for the nodes change depending on both your infrastructure tier and the amount of data that you plan to store in Data Grid. When building out your Data Grid infrastructure, use the following formula to calculate the total required number of data nodes:

**Size of Data in TBs * .8**

This formula is a shortened version of the following formula

**\((\text{Size of Data in TBs } \times 3 \times 1.2)/4.5\)**

Where:

- 3 refers to the total copies of the data
- 1.2 accounts for indexed data
4.5 refers to the maximum data allowed per node

Note the following considerations:

<table>
<thead>
<tr>
<th>Maximum shards per node</th>
<th>150</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recommended GBs per shard</td>
<td>30</td>
</tr>
<tr>
<td>Total copies of the data</td>
<td>3</td>
</tr>
</tbody>
</table>

Contact support@relativity.com so that our infrastructure team can review the amount of activity and monitoring data you want to store in Data Grid and provide a personalized recommendation of monitoring nodes required.

### 2.3 Java version compatibility

See the Elastic website for compatible Java versions.
3 Installing Data Grid

The procedure for installing Data Grid to your Relativity environment is the following:

1. Complete the pre-installation steps. See the Pre-Installation Guide.
2. Install the software on which Data Grid runs. This includes:
   a. Master, client and data nodes. See Creating master, client, and data nodes below
   c. A monitoring cluster. See Installing a monitoring cluster on page 21
3. Install and configure the Relativity components. These include the following:
   a. The license and plugins. See Installing the license on page 23
   b. The Instance Settings that accompany the Data Grid Core application. See Linking Relativity to the Data Grid cluster on page 27.
   c. Data Grid plugins. See Setting up the Kibana server on page 24.
   d. Shield authentication. See Configuring Shield authentication on page 31

Once all installation steps are completed, you must configure the Relativity core application. See Configuring Data Grid on page 39.

Note: Ensure that your antivirus software excludes the Data Grid data location (data node) from scanning. See the Environment Optimization Guide.

3.1 Creating master, client, and data nodes

Three different types of nodes are required to run Data Grid in your environment:

- Master - the node within a cluster that manages changes across an entire cluster.
- Client - the node within a cluster that either serves as the gateways through which data enters a cluster.
- Data - the node that stores data within a cluster.

The Tier 1 installation of the Data Grid data store involves configuring one or more dedicated master nodes, one dedicated client node, one monitoring node, and two data nodes. The master node(s) manage the overall state of the cluster, delegates the structure by which data is stored and replicated across a cluster, and facilitates resources returned during searches. The client node serves as the endpoint through which you import data into the Data Grid cluster. The data nodes store primary shards and replica shards.

The account running the Data Grid service requires access to SQL Server, and specifically needs to have read, write, and bulk permissions for all workspace databases.

For more information on cluster specifications for each tier, see Data Grid system requirements on page 11.
To install a master, client, or data node, perform the following steps. Note that the settings within the Elasticsearch.yml determine which type of node you’re installing, but the remaining steps apply to all node types.

1. Install the Java Runtime Environment. See the Elastic website for compatible Java versions. For questions, email support@relativity.com.

2. Use the following steps to insert an environment variable (KCURA_JAVA_HOME).
   a. Click Start.
   b. Right-click on Computer and select Properties.
   c. Click Advanced system settings.
   d. Select the Advanced tab.
   e. Click Environment Variables…
   f. Click New under System Variables.
   g. Name the variable KCURA_JAVA_HOME.
   h. Copy the file path to C:\Program Files\Java\jdk(version_number)

3. Contact your Relativity Account Manager to download the Data Grid installer package.

4. Extract the Elasticsearch zip folder to a root directory (Example: C:\RelativityDataGrid).
5. Rename the default directory `\RelativityDataGrid\elasticsearch-2.x.x` to `\RelativityDataGrid\elasticsearch-main`. This allows you to make upgrades to Data Grid without having to modify the folder to accommodate future version numbers.

6. Navigate to `\RelativityDataGrid\elasticsearch-main\config\elasticsearch.yml` and update the settings in the following Elasticsearch.yml file in a text editor. This is where you'll specify what type of node you're creating:

   **Note:** When entering these values, you must enter a single space between the field name and the field value.

   - `cluster.name` - enter the name of the cluster. All nodes on the cluster must share a common name in order to communicate with one another.
     
     **Note:** You can't use hyphens to name a cluster. Use underscores if necessary.

   - `node.name` - enter the fully qualified domain name.

   - `node.master` - enter `true` because this is a master node. Enter `false` for client and data nodes.

   - `node.data` - enter `false` for a master and client node. Enter `true` for a data node.

   - `discovery.zen.minimum_master_nodes` - enter 1

   - `discovery.zen.ping.multicast.enabled` - enter `false`

   - `discovery.zen.ping.unicast.hosts` - enter the full computer name of all nodes on the cluster using the following format: `"DN_FULL_COMPNAME_1","DN_FULL_COMPNAME_2","DN_FULL_COMPNAME_3"`. Always include the master node in the list of hosts.

     **Note:** Don't put spaces between the commas when specifying multiple nodes. The default port number is 9300, but you can specify a port number after the computer name with the following format: "DN_FULL_COMPNAME_1:9500". The specified port must be free.

   - `marvel.agent.enabled` - enter `false`

   - `action.destructive_requires_name` - enter `true`. This prevents destructive actions w/ wild-cards. For example, DELETE /*.

   - `action.auto_create_index` - enter `false`. This disables automatic index creation.

   - `format` - enter `json`. This is for global cluster state file to be readable.

   - `transport.tcp.compress` - enter `true`

   - `http.max_content_length` - enter `201mb`

   - `http.cors.enabled` - enter `false`

   - `gateway.expected_master_nodes` - enter 0. This delays cluster recovery, providing additional time for all nodes to first come online.

   - `gateway.expected_data_nodes` - enter 0

   - `gateway.recover_after_time` - enter 5m
- `script.default_lang` - enter `groovy`
- `script.groovy.sandbox.enabled` - enter `true`
- `path.data` - enter the path of the location in which you want to store the allotted index data for this node. For example, `C:\RelativityDataGrid\data`.
- `path.repo` - enter the path of the location in which you want to store backups. For example, `C:\RelativityDataGrid\backups`.

**Note:** You can specify multiple backup locations with the following format: `[/mount/backups, /mount/longterm_backups]`. You can also specify a network share with the following format: `[/\\my_server\\snapshots]`.

- Configure the Shield settings as follows:

  **Note:** To disable Shield, remove the number sign (`#`) in front of `shield.enabled: false`.

```groovy
#shield.enabled: false
shield.authc.realms:
  custom:
    type: kCuraBearerRealm
    order: 0
    publicJwksUrl: https://<server>/Relativity/Identity/.well-known/jwks
  esusers1:
    type: esusers
    order: 1
```

**Note:** The URL must point to the Relativity installation where Identity Server can be found. This should be the same URL used to log in to Relativity.

7. If using Shield, see Configuring Shield authentication on page 31 for steps on configuring users and roles.

8. Use the following steps to launch the command prompt:
   a. Click **Start**.
   b. Search for "**cmd**".
   c. Right-click on **cmd.exe** and select **Run as Administrator**.

9. Navigate to the bin directory in the RelativityDataGrid folder (`\RelativityDataGrid\elasticsearch-main\bin`) by running `cd \RelativityDataGrid\elasticsearch-main\bin` and enter the following commands:
   a. Install the Windows service: `\kservice.bat install`
   b. Run the GUI manager: `\kservice.bat manager`
c. Use the drop down menu for **Startup Type** to select **Automatic**.

![Elasticsearch (elasticsearch-service-x64) Properties](image)

- **Service Name**: elastsearch-service-x64
- **Display name**: Elasticsearch (elasticsearch-service-x64)
- **Description**: Elasticsearch Windows Service - http://elasticsearch.org
- **Path to executable**: C:\RelativityDataGrid\bin\elasticsearch-service-x64.exe /RS://elasticsearch
- **Startup type**: Automatic

**Service Status**: Started

- **Start**
- **Stop**
- **Pause**
- **Restart**

**OK**  **Cancel**  **Apply**

---

d. Click the **Java** tab and configure the **Maximum memory pool** to be **30 GB** or half of the total RAM available (whichever is less). You can enter identical values for the **Initial memory pool** and **Maximum memory pool** settings. It's recommended that you keep the **Thread stacks size** value at its default of 256.
e. Select the Log On tab. In the Log on as setting, select This account. Enter a valid Relativity service account domain name and password and confirm the password.

![Elasticsearch Properties](image)

- Local System account
- This account: domain\relativity.service
- Password: ********
- Confirm Password: ********

f. Click OK.

g. Return to the command prompt.

h. Start the Windows service: \kservice.bat start

**Note:** Don’t start the Windows service on a node until you complete step 5 above to configure the elasticsearch.yml file on the node.

10. Verify the installation is complete by navigating to the following address in a Chrome browser:  
    http://computename:9200/_plugin/head or http://localhost:9200/_plugin/head  

**Note:** This page might not properly display in Internet Explorer. Use Chrome for verification.

### 3.1.1 Adding additional masters

Data Grid 2.x allows for multiple masters with a minimum of 3 masters. With multiple masters, your cluster is highly available. If a master node goes down, one of your additional master nodes can be elected and your cluster will continue in an active green state.

Set up your additional two master nodes per the instruction for installing Data Grid nodes.

The following needs to be updated in the elasticsearch.yml file for all master nodes:  
**discovery.zen.minimum_master_nodes: 2**
This will ensure that if any master goes offline, the third one joins the cluster forming a quorum.

For more information please see Elastic documentation here.

**Note:** Enabling multiple masters is optional. If you would like to stay at 1 master, move on to the next step to install plugins.

### 3.1.2 Java Virtual Machine (JVM) settings

You must also configure the following settings on each JVM for JVM logging:

1. Run the Elasticsearch GUI manager: `.\kservice.bat manager`
2. Select default logging to "Error" instead of "Info".
3. Update JVM memory to half of total RAM of the machine or 30GB (whichever is less).
4. Start each of the nodes services: `.\kservice.bat start`

**Note:** Never restart a node if data already exists on the cluster. If there is no data in the cluster, you can restart the service if it's already started and configuration settings change. See [Restarting nodes and clusters](#) for more information.

Clusters automatically form between nodes on a network if the cluster names are the same across nodes.

### 3.2 Creating a cluster

Clusters automatically form between multiple nodes when the same cluster name is specified across multiple nodes in the `elasticsearch.yml` file. You can check to see if your various nodes have joined the cluster by checking Head. Navigate to [http://computername:9200/_plugin/head](http://computername:9200/_plugin/head) or [http://localhost:9200/_plugin/head](http://localhost:9200/_plugin/head) and see if all of your nodes are present.
3.3 Installing a monitoring cluster

A monitoring cluster allows you to store Marvel data from the production cluster for analysis. A monitoring cluster for Data Grid only needs one node, but you can set up a multi-node monitoring cluster if you prefer. We recommend using no more than three monitoring cluster nodes.

Use the following steps to install Data Grid 2.x on a machine that you want to use as a single node monitoring cluster:

1. Install the Java Runtime Environment. You can acquire the correct version by emailing sup-port@relativity.com.

2. Use the following steps to insert an environment variable (KCURA_JAVA_HOME):
   1. Click Start.
   2. Right-click on Computer and select Properties.
   3. Click Advanced system settings.
   4. Select the Advanced tab.
   5. Click Environment Variables...
6. Click **New** under **System Variables**.
7. Name the variable **KCURA_JAVA_HOME**.
8. Copy the file path to `C:\Program Files\Java\jdk(version_number)`

---

3. Contact your Relativity Account Manager to download and unzip the Data Grid installer package.
4. Extract the Elasticsearch zip folder to a root directory (Example: C:\RelativityDataGrid).
5. Rename the default directory (C:\RelativityDataGrid\elasticsearch-2.1.x) to `RelativityDataGrid\elasticsearch-main`. This allows you to make upgrades to Data Grid without having to modify the folder to accommodate future version numbers.
6. Extract and copy the Relativity Data Grid package to each node on your monitoring cluster.
7. Navigate to `RelativityDataGrid\elasticsearch-main\config\elasticsearch.yml` and update only the following settings in the `elasticsearch.yml` file in a text editor.

**Note:** When entering these values, you must enter a single space between the field name and the field value.

- **cluster.name** - enter the name of the monitoring cluster. All nodes on the cluster must share a common name in order to communicate with one another. Do not confuse this cluster name with the name of the production cluster. This cluster name should be the name of the
monitoring cluster.

- **node.name** - enter the name of the monitoring node.
- **node.master** - enter **true** because this node performs the same tasks as a master node.
- **node.data** - enter **true**. You can use the master node to store data for a single node monitoring cluster, but don’t store data on the master node for Tier 2 or Tier 3 setups on a production cluster.
- **action.auto_create_index** - enter **true**.

8. If you have Shield enabled on your cluster, create a user with a kibana4_server role and marvel_remote_agent role, both of which are defined in roles.yml. For more information on defining roles, see [Configuring Shield authentication on page 31](#).

### 3.4 Installing the license

Install the Elasticsearch license from the search/bin folder. If you haven’t already, submit a support ticket to Relativity for a production grade license for Elasticsearch plugins. While you wait for a license, you can continue using the plugins on a trial period.

To install the license, complete the following in Sense or Head:
1. Copy the contents of the license and run `PUT/_license<<contents_of_license_file.json>>`

2. Verify that the expiration date is set to a year from now by running the command `GET/_license`.

3.4.1 Setting up the Kibana server

Extract the Kibana 4.5.3 package on your monitoring cluster, and then use the following steps to set up the Kibana server on your monitoring cluster.

The following table breaks down the compatibility of different versions of Data Grid, Marvel, and Kibana.

<table>
<thead>
<tr>
<th>Data Grid version</th>
<th>Marvel version</th>
<th>Kibana version</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3.3</td>
<td>2.3.3</td>
<td>4.5.3+</td>
</tr>
</tbody>
</table>
3.4.2 Installing Kibana

Kibana is a requirement for Marvel which allows you to manage your Data Grid indexes. To use Kibana with Data Grid, you need the following:

- Data Grid 2.1 or later.
- One of the following supported browsers: IE9, IE10+, Firefox, Chrome, Safari (Mac).
- The endpoint of the master node of the Data Grid instance you’re connecting to.
- Which Data Grid indexes you want to search.

To install and start Kibana:

1. Download the Kibana 4.5.x binary package for your platform.
2. Extract the .zip or tar.gz archive file.
3. After installing, run Kibana from the install directory: bin/kibana (Linux/MacOSX) or bin\kibana.bat (Windows).
4. Navigate to the kibana.yml file, and update the following in a text editor:
   - `elasticsearch.url` - enter the name of your monitoring cluster.
5. If you have Shield enabled on your cluster, also update the following kibana.yml settings in a text editor:
   - `kibana_elasticsearch_username` - enter the name of the user with a kibana4_server role defined in roles.yml.
   - `kibana_elasticsearch_password` - enter the password for the user who has the kibana4_server role defined in roles.yml.
   - `elasticsearch.ssl.ca` - enter PEM file path.
   - `elasticsearch.ssl.verify` - enter false.
   - `path.data` - enter the path of the location in which you want to store the allotted data for this node. For example, C:\RelativityDataGrid\data.
   - `path.repo` - enter the path of the location in which you want to store backups. For example, C:\RelativityDataGrid\backups. You can specify multiple backup locations with the following format: ["/mount/backups", "/mount/longterm_backups"]. You can also specify a network share with the following format: ["\\my_server\snapshots"].
   - `server.port` - enter the port where the Kibana server runs. Defaults to 5601.
   - `server.host` - enter the IP address where the Kibana server runs. Defaults to 0.0.0.0 (localhost).
6. Browse to the kibana/bin folder and enter the following:
   - kibana plugin -i elasticsearch/marvel/2.3.3
   - kibana plugin -i elastic/sense
7. Start the Kibana server by running the following:
   - bin/kibana.bat


### 3.4.3 Installing Marvel and Sense

Marvel is a management and monitoring tool for Data Grid. Marvel aggregates cluster wide statistics and events and offers a single interface to view and analyze them.

Sense is a Kibana plugin and visual console that provides auto-complete, auto-indentation, and syntax checking.

To install Marvel and Sense:

1. Refer to the [Marvel requirements](#) and [Sense requirements](#).
2. Open the command prompt and go to the `elasticsearch/bin` directory. Enter the following command: `plugin install marvel-agent`
3. Browse to the Kibana/bin folder and enter the following:
   - kibana plugin --install elasticsearch/marvel/2.3.3
   - kibana plugin --install elastic/sense

### 3.4.4 Configuring Shield

When you install Data Grid 2.3.3, Shield is enabled by default; however, you still need to configure Shield's authentication on every node. For more information, see [Configuring Shield authentication](#).

**Note:** With Shield on by default, other plugins like Marvel or Head are not supported. In order to use your other plugins, you need to provide the Kibana server with credentials so it can access the .kibana index and monitor the cluster. See [Configuring Shield authentication](#) for more information.

### 3.4.5 Updating your Data Grid cluster

To finish setting up your monitoring cluster, the following changes need to be made on all nodes in the Data Grid cluster.

Complete the following on all nodes in the Data Grid cluster:

1. Install the marvel-agent plugin by running the command below.
   - bin\plugin install marvel-agent
2. If your monitoring cluster has Shield enabled, navigate to `C:\RelativityDataGrid\elasticsearch-main\config\elasticsearch.yml` and update the elasticsearch.yml file in a text editor with the following. Set the `auth.username` and `auth.password` to the user who has the remote_marvel_agent role defined in roles.yml

```yaml
marvel.agent.exporters:
  id1:
    type: http
```
3. Restart the Data Grid service for the changes to take effect.

3.5 Linking Relativity to the Data Grid cluster

After you've installed the Data Grid Core application and added the accompanying agents in the Agents tab, you'll need to configure all the entries in the Instance Settings tab that allow Relativity to communicate with the Data Grid service.

This will enable Relativity to save data into the cluster you created as part of the Data Grid component installation.

Use the following steps to link Relativity to your Data Grid service:

1. Specify the URL and port of the client node in the DataGridEndPoint instance setting value (i.e., http://CLIENT_NODE_COMPNAME:9200). This is the endpoint that Relativity uses to ingest data into the cluster. If you have multiple client nodes, you can specify all of them as endpoints. Separate your endpoints by commas.

2. Enter a unique prefix for this instance of Relativity in the DataGridIndexPrefix instance setting value. The prefix allows you to differentiate data from instance to instance within the cluster.

3. Set the number_of_shards and number_of_replicas fields in the DataGridIndexCreationSettings and DataGridIndexCreationSettings instance setting value to match your cluster setup. The number_of_shards field should be one shard per data node, and the number_of_replicas field should be two. The default instance setting value for number_of_shards is set to four, which assumes four data nodes. The default instance setting value for number_of_replicas value is two.

Use the following steps to edit the DataGridIndexCreationSettings instance setting value to match your particular cluster setup. The instance setting value holds the template for creating a text index (DataGridIndexCreationSettings) in Data Grid.

a. Enter the following for the DataGridIndexCreationSettings entry in Instance Settings, edit the number_of_shards and number_of_replicas fields to match your preferred settings, and paste the updated text into the text box.

```json
{
    "template": "relativity_*",
    "aliases": {
        "{index}_read": {},
        "{index}_write": {},
        "{index}_verify": {}
    },
    "settings": {
        "index": {
            "number_of_shards": 12,
            "number_of_replicas": 2
        },
        "analysis": {
```
b. You must install the Data Grid for Audit application from the library in at least one workspace in order to use Data Grid for Audit and to create audit indexes. This installation creates your necessary Data Grid for Audit entries in the Instance Settings tab.

c. Enter a unique prefix for this instance of Relativity in the ESIndexPrefix instance setting value. The prefix allows you to differentiate data from instance to instance within the cluster.

d. Set the number of shards and number of replicas fields in the ESIndexCreationSettings instance setting value to match your cluster setup. The number of shards field should be twice the number of nodes on the cluster, and the number of replicas field should be two. Use the following steps to edit the ESIndexCreationSettings and instance setting value to match your particular cluster setup. The instance setting value holds the template for creating an audit index (ESIndexCreationSettings) in Data Grid.

e. The ESIndexCreationSettings value populates with the default template that includes settings for four shards and two replicas. Edit the ESIndexCreationSettings entry in Instance Settings to update the template if your setup requires shard or replication modifications. Always update the template through these instance settings to keep templates consistent across
Relativity and Data Grid.

**Note:** If you already migrated audit data into Data Grid through the Data Grid for Audit application, you must manually delete the template in Data Grid (DELETE /_template/audit and DELETE /_template/document) before updating the template with any new changes. Contact support@relativity.com if you require assistance.

f. Use the following command to edit the ESIndexCreationSettings instance setting value to match your particular cluster setup:

```json
{
  "template": "audit_*",
  "aliases": {
    "{index}_read": {
      },
    "{index}_write": {
      },
    "{index}_verify": {
      }
  },
  "settings": {
    "index": {
      "number_of_shards": 4,
      "number_of_replicas": 2
    },
    "analysis": {
      "analyzer": {
        "str_search_analyzer": {
          "tokenizer": "keyword",
          "filter": ["lowercase",
                      "substring"]
        },
        "str_index_analyzer": {
          "tokenizer": "keyword",
          "filter": ["lowercase",
                      "substring"]
        },
        "lwhitespace": {
          "tokenizer": "whitespace",
          "filter": ["lowercase"]
        }
      },
      "filter": {
        "substring": {
          "type": "nGram",
          "min_gram": 1,
          "max_gram": 20
        }
      }
    },
    "mappings": {
      "audit": {
        "dynamic_templates": [{
          "raw": {
            "match_pattern": "regex",
            
```
"path_match": "Details\\.auditElement\\..*",
"mapping": {
    "type": "string",
    "fields": {
        "Raw": {
            "analyzer": "lwhitespace",
            "type": "string"
        }
    }
},
"match_mapping_type": "string"
},

"newvalue": {
    "match_pattern": "regex",
    "path_match": ".*\\.newValue$",
    "mapping": {
        "type": "string",
        "fields": {
            "Raw": {
                "analyzer": "lwhitespace",
                "type": "string"
            }
        }
    }
}
},

"oldvalue": {
    "match_pattern": "regex",
    "path_match": ".*\\.oldValue$",
    "mapping": {
        "type": "string",
        "fields": {
            "Raw": {
                "analyzer": "lwhitespace",
                "type": "string"
            }
        }
    }
}
},

"analytics_text": {
    "match_pattern": "regex",
    "path_match": ".*\\.#text$",
    "mapping": {
        "type": "string",
        "fields": {
            "Raw": {
                "analyzer": "lwhitespace",
                "type": "string"
            }
        }
    }
}
},

"_timestamp": {
}
4. Change the **Enable Data Grid** setting on the workspace details tab from **No** to **Yes**.

5. Change the **Enable Data Grid** setting for the long text field for extracted text from **No** to **Yes**. On import, all data mapped to this field saves to the Data Grid data store.

6. For information on setting up Data Grid for Audit, see Installing the Data Grid Core and Data Grid for Audit applications on page 39.

**Note:** You can't migrate long text fields that already contain data from the SQL Server to the Data Grid data store.

### 3.6 Configuring Shield authentication

Shield allows you to easily protect your data from unintentional modification or unauthorized access while simplifying your architecture. Without Shield, anyone who knows the correct endpoint or URL can query, update, or delete all data in Data Grid. With default Shield installation, only Relativity can access Data Grid data, and all traffic between servers is encrypted. In addition, system admins can specify custom authentication realms so that targeted users can interact with Data Grid data outside of Relativity (via plugins and endpoints).

When you install Data Grid 2.3.3, Shield is enabled by default; however, you still need to configure Shield’s authentication on every node.

The basic procedure for configuring Shield authentication is the following:

1. Install the authenticate Shield plugin on every node. Installing the Shield plugins on the next page.
2. Configure custom authentication. See Configuring custom authentication on the next page.
4. Enable SSL in the node configuration. See Enabling SSL in the node configuration on page 37.
5. Restart Data Grid.
Note: With Shield on by default, other plugins like Marvel or Head aren't operational until you configure Kibana to use your other plugins. For information on configuring Kibana, see Configuring Kibana to work with Shield.

3.6.1 Installing the Shield plugins
To install the authenticate and Shield plugins, perform the following steps on every node:

1. Copy the authenticate and Shield plugins from the elasticearch-main\plugins folder in the zip file to the plugins folder.
2. Restart Data Grid.

3.6.2 Configuring custom authentication
With Shield authentication, you can also specify custom authentication that allows targeted users to interact with Data Grid data outside of Relativity. To specify custom authentication, you must first define roles in the roles.yml file, located in `<installDirectory>\plugins\shield\config\roles.yml`, and assign users to roles using your preferred authentication realm.

3.6.2.1 Defining roles
The roles.yml file contains a default role called admin, which Relativity uses to access the data stored in Relativity. The admin role has complete system admin rights to all data stored in Data Grid.

Note: The node manages the roles.yml file locally; the cluster doesn't manage the yml file globally. This means that with a typical multi-node cluster, you need to apply the exact same changes on each and every node in the cluster. A safer approach would be to apply the change on one of the nodes and have the roles.yml distributed/copied to all other nodes in the cluster (either manually or using a configuration management system such as Puppet or Chef).

If you want to create a new role with a different permission set, you must configure the roles.yml file to specify the unique role name(s) and the cluster and indices permissions associated with it. For example:

```yaml
# All cluster rights
# All operations on all indices
admin:
  cluster: all
  indices:
    '*':
      privileges: all

# Monitoring cluster privileges
# All operations on all indices
power_user:
  cluster: monitor
  indices:
    '*':
      privileges: all

# Only read operations on indices
user:
  indices:
    '*':
      privileges: read
```
# Only read operations on indices named events_*

```bash
events_user:
    indices:
        'events_*':
            privileges: read
```

See the complete list of cluster and indices privileges [here](#).

Once you create your roles, you can create users using any of the following authentication realms and associate them to a role to determine the level of permissions they have to access data stored in Data Grid.

## 3.6.2.2 Native realm

The default Shield realm is the native realm. In the native realm, all user operations occur within the Elasticsearch cluster. The native realm stores and distributes users with the Elasticsearch cluster. Users authenticate with a username and password.

To add or update a user to the native realm, submit a PUT or POST request to the `/_shield/user/<username>` endpoint. The username must be at least 1 character and no longer than 30 characters. The first character must be a letter (a-z or A-Z) or an underscore (_). Subsequent characters can be letters, underscores (_), digits (0-9), or any of the following symbols: @, -, . or $.

```json
POST /_shield/user/username
{
    "password": "P@ssword1",
    "roles": [ "admin", "other_role1" ],
    "full_name": "Jane Smith",
    "email": "jsmith@example.com",
    "metadata": {
        "intelligence": 7
    }
}
```

Note the following considerations:

- **password** - the password for the user. Passwords must be at least 6 characters long.
- **roles** - determines the user's access privileges. You must assign at least one role to the user.
- **full_name** - (Optional) the user's full name.
- **email** - (Optional) - the user's email address.
- **metadata** - (Optional) arbitrary metadata you want to associate with the user.

## 3.6.2.3 Elastic Shield realm

Using the Elastic Shield realm allows you to add a user to your cluster and specify a password. This realm is not recommended because it potentially undermines Shield authentication.

To configure the Elastic Shield realm:

1. Open the command prompt and navigate to `elasticsearch/bin/shield`.
2. Enter the following command to add a new user to your cluster and specify a password for that user:
   - `esusers useradd <username> -p <secret>`
3. To add users to roles, enter the following command.

   - esusers roles <username> -a <comma-separated list of roles> -r <comma-separated list of roles>

   The -a option adds a comma-separated list of roles to a user. The -r option removes a comma-separated list of rules from a user.

### 3.6.2.4 LDAP realm

To integrate with LDAP you must configure an LDAP realm and assign LDAP groups to Shield roles in the role mapping file.

To configure the LDAP realm with User search, add the following settings to the elasticsearch.yml file in a text editor:

```yaml
shield.authc.realms:
  ldap1:
    type: ldap
    order: 1
    url: "LDAP://Test-DC01.testing.corp:389"
    bind_dn: "CN=Administrator,CN=Users,DC=testing,DC=corp"
    bind_password: P@ssw0rd@1
    user_search:
      base_dn: "DC=testing,DC=corp"
      attribute: sAMAccountName
    group_search:
      base_dn: "DC=testing,DC=corp"
```

Settings with User template

To configure the LDAP realm with User templates, add the following settings to the elasticsearch.yml file in a text editor:

```yaml
ldap2:
  type: ldap
  order: 2
  url: "LDAP://Test-DC01.testing.corp:389"
  user_dn_templates:
    - "cn={0},OU=Testing - Users,DC=testing,DC=corp"
  group_search:
    base_dn: "DC=testing,DC=corp"
```

To assign LDAP groups to Shield roles, add the following settings for your user to the role_mapping.yml file in a text editor:

```yaml
admin:
  - "OU=Testing - Admins,DC=testing,DC=corp"
  - "OU=Testing - Users,DC=testing,DC=corp"
  - "CN=Jane Smith,OU=Testing - Users,DC=testing,DC=corp"
```

### 3.6.2.5 Active Directory (AD) realm

To configure the Active Directory realm, add the following settings to the elasticsearch.yml file in a text editor:

```yaml
active_directory:
  type: active_directory
  order: 2
  domain_name: testing.corp
```
3.6.3 Installing an X.509 certificate

Installing an X.509 certificate consists of the following tasks:

1. Create a keystore and generate a node certificate.
2. Create a certificate signing request (CSR).
3. Send the certificate to your certificate authority (CA) for signing OR set up your certificate authority to sign the certificate.
4. Add the signed certificate to the node's keystore.

3.6.4 Create a keystore and generate a keypair

1. Create a keystore and import your certificate authority's (CA) certificate or a trusted certificate using Java Keytool. This process configures the node to trust certificates signed by the CA. For example, the following command creates a keystore for node01 and imports the CA certificate cacert.pem. Local self-signed certificates are not recommended. The keytool can be found in the following:

   ```
   cd \Program Files\Java\jdk1.8.*\bin keytool -importcert -keystore node01.jks -file cacert.pem -alias my_ca
   ```

   The Java keystore file (.jks) securely stores certificates for the node. The CA cert must be a PEM encoded certificate. If you need to convert your certificate to PEM file, you can use a tool such as OpenSSL to convert a certificate.

   When you create a keystore, you are prompted to set a password. This password protects the integrity of the keystore. You need to provide it whenever you interact with the keystore.

   **Note:** When the CA certificate expires, you must update the node's keystore with the new CA certificate.

   You can also store the CA certificate in a separate truststore. For more information, see [Configuring a truststore](#).

2. Generate a private key and certificate for the node with Java Keytool. For example, the following command creates a key and certificate for node01:

   ```
   keytool -genkey -alias node01 -keystore node01.jks -keyalg RSA -keysize 2048 -validity 712 -ext san=dns:node01.example.com,ip:192.168.1.1
   ```

   This command creates an RSA private key with a key size of 2048 bits and a public certificate that is valid for 712 days. The key and certificate are stored in the node01.jks keystore.

   The **san** value specifies all alternative names for the node. The generated certificate is valid for the DNS names and IP addresses specified as alternative names. You can specify multiple DNS or IP address entries as a comma-separated list.
When you run `keytool -genkey`, Keytool prompts you for the information needed to populate the node’s distinguished name that’s stored the certificate. Use a trusted domain wildcard certificate that can be trusted among many nodes. For example:

```plaintext
What is your first and last name?
[Unknown]: *.domain.corp

What is the name of your organizational unit?
[Unknown]: test

What is the name of your organization?
[Unknown]: Relativity

What is the name of your City or Locality?
[Unknown]: Chicago

What is the name of your State or Province?
[Unknown]: Illinois

What is the two-letter country code for this unit?
[Unknown]: US

Is CN=*.domain.corp, OU=test, O=Relativity, L=Chicago, ST=Illinois, C=US correct?
[no]: yes

Enter key password for <node01>
(RETURN if same as keystore password):
```

### 3.6.5 Create a certificate signing request (CSR)

A node’s certificate needs to be signed by a trusted CA for the certificate to be trusted. To get a certificate signed, you need to create a certificate signing request (CSR) and send it to your CA.

To create a CSR with Java Keytool, use the `keytool t-certreq` command. You specify the same alias, keystore, key algorithm, and DNS names and IP addresses that you used when you created the node certificate. Specify where you want to store the CSR with the `-file` option.

```bash
keytool -certreq -alias node01 -keystore node01.jks -file node01.csr -keyalg rsa -ext san=dns:*domain.corp,ip:192.168.1.1
```

### 3.6.6 Send the certificate to your certificate authority (CA) for signing

To get a signed certificate, send the generated CSR file to your CA. The CA will sign it and send you the signed version of the certificate.

### 3.6.7 Install the signed certificate

To install the signed certificate, use `keytool -importcert` to add it to the node’s keystore. You specify the same alias and keystore that you used when you created the node certificate.

```bash
cd CONFIG_DIR/shield
keytool -importcert -keystore node01.jks -file node01-signed.crt -alias node01
```
3.7 Enabling SSL in the node configuration

Once you've added the signed certificate to the node's keystore, you need to modify the node configuration to enable SSL then restart Data Grid.

To enable SSL, make the following changes in elasticsearch.yml:

1. Specify the location of the node's keystore and the password(s) needed to access the node's certificate. For example:

   ```yaml
   shield.ssl.keystore.path: /home/es/config/shield/node01.jks
   shield.ssl.keystore.password: myPass
   shield.ssl.keystore.key_password: myKeyPass
   shield.ssl.hostname_verification: false
   ```

   The first line indicates the full path to the node keystore file. This must be a location within the Data Grid configuration directory.

2. Enable SSL on the transport networking layer to ensure that communication between nodes is encrypted:

   ```yaml
   shield.transport.ssl: true
   ```

3. Enable SSL on the HTTP layer to ensure that communication between HTTP clients and the cluster is encrypted:

   ```yaml
   shield.http.ssl: true
   ```

4. Restart Data Grid so these configuration changes take effect.

3.8 Configuring Kibana to work with Shield

When you install or update to Relativity 9.4, Shield is enabled by default. With Shield on by default, other plugins like Marvel or Head are not supported. In order to use your other plugins, you need to provide the Kibana server with credentials so it can access the .kibana index and monitor the cluster.

To configure credentials for the Kibana server:

1. Assign the `kibana4_server` role to a user in Shield. For more information, see [Configuring a Role for the Kibana 4 Server](#) in the Shield documentation.

2. Set the `kibana_elasticsearch_username` and `kibana_elasticsearch_password` properties in `kibana.yml` to specify the credentials of the user you assigned to the `kibana4_server` role:

   ```yaml
   kibana_elasticsearch_username: kibana4-user
   kibana_elasticsearch_password: kibana4-password
   ```

3. Update the following setting in kibana.yml to false:

   ```yaml
   elasticsearch.ssl.verify: false
   ```
Kibana 4 users also need access to the .kibana index so they can save and load searches, visualizations, and dashboards. For more information, see Configuring Roles for Kibana 4 Users in the Shield documentation.
4 Configuring Data Grid

The Data Grid Core application contains all the required infrastructure components you need to use Data Grid in your environment. Install the Data Grid Core application a workspace in your environment.

Before configuring the Data Grid application, make sure you have completed the following:

- Completed the pre-installation steps. For more information, see .
- Installed Data Grid to your Relativity environment. For more information, see Installing Data Grid on page 13.

4.1 Installing the Data Grid Core and Data Grid for Audit applications

To install the Data Grid Core application, perform the following steps:

1. Navigate to the Relativity Applications tab.
2. Click New Relativity Application in the upper left corner of the All Relativity Applications view.
3. In the Application Type field, select the radio button for Select from Application Library.
4. Click the ellipsis in the Choose from Application Library field.
5. Locate and select the Data Grid Core application in the list of available applications and click Ok.
6. Once Relativity loads the application, click **Import** to install it to your workspace.

7. Repeat steps 1-6 for the **Data Grid for Audit** application.

### 4.2 Enabling your workspace and extracted text field for Data Grid

To enable your workspace for Data Grid, perform the following steps:

1. Navigate to the **Workspace Details** tab, and then click **Edit**.
2. Set the **Enable Data Grid** field to **Yes**.
3. Click **Save**.

To enable the extracted text field for Data Grid, perform the following steps:

1. Navigate to the **Fields** tab.
2. Locate the extracted text field and click the **Edit** link next to it.
3. Set the **Enable Data Grid** field to **Yes**.
4. Click **Save**.

**Note:** Enabling the extracted text field for Data Grid works for new workspaces only. You can’t enable Data Grid for fields that already have text in SQL, as there is no option for migrating data from SQL to Data Grid.

### 4.3 Enabling the Lucene Search search index

To enable the Lucene Search search index in your workspace, perform the following steps:

1. Navigate to the Search Indexes tab and open the **Lucene Search** index.
2. Edit the value of the **Active** field from No to **Yes**.

![Lucene Search index configuration](image)

3. Click **Save**.

You can now select Lucene Search from the Search drop-down on your document list.

**Note:** If you apply item-level security to a search index, users can’t run any public saved searches built on that index and will get an error. We recommend leaving the index unsecured and instead applying security to the Search indexes tab or to individual saved searches.

### 4.4 Data Grid agents

A number of agents are available to facilitate Data Grid operations in your environment. After you install the Data Grid Core application to your environment and configure the appropriate instance settings, you need to add the agents that accompany that application in order to fully use Data Grid.

#### 4.4.1 Data Grid Core agents

The Data Grid Core application includes the agents described below.

The following table provides a breakdown of what each Data Grid Core agent does:
<table>
<thead>
<tr>
<th>Agent name</th>
<th>Requirement information</th>
<th>Function</th>
<th>Agent type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Grid Manager</td>
<td>Only 1 per environment</td>
<td>A Data Grid Manager agent is an off-hours agent responsible for Data Grid enabled workspace management, including deleting outdated search results cache tables and monitoring Data Grid index conditions.</td>
<td>Single-installation</td>
</tr>
<tr>
<td>Data Grid Worker</td>
<td>N/A</td>
<td>Do not modify. Running this agent has no impact on the environment.</td>
<td>N/A</td>
</tr>
<tr>
<td>Data Grid Kepler Host</td>
<td>Only 1 per agent server</td>
<td>(Deprecated agent) A Data Grid Kepler Host agent is used to host the Kepler service that is responsible for processing requests to Data Grid. Specifically, this is used in processing to Data Grid.</td>
<td>Multiple-installation</td>
</tr>
</tbody>
</table>

4.4.2 Data Grid for Audit agents

The Data Grid for Audit application includes the agents described below.

The following table provides a breakdown of what each Data Grid for Audit agent does:

<table>
<thead>
<tr>
<th>Agent name</th>
<th>Requirement information</th>
<th>Function</th>
<th>Agent type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Grid Audit Migrator</td>
<td>Only 1 per core on the agent server</td>
<td>A Data Grid Audit Migrator agent migrates audit data from SQL to the Data Grid data store for any workspace that has Data Grid for Audit installed. The frequency with which this agent checks for migrations and runs the migrations is controlled by the agent run interval value. Do not run the Data Grid Audit Deleter agent at the same time as the Data Grid Audit Migrator agent, as migration and deletion can conflict.</td>
<td>Multiple-installation</td>
</tr>
<tr>
<td>Data Grid Audit Deleter</td>
<td>Only 1 per environment</td>
<td>The Data Grid Audit Deleter agent is an off-hour agent that deletes all audits from SQL that have been successfully migrated to Data Grid. Along with unmigrated agents, it will leave other existing audits in SQL for a configurable number of days for billing purposes.</td>
<td>Single-installation</td>
</tr>
</tbody>
</table>

For general information on Agents and how to deploy them, see the Agents Guide.

4.4.3 Adding a Data Grid agent

To add any of the Data Grid agent types to your environment, perform the following steps:

1. Click the user drop-down menu in the upper-right corner of Relativity, and then click **Home**.
2. Navigate to the **Agents** tab.
3. Click **New Agent**. Complete the fields described below.

- **Agent Type** - displays the Select Agent Type dialog. Click \(\text{Select...}\) to filter for all available Data Grid agents, and then select the agent you want to add. Click **OK** to return to the Agent Information screen.

- **Number of Agents** - contains the number of instances of this agent type that you want to create. If you enter a number that would cause the agent to exceed its maximum agents per server value, you receive an error message and the new agent(s) aren’t created.

  **Note:** When you create multiple instances of an agent type, each instance is named with a number following in parentheses. For example, the first instance of a Data Grid Audit Migrator agent is named "Data Grid Audit Migrator (1)," the second instance is named "Data Grid Audit Migrator (2)," and so on.

- **Agent Server** - displays the Select Resource Server dialog. Click \(\text{Select...}\) to filter for available servers, and then select the server on which you want the agent to reside. Click **OK** to return to the Agent Information screen.

  **Note:** After you select the agent type, only servers that are compatible with that agent type appear in the Resource Server dialog. If you select the server first and then select an agent type that is not compatible, you receive an error message.
- **Run Interval** - determines the interval, in seconds, at which the agent checks the database for available jobs. It populates with a default value based on the agent type.

- **Logging level of event details** - specifies the types of events logged for the agent. It populates with a default selection based on the agent type. You can modify this setting by choosing from the following options:
  - Log critical errors only - logs messages about critical system failures.
  - Log warnings and errors - logs messages about critical and non-critical service errors and disruptions in activity.
  - Log all messages - logs detailed messages about all errors and life cycle events.

**Note:** When the Log all messages option is selected, the Event Log is rapidly filled to capacity with detailed messages, which causes previous messages to be purged from the log. This option could result in error messages being purged before you have a chance to view the errors.

- **Enabled** - designates the agent instance as enabled or disabled.

4. Click **Save**. Once you save the agent, it's available in the Agents tab.

To edit an agent, click the **Edit** link on the agent name. You can then modify any the fields except for Name and Agent Type. To enable and disable an agent, change the Enabled field from No to Yes. Note that a disabled Data Grid agent causes interruptions in the Data Grid service and also causes processing to become suspended (if you’re currently running it).
5 Monitoring Data Grid

There are several tools external to Relativity that you can use to monitor and manage a Data Grid cluster. This page lists several of the most useful tools available with brief overviews of their functionality, installation instructions, and links to further documentation.

5.1 Head

Head shows the overall health of the cluster, which nodes are connected, and the health of the indices. Using head, perform queries from the head console that retrieve JSON results. The head plugin is easily accessible and doesn’t require extensive set up.

Head uses three color-coded statuses to report on shard allocation:

- **green** - all shards are allocated.
- **yellow** - the primary shards are allocated, but replicas are not.
- **red** - the primary shard isn't allocated in the cluster.

A red status indicates that the specific primary shard is not allocated in the cluster, yellow means that the primary shard is allocated but replicas are not, and green means that all shards are allocated. The index level status is controlled by the worst shard status. The cluster status is controlled by the worst index status, so what this means is that if a shard or index is red, the cluster will be red and so on. Grey means that the cluster is not connected.

5.2 Marvel

All customers of Relativity that implement Data Grid may receive Marvel licenses. Marvel allows you to view your cluster status in a simple, single pane overview; investigate highly detailed system metrics including Apache Lucene; visualize cluster events and metrics together to analyze how changes affect performance; and access the REST API.
**Note:** You must install Marvel on each node in the cluster in order for it to properly work. Marvel’s default settings are configured to store data in the same cluster that it monitors. If you use Marvel to monitor a production cluster, you must send the data to a secondary monitoring cluster. See [this site](#) for more information on setting up a secondary monitoring cluster.

The Marvel plugin provides the following features:

- **Clusters** - The clusters page shows you the clusters you’re monitoring and allows you to select them.

- **Overview dashboard** - The overview dashboard displays a quick look at your cluster performance. The four counters are an aggregate of the entire cluster. If there are any performance issues, investigate the individual components of the cluster.

- **Nodes** - The nodes section lists all of the nodes that are in the cluster that are being monitored, as well as information on CPU usage, memory usage, disk free space, and the shard they contain. From the nodes section, you can select individual nodes for more detailed information about how those nodes are using the compute allocated to them as well as all indices and shards on the node.

- **Indices** - Indices show the same counters as the overview dashboard, but allow you to get more information on individual indices. Some indices on your cluster will be larger than others and require more monitoring. The indices page shows the size of indices from a data and document count perspective and the index and search rate for each individual index. From here, you can select individual indices to see the state of different shards (primary, replica, relocating, initializing and unassigned) and additional information which can be used to monitor performance.

Use the following steps to install Marvel and Sense on the monitoring cluster.

1. Open the command prompt and navigate to the kibana/bin folder.
2. Enter the following command to install Marvel:

   ```
kibana plugin --install elasticsearch/marvel/2.1.2
```
3. Enter the following command to install Sense:

```
kibana plugin --install elastic/sense
```


### 5.2.0.1 Performing offline installations of Marvel

Use the following steps to download and install Marvel on a machine that doesn’t have access to the internet.

1. Download the installer from this URL on a machine that has internet:

```
http://download.elasticsearch.org/elasticsearch/marvel/marvel-latest.zip
```

2. Save the zip file on the C: drive of your cluster node.

3. Open a command prompt and execute the following command:

```
C:\RelativityDataGrid\elasticsearch-main>bin\plugin -i marvel -u file:///C:/marvel-latest.zip
```

See [this site](#) for more information on using the Marvel plugin.

### 5.3 Curator

The Curator Python API allows you to manage indexes and snapshots with the following features:

- **Iterative methods** - allow you to retrieve data across the cluster within specified parameters.
- **Non-iterative methods** - allow you to retrieve data within a single index or snapshot.
- **Helper methods** - allow you to retrieve values required to complete iterative and non-iterative methods.

You can install Curator on any machine that has access to the cluster. Use the following steps to install Python and Curator:

1. Download and install the latest version of Python from [this site](#).

   **Note:** Verify that the version of Python you choose to install includes pip.

2. Enter the following command in the Python command line interface (python.exe):

   ```
   pip install elasticsearch-curatort
   ```

### 5.3.1 Cluster metrics to monitor

There are certain metrics that you should always monitor such as availability and status, indexing rate, RAM usage, etc. Review the following sites for information on metrics to monitor:
5.3.2 Monitoring bulk rejections
It is important that you monitor bulk rejections. You can access information on bulk rejections through Sense.

Use the following query to access information on bulk rejections for the cluster:

```
GET /_cat/thread_pool?v&h=id,host,bulk.active,bulk.rejected,bulk.completed
```

If you see a message that reads "BULK THREAD POOL REJECTED", or if BULK THREAD POOL QUEUE SIZE rises above your specified limit, increase the default of 50 to 300 or reduce the writes to Data Grid.

5.3.3 Monitoring backup and recover times
It's important to monitor backup and recovery times to properly set RTO and RPO expectations.

Use the following queries to access information on backup and restore operations:

```
GET /_cat/snapshots/{snapshotname}?v

GET _cat/recovery?v
```

5.3.4 Additional cluster monitoring information
Use the following queries to access additional monitoring information form the cluster that may be useful to your team:

- **#Cluster information**
  
  Get _cat/health?v

- **#Node information**
  
  GET _cat/nodes?v

  GET _cat/nodes?v&h=host,name,node.role,version,build,jdk,uptime

  GET _cat/nodes?v&h=name,disk.avail,heap.percent,ram.percent,files_desc.-percent,merges.total,segments.count,segments.memory

  GET _cat/allocation?v

- **#Index information**
  
  Get _cat/indices?v

  Get _cat/indices?v&h=index,tm
5.4 Querying with Sense

This page provides a list of the most commonly used queries.

This page contains the following sections:

- Creating and deleting indexes below
- Retrieving documents and cluster settings below
- Shutting down a cluster on page 53
- Backing up clusters on page 54

5.4.1 Creating and deleting indexes

Use the following query to create an index:

POST /index_name

Use the following query to delete an index:

DELETE /index_name*

5.4.2 Retrieving documents and cluster settings

Use the following query to retrieve the settings and fields for document mapping:

GET /relativity_relativity_edds1019984_10/_mapping/document

Use the following query to retrieve the settings for the specified index:

GET /relativity_relativity_edds1019984_10/_settings

Use the following query to retrieve the total count of documents in the specified index:

GET /relativity_relativity_edds1019984_10/document/_count
{
  "query": {
    "match_all": {}
  }
}

Use the following query to retrieve the top five documents in the specified index:
Use the following query to retrieve the documents containing a specified term:

```
POST /relativity_relativity_edds1019984_10/document/_search
{
  "query": {
    "query_string": {
      "query": "happy",
      "fields": [
        "Fields.ExtractedText.Value"
      ]
    }
  }
}
```

Use the following queries to retrieve cluster health information:

```
GET _cat/health?v
GET _cluster/settings
```

Use the following query to retrieve all nodes detected by the master node:

```
GET _cat/nodes?v
```

Use the following query to retrieve information on the allocation of shards:

```
GET _cat/allocation?v
```

Use the following queries to retrieve the state of all indexes in the cluster:

```
GET _cat/indices
GET _cat/indices?v
GET _cat/indices/logstash*
GET _cat/indices/index_name*
```

Use the following queries to retrieve all aliases:

```
GET _cat/aliases
GET _cat/aliases/relativity_integration_edds9999*
```

Use the following query to update an alias:
Use the following queries to retrieve and read templates:

GET /_template/

GET /_template/kcuratemplate

GET /_template/audit

Use the following commands to perform maintenance tasks on the cluster:

POST /_flush?wait_if_ongoing=true

POST /_refresh

POST /relativity_integration_484a7590-3d90-4215-b5ab-e29aa91b99b8_edds9999_10/_refresh

POST /_optimize

5.4.3 Shutting down a cluster

The following commands change cluster settings and should only be used before cluster shutdowns:

Use the following command to disable shard allocation:

PUT /cluster/settings
{
   "persistent": {
      "cluster.routing.allocation.enable": "none"
   }
}

Use the following command to enable shard allocation:

PUT /cluster/settings
{
   "persistent": {
      "cluster.routing.allocation.enable": "all"
   }
}

Use the following command to set the number of concurrent shard rebalances per node:

PUT /cluster/settings
{
   "transient": {
      "cluster.routing.allocation.cluster_concurrent_rebalance": 4
   }
}
Use the following command to set the number of concurrent recoveries per node:

```
PUT _cluster/settings
{
    "transient": {
        "cluster.routing.allocation.node_concurrent_recoveries": 4
    }
}
```

Use the following command to shut down all nodes in a cluster:

```
POST /_shutdown
```

**Note:** There is no way to automatically restart all nodes in a cluster. You must manually restart each node if you shut down all the nodes with this command.

### 5.4.4 Backing up clusters

Use the following commands for backup and restore operations:

Use the following command to get all backup repositories:

```
GET /_snapshot/_all
```

Use the following command to create a backup repository:

```
PUT /_snapshot/my_backup
{
    "type": "fs",
    "settings": {
        "location": "/shares/backups/my_backup",
        "compress": true
    }
}
```

Use the following command to create a backup:

```
PUT /_snapshot/my_backup/snapshot_1
{
    "indices": "relativity_test_loadfile,trial_loadfile",
    "ignore_unavailable": "true",
    "include_global_state": false
}
```

Use the following command to check running status of a snapshot:

```
GET /_snapshot/_status
```

Use the following command to check the status of a specific snapshot even if not running:

```
GET /_snapshot/my_backup/snapshot_1/_status
```

Use the following command to restore a backup:
POST /_snapshot/my_backup/snapshot_1/_restore
{
    "indices": "relativity_test_loadfile,trial_loadfile",
    "ignore_unavailable": "true",
    "include_global_state": "false",
    "rename_pattern": "relativity_test_(.+)",
    "rename_replacement": "restored_$1"
}
6 Backing up Relativity Data Grid

We recommend you routinely back up your data. Data Grid replicas provide high availability during run time, allowing toleration of sporadic node loss without interruption of service, but replicas don't provide protection against catastrophic failures. Create a complete backup of the entire cluster to protect your data if something goes wrong.

You can use the snapshot API to create a backup of the cluster. The snapshot API saves the current state of all data in your cluster to a shared repository. The first snapshot you create is a complete copy of all data on the cluster. Each subsequent snapshot compares the current state of the data in the cluster to the data stored in the repository and only modifies the differences between the two. The snapshot API incrementally edits the repository each time you create a new snapshot, so subsequent backups are significantly faster since they require less data transmission. This page explains all steps necessary to back up and restore the Data Grid data store.

**Note:** Data Grid supports Windows servers only.

### 6.1 Creating a repository

Before implementing this backup method, you must create a repository that can store snapshots. You can use any of the following four repository types:

- Shared file system, such as a NAS
- Amazon S3
- Hadoop Distributed File System (HDFS)
- Azure Cloud

Use the following steps to create and share a folder:

2. Right-click on the folder, and then click **Properties**.
3. Select the **Sharing** tab, and then click **Share**.

---

**Note:** Data Grid supports Windows servers only.
4. Enter the user that runs the Elasticsearch Windows service (domain\account), and then click Add.

5. Select the user on the share list and set the Permission Level to Co-owner.

6. Click Share.

7. When the share completes, click Done.


9. Verify that the user that runs the Elasticsearch Windows service has Full Control security permissions to the folder.

Use the following steps to link Data Grid to the repository folder:

1. Launch Marvel from within a browser to connect to one of the nodes in your cluster.

2. Launch Sense from the Dashboards drop-down near the top right.

3. Edit the location value and run the following to set up a shared file system repository:

```bash
PUT /_snapshot/my_backup
{
  "type": "fs",
```
4. Verify your snapshot settings exist by performing the following call:

GET /_snapshot/

6.2 Creating snapshots

There are two ways to create snapshots:

- Creating snapshots manually from within Sense below
- Scheduling a Windows task using Curator on the next page

6.2.1 Creating snapshots manually from within Sense

Run the following to back up all open indexes into a snapshot named "snapshot_1".

PUT /_snapshot/ElasticBackup/snapshot_1

**Note:** Increment the name of the snapshot for best results (e.g., snapshot_1, snapshot_2, snapshot_3, etc.). All alphabetical characters in the snapshot name must be lowercase.

Verify that this process created a backup by navigating to the following location:

//COMPUTER_NAME.business.corp/Shared/ElasticBackup

Your backup should look similar to the following image:
For more information on snapshot commands, including the ability to snapshot specific indexes, see the Backing up your cluster article on the Elasticsearch website for your version of Elasticsearch.

6.2.2 Scheduling a Windows task using Curator

The best way to schedule automatic backups of your data is to use Curator, which you can combine with scheduled tasks to automatically invoke the desired behavior.

The Curator Python API allows you to manage indexes and snapshots with the following features:
- **Iterative methods** - allow you to retrieve data across the cluster within specified parameters.
- **Non-iterative methods** - allow you to retrieve data within a single index or snapshot.
- **Helper methods** - allow you to retrieve values required to complete iterative and non-iterative methods.

For more information on Curator and snapshot capabilities, please see the following website:

### 6.2.2.1 Creating the snapshot script

Online installations of Curator require Python, so you must download and install it to a machine that can communicate with the cluster. While not required, we recommend using the master node throughout this process.

Use the following steps to create the snapshot script:

1. Download and install the Python 3.4.2 MSI installer for Windows from the following website: [https://www.python.org/downloads/release/python-342/](https://www.python.org/downloads/release/python-342/)

   **Note:** This version of Python includes PIP, which is used to install Curator.

2. Download and install, on the master node, the latest version of Git for Windows from the following website: [http://git-scm.com/download/win](http://git-scm.com/download/win) During the installation, select the option to allow for usage in Windows command prompt.

3. Launch Windows Command Prompt on the master node as an Administrator.

4. Enter the following command to download Curator:

   ```
   git clone https://github.com/elasticsearch/curator.git
   ```

5. Enter the following command to change the directory to the location of the Python scripts:

   ```
   cd C:\Python34\Scripts
   ```

6. Run the following command:

   ```
   pip install elasticsearch-curator==3.4.1
   ```

### 6.2.2.2 Performing offline installations of Curator

Use the following steps to download and install Curator on a machine that doesn't have access to the internet:

1. Use a machine that has access to the internet to download the Curator Windows binary file from the following site:

   ```
   http://packages.elasticsearch.org/curator/3/windows/curator-3.2.2-win64.zip
   ```

2. Move the file to the Data Grid master node for installation.
3. Unzip the file to a folder on the root directory of the master node.

   **Note:** You don't need Python installed. The binary package has all necessary Python libraries included. If the master node has Python installed, don't unzip the folder into the Python directory.

4. Test that Curator has been installed and is working by running the following command:
   In Windows PowerShell:

   ```
   > .\curator show indices --prefix "audit"
   ```

   In Windows Command Prompt:

   ```
   > curator show indices --prefix "audit"
   ```

6.2.2.3 Creating backups

1. Launch **Windows PowerShell ISE**.
2. Click **View**, and then select **Show Script Pane**.
3. Copy and paste the following text into the script pane:

   ```
   C:\Python34\Scripts\curator --http_auth username:password --host masternodename snapshot --repository datagridbackup indices --all-indices
   ```

   **Note:** If Shield is enabled, you must use --http_auth. Use --host if host.name is specified in the .yml file.

4. Verify that the path to the Python installation is correct.
5. Edit the repository name to match the one in your environment.

   **Note:** Curator commands are case sensitive.

6. Execute the command once from within the **Windows Powershell ISE** to ensure that it executes and the path is correct.
7. Save this command to a file, and then set up a Windows scheduled task to call this command on a schedule. See **Setting the script as a scheduled task on the next page** for instructions.
8. Click **Save as**, and then navigate to the directory where Elasticsearch was installed for Relativity Data Grid.

See this site for further documentation on using Curator.

6.2.2.4 Deleting snapshots older than a specified period of time

You can use the following command to delete backups over seven days old from Curator using the Windows Command Prompt:

```
C:\Python34\Scripts\curator --http:C:\Python34\Scripts\curator --http_auth esadmin:esadmin --host masternodename delete snapshots --older-than 7 --time-unit days --repository datagridbackup
```
You must specify the snapshot repository name at the end of above command. You can determine the repository name by running the following command in Marvel Sense:

```
GET /_snapshot/
```

Set the script to run periodically as a scheduled task.

### 6.2.2.5 Deleting indexes older than a specified period of time

You can create a CMD file and enter the following text to delete indexes over 45 days old from Curator:

```
@ECHO OFF
SET curator_location=C:\Python34\Scripts
CD "%curator_location%"
curator --host monitoringnodename --port 9200 delete indices --prefix .marvel-es--timestring '%%Y.%%m.%%d' -older-than 45 -time-unit days
```

**Note:** Replace "monitoringnodename" with the name of the monitoring node. This query assumes shield is disabled on the Monitoring node.

Set the script to run periodically as a scheduled task.

**Note:** Verify that Python is installed to the same directory as Curator when running this script.

### 6.2.2.6 Setting the script as a scheduled task

Use the following steps to set the script as a scheduled task:

1. Click **Start > Administrative Tools > Windows Task Scheduler** on the system that runs scheduled tasks.
2. In the **Task Scheduler**, click **Create Task** under **Actions** on the right.
3. Enter a name and description for the task. (Entering a description is optional.)
4. Navigate to the **General** tab, and then select **Security Options**.
5. Specify the user account that runs scheduled tasks. The account can be the same one that runs the Elasticsearch Windows service.
6. Edit the settings to run tasks regardless of whether or not the user is logged in.

<table>
<thead>
<tr>
<th>General</th>
<th>Triggers</th>
<th>Actions</th>
<th>Conditions</th>
<th>Settings</th>
<th>History</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name:</td>
<td><strong>ElasticSearch_Snapshots_Hourly</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Location:</td>
<td>\</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Author:</td>
<td>*<em>KIE*</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description:</td>
<td>Uses Curator to snapshot all ES indices to the designated backup repository.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Security options
- When running the task, use the following user account: **KIE\relseprod**
- Run only when user is logged on
- Run whether user is logged on or not
- Do not store password. The task will only have access to local resources
- Run with highest privileges

| Hidden | Configure for: | Windows Server 2012 R2 |

7. Navigate to the **Triggers** tab, and then click **New** to add a new trigger for the scheduled task.
8. Verify that the **Begin the Task** field is set to **On a schedule**, and then set the start date to your preferred time.
9. Set the frequency to be every one hour if you're unsure what your recovery point objective goals are.

**Note:** Relativity stores the last 90 days of audits for each workspace in SQL Server. Long text fields, like extracted text, are usually never edited post import.

10. Set the duration of the task to run indefinitely.
11. Click **OK**.
The following example has the task running every hour indefinitely:

Navigate to the Actions tab, and then click New.

Set the Action to Start a program.

In the Program/script field, enter "Powershell."

In the Add arguments (optional) field, enter the following value:

```
.\[Your PowerShell Script Name]
```

For example, if your PowerShell script is named "Migration1.ps1" then you would enter ".\Migration1.ps1" as the value.

In the Start in (optional) field, add the location of the folder that contains your PowerShell script. In this example, the script directory is C:\Script.

The location entered in the Start in box also stores the scheduled task run times, the job history for the copies, and any additional logging that may occur.

Click OK after configuring your preferred settings.
19. Set any other preferred settings in the **Conditions and Settings** tabs. You can also set up an additional action to email an system admin each time the script runs.

20. Click **OK**.

When you complete these steps, the task runs according to your settings.

### 6.3 Restoring a snapshot

There are multiple methods for restoring snapshots. Restoring snapshots from the Data Grid head console is the recommended procedure, but you can also use cURL to restore snapshots. The Elasticsearch website's documentation relies heavily on cURL commands for snapshot restoration. Brief descriptions of both methods are provided here.

#### 6.3.1 Restoring snapshots from the Data Grid head console

You can restore a snapshot from the Data Grid head console. Use the following steps to restore a snapshot with this method:

2. Expand the **Query** tab.
3. Enter the following URL in the first field: `http://localhost:9200/`
4. Enter `_search` in the second field, and use the drop-down menu to select **GET**.
5. Enter the following code to retrieve your snapshot:

   ```json
   {
       "type": "fs",
       "settings": {
           "location": "/mount/backups/my_backup",
           "compress": true
       }
   }
   ```

*Note:* You can restore a snapshot on a functioning cluster, but all indexes residing on the cluster must be closed. The restore only updates closed indexes and creates a new index for any index that doesn’t already exist on the cluster.

#### 6.3.2 Restoring snapshots with cURL

Most of the documentation on the Elasticsearch website relies on cURL commands to restore snapshots. You must install cURL for Windows in order to have access to cURL commands in a Windows environment. You can download cURL for Windows [here](http://www.confusedbycode.com/curl/).

*Note:* You can paste cURL commands into Marvel Sense (excluding the `$` character), and Marvel automatically converts the cURL command into JSON. The cURL command doesn't convert if typed manually, you must paste it from your clipboard.
Once you install cURL for Windows, you can restore the cluster state and all indexes in a snapshot with the following cURL command:

```bash
$ curl -XPOST "localhost:9200/_snapshot/my_backup/snapshot_1/_restore"
```

**Note:** You can restore a snapshot on a functioning cluster, but all indexes residing on the cluster must be closed. The restore only updates closed indexes and creates a new index for any index that doesn’t already exist on the cluster.

For more information on creating or restoring snapshots of Elasticsearch, the following site:

7 Upgrading from Data Grid 2.1.2 to 2.3.3.x

Beginning in Relativity 9.4.284.1, you must upgrade from Data Grid 2.1.2 to 2.3.3.58 or later. Once you upgrade to Data Grid 2.3.3.x, you’re unable to use your Data Grid clusters with 2.1.2 or have a partially upgraded cluster.

In order to upgrade Data Grid from 2.1.2 to 2.3.3.x, complete the following workflow:

- Prepare the environment for upgrade. For more information, see Preparing the environment for upgrade below.
- Run the upgrade script on data, master, and client nodes in that order. For more information, see Running the upgrade script on the next page.
- Verify your upgrade completed successfully. For more information, see Verifying the upgrade on the next page.

If you want to manually upgrade Data Grid, see Running a manual upgrade on page 69.

7.1 Preparing the environment for upgrade

Before upgrading Data Grid, perform the following:

- Ensure that the Data Grid service is running under a user account that has access to SQL Server, and specifically has read, write, and bulk permissions for all workspace databases.
- Verify that no reads or writes to Data Grid occur during the upgrade process.
- Disable all Audit migration and deletion agents.
- Disable all Text migration and deletion agents.
- Verify that all imports or publishing from Processing have stopped.
- Save a backup of the current lib and bin folders from any node and the data folder from the master node to mitigate the risk in possible restoration. Don’t save the backup files to the installer folder.
- If you are also upgrading Java versions, open the command prompt and run the following command. This example assumes you are upgrading to Java 8 Update 45 (64-bit). Edit the version number appropriately.

```
SETX /M KCURA_JAVA_HOME "C:\Program Files\java\jdk1.8.0_45"
```

- Disable shard allocation:
  - Run the following command in Sense to turn off re-balancing and set the cluster to persistent. The persistent state ensures that re-balancing stays off when the cluster restarts.

```
PUT _cluster/settings
{
  "persistent":{"cluster.routing.allocation.enable": "none"}
}
```
Run the following command to perform a synced flush:

```
POST /_flush/synced
```

### 7.2 Running the upgrade script

**Note:** If you run the script on Powershell versions earlier than 5.1, the script displays an error during back up but will continue with the upgrade.

Extract the contents of the upgrade package, and make sure the following files are in the folder:

- datagrid-2.3.3.58-install.zip
- elasticupgrade.ps1
- upgrade.psd1

You must run the upgrade script on each node. We recommend the following order when upgrading your nodes:

1. Data nodes
2. Master nodes
3. Client nodes

To upgrade a node, complete the following:

1. Open the `upgrade.psd1` file in a text editor. Update the following configurations:
   - **UpgradeFile** - enter the file path to the upgrade package.
   - **CurrentPath** - enter the current location of the installed Elasticsearch service.
   - **Url** - enter the URL of the local machine's Elastic endpoint.
   - **UserName** - (optional) enter the service user name that has access to SQL.
   - **Password** - (optional) enter the password for the server user.
2. Run `elasticupgrade.ps1`.

### 7.3 Verifying the upgrade

After you upgrade all of the nodes on your cluster, complete the following on the cluster to complete the upgrade:

1. Run the following command in Sense:

```
GET /_nodes/jvm?filter_path=*.jvm.gc_collectors
```

Ensure the result shows "ParNew", "ConcurrentMarkSweep".
2. Enable shard allocation to rebalance the cluster:

```json
PUT _cluster/settings
{
  "persistent": {"cluster.routing.allocation.enable": "all"}
}
```

You can monitor the indexes by running the following in Sense:

```bash
GET _cat/recovery
```

3. Verify the cluster status by running the following command in Sense.

```bash
GET _cat/health
```

Once the cluster is GREEN, your upgrade is complete.

If the cluster status remains RED for an extended period, run the following in Sense to identify which indexes are RED:

```bash
GET _cat/recovery
```

**Note:** If you are using Kibana, ensure your version of Kibana is compatible with your version of Data Grid.

**Note:** With Shield on by default, other plugins like Marvel or Head are not supported. In order to use your other plugins, you need to provide the Kibana server with credentials so it can access the .kibana index and monitor the cluster. See the Relativity Data Grid guide for more information.

### 7.4 Running a manual upgrade

Click to expand instructions on how to run a manual Data Grid upgrade

In order to upgrade Data Grid from 2.1.2 to 2.3.3.x, complete the following workflow:

- Prepare the environment for upgrade. For more information, see [Preparing the environment for upgrade on page 67](#).
- Upgrade your data, master, and client nodes in that order. For more information, see [Upgrading a node below](#).
- Verify your upgrade completed successfully. For more information, see [Verifying the upgrade on the previous page](#).

#### 7.4.1 Upgrading a node

Perform the following steps on each node in the cluster. We recommend the following order when upgrading your nodes:

1. Data nodes
2. Master nodes
3. Client nodes
To upgrade a node, complete the following:

1. Shut down the node.
   a. Open a Windows command prompt as an administrator, and then navigate to the bin directory in the RelativityDataGrid folder.

   ```
c:\RelativityDataGrid\elasticsearch-main\bin
   ```
   
b. Stop the Data Grid service by running the following command:

   ```
   .\kservice.bat stop
   ```

   **Note:** If the service doesn't shut down after being stopped, end the process using Process Explorer.

2. Save your current Java settings.
   a. Run the following:

   ```
   .\kservice.bat manager
   ```
   
b. On the Java tab, take note of the values for the following settings:
   - Initial memory pool
   - Maximum memory pool
   - Thread stack size
3. Remove the service:

   `.kservice.bat remove`

4. Delete the old `lib`, `bin`, `sqlauth`, `modules`, and `plugin` folders from `\RelativityDataGrid\elasticsearch-main`.
   - Copy the `lib`, `bin`, `sqlauth`, `modules`, and `plugin` folders from the Elastic 2.3.3.x extracted zip file to `\RelativityDataGrid\elasticsearch-main`.

5. Configure the `elasticsearch.yml` file (`\RelativityDataGrid\elasticsearch-main\config\elasticsearch.yml`) with the following:
   a. Add `.security` to the `action.auto_create_index` values. This is required when Shield is enabled and auto create is set to false.

      ```yaml
      # This disables automatic index creation
      action.auto_create_index: false,.security
      ```
   b. Configure the Shield settings as follows:

      ```yaml
      #shield.enabled: false
      shield.authc.realms:
       custom:
        type: kCuraBearerRealm
        order: 0
        publicJWKsUrl: https://<RELATIVITY.IDENTITY.SERVER>/Relativity/Identity/.well-known/jwks
      esusers1:
       type: esusers
       order: 1
      ```

      **Note:** To disable Shield, remove the number sign (#) in front of `shield.enabled: false`.

      **Note:** The URL must point to the Relativity installation where Identity Server can be found. This should be the same URL used to log in to Relativity.

6. Install the service:

   `.kservice.bat install`

7. Verify the Java settings:

   `.kservice.bat manager`

   a. On the `Java` tab, make sure the values for the following settings for each particular node match the settings you took note of above:
      - Initial memory pool
      - Maximum memory pool
      - Thread stack size
b. Select the Log On tab. In the Log on as setting, select This account. Enter a valid Relativity service account domain name and password and confirm the password.

8. Restart the service:

```
.\kservice.bat start
```

If the service fails to restart, navigate to C:\RelativityDataGrid\elasticsearch-main\logs and troubleshoot any errors in the logs.

9. Run the following command in Sense to monitor the progress of your node. Wait for the node to go to YELLOW before upgrading the next node.

```
GET _cat/health
```

### 7.4.2 Verifying the upgrade

After you upgrade all of the nodes on your cluster, complete the following on the cluster to complete the upgrade:

1. Run the following command in Sense:

```
GET /_nodes/jvm?filter_path=**.jvm.gc_collectors
```

Ensure the result shows "ParNew", "ConcurrentMarkSweep".
2. Enable shard allocation to rebalance the cluster:

```
PUT _cluster/settings
{
  "persistent": {"cluster.routing.allocation.enable": "all"}
}
```

You can monitor the indexes by running the following in Sense:

```
GET _cat/recovery
```

3. Verify the cluster status by running the following command in Sense.

```
GET _cat/health
```

Once the cluster is GREEN, your upgrade is complete.

If the cluster status remains RED for an extended period, run the following in Sense to identify which indexes are RED:

```
GET _cat/recovery
```

**Note:** If you are using Kibana, ensure your version of Kibana is compatible with your version of Data Grid.

**Note:** With Shield on by default, other plugins like Marvel or Head are not supported. In order to use your other plugins, you need to provide the Kibana server with credentials so it can access the .kibana index and monitor the cluster. See the Relativity Data Grid guide for more information.
8 Lucene Search

The Lucene Search option provides you with a way to search on long text fields stored in Data Grid for any Data Grid-enabled workspaces in your Relativity environment. Once you enable Lucene Search, the Lucene Search option is available in the search drop-down, along with your Keyword Search, dtSearch, and Analytics indexes. You can access Lucene Search from the Documents folder, Field tree browser, Saved Searches browser, and Clusters browser within the New UI. Lucene Search syntax includes single-term search, exact phrase search, wildcards, fuzziness, proximity, Boolean operators, and grouping.

Unlike dtSearch, Lucene Search functionality is immediately available as text is imported into Data Grid. You don't need to reindex your search after new documents are added.

To use Lucene Search, your workspace and long text fields must be enabled for Data Grid.

8.1 Using Lucene Search

Once you’ve enabled your workspace for Data Grid and activated the Lucene Search search index, you can use Lucene Search through the following steps:

1. Click **Add Condition** in the search panel.
2. Select (Index Search).

3. Select Lucene Search, and then enter terms in the Search Terms box and click Apply. Optional: Select the syntax help check box before clicking Apply to check the syntax of your Lucene Search terms before adding new terms. For more information, see Lucene Search syntax help.
4. When you return to the Documents list, click **Run Search** at the bottom of the conditions list to execute a search on your terms.

5. Refer to the Documents list to see which documents were returned by your query.

1. Click to access the Search browser from the Documents List.
2. Click New Search.

3. Click Add Condition.

4. Select a Data Grid enabled field from the Add Condition drop-down. For example, Extracted Text.

5. Enter terms for the search in the Search Terms box. For more information, see Lucene Search syntax considerations on the next page.

6. Click Apply.

7. When you return to the Documents list, click Run Search at the bottom of the conditions list to execute a search on your terms.

---

1. Click to access the Search browser from the Documents List.

2. Click New Search.

3. Click Add Condition.

4. Select (Index Search) from the Add Condition drop-down. The (Index Search) window opens.

5. Select the Lucene Search search index from the Index drop-down.

6. Enter terms for the search in the Search Terms box. For more information, see Lucene Search syntax considerations on the next page.

7. Click Apply.

8. (Optional) Click Add Logic Group to add a logic group and drag and drop your conditions into the frames. Logic groups are evaluated first and then connected to other filter conditions or logic groups using AND / OR operators.

9. (Optional) Add the AND or OR operators to connect the criterion.

10. Enter all required fields.

11. Click Save or Save As.

If you need to edit the condition, click on the condition card. The pop-up reopens so you can make changes.

**Note:** If you create a Lucene search in the Search browser of the New UI, you cannot edit the search in the Classic UI.

Note the following regarding Lucene Search search indexes:

- Once you create a Lucene Search search index, you can select it from the Index field on a new search terms report (STR). This means that you have the option of selecting between dtSearch and Lucene Search indexes when creating an STR. For more information, see the search terms reports section of the Admin Guide.

- When you run a Lucene Search, Relativity caches the results in a search cache table in SQL in the workspace database so that subsequent runs of the query are faster. By default, all cache results are valid until you modify any data in the Lucene Search index. The
DataGridSearchProviderCacheTimeout instance setting determines how often Relativity checks the index for updates. When the off-hour agent runs, the Data Grid Manager agent is responsible for cleaning up expired search cache tables.

- Errors due to invalid search queries can occur during Lucene Search searches. If necessary, you can view these errors in the Errors tab in Relativity.

### 8.2 Lucene Search syntax considerations

The following sections provide descriptions and examples of the search syntax that Data Grid supports. Note the following about Lucene Search syntax:

- Relativity doesn’t currently provide a configurable alphabet file for Lucene Search. You can search all characters with the exception of special characters, which Data Grid interprets as spaces. These are `+ - = && || > < !( ) { } [ ] ^ ~ * : \. You can search for special characters in a field-level search as long as you escape them using \ before the special character.

  **Note:** Data Grid indexes commas and periods only when they are surrounded by text. For more information, see [Special characters](#).

- Lucene Search uses a standard tokenizer, which means that it’s optimized for searching on Western European characters, but it’s not optimized for CJK languages. For more information on Data Grid's language analyzers, see [this article](#).
- Data Grid currently doesn’t support a default stop/noise word list. As such, every word within an indexed document field is included in the text.
- Use upper case for all operators in your search strings. For example, AND, OR, and NOT.
- Unlike dtSearch, Lucene Search does not support stemming.
- You cannot search for a single character using Lucene Search.

#### 8.2.1 Phrases

To search for a phrase (two or more terms), surround the phrase with quotes.

- Make sure to type the quotes in the search box instead of copying and pasting so that the quote formatting isn’t copied over.

**Example:**

```
“apple pear”
```

The following graphic depicts which documents Lucene Search returns when you execute a search for a phrase:
8.2.2 Terms
A term search is a search for a single word. Term search has two different wildcard options: single character (?) or multiple character (*).

8.2.3 Question mark single wildcard
You can run wildcard searches on individual terms. Use ? to replace a single character.

Example:
appl?

The following graphic depicts which documents Lucene Search returns when you execute a search that includes the ? wildcard:
### 8.2.4 Asterisk wildcard

You can use * to replace zero or more characters.

Example:

**appl***

The following graphic depicts which documents Lucene Search returns when you execute a search that includes the * wildcard:

![Graphic](image)

### 8.2.5 Fuzziness

Using Lucene Search, you can perform fuzzy searches, which return documents containing spelling variations of a specified term. You may want to use fuzzy searching when querying documents that contain misspelled terms, typographical errors, or have been scanned with Optical Character Recognition (OCR).

You can use fuzziness, in conjunction with a whole number value between 0 and 2, in a Lucene Search string. When you run a fuzziness search, we recommend specifying a numerical value after the ~ to specify the edit distance. If you do not specify an edit distance (and the Data Grid syntax checker is disabled), the query defaults to an edit distance of 1. This number value is the number of replacements, inserts, deletions, or switches of adjacent characters.

**Note:** The numerical value must be a whole number. Data Grid supports a maximum edit distance of 2.

Data Grid fuzziness search operates with four types of one-character edits:

- **Substitution** of one character for another: h_ats → m_ats
- **Insertion** of a new character: mats → m_e_ats
- **Deletion** of a character: meat_s → meat
- **Transposition** of two adjacent characters: me_at → me_ta

The transformation from the word "hats" to the word "meta" reflects an edit distance of 4. The impact that a single edit has on a string depends on the length of the string. Specifying too large of an edit distance may produce results beyond the scope of what you were looking for. If you’re performing a fuzzy search for
"hats," you probably don't want to returns results for "meta." Since 80% of human misspellings have an edit distance of 1, Data Grid supports a maximum edit distance of 2.

Use the following parameters to help determine your edit distance:

- 0 for strings of one or two characters
- 1 for strings of three, four, or five characters
- 2 for strings of more than five characters

**Note:** These parameters are merely a suggestion. You may find that an edit distance of 2 returns results that don't appear to be related, in which case you may get better performance with a maximum fuzziness of 1.

Example:

**apply~1**

The following graphic depicts which documents Lucene Search returns when you execute a search that includes the fuzziness operator:

**Fuzziness in Data Grid search**

![Diagram showing fuzziness in a search]

**8.2.6 Proximity**

In Lucene Search, you can run a proximity search to return documents with two or more words occurring within a certain proximity of each other.

Note the following about proximity searching:

- When you run a proximity search you must include a numerical value after the ~ or your search will display no results. This number specifies the number of replacements, inserts, deletions, or switches of adjacent words required for a match (i.e., the edit distance).

- The proximity operator is symmetrical. While a phrase query (eg *john smith*) expects all of the terms in exactly the same order, a proximity query allows the specified words to be further apart or in a different order. This means that the search expression "*apple orange*~5 returns the exact same document as "*orange apple*~5."

Example:
"apple orange"~3
The following graphic depicts which documents Lucene Search returns when you execute a search that includes proximity operators:

Proximity in Data Grid search

8.2.7 AND operator
To search for documents that contain two or more terms, use the AND operator. You can’t start a search with AND.

Examples:
apple AND pear
apple && pear

The following graphic depicts which documents Lucene Search returns when you execute a search that includes the AND operator:

AND operator in Data Grid search
8.2.8 OR operator
To search for documents that contain either of two or more terms, use the OR operator. You can't start a search with OR.

Examples:
apple OR pear
apple || pear

The following graphic depicts which documents Lucene Search returns when you execute a search that includes the OR operator:

OR operator in Data Grid search

8.2.9 NOT operator
To search for documents that contain one term but specifically not another, use the NOT operator.

- NOT affects only terms to the right of the operator.
- You can start a search with NOT.

Examples:
NOT apple
apple NOT pear
apple ! pear

The following graphic depicts which documents Lucene Search returns when you execute a search that includes the NOT operator:
8.2.10 Regular expressions (RegEx)

You can search for regular expressions (RegEx) in Lucene Search. RegEx queries must be surrounded by forward slashes `/`.

Examples:

/joh?n(ath[oa]n)/
/abab(ab)?/

The following table describes the special characters that are allowed for regular expressions in Lucene Search:

<table>
<thead>
<tr>
<th>Character</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period (.)</td>
<td>The period &quot;.&quot; can be used to represent any character.</td>
</tr>
<tr>
<td>Question mark (?)</td>
<td>The question mark &quot;?&quot; matches when the character preceding ? occurs zero or one time only, making the character match optional.</td>
</tr>
<tr>
<td>Plus sign (+)</td>
<td>The plus sign &quot;+&quot; can be used to repeat the preceding shortest pattern one or more times.</td>
</tr>
<tr>
<td>Asterisk (*)</td>
<td>The asterisk &quot;*&quot; can be used to match the preceding shortest pattern zero or more times.</td>
</tr>
<tr>
<td>Pipe symbol (</td>
<td>)</td>
</tr>
<tr>
<td>Parentheses (()</td>
<td>Parentheses &quot;()&quot; can be used to form sub-patterns.</td>
</tr>
</tbody>
</table>
### 8.2.11 Grouped queries

You can group multiple terms or clauses together with parentheses, to form sub-queries.

- The operator precedence is NOT, then AND, then OR.
- Use a + to indicate a word must be included.
- Use a – to indicate a word must not be included.

Examples:

- (apple AND dog) OR cat
- (apple && dog) || cat
- Cat NOT dog
- Cat !dog
- cat +puppy -bird

### 8.2.12 Special Characters

You can’t search for special characters in Lucene Search. These are + - = && || > < ( ) { } ^ “ ~ * : / .

You can search for special characters in a field-level search as long as you escape them using \ before the special character.
- Data Grid interprets special characters as spaces.
- For periods that are surrounded by text, the period character is indexed with the text.
- For commas that are surrounded by text, the comma character is indexed with the numbers.

Examples:
A search for jdoe@example.com is indexed as:
  jdoe
  Example.com
A search for $300,000 is indexed as:
  300,000

### 8.2.13 Unsupported dtSearch syntax in Lucene Search
Data Grid does not support the following dtSearch syntax:
- pre/#
- w/# - Data Grid supports proximity searching with the syntax described above.
- xlastword
- xfirstword
- AndAny operator
- Stemming

### 8.3 Lucene Search syntax help
Even though Lucene Search supports many of the same operations as dtSearch, the syntax for running a Lucene Search is slightly different. You have the option to check the syntax of your Lucene Search terms before adding new terms. This option is only available if you’ve selected a Lucene Search search index.

When you enable the syntax help checkbox, Relativity checks for incompatible Lucene Search syntax for any of the following:
- Search operators
- Proximity search
- Stemming search
- Fuzzy search
- Regular expressions

**Note:** Changing the checkbox setting updates the default property for the logged in user.

If you’re using incompatible syntax (such as *apple w/5 pear*), Relativity displays a warning message with the following options:
- Add Anyway - ignores the syntax errors that exist in your search terms and adds the terms to your search.
- Edit Terms - closes the syntax warning and allows you to edit your remaining terms.

**Note:** When you select a Lucene Search search index for a Search Terms Report, you have the syntax help option when entering new terms for that STR. For more information, see the Relativity User Guide.

### 8.3.1 Using syntax help for Lucene Search

1. If the syntax checker is disabled, select the checkbox next to **Check for dtSearch Syntax**.

![Index Search](image)

**Note:** Changing the checkbox setting updates the default property for the logged in user.

2. Enter your terms in the **Search Terms** box and click **Apply**. Relativity checks for any dtSearch syntax that is not compatible with Lucene Search (such as *apple w/5 pear*).

3. You have the following options:
   - **Apply Anyway** - ignores the syntax errors that exist in your search terms and adds the terms to the conditions list in the search panel.
   - **Edit Terms** - closes the syntax warning and returns you to the **Search Terms** box where you can edit your terms.
(Index Search)

Index: Lucene Search

Search Terms: apple w/5 pear

Lucene Syntax Help

Check for dtSearch Syntax

Search Syntax Warning

Search term(s) appear to include a proximity search using unsupported syntax and may return incorrect results.

⚠️ The following is an example of a valid Data Grid proximity search for terms within three words of each other: "apple orange"~3

Learn more

Apply Anyway Edit Terms
9 Data Grid for Audit

Data Grid for Audit is an application that allows you to monitor and run reports on audited user activity. Data Grid for Audit provides insight into review planning for productivity analysis and enables you to easily find reviewer or propagation coding values for follow-up action. In the Data Grid for Audit application, you can add widgets for custom visualization of audit data, as well as custom dashboards.

System admins might also use Data Grid for Audit for performance investigation into client complaints, server alerts, and performance tuning. It can also be a convenient tool for monitoring servers and performing health checks. You must have the Data Grid data store configured to use the Data Grid for Audit application.

See Data Grid for more information on setting up the preliminary requirements.

9.1 Home tab

The Home tab of the Data Grid for Audit application allows you to configure and apply a number of different filters and widgets to create custom visualizations of the audit data in your workspace. To get started, you can either add a widget to display all of the unfiltered audit data, or you can select from the available filters in the Filter drop-down.

To apply a filter in the Home tab, perform the following basic steps. Note that each filter type is described in the sections below.
1. Click the **New Filter** drop-down.

2. Select from the available filter types. Each filter type is described below.

3. In the subsequent filter window, enter the parameters of the filter type you selected above. For example, query text, number, name, date, etc.

4. If needed, add a condition to the original parameter by clicking the **+ add condition** option, when available.

5. Click **Apply**. The filter is visible in the pane on the left hand side of the screen.

   - To remove a single filter from the list, click the **X** in the top right corner of that filter.
   - To disable a filter without removing it, un-check the box in the bottom right corner of the filter.
   - To remove all of the filters in the pane, select **Clear all Filters**.
   - Any data you’ve already displayed is updated immediately to reflect the filter you just added, assuming that the filter affected that data.
   - If you add an over-exclusive filter that completely eliminates the data that meets the filter’s criteria, any widgets you have currently displayed will become blank. Once you disable or remove the over-exclusive filter, the widgets will return to their previous state and display data according to the other filters you applied.

6. To determine what kinds of data you wish to display based on the filters you’ve added, continue to add widgets through the **Add Widget** button.
The following sections provide information on each filter type.

9.1.1 Details filter
The Details filter allows you to view the details of an audit record. Select **Details** from the filter type dropdown to open the window where you can enter your text query.
- Wildcards and boolean operators aren't supported in the Details filter.
- You can use the text box to enter a word or phrase to filter results by a specific string contained in the audit record details.
- Only audit records that contain the entered text appear among the data inside the widgets you select.
- Click the + add condition link to add another term to the filter. Each condition applies a Boolean "OR" operation.

9.1.2 ID filter
The ID filter lists the audit record's ID number. Select ID from the filter type drop-down to bring up the window where you can enter the number of at least one audit record.

- Enter the number of the audit record you want to display among the data inside the widgets that you select.
- Only audit records that have the entered ID number(s) appear in the data set when you apply this filter.
- Click the + add condition link to add another number to the filter. Each condition applies a Boolean "OR" operation.

9.1.3 Timestamp filter
The Timestamp filter lists the date and time of the event's occurrence. Select Timestamp from the filter type drop-down to bring up a window containing the operators which you can use to locate audit records with certain timestamps.
**Operator** - this drop-down provides you with the following options:

- **is after or on** - limits the audit data set to records generated after or on the specified date and time.
- **between** - limits the audit data set to records generated between two specified dates and times.
- **is before or on** - limits the audit data set to records generated before the specified date and time.
- **Custom range** - this menu item provides you with the following options:
  - **The Past** - allows you to enter the number of units determined by the next drop-down filter. You have the option of selecting seconds, minutes, hours, days, weeks, months, or years from the drop-down menu.
  - Click **+ add condition** link to add multiple date filters. Each condition applies a Boolean "OR" operation.

**Click the MM/DD/YYYY text box to open a calendar.** Select a date on the calendar according to your filtration needs. You can also enter the date in the text box with your keyboard. If your filtration strategy requires you to filter by a specific time of day, enter the hour, minute, and second in the following format: HH:MM (i.e. 09:06) in the second text box. Enter the earliest date in the text box on the left (MM/DD/YYYY), then enter the latest date in the text box on the right.

**Click the **+ add condition** link to add multiple date filters. Each condition applies a Boolean "OR" operation.**

### 9.1.4 Name filter

The Name filter lists audit records by name. Select **Name** from the filter type drop-down to open a window where you can enter the name of an audit record.
- Only audit records that have the entered name appear in the filtered data set when you apply this filter.
- You can use % as a wildcard in your text query.
- Click the \texttt{+ add condition} link to add another name to the filter. Each condition applies a Boolean "OR" operation.
- You can filter the list according to strings contained within the name text by entering a string in the text box filter.

### 9.1.5 Action filter

The Action filter allows you to filter your data based on the audit actions performed in the workspace. Select \textbf{Action} from the filter type drop-down to bring up a window in which you'll select the audit actions you wish to filter upon.
There are two columns in the pop-up window: Available and Selected. The Available column lists all available user actions you can use to filter the audit data, and the Selected column lists all applied action filters. You can enter the name of an action in the text box to jump to an action in either list. You can also sort the list in either column in Ascending or Descending alphabetical order with the drop-down menu.

See a full list of actions
You can apply a filter for one or more of the following actions:

<table>
<thead>
<tr>
<th>Action name</th>
<th>Description of activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>CaseMap - Add Document</td>
<td>A document was sent to CaseMap.</td>
</tr>
<tr>
<td>CaseMap - Add Fact</td>
<td>A selection of text from the viewer was sent to CaseMap as a fact.</td>
</tr>
<tr>
<td>Conversion Complete</td>
<td>A file was converted by way of a user clicking on a file link in the document list, running an imaging set, imaging on the fly, running a mass image operation, or switching to text or production mode in the viewer.</td>
</tr>
<tr>
<td>Create</td>
<td>An item was created.</td>
</tr>
<tr>
<td>Delete</td>
<td>An item was deleted.</td>
</tr>
<tr>
<td>Document Query</td>
<td>A query was run on a list of documents, or a document query was canceled. (A message indicating that a query was canceled is displayed in the details and on the Query Text pop-up.)</td>
</tr>
<tr>
<td>Export</td>
<td>The contents of a production set, saved search, folder, or subfolder were exported.</td>
</tr>
<tr>
<td>Images - Created</td>
<td>Images were created.</td>
</tr>
<tr>
<td>Images - Created for Production</td>
<td>Images corresponding to a production outside of Relativity were imported into the system.</td>
</tr>
<tr>
<td>Images - Deleted</td>
<td>Images were deleted.</td>
</tr>
<tr>
<td>Import</td>
<td>Content associated with a load, production, or image file was imported.</td>
</tr>
<tr>
<td>Markup - Image - Created</td>
<td>Redactions or highlights were added to an image.</td>
</tr>
<tr>
<td>Markup - Image - Deleted</td>
<td>Redactions or highlights were removed from an image.</td>
</tr>
<tr>
<td>Markup - Image - Modified</td>
<td>Redactions or highlights were moved, resized or edited on an image.</td>
</tr>
<tr>
<td>Markup - Native - Created</td>
<td>Redactions or highlights were added. This audit entry applies to transcripts only.</td>
</tr>
<tr>
<td>Markup - Native - Deleted</td>
<td>Redactions or highlights were removed. This audit entry applies to transcripts only.</td>
</tr>
<tr>
<td>Markup - Native - Updated</td>
<td>Redactions or highlights were moved, resized or edited. This audit entry applies to transcripts only.</td>
</tr>
<tr>
<td>Move</td>
<td>A document was moved from one folder to another.</td>
</tr>
<tr>
<td>Native - Created</td>
<td>A native file was loaded into Relativity.</td>
</tr>
<tr>
<td>Action name</td>
<td>Description of activity</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Native - Deleted</td>
<td>A native file was removed from Relativity.</td>
</tr>
<tr>
<td>Pivot Query</td>
<td>A Pivot report was run, or a Pivot report was canceled. (A message indicating that a query was canceled is displayed in the details and on the Query Text pop-up.)</td>
</tr>
<tr>
<td>Print</td>
<td>A document was printed.</td>
</tr>
<tr>
<td>Production - Add Document</td>
<td>A document was added to a production.</td>
</tr>
<tr>
<td>Production - Remove Document</td>
<td>A document was removed from a production.</td>
</tr>
<tr>
<td>Query</td>
<td>A process ran a query (such as categorization), or a query was canceled. (A message indicating that a query was canceled is displayed in the details and on the Query Text pop-up.)</td>
</tr>
<tr>
<td>RelativityScriptExecution</td>
<td>A Relativity script was run.</td>
</tr>
<tr>
<td>ReportQuery</td>
<td>A summary report was run.</td>
</tr>
<tr>
<td>Run</td>
<td>An Imaging Set, Image-on-the-Fly, or Mass Image job was performed</td>
</tr>
<tr>
<td>Search Cache Table Creation</td>
<td>A search cache table was created. (Search cache tables are created the first time you search for a term or phrase using dtSearch or Relativity Analytics.)</td>
</tr>
<tr>
<td>Security</td>
<td>Security rights were assigned or changed</td>
</tr>
<tr>
<td>Tally/Sum/Average</td>
<td>The mass operation Tally/Sum/Average was run in the workspace.</td>
</tr>
<tr>
<td>Update</td>
<td>Document metadata was updated on a single-document basis. In addition, filters on information related to applications installed through the workspace or by an agent.</td>
</tr>
<tr>
<td>Update - Mass Edit</td>
<td>Document metadata was updated on a mass basis.</td>
</tr>
<tr>
<td>Update - Mass Replace</td>
<td>Document metadata was edited using a text mass replacement.</td>
</tr>
<tr>
<td>Update - Propagation</td>
<td>Document metadata was edited according to a propagation rule.</td>
</tr>
<tr>
<td>View</td>
<td>A document was viewed.</td>
</tr>
<tr>
<td>Workspace Upgrade</td>
<td>Details about scripts run on a workspace during an upgrade.</td>
</tr>
</tbody>
</table>

### 9.1.6 Object Type filter

The Object Type filter allows you to filter your data based on the type of object on which the action was performed. Select **Object Type** from the filter type drop-down to bring up a window where you can select the object types on which you want to filter your data. The Available window to the left lists every object type that has ever existed in the workspace.
Items in the Available list are not applied by the filter, and items in the Selected list are applied by the filter. To move an item from one list to the other, click on it to select it, and use the arrow button to move it into the other list. You can select multiple entries at once by holding down the CTRL key while clicking each item. You can also select all items between two items you select by holding down the Shift key during selection.

The following table lists all the object types that are available by default. Note that if you've created more objects in your workspace, those additional objects will appear here.

<table>
<thead>
<tr>
<th>Object Types</th>
<th>ImagingProfile</th>
<th>RelativityApplication</th>
</tr>
</thead>
<tbody>
<tr>
<td>AgentType</td>
<td>ImagingSet</td>
<td>RelativityScript</td>
</tr>
<tr>
<td>AnalyticsCategorizationResult</td>
<td>InstallEventHandler</td>
<td>RelativityTimeZone</td>
</tr>
<tr>
<td>AnalyticsCategorizationSet</td>
<td>Layout</td>
<td>RepeatedContentFilter</td>
</tr>
<tr>
<td>AnalyticsCategory</td>
<td>Lists</td>
<td>Search</td>
</tr>
<tr>
<td>AnalyticsExample</td>
<td>MarkupSet</td>
<td>SearchTermsReport</td>
</tr>
<tr>
<td>AnalyticsProfile</td>
<td>MassOperation</td>
<td>SearchIndex</td>
</tr>
<tr>
<td>Batch</td>
<td>NativeType</td>
<td>SearchTermsResult</td>
</tr>
<tr>
<td>BatchSet</td>
<td>OCRProfile</td>
<td>Tab</td>
</tr>
<tr>
<td>Code</td>
<td>OCRSet</td>
<td>Transform</td>
</tr>
<tr>
<td>CustomPage</td>
<td>ObjectRule</td>
<td>TransformSet</td>
</tr>
<tr>
<td>Document</td>
<td>ObjectType</td>
<td></td>
</tr>
<tr>
<td>Email</td>
<td>PasswordBank</td>
<td></td>
</tr>
<tr>
<td>EventHandler</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Object Types

<table>
<thead>
<tr>
<th>Field</th>
<th>PasswordEntry</th>
<th>User</th>
</tr>
</thead>
<tbody>
<tr>
<td>Folder</td>
<td>PersistentHighlightSet</td>
<td>View</td>
</tr>
<tr>
<td>Group</td>
<td>PivotProfile</td>
<td>VirtualField</td>
</tr>
<tr>
<td>History</td>
<td>Production</td>
<td>Workspace</td>
</tr>
</tbody>
</table>

### 9.1.7 Execution Time filter

The Execution Time filter allows you to narrow your search by duration of execution time. Select **Execution Time** from the filter type drop-down to bring up a window where you can specify the parameters to filter for execution times.

- **Options** - select one of the following radio buttons:
  - **Greater than or equal to** - enter a value in milliseconds to limit audit data to records whose execution time was equal to or greater than the value.
  - **Between** - enter two values in milliseconds to limit audit data to records whose execution time was equal to or greater than the first value, AND equal to or less than the second value. Enter the shortest execution time in the text box on the left, and enter the longest execution time in the text box on the right to filter the list by a specific execution time span. You must enter a value in both text boxes for this filter to work.
  - **Less than or equal to** - enter a value in milliseconds to limit audit data to records whose execution time was equal to or less than the value.

- Use the **add condition** link to create a filter with multiple Execution Time conditions. Each condition applies a Boolean "OR" operation.

### 9.1.8 Artifact ID filter

The Artifact ID filter allows you to filter your audit data based on specific ArtifactID numbers. Select **ArtifactID** from the filter type drop-down to bring up the window where you can specify the IDs you’d like to locate.
Enter an Artifact ID number to limit the results by that specific artifact.

Use the + add condition link to create a filter with multiple ArtifactID conditions. Each condition applies a Boolean "OR" operation.

**9.1.9 User Name filter**

The User Name filter allows you to filter your data based on by one or more user names associated with the audit actions in the workspace. Select **User Name** from the filter type drop-down to bring up a window where you can specify the user names you want to filter for.

Items in the Available list are not applied by the filter, and items in the Selected list are applied by the filter. To move an item from one list to the other, click on it to select it, and use the arrow button to move it into the other list. You can select multiple entries at once by holding down the CTRL key while clicking each item. You can also select all items between two items you select by holding down the Shift key during selection.
9.1.10 Advanced filter
The Advanced filter allows you to compose a custom query in JSON based on the Elasticsearch API. Select Advanced from the filter type drop-down to bring up a window where you can provide a description and specify a query to filter your data.

- **Description** - provide a description of the query here.
- **Query** - enter the actual text of the query here.

9.1.11 Saved Search filter
The Saved Search filter allows you to filter your audit data based on documents returned by any saved search available in your workspace. Select Saved Search from the filter type drop-down to bring up a window in where you can select the saved searches you want to apply to the audit data.
Items in the Available list are not applied by the filter, and items in the Selected list are applied by the filter. To move an item from one list to the other, click on it to select it, and use the arrow button to move it into the other list. You can select multiple entries at once by holding down the CTRL key while clicking each item. You can also select all items between two items you select by holding down the Shift key during selection.

### 9.2 Saving and loading filter sets

You have the option of saving one or more applied filters as a filter set, which you can then treat as a template and load at a future date to filter your audit data. This prevents you from having to manually configure and apply the same individual filters you created before.

To create a filter set, perform the following steps:
1. Click the save icon in the upper right corner of the filter pane.

![Filter Pane Image]

2. In the Filter Set layout, complete the following fields.

![Filter Set Layout Image]

- **Name** - enter the name of the filter set.
- **Order** - enter the order in which you'd like this filter set to appear in the Load Filter Set drop-down.

3. Click **Save** to save the filter set and make it available for selection in the Load Filter Set drop-down. Select **Save As** to save the set as a copy of one that you might have chosen to edit rather than overwriting the filter set you initially used as a template.

The new filter set is now available for selection in the drop-down.
9.3 Adding widgets

Widgets allow you to visualize audit data according to your preferences. To add a widget, perform the following steps:

1. Click Add Widget in the Home tab.

   **Note:** You need to save the dashboard to save a widget to the dashboard. If you don't save the dashboard and you refresh the page or leave the page and come back, the new widget will be gone.

1. Select from the available widget types.
2. Click **Add Widget** again to bring up the visualization of the audit data for that type. Once you add a widget you can re-size, drag, and drop it to a new location in the Home tab. The following sections provide information on how to interact with each widget type.

**9.3.1 User Actions widget**

The User Actions widget displays frequency metrics in a pie chart, bar graph, or grid format. You can switch between data visualization formats from the menu that appears on hover over the widget.

Click 👉 to display the user actions as a pie chart. Hovering over any segment of the pie chart lists the user action, the audit count, and the percentage of the data set, and hovering over any action in the list in the right highlights the corresponding segment of the pie chart. You can click on any region of the pie chart to limit your results to user actions of that particular type. Relativity adds a filter indicator to the filter list when you use this method to limit the data set.

Click ➕ to display the user actions as a bar graph. Hovering over any bar displays the count. You can click on any bar to limit your results to user actions of that particular type. Relativity adds a filter indicator to the filter list when you use this method to limit the data set.
Click ☐ to display the user actions in table format. The User Actions column displays the name of the user action, and the Count column displays the number of user actions that occurred within the time span limitations of your data set. You can click on any row in the list to limit your results to user actions of that particular type. Relativity adds a filter indicator to the filter list when you use this method to limit the data set.
The following fields appear in the User Actions widget:

- **Create** - the number of create commands executed on the data set.
- **Update** - the number of updates to objects in the data set.
- **Document Query** - the number of document queries executed on the data set.
- **Query** - the number of SQL queries executed on the data set.
- **Search Cache Table Creation** - the number of search cache tables created in accordance with Lucene Search provider usage.
- **Delete** - the number of deletions executed on the data set.
- **Update - Import** - the number of updates made as a result of imports into Data Grid.
- **Workspace Upgrade** - the number of workspace upgrades on the data set.
- **Import** - the number of imports into Data Grid that were executed.
- **Security** - the number of security edits applied to the data set.

### 9.3.2 User Names widget

The User Names widget displays frequency metrics in a pie chart, bar graph, or grid format. You can switch between data visualization formats from the menu that appears on hover over the widget.
Click 📊 to display the user names as a pie chart. You can click on any region of the pie chart to limit your results to a particular user. Hovering over any segment of the pie chart lists the user name, the audit count, and the percentage of the data set, and hovering over any object type in the list in the right highlights the corresponding segment of the pie chart. Relativity adds a filter indicator to the filter list when you use this method to limit the data set.

Click 📊 to display the user names as a bar graph. Hovering over any bar displays the audit count. You can click on any bar to limit your results to a specific user. Relativity adds a filter indicator to the filter list when you use this method to limit the data set.
Click to display the user names in two columns. The User Name column displays the name of the user, and the Count column displays the number of times an action was performed by that user within the time span limitations of your data set. You can click on any row in the list to limit your results to a particular user. Relativity adds a filter indicator to the filter list when you use this method to limit the data set.

<table>
<thead>
<tr>
<th>User Name</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steinberg - Paralegal, Jeff</td>
<td>73993</td>
</tr>
<tr>
<td>Service Account, Relativity</td>
<td>10275</td>
</tr>
<tr>
<td>Mecci, Zehra</td>
<td>704</td>
</tr>
<tr>
<td>Thorpe, David</td>
<td>465</td>
</tr>
<tr>
<td>Admin, Relativity</td>
<td>254</td>
</tr>
<tr>
<td>Smith, Jane</td>
<td>229</td>
</tr>
<tr>
<td>Ye, Crystal</td>
<td>165</td>
</tr>
</tbody>
</table>
9.3.3 Object Types widget

The Object Types widget displays frequency metrics in a pie chart, bar graph, or grid format. You can switch between data visualization formats from the menu that appears on hover over the widget.

Click to display the object types as a pie chart. You can click on any region of the pie chart to limit your results to objects of that particular type. Hovering over any segment of the pie chart lists the object type, the audit count, and the percentage of the data set, and hovering over any object type in the list in the right highlights the corresponding segment of the pie chart. Relativity adds a filter indicator to the filter list when you use this method to limit the data set.

Click to display the object types as a bar graph. Hovering over any bar displays the count. You can click on any bar to limit your results to objects of that particular type. Relativity adds a filter indicator to the filter list when you use this method to limit the data set.
Click to display the object types in two columns. The Object Types column displays the name of the object type, and the Count column displays the number of times an action occurred on that object type within the time span limitations of your data set. You can click on any row in the list to limit your results to objects of that particular type. Relativity adds a filter indicator to the filter list when you use this method to limit the data set.
9.3.4 Actions over Time widget

The Actions over Time widget displays a line graph that shows the number of audited user actions over a specified period of time.
The drop-down menu that appears in the lower left corner of the Actions over Time widget allows you to configure the units of time in the following increments:

- Every 1 second
- Every 1 minute
- Every 5 minutes
- Every 10 minutes
- Every 30 minutes
- Every 1 hour
- Every 3 hours
- Every 1 day
- Every 1 week
- Every 1 month
- Every 1 year

You can also click and drag over a range within the Actions over time widget to apply a filter that narrows your results to the selected range.
9.3.5 Results widget

The Results widget displays all filtered results in a grid format. Select Results from the filter type drop-down to bring up a window resembling a standard Relativity item list.

The following information is listed in the Results widget view:

- **Details** - provides the option of viewing the query text of an audit record, as well as the full details of a that record in both table and JSON formats. These options are discussed in detail below.
- **ID** - lists the audit ID number.
Time Stamp - lists the date and time of the event’s occurrence.

Name - lists the title of the artifact on which the event was performed.

Action - lists the name of the action performed. For example, Query.

ObjectType - lists the type of object on which the event was performed.

ExecutionTime - lists the duration of the event.

ArtifactID - lists the artifact ID of the audit.

UserName - lists the user name of the user who performed the action.

9.3.5.1 Expanded record view

To display the query text of an audit record, click + in the Details field for that record.

The Name and Value columns display the following information in this view:

- **auditElement.Query Text** - provides the code comments and select parameters of the query text associated with the audit action.

- **auditElement.Milliseconds** - provides the time it took the system to execute the audit action.

9.3.5.2 Full Details window

To view the full details of an audit record from the Results widget, click in the Details field for that record.
The names and values of the each field display on the table view of the Full Details window, but the specific field types vary according to audit type.

- **ID** - lists the audit ID number.
- **Time Stamp** - lists the date and time of the event's occurrence.
- **ArtifactName** - lists the title of the artifact on which the event was performed.
- **Action Name** - lists the name of the action performed.
- **ActionID** - lists the action ID number of the action performed.
- **ObjectTypeName** - lists the type of object on which the event was performed.
- **ObjectTypeID** - lists the Object Type ID number of the object on which the action was performed.
- **ExecutionTime** - lists the duration of the event.
- **ArtifactID** - lists the artifact ID of the audit.
- **UserName** - lists the user name of the user who performed the action.
- **UserID** - lists the UserID number of the user who performed the action.
- **Details.auditElement.QueryText** - lists the SQL query text of the action performed.
- **Details.auditElement.Milliseconds** - the time it took the audit action to run.

Click the **Display in JSON format** link at the bottom of the window to display all of the above information related to the audit record in JSON format.
The Display in table format link reverts the display to the original formatting.

Click Close in the upper right corner of the window to exit the Full Details window.

### 9.4 Creating dashboards

You can organize any of the above widgets according to your preferences by creating a custom dashboard. Use the following steps to create your own custom dashboard:
1. Click the pencil icon to bring up the **Dashboard** layout.

2. Enter the name of the dashboard in the **Name** field.

3. Enter an order number in the **Order** field to determine the order the dashboard appears in the drop-down menu.

4. Click **Save** when you've configured your dashboard according to your preferences. You can also use the drop-down menu to **Save As** if you want to save your dashboard as a copy of one that you might have chosen to edit rather than overwriting the dashboard you initially used as a template.

After you save it, the new dashboard is available for you to select from the dashboard drop-down next to the pencil icon.

9.5 **Admin tab**

The Admin tab provides you with the option of generating a migration error and migration status report for you to monitor the status of audit data as you migrate it into Data Grid. You also have the option of running a migration error retry script.

9.5.1 **Migration Error Report**

The Migration Error Report generates a report that provides information on all errors that have occurred while migrating audit data into the Data Grid repository, including batch and document errors.
To preview the script, select the radio button for this script and click **Preview**. To run this report, click **Run**. The following results populate the bottom of the window.

**Note:** Data Grid for Audit automatically retries certain audit migration errors, like SQL timeouts, invalid XML, and inability to connect to Elasticsearch. With this improvement, the migration error report may display a total of 0 errors because the system automatically retried and resolved those common errors. In this situation, there is no need to run the Migration Error Retry Script.
## Total Error Counts

<table>
<thead>
<tr>
<th>Batch Errors</th>
<th>Document Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>

## Batch Errors

<table>
<thead>
<tr>
<th>Batch ID</th>
<th>Audit ID Range</th>
<th>Time Stamp</th>
<th>Error Message</th>
</tr>
</thead>
</table>
- **Total Error Counts** - this section provides a summary of all errors that occurred during migration.
  - **Batch Errors** - lists the total number of batch errors that occurred during migration.
  - **Document Errors** - lists the total number of document errors that occurred during migration.

- **Batch Errors** - this section provides a summary of all batch errors that occurred during migration.
  - **Batch ID** - lists the batch ID number of the batch that received an error.
  - **Audit ID Range** - lists the first and last Audit ID numbers contained in the batch that received an error.
  - **Time Stamp** - lists the time at which the error occurred.
  - **Error Message** - lists the text of the batch error.

- **Document Errors** - this section provides a summary of all document errors that occurred during migration.
  - **Audit ID** - lists the audit ID number of the document that received an error.
  - **Batch ID** - lists the batch ID number of the batch in which the document that received an error was contained.
  - **Time Stamp** - lists the time at which the error occurred.
  - **Error Message** - lists the text of the document error.

### 9.5.2 Migration Error Retry Script

After identifying errors with the Migration Error Report, an admin can investigate and fix specific audit issues. Once the issues have been fixed, this script will resubmit audit records with errors to be migrated into the Data Grid repository.
In the **Error type to retry** drop-down, select from the following options:

- **Batch Errors** - retries all batches with errors as reported in the Migration Error Report.
- **Document Errors** - retries all documents with errors as reported in the Migration Error Report
- **All Errors** - retries all batch errors and document errors as reported in the Migration Error Report.

Click **Preview** to see a sample of the report, or select the appropriate option and click **Run** to execute the script.

### 9.5.3 Migration Status Report

The Migration Status Report generates a report that shows the progress of the migration of audit data into the Data Grid repository. This includes audits in SQL, audits pending migrations, and migration errors. To preview the script, select the radio button for this script and click **Preview**.
Click Run to generate this report and display the following information:

- **Audits in SQL** - lists the number of audits in the SQL database.
- **Audits Pending Migration** - lists the number of audits in queue for migration into Data Grid.
- **Migration Errors** - lists the number of errors that occurred during migration.

### 9.6 Audit Workspace Setting tab

The Audit Workspace Setting tab of the Data Grid for Audit application allows you to configure whether Relativity displays audits from SQL or Data Grid on the document history pane in the viewer and the View Audit button on object layout pages.

**Note**: You must have permission to view this tab in order to configure the setting.

If you set audits to display from Data Grid, there may be a slight delay in viewing the most recent audits. A "Last Updated" value next to audits indicates the last time all audits were verified as migrated for that object. This let's you know that several of a document's most recent audits might not be migrated yet.

The Audit Workspace Setting tab contains one setting, DisplayAuditDataSource.
You can configure the DisplayAuditDataSource setting with one of three values:

- **0** - displays audits from SQL until the first point that the audit migration has no more audits to migrate. At this point, the system automatically updates this value to "1," and audits are displayed from Data Grid going forward.
- **1** - displays audits from Data Grid only.
- **2** - displays audits from SQL only.

By default, DisplayAuditDataSource is set to display audits from SQL only. To configure DisplayAuditDataSource, click the **Edit** link and modify the **Value** field in the subsequent layout.
10 Searching documents over 100MB in a Data Grid enabled workspace

This page describes workflows for identifying documents with long text fields that are too large for Data Grid and searching for those documents in SQL. The basic workflow is as follows:

1. Attempt to **ingest** long text fields into Data Grid
2. Identify documents with long text errors.
3. Store long text fields with errors in SQL
4. Search against your Data Grid and SQL long text fields using dtSearch with saved searches.

The workflow you follow depends on how you're ingesting long text fields into Data Grid. This workflow assumes your workspace and extracted text field are Data Grid enabled. You can ingest long text fields into Data Grid using one of four methods:

- Importing documents using the Relativity Desktop Client
- Publishing a Processing Set
- Migrating text using the Data Grid Text Migration application
- Running an OCR set

### 10.1 Importing documents using the Relativity Desktop Client

#### 10.1.1 Ingesting long text fields into Data Grid

Use the Relativity Desktop Client (RDC) to import documents into Data Grid. If you encounter any errors during your import, export the error file and error report. For more information, see the Desktop Client guide.

#### 10.1.2 Identifying documents with long text errors

Documents with text larger than 100MB aren't imported to Data Grid, and the following message appears in the error report exported with your error file:

"This document has a Data Grid field containing data that exceeds the maximum size threshold. Relativity skipped this document's Data Grid upload."

Correct any non-Data Grid errors in your error file and re-import your documents until you’re left with a load file that contains only documents with Data Grid long text errors. The long text for these documents will be stored in SQL. For more information, see the Desktop Client guide.

Note: If you don’t correct and re-import documents with non-Data Grid errors, those documents will never be imported into Data Grid, and you won’t be able to search on them.

10.1.3 Storing long text fields with errors in SQL

Use the following workflow to import long text fields too large for Data Grid to SQL:

1. In Relativity, create a long text field in SQL to store text fields larger than 100MB. Name it something specific, for example, Extracted Text Large.

2. Return to the RDC to import long text fields too large for Data Grid to SQL. Perform an Append/Overlay load using the error file with only Data Grid errors.
3. Map the extracted text load file field to the long text field in SQL, for example, Extracted Text Large.

4. On the import menu, click **Import File**.

5. Relativity stores the large text in SQL in the long text field you created, for example, Extracted Text Large.
10.1.4 Searching against your Data Grid and SQL Long Text fields

Once you've imported your documents into Relativity, you can search against all of your Data Grid and SQL long text fields, including fields with oversized text. For more information, see Searching against your Data Grid and SQL long text fields on page 140.

10.2 Publishing a processing set

In order to complete this workflow, you must have access to the following:

- Administrative access to the Invariant database server
- Microsoft SQL Server Management Studio
- Microsoft Excel
- Relativity Desktop Client.

If you don't have access to any of the above, contact your network administrator or Relativity Client Services.

10.2.1 Ingesting long text fields into Data Grid

Use Relativity Processing to publish files in a processing set to Data Grid. If any errors occur during the publish process, view them on the Errors tab. For more information, see the Processing User Guide.
10.2.2 Identifying documents with long text errors

Documents with text larger than 100MB aren’t imported to Data Grid, and the following message appears on the Errors tab:

"This document has a Data Grid field containing data that exceeds the maximum size threshold. Relativity skipped this document’s Data Grid upload."

Create a new view in your workspace on the Errors tab to store processing errors. Once you create the view, export a list of documents with large text errors.

To create this view and export the file, complete the following:

1. On the Errors tab, click +.
2. Complete the fields on the Basic Information form.
3. On the Fields form, remove all fields except File ID
4. On the Conditions form, add the following conditions:
   1. Click Add Condition.
   2. Set Field to Processing Set and Operator to any of these.
   3. Click [icon], and then select your processing set.
   4. Click Set.
   5. Click Add Condition.
   6. Set Field to Message and Operator to is like.
   7. Copy and paste the following into the textbox: This document has a Data Grid field containing data that exceeds the maximum size threshold.
5. Click Save.
6. On the Processing Errors tab, select your new view.
7. Select all documents in the view, and then click Export to File. The Exporting Documents form appears.
8. Next to Format, select Excel spreadsheet via HTML (.xls).
9. Click Run.
11. Open a remote desktop connection to the Invariant database server.
12. Copy the CSV file to the Invariant database server.
10.2.3 Storing long text fields with errors in SQL

Use the following workflow to process documents too large for Data Grid to SQL:

1. Open SQL Server Management Studio.
2. Run the following SQL script on the **Invariant stores database** to create a load file to import text too large for Data Grid to SQL. Update the file path to the location where you saved CSV file. Update other user inputs per your environment.

```sql
-- To Be Run On Invariant Stores SQL Server

--***USER INPUTS***
DECLARE @FilePath AS NVARCHAR(255) = 'c:\export.csv'  --Path to file exported from Relativity containing Processing Set File Ids
USE [INV1234567]  --Invariant database for workspace (format: INV + Workspace ID)
IF OBJECT_ID('dbo.DataGridRelImport') IS NOT NULL
    DROP TABLE [dbo].[DataGridRelImport]
CREATE TABLE [dbo].[DataGridRelImport] (  
    [FileId] [bigint],
    CONSTRAINT [PK_DataGridRelImport] PRIMARY KEY CLUSTERED ([FileId] ASC)
)
DECLARE @sqlBulkInsert NVARCHAR(MAX) = 'BULK INSERT [dbo].[DataGridRelImport] FROM ''' + @FilePath + ''' WITH (FIRSTROW = 2, FIELDCOLUMNQUOTE = '''',''', ROWTERMINATOR = '''

EXEC(@sqlBulkInsert);
SELECT [EDN].[BatesNumber] AS [ControlNumber],  
    [S].[ExtractedTextLocation] AS [ExtractedText]  
FROM [dbo].[DataGridRelImport] AS [DGR]  
INNER JOIN [dbo].[ExportDocNumbers] AS [EDN] ON [EDN].[FileId] = [DGR].[FileId]  
INNER JOIN [dbo].[Matter] AS [M] ON [M].[FileId] = [DGR].[FileId]  
INNER JOIN [dbo].[Storage] AS [S] ON [S].[StorageId] = [M].[StorageId]  
DROP TABLE [dbo].[DataGridRelImport]
```

3. Click **Execute**. The query returns a table in the **Results** pane that contains the control number of the documents with Data Grid long text errors as well as the path to the extracted text for those documents.
4. Select all items in the table, and then **Copy with Headers**.

5. Paste the results into a new text file, and then save the file as **ProcessingLoadFile.dat**.

6. Navigate to the following path, and open kCura.EDDS.Winform.exe.config in a text editor: `C:\Program Files\kCura Corporation\Relativity Desktop Client\kCura.EDDS.Winform.exe.config`

7. Update **kCura.WinEDDS** with the following:

```xml
<kCura.WinEDDS>
  <add key="WebAPIOperationTimeout" value="60000000"/>
  <add key="ImportBatchMaxVolume" value="200485760"/>
  <add key="ImportBatchSize" value="1"/>
  <add key="ExportBatchSize" value="1000"/>
  <add key="DynamicBatchResizingOn" value="False"/>
  <add key="MinimumBatchSize" value="1"/>
  <add key="CreateErrorForEmptyNativeFile" value="False"/>
</kCura.WinEDDS>
```

These settings help the RDC better handle large documents. Revert these changes after you've successfully imported your large documents.

8. Click **Save**.

9. In Relativity, navigate to the **Instance Settings** tab.

10. Filter for the **MassImportSqlTimeout**, and then click **Edit**.
11. Update the instance setting **Value** from 60 to 600 to better handle large documents. Revert this change after you’ve successfully imported your large documents.

![Instance Setting Information](image1)

12. Navigate to your workspace, and then create a long text field in SQL to store text fields larger than 100MB. Name it something specific, for example, Extracted Text Large.

![Object](image2)

13. Use the Relativity Desktop Client to import long text fields too large for Data Grid to SQL using the `ProcessingLoadFile.dat` you created in step 8.
- Map the control number fields.
- Map the extracted text load file field to the long text field in SQL, for example, Extracted Text Large.
- Set Overwrite to Overlay Only.
- Set Overlay Identifier to Control Number.
- Select the checkbox next to Cell contains file location, and then select the long text field in SQL from the drop-down. For example, Extracted Text Large.


15. Relativity stores the large text in SQL in the long text field you created, for example, Extracted Text Large.
Note: After you've successfully imported your data, revert the `MassImportSqlTimeout` instance setting and `kCura.EDDS.Winform.exe.config` to the original settings.

10.2.4 Searching against your Data Grid and SQL Long Text fields
Once you've imported your documents into Relativity, you can search against all of your Data Grid and SQL long text fields, including fields with oversized text. For more information, see Searching against your Data Grid and SQL long text fields on page 140.

10.3 Migrating text using the Data Grid Text Migration application

10.3.1 Ingesting long text fields into Data Grid
Migrate text from SQL to Data Grid using the Data Grid Text Migration application. For more information, see Data Grid Text Migration.

10.3.2 Identifying documents with long text errors
Document long text fields text larger than 100MB aren't imported to Data Grid, and the following message appears on the errors tab:

"Document [Number] contains field(s) ExtractedText with size exceeding maximum allowed byte size (DataGridMaximumFieldSizeInBytes instance setting)."
You can also use the DG Long Text Size Exceeded field to filter for documents with long text fields larger than 100MB. In the following example, AZIPPER_0007747, AZIPPER_0007748, and AZIPPER_0007749 each have an extracted text field whose size exceeds the 100MB limit.

Create a saved search of documents with Data Grid long text errors. Later, you can use this saved search to perform a mass replace to copy oversized text to a new long text field in SQL.

To create the saved search, complete the following:

1. Click 🎨 at the top of the browser.
2. Click New Search.
3. Fill out the information in the Information section.
4. On the Conditions tab, click Add Condition, and then select the DG Long Text Size Exceeded field.
5. Next to Operator, select Is set, and then click Apply.
6. On the **Fields** tab, remove all fields except **Extracted Text**.

7. Click **Save & Search**.

---

10.3.3 Storing long text fields with errors in SQL

Use the following workflow to perform a mass replace operation to migrate too large for Data Grid to SQL:

1. In Relativity, create a long text field in SQL to store text larger than 100MB. Name it something specific, for example, Extracted Text Large.
2. Return to your saved search of documents with oversized long text fields.

3. Perform a mass replace operation on the documents with oversized fields to copy oversized text to your new long text field in SQL. For example, from Extracted Text to Extracted Text Large.
   a. From the mass operations bar on the document list, choose whether to replace text in All or Checked items in the current returned set.
   b. Select Replace in the drop-down menu. The Replace Document form appears.
   c. Complete the fields on the Replace Documents form:
      - Field to Update - select the long text field you created, for example, Extracted Text Large.
      - Action - select Replace Entire Field.
      - Update With - select Field, and then select Extracted Text - Long Text.

4. Run the Text Migration Complete script to finalize the text migration process. The Text Migration Complete script configures the long text field(s) on the Document object to be Data Grid enabled in order to use the migrated data in Data Grid. For more information, see Running the Text Migration Complete script.

10.3.4 Searching against your Data Grid and SQL Long Text fields

Once you’ve imported your documents into Relativity, you can search against all of your Data Grid and SQL long text fields, including fields with oversized text. For more information, see Searching against your Data Grid and SQL long text fields.
### 10.4 Running an OCR set

#### 10.4.1 Ingesting long text fields into Data Grid

Run an OCR set to save OCR text to a Data Grid-enabled long text field. For more information, see the Admin Guide.

#### 10.4.2 Identifying documents with long text errors

Document long text fields text larger than 100MB aren’t imported to Data Grid, and the following message appears in the OCR status field:

"This field has data that exceeds the maximum size threshold."

<table>
<thead>
<tr>
<th>OCR Status</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Status:</strong></td>
</tr>
<tr>
<td><strong>Image Completion:</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Last Run Error:</strong></td>
</tr>
<tr>
<td><strong>Response Result:</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

#### 10.4.3 Storing long text fields with errors in SQL

Use the following workflow to store OCR text too large for Data Grid in SQL.

1. In Relativity, create a long text field in SQL to store OCR text larger than 100MB. Name it something specific, for example, OCR Text Large.
2. Return to your OCR set and click **Edit**.

3. Update the **Destination Field** to the long text field you created, for example, **OCR Text Large**. When you re-run your OCR job, documents with text too large for Data Grid are stored in this SQL field.

4. Click **Save**.
5. Click **Retry Errors**.

When you click **Retry Errors**, Relativity re-runs your OCR job with the images that previously ran with errors, in this case text too large for Data Grid. Because you updated your destination field to a SQL long text field (for example, OCR Text Large), Relativity stores this OCR text in SQL.

### 10.4.4 Searching against your Data Grid and SQL Long Text fields

Once you’ve imported your documents into Relativity, you can search against all of your Data Grid and SQL long text fields, including fields with oversized text. For more information, see [Searching against your Data Grid and SQL long text fields](#).

### 10.5 Searching against your Data Grid and SQL long text fields

As you import your data into Data Grid, you can immediately search on a majority of your documents using Lucene Search. For more information, see [Lucene Search on page 74](#).

You can’t use Lucene Search on the small percentage of documents with long text fields stored in SQL. Once you migrate fields too large for Data Grid to SQL, you must create a dtSearch index to search on these documents.

You have two options when creating a searchable set for your dtSearch:

1. Create a saved search of only documents with long text stored in SQL. This will likely be a small percentage of documents.

2. Create a saved search of all documents in your workspace, including both long text stored in SQL and long text stored in Data Grid. This searchable set may take longer to index.

Once you create your dtSearch, you can still use Lucene search on a majority of your data. We recommend using dtSearch only when searching on the small subset of data too large for Data Grid.

To search only the long text fields stored in SQL, complete the following:

Click to expand

1. Click **at the top of the browser.**

2. Click **New Search.**
3. Fill out the information in the **Information** section.

4. On the **Conditions** tab, click **Add Condition**, and then select the Extracted Text Large field.

5. Next to **Operator**, select **Is set**, and then click **Apply**.

6. On the **Fields** tab, remove all fields except **Extracted Text** and **Extracted Text Large**.

7. Click **Save & Search**.

8. Navigate to the **Search Indexes** tab and click **New dtSearch Index**.

9. Complete the fields on the dtSearch index form.
   a. Set **Searchable set** to your saved search long text in SQL.

10. Click **Save**.

Once you index your dtSearch, you can search all of your Data Grid and SQL long text. For more information, see the Searching Guide.

In order to search against all of your data, including long text stored in Data Grid data and long text stored in SQL, you must create a saved search containing Data Grid and SQL long text fields, and then build a dtSearch index on the saved search.

To create a dtSearch on Data Grid and SQL long text fields, complete the following:

> Click to expand

   1. Click ![icon] at the top of the browser.

   2. Click **New Search**.

   3. Fill out the information in the **Information** section.

   4. On the **Conditions** tab, click **Add Condition**, and then select the Extracted Text field.

   5. Next to **Operator**, select **Is set**, and then click **Apply**.

   6. Click **Add Condition**, and then select the long text field stored in SQL, for example, Extracted Text Large.

   7. Next to **Operator**, select **Is set**, and then click **Apply**.
8. Change the boolean operator to OR.

9. On the Fields tab, remove all fields except Extracted Text and Extracted Text Large.

10. Click Save & Search.

11. Navigate to the Search Indexes tab and click New dtSearch Index.

12. Complete the fields on the dtSearch index form.
   a. Set Searchable set to your saved search containing Data Grid and SQL long text.

Once you index your dtSearch, you can search all of your Data Grid and SQL long text. For more information, see the Searching Guide.
Proprietary Rights

This documentation ("Documentation") and the software to which it relates ("Software") belongs to Relativity ODA LLC and/or Relativity's third party software vendors. Relativity grants written license agreements which contain restrictions. All parties accessing the Documentation or Software must: respect proprietary rights of Relativity and third parties; comply with your organization’s license agreement, including but not limited to license restrictions on use, copying, modifications, reverse engineering, and derivative products; and refrain from any misuse or misappropriation of this Documentation or Software in whole or in part. The Software and Documentation is protected by the Copyright Act of 1976, as amended, and the Software code is protected by the Illinois Trade Secrets Act. Violations can involve substantial civil liabilities, exemplary damages, and criminal penalties, including fines and possible imprisonment.

©2019. Relativity ODA LLC. All rights reserved. Relativity® is a registered trademark of Relativity ODA LLC.