# Table of Contents

How Fusion Works ................................................................. 1
Fusion Concepts and Components ........................................ 2
  Apps .................................................................................. 2
    Collections ................................................................. 2
    Datasources .............................................................. 2
      Connectors .............................................................. 2
      Parsers ................................................................. 2
      Index pipelines ....................................................... 2
    Query Pipelines ......................................................... 2
Fusion Components ............................................................... 3
  Apache Solr ..................................................................... 3
    Solr configuration ....................................................... 3
    Solr logs ................................................................... 3
    Accessing the Solr UI .................................................. 3
    Solr documentation .................................................... 3
  Apache Spark ................................................................... 3
  Apache ZooKeeper .......................................................... 4
    ZooKeeper Terminology ............................................... 4
    Fusion ZooKeeper Nodes ............................................. 5
Jetty .................................................................................... 8
Getting Data In ..................................................................... 9
Getting Data Out .................................................................. 10
Building Applications ........................................................ 11
Fusion UI .............................................................................. 12
Logging in ............................................................................ 13
  A brief tour ....................................................................... 15
    The Fusion launcher .................................................... 15
    The Fusion workspace ................................................ 15
      App tool menus ....................................................... 16
      Finding objects in Object Explorer ............................ 17
    System-wide tools ........................................................ 17
Working With Apps ............................................................... 18
  Create an app ................................................................... 19
  Share objects between apps ............................................. 20
  Export an app .................................................................. 22
    Export an app with the Fusion UI ................................. 22
    Export an app with the Objects API .............................. 23
  Import an app .................................................................... 24
    Import an app with the Fusion UI .................................. 24
    Import an app with the Objects API .............................. 26
  Delete an app .................................................................... 27
Index Workbench ................................................................. 28
How Fusion Works

These topics give you a high-level view of how Fusion works:

• Fusion Concepts and Components
• Getting Data In
• Getting Data Out
• Building Applications
Fusion Concepts and Components

Basic Fusion concepts are explained below. Since the core of Fusion Server is Solr, you may also find it useful to familiarize yourself with Solr terminology.

Apps

Fusion apps provide tailored search functionality to specific groups of users.

An app is a named set of linked objects, including collections, datasources, index and query pipelines, index and query profiles, parsers, and more. Using roles and security realms, you can define security on a per-app basis.

See App Management in the Search Development Guide.

Collections

Collections consist of stored data and the datasources that determine how the data is ingested and indexed. Collections are a way to logically group your data sets. Fusion’s concept of collections is the same as Solr collections. See Collection Management in the Search Development Guide.

Datasources

Datasources are the configurations that determine how data is ingested and indexed. Each datasource includes a connector configuration, a parser configuration, and an index pipeline configuration. See Datasource Configuration in the Search Development Guide.

Connectors

Connectors are the conduit between Fusion and your external data sources. Connectors retrieve your data and import it into Fusion Server. See the Connectors Reference Guide for a complete list of available connectors.

Parsers

Parsers interpret incoming data in order to determine its format and fields. A parser consists of a sequence of parsing stages, each designed to parse a different data format, sometimes recursively. See the Parser Stages Reference Guide for complete details about all available parsing stages.

Index pipelines

Index pipelines format the incoming raw data data into fielded documents that it can be indexed and searched by the Solr core. A pipeline consists of a sequence of stages, and each stage performs a different kind of processing based on user-configured logic. See the Index Pipeline Stages Reference Guide for a complete list of available index pipeline stages.

Query Pipelines

Query pipelines manipulate incoming queries and return an ordered list of matching results from Solr. Individual search results are called documents. See Query Pipeline Configuration in the Search Development Guide.
Fusion Components

Apache Solr

Solr is the search platform that powers Fusion. There are multiple aspects to Fusion’s use of Solr:

- Fusion components manage Solr search and indexing and provide analytics over these collections. Fusion’s analytics components depend on aggregations over information which is stored in a Solr collection.
- Fusion collections are all Solr collections.
- Application data is stored as one or more Solr collections.
- Fusion’s own logs are stored as Solr collections.
- A few Fusion service APIs use Solr as a backing store, notably Parameter Sets.

Solr configuration

Fusion requires that Solr run with SolrCloud enabled.

Configuration for Solr's Web service is in fusion/5.0.x/apps/jetty/solr.

Solr logs

Solr log files are in fusion/5.0.x/var/log/solr.

Accessing the Solr UI

With Fusion installed out of the box, you can still access the Solr UI at http://localhost:8983/solr/.

Solr documentation

Solr documentation and additional resources are available at http://lucene.apache.org/solr/resources.html.

You can also find plenty of Solr tips and technical discussions in our knowledge base, blog, and webinars. Lucidworks also maintains a search interface to Solr’s community discussions at searchhub.org.

Apache Spark

Apache Spark is a fast and general execution engine for large-scale data processing jobs that can be decomposed into stepwise tasks which are distributed across a cluster of networked computers. Spark provides faster processing and better fault-tolerance than previous MapReduce implementations. The following schematic shows the Spark components available from Fusion:
Apache ZooKeeper

Apache ZooKeeper is a distributed configuration service, synchronization service, and naming registry. Fusion uses ZooKeeper to configure and manage all Fusion components in a single Fusion deployment, therefore a ZooKeeper service must always be running as part of the Fusion deployment. For high availability, this should be an external 3-node ZooKeeper cluster. All Fusion Java components communicate with ZooKeeper using the ZooKeeper API.

For ZooKeeper installation instructions, see the ZooKeeper documentation.

You can find ZooKeeper's logs at fusion/5.0.x/var/log/zookeeper.

ZooKeeper Terminology

- **znode**: ZooKeeper data is organized into a hierarchal name space of data nodes called znodes. A znode can have data associated with it as well as child znodes. The data in a znode is stored in a binary format, but it is possible to import, export, and view this information as JSON data. Paths to znodes are always expressed as canonical, absolute, slash-separated paths; there are no relative reference.

- **ephemeral nodes**: An ephemeral node is a znode which exists only for the duration of an active session. When the session ends the znode is deleted. An ephemeral znode cannot have children.

- **server**: A ZooKeeper service consists of one or more machines; each machine is a server which runs in its own JVM and listens on its own set of ports. For testing, you can run several ZooKeeper servers at once on a single workstation by configuring the ports for each server.

- **quorum**: A quorum is a set of ZooKeeper servers. It must be an odd number. For most deployments, only 3 servers are required.

- **client**: A client is any host or process which uses a ZooKeeper service.

See the official ZooKeeper documentation for details about using and managing a ZooKeeper service.
Fusion ZooKeeper Nodes

Fusion configuration data is stored in ZooKeeper under two znodes:

- Node **lucid** stores all application-specific configurations, including collection, datasource, pipeline, signals, aggregations, and associated scheduling, jobs, and metrics.
- Node **lucid-apollo-admin** stores all access control information, including all users, groups, roles, and realms.

The Solr Admin tool provides a ZooKeeper node browser tool. In the case of the Fusion default developer deployment, the Fusion runs scripts are configured to run the instances of both Solr and ZooKeeper which are included with the Fusion distribution, and therefore we take a fresh installation of a Fusion developer instance and use the embedded Solr’s Admin tool to explore how Fusion’s configurations are managed in ZooKeeper.

On initial install, the "lucid" znode contains the set of default configurations used by Fusion’s services:

![ZooKeeper Node Diagram]

The "lucid-apollo-admin" znode contains the set of nodes used by Fusion’s access control services:
In the above screenshot, the ZooKeeper node browser is browsing the contents of znode "lucid-apollo-admin/users" which is empty. The Fusion distribution ships without any user accounts. The initial user added to Fusion is the Fusion native realm "admin" user. This entry is only created on initial startup via the Fusion UI "set admin password" panel. Once you submit the admin password, the admin user account is created. Until Fusion contains at least the admin user account, you cannot use the system, because all Fusion requests require proper authorization.

Once the admin password is set, and you have created one or more Fusion collections and have populated them by running one or more datasources, these collections, datasources, pipelines, and other application configuration settings are stored under the "lucid" znode:
In the above screenshot, the ZooKeeper node browser is browsing the contents of znode “lucid/connectors/datasources/ds1”. This datasource was used to populate a Fusion collection with documents retrieved via a webcrawl. Note that in the initial screenshot for znode “lucid”, there is no “connectors” node at all.

The "lucid-apollo-admin" znode now contains one user accounts for user "admin":

```json
{
    "created": "2016-06-01T22:46:54.195465Z",
    "modified": "2016-06-01T22:46:54.195465Z",
    "connector": "lucid-apollo",
    "type": "web",
    "pipeline": "Datasources_Parsing",
    "properties": {
        "refreshOlderThan": -1,
        "f.appendTimestampToLinks": false,
        "refreshErrors": false,
        "restrictToTreeIgnoreHostPrefixes": [ "www." ],
        "deduplicateSignature": true,
        "crawlType": "in-memory",
        "crawWithHeader": true,
        "f discardLinksHREF": false,
        "f respectsHostEq zesRedirects": false,
        "f catchXmlType": "html",
        "splitArchive": true,
        "refreshAll": true,
        "f.defaultMIMEType": "application/octet-stream",
        "restrictToTreeIgnoreSubdomains": false,
        "lookAndFeel": "default"
    }
}
```
Jetty provides Web services for Fusion's UI, APIs, and Connectors, plus Solr. Each of those components runs inside its own instance of Jetty, using a separate configuration. Configurations for each component are located in fusion/5.0.x/apps/jetty.

Securing Fusion using SSL requires configuring Jetty to use SSL. For example, to secure the UI you need to modify the configuration in fusion/5.0.x/apps/jetty/admin-ui. See SSL Security (Unix) or SSL Security (Windows).

Log messages about Jetty are written to the log files for the components that use it: * fusion/5.0.x/var/log/ui * fusion/5.0.x/var/log/api * fusion/5.0.x/var/log/connectors * fusion/5.0.x/var/log/solr
Getting Data In

Data ingestion gets your data into Fusion Server, and data indexing stores it in a format that's optimized for searching. These topics explain how to get your data into Fusion Server in a search-optimized format.

- **Collections** are a way of grouping data sets so that related data sets can be managed together. Every data set that you ingest belongs to a collection. Any app can contain one or more collections. See Collection Management.

- **Datasources** are configurations that determine how your data is handled during ingest by Fusion Server’s connectors, parsers, and index pipelines. When you run a fully-configured datasource, the result is an indexed data set that's optimized for search, depending on the shape of your data and how you want to search it. See Datasource Configuration.

- In some cases, you might find that it's best to use other ingestion methods, such as the Fusion Bulk Loader, Hive, Pig, or pushing data to a REST API endpoint.

- Blob storage is a way to upload binary data to Fusion Server. This can be your own data, such as images or executables, or it can be plugins for Fusion Server, such as connectors, JDBC drivers, and so on.
Getting Data Out

• Query pipelines process search requests before sending them to Fusion Server’s Solr core. A query pipeline can perform transformations on a search request in order to customize the search results that Solr returns. See Query Pipeline Configuration.

• Search applications are the front-end interfaces that you build on top of Fusion. Your application makes calls to Fusion’s REST API in order to retrieve search results or perform other actions. Certain features, like autocomplete and synonyms, require some configuration on the Fusion back end. See Application Development.

• Whenever you’re getting data out of Fusion Server, you may find it handy to consult the Query Language Cheat Sheet.

If you are using Fusion AI with Fusion Server, you can generate and retrieve additional data for analysis or to enhance the end-user experience. For example, Fusion AI can produce sophisticated recommendations to guide end users to the best available results, including results that don’t exactly match the original user-submitted query. It can also perform machine learning functions that automatically improve search results based on the past activities of users. See the Fusion AI documentation.
Building Applications

Ultimately, Fusion Server is the back end for your search applications.

If you develop your own search application from scratch, it must make calls to Fusion’s REST API in order to retrieve search results or perform other actions, such as uploading blobs, modifying user roles, or sending messages.

Certain features require configuration on the Fusion back end before you can surface them in your application. The Search Development Guide explains these in detail:

- autocomplete
- faceting
- synonyms
- stopwords

If you are using Fusion AI, then your application can also send signals to Fusion and retrieve recommendations to personalize each user’s search experience.
Fusion UI

The Fusion UI gives you easy access to all of Fusion Server’s features.

The UI service utilizes the API service, so each UI feature has an API equivalent. For programmatic access to the features you see in the UI, see the API Reference Guide.
Logging in

The Proxy service runs on port 8764 by default. Access the Fusion UI through the Proxy service. To log in, point your browser to this port on the Fusion host, for example, http://localhost:8764/.

When you are the first user to log in:

Fusion Server prompts you to set the password for the "admin" account and agree to the license terms. All other accounts must be created by an admin.

Fusion Server also prompts you to register to send metrics to Lucidworks. You can click Skip to send no metrics; you can also register later.

The first time you log in:

Fusion Server opens the Quickstart, to give you a quick demonstration of Fusion’s most basic features. You can click Exit the Quickstart to close it and go to the Fusion launcher.

You can open the Quickstart at any time by navigating to the Fusion launcher and clicking New here? Get started....
The brief tour below shows you how to find your way around the interface. For a more detailed tour, try the Tutorials.
A brief tour

- The launcher is where you can view all of the apps in your Fusion environment and create new ones.
- The Fusion workspace is where you configure how an app ingests, indexes, queries, and analyzes data.

The Fusion launcher

When you log in to Fusion Server, the first page is the Fusion launcher. This is where you create and launch apps:

![Fusion launcher screenshot]

To return to the Fusion workspace, hover over [icon] and then click Return to launcher.

See Working With Apps for details about creating, configuring, and managing apps.

The Fusion workspace

You can reach the Fusion workspace by clicking on any app in the launcher. When you enter the Fusion workspace, you find all the tools you need to configure the currently-selected app:
• On the left side are the app tool menus, where you can find tools for working with your app.

• At the top, you can use Object Explorer to search for and browse objects that are in the current app, in any app, in no apps, and in any app or no apps.

• System-wide tools are in the upper right corner.

App tool menus

On the left side of the screen are five menus where you can find tools for working with your app:

• Indexing (getting data in)

The Index Workbench is the primary tool for getting data into your app through the Fusion UI. It provides an all-in-one view of the components of a datasource. You can also access those components individually through this menu.

See also Getting Data In, in the Search Development Guide.

• Querying (getting data out)

For getting data out, the Query Workbench is the main tool. It shows you a live preview of search results while you configure and fine tune your query pipelines.

See also Getting Data Out, in the Search Development Guide.

• Collection management

Manage your data collections and configure their features.

See also Collection Management in the Search Development Guide.

• System administration

System administration for your apps.

See the System Administration Guide.
Finding objects in Object Explorer

At the top of the Fusion workspace is a Find Objects search field. When you click **Find Objects**, Object Explorer opens.

In the **Filter** search box, you can search by name or type for all types of Fusion objects, such as collections, apps, pipelines, and more. Browse objects by clicking them. You can use Object Explorer to view and configure the links between objects. For example, you can use it to share an object between multiple apps.

System-wide tools

On the right are menus for system-wide actions:

- **Help**
  
  Click here to access the documentation from the Fusion UI.

- **Alerts**
  
  New alerts are briefly displayed in the foreground. Open this menu to see past alerts, or to clear the alert history.

- **Account**
  
  Here you can edit your profile, log out, or manage your Fusion license.
Working With Apps

Fusion apps provide tailored search functionality to specific groups of users.

An app is a named set of linked objects, including collections, datasources, index and query pipelines, index and query profiles, parsers, and more. You can define security on a per-app basis using roles and security realms.

Generally, you will create multiple apps for different purposes. You can view all of your apps in the launcher when you log in to the Fusion UI:

To enter the Fusion workspace for any app, click the app name.

In the Fusion workspace, you can hover over the top left corner to switch to another app, create a new app, or return to the launcher:
Create an app

To create an app:

1. Navigate to the launcher and click Create new app.
2. Enter an app name.

   Optionally, you can also enter a description and select the app tile color.

   If you want to choose the name of the first collection in the app, then deselect **Create new app default collection**. When you do this and click **Create app** (in the next step), Fusion Server prompts you to enter a name for the first collection. Otherwise, Fusion Server creates a default collection whose name is the same as the app name.

3. Click **Create App**.

   The Fusion workspace appears, and you are ready to work with your app’s first collection.

**Share objects between apps**

Sharing objects between apps lets you reuse parts of apps and synchronize apps without duplicating objects. For example, after you have developed a query pipeline that is well-suited to your data and search application, you can reuse it in another app with a different set of datasources.
Important

When a shared object is modified in one app, the changes affect all other apps that share the object.

To share an object between apps:

1. Navigate to the app that does not yet include the object.
2. Navigate to System > Object Explorer or click Find Objects.
3. Click In Any App.
4. Select the object you want to include in this app.
5. Hover over the object to reveal the app icon.
6. Click the app icon.
7. Select Add to this app.

Object Explorer now displays a link between the object and this app:
Now you'll see the shared object in the Fusion workspace for this app.

**Export an app**

To export an app, you can use either the Fusion UI or the Objects API.

| Important | When upgrading from one Fusion version to a later one, you must use the migrator to migrate objects. The migrator automates the process of translating relevant objects to the new version. Exporting an object from one Fusion version and importing it into a different Fusion version *isn’t* supported. |

**Export an app with the Fusion UI**

How to export an app with the Fusion UI

1. Navigate to the launcher.
2. Hover over the app you want to export and click the Configure icon:
3. In the app config window, click **Export app to zip**:

This downloads a zip file that you can import into other instances of Fusion Server.

**Export an app with the Objects API**

The examples below show how to export one or more apps:

Export all apps

```bash
curl -u user:pass http://localhost:8764/api/objects/export?type=app > all-apps.zip
```

Get all app IDs, then export one app by ID
> curl -u user:pass http://localhost:8764/api/apps

[{
  "id": "movies",
  "name": "Movies",
  "description": "Search the movielens database.",
  "dataUri": "/App-Tile-01-460x160.png",
  "properties": {
    "headerImageName": "headerImage1",
    "tileColor": "apps-darkblue",
    "previousCollectionId": "movies"
  }
}, {
  "id": "tech-pubs",
  "name": "TechPubs",
  "description": "Search the documentation.",
  "dataUri": "/App-Tile-02-460x160.png",
  "properties": {
    "headerImageName": "headerImage2",
    "tileColor": "apps-darkblue",
    "previousCollectionId": "tech-pubs"
  }
}]


Note

When you export an app that includes objects that are shared with other apps, then all apps linked to that object are also exported.

Export two apps by ID


Tip

For the app object type, the deep parameter is not used. Linked objects are always included when exporting apps.

Import an app

To import an app, you can use either the Fusion UI or the Objects API.

Import an app with the Fusion UI

How to import an app with the Fusion UI

1. Navigate to the launcher.
2. Click Import app.
3. Under **Data File**, click **Choose File** and select the zip file containing the app you want to import.

4. If your app has usernames and passwords in a separate file, select it under **Variables File**.

<table>
<thead>
<tr>
<th>Tip</th>
<th>If the variables file is included inside the zip file, then you don’t need to upload it separately.</th>
</tr>
</thead>
</table>

5. In some cases, you can edit parameter values to use the new values in the imported app. If this is the case, Fusion displays a dialog box that lets you edit the parameter values.
Make desired changes, and then click **Import**.

**Import an app with the Objects API**

Like any other Fusion object, an exported app is contained in a zip file which Fusion can consume upon import. The zip file does not need to be expanded first.

**Import objects from a file and stop if there are conflicts**

```bash
```

**Import objects, substitute the password variables, and merge any conflicts**

```bash
```

**Note**

`password_file.json` must contain plaintext passwords.

**Import a zip file of Fusion objects and merge any conflicts**

```bash
```
Delete an app

How to delete an app

1. Navigate to the launcher.

2. Click the Configure icon on the app you want to delete:

3. Click **Delete app**:

Fusion Server prompts you to confirm that you want to delete the app.

4. Click **Yes, delete**.
Index Workbench

The Index Workbench is a powerful tool that combines key aspects of the data indexing configuration process into one user-friendly panel. It guides the user through the workflow for configuring datasources, parsers and index pipelines, and then running a datasource job to index your data.

Datasources
To set up a new datasource in the Index Workbench, click **New**. The **Add A New Datasource** panel appears, offering three simplified methods of configuring how Fusion fetches your data:

- **By URL**
- **By local file upload**
- **By selecting an installed connector to configure**

See the Connectors Reference Guide for a complete list of available connectors. Some are installed by default, and others require separate download and installation.

**Parsers**

Parsers are a configurable component of the indexing workflow, for flexibility and specificity when parsing inbound data.
A parser consists of an ordered list of parser stages that is completely customizable. The same parser stage can be added to a given configuration multiple times if the different specified settings within those stages best suits the parsing of the data. There is no limit to the number of parser stages that can be included in a parser, and the order in which they run is also completely flexible.

In a parser, after all of the doctype-specific parser stages have run, the Tika and Fallback stages are useful catch-all stages that can attempt to parse anything that has not yet been matched. Tika is used for parsing many types of unstructured documents like PDFs, DOCX, and many more. If all of the other stages in the parser fail to completely parse the data, the Fallback stage can copy the raw bytes directly to Solr.

**Index pipelines**

An Index Pipeline transforms incoming data into a document suitable for indexing by Solr via a series of modularized operations called stages. Fusion provides a variety of specialized index stages to index data effectively. Stages can be selected, configured, and enabled or disabled in the Index Pipeline section of the Index Workbench.
Once you finish configuring a datasource using the Index Workbench, you can move on to setting up queries using the Query Workbench, which provides a similar workflow for configuring and previewing search results.

Adding a pipeline stage

Click Add a Stage to add index pipeline stages that can perform data filtering, transformations, and more.
The Solr Indexer stage is the only pipeline stage that is required to complete the indexing process, and therefore exists in every query pipeline. It is always the last stage in a series.

Re-ordering pipeline stages

The order of the pipeline stages matters, because the output from one stage becomes the input to the next stage. For example, the Solr Indexer stage must always come last in the sequence, so that data is indexed only after it has been processed by all other stages. Putting this stage first in the sequence means that subsequent stages have no effect on the indexed data.
Drag any stage in the pipeline to move it up or down in the sequence of stages. The preview panel automatically updates the results to reflect the output of the new sequence.

**Enabling and disabling pipeline stages**

By default, every stage in an index pipeline is enabled. While working with a pipeline, it can be helpful to disable stages without removing them completely. This allows you to preserve a stage's configuration while observing how the search results change in its absence. You can re-enable the stage at any time. When you save a query pipeline, the enabled/disabled state of each stage is also saved.
Click the circle next to any stage in order to enable or disable it.
Query Workbench

The Query Workbench is where you edit query pipeline stages, results parameters, and more, then preview the impacts of the changes in real time before saving them. Relevancy tools such as boosting and blocking are accessible directly from the Workbench. The Query Workbench streamlines the process of querying indexed results and fine-tuning Query Pipelines to surface the results that best satisfy the end user’s needs.

See the Query Language Cheat Sheet for help constructing queries.

With this sandbox paradigm, you can experiment with query pipelines without consequence because Fusion uses a copy of a pipeline to simulate the new results, and it is not permanently modified in the collection until you click Save.

Query pipelines and pipeline stages

Query pipelines work by processing search requests and returning an ordered list of matching documents. Each pipeline consists of a series of query stages that can be added, ordered, and configured using the Query Workbench.

Select any stage in the pipeline to open its configuration panel:
Adding a pipeline stage

Click **Add a Stage** to add query pipeline stages that can perform query setup, results relevancy, troubleshooting, and more.

The Solr Query stage is the only pipeline stage that is required for querying processes to complete, and therefore exists in every query pipeline. It is always the last stage in a series.
**Re-ordering pipeline stages**

The order of the pipeline stages matters, because the output from one stage becomes the input to the next stage. For example, the Solr Query stage must always come last in the sequence, so that data is indexed only after it has been processed by all other stages. Putting this stage first in the sequence means that subsequent stages have no effect on the indexed data.

Drag any stage in the pipeline to move it up or down in the sequence of stages. The preview panel automatically updates the search results to reflect the output of the new sequence.

**Enabling and disabling pipeline stages**

By default, every stage in a query pipeline is enabled. While working with a query pipeline, it can be helpful to disable stages without removing them completely. This allows you to preserve a stage's configuration while observing how the search results change in its absence. You can re-enable the stage at any time. When you save a query pipeline, the enabled/disabled state of each stage is also saved.
Click the circle next to any stage in order to enable or disable it.

**Boosting and blocking**

As you search your data and inspect the results, you can manipulate the rankings of individual documents. Boosting a document raises its ranking, while blocking a document removes it from search results.

Boosting and blocking affects the results for specific queries. For example, when you search for "citizen" and boost the document for "Citizen Kane", your end users will see that title boosted to the top when they also search for "citizen", but not necessarily when they search for "kane".

The Query Workbench provides convenient buttons for instantly blocking or boosting the documents that you see in the preview panel. Clicking **Boost** automatically adds the Boost Documents stage to your query pipeline, with a boost rule that matches your query and the document you clicked. Likewise, clicking **Block** adds and configures the Block Documents stage.

**How to boost a document in the Query Workbench**

1. In the preview panel, hover over the document you want to boost.
2. Click **Boost**.

The preview panel automatically updates the rankings of the search results, and tags the differences:

Notice that the Boost Documents stage is now in our pipeline. You can click this stage to view the boost rules you've added. This is also where you can remove boost rules:
How to block a document in the Query Workbench

1. In the preview panel, hover over the document you want to block.

2. Click **Block**

   The preview panel automatically updates the rankings of the search results, and tags the differences.
Notice that the Block Documents stage is now in our pipeline. You can click this stage to view the block rules you’ve added. This is also where you can remove block rules:

**Compare mode**

Compare mode allows you to compare search results using two different query pipelines, or compare search results from the original query pipeline and the copy being modified in the Query Workbench. On the right is the working pipeline, which you can edit. On the left is one of your saved pipelines.

**How to use compare mode**

1. Click **Compare**.
Another preview panel opens. In this view, you can compare results from one query pipeline side by side with another query pipeline.

2. In the left panel, select a pipeline to compare to your working pipeline.

Now you can see how the search results differ between the two pipelines:

Tip

When you click a document in one panel, the other panel automatically scrolls to the same document.

To exit compare mode, click the Close icon.

Editing parameters

The Query Workbench allows you to edit the search parameters. Click Parameters to display the current search parameters and the #dit parameters button:
Then, click **Edit parameters** to open the "Parameters and Values" window where you can add, modify, or remove search parameters:

![Parameters and Values window](image)

**Formatting the search results**

Ultimately, your search application will format the search results that end users see. The Query Workbench provides some formatting options for the preview panel.

At the bottom of the screen, click **Format Results** to configure how results are displayed in the QWB:
These options only affect how the Query Workbench displays results; they have no effect on how your search application displays them.

**Selecting the fields to display**

While the Query Fields stage configures which fields are matched against incoming queries, the search application itself determines which fields to display in the search results. In this case, the Query Workbench is our search application.

For example, the search results below use the `title_txt` field as the document name and the `_lw_data_source_s` field as the document description:

The `_lw_data_source_s` field is an internal field added by Fusion during ingest; our end users will not find it useful. To change it, select a different field from the drop-down list. In this example, we selected the `year_title` field. The preview panel automatically updates, and now we see the year of release for each movie title in our search results:
Configuring highlighting

Search results can be displayed with the search terms highlighted, by adding these Solr query parameters to the Additional Query Parameters stage of the query pipeline:

- \( \text{hl=true} \)
- \( \text{hl.fl=*} \)

By default, the Query Workbench ignores these parameters when rendering search results. To view highlighted search results in the preview panel of the Query Workbench, you must configure the parameters above and enable the Display highlighting option.

How to enable highlighting in the preview panel

1. Add the Additional Query Parameters stage to your query pipeline, or select it if it is already there.
2. Under Parameters and Values, add the following query parameters and values:

   ![hl true append]  ![true]  ![append]
   ![hl fl]  ![true append]

3. Click Apply.

   At this point, no highlighting appears in the preview panel.

4. At the bottom of the window, click Format Results.
5. Select Display highlighting.
6. Click **Save**.

Now our search results include highlighted search terms:

---

**Step-by-step Query Workbench workflow**

1. Use the Index Workbench to set up datasource(s).
2. Run a crawl on the data.
3. Open the Query Workbench.
4. Select a query pipeline to use or create a new, blank pipeline (the Default pipeline is created automatically and can be used as-is or with modification).
5. Modify the pipeline stages and set the order in which the queries will run.
6. Examine the live results in the Search panel.
7. Modify any pertinent stage settings to fine tune your results.
8. Click Save to commit the new settings to the pipeline

After a custom Query Pipeline configuration is saved, it becomes available throughout Fusion and can be re-used and modified.
**Object Explorer**

Object Explorer lets you easily find and manage Fusion objects within a specific app and across multiple apps in an entire Fusion deployment.

**Fusion apps** provide tailored search functionality to specific groups of users.

An app is a named set of linked objects, including collections, datasources, index and query pipelines, index and query profiles, parsers, and more. Using roles and security realms, you can define security on a per-app basis.

An object is something that has a specific function in a Fusion app, such as an index pipeline or Spark job. It is often something that you configure. An app is an object. Several object types are groups that group other objects. For more information about object types, see Object types.

**What you can do in Object Explorer**

In Object Explorer, you can explore objects.

<table>
<thead>
<tr>
<th>Note</th>
<th>Which objects you can explore depends on the permissions that your security realm and user definition grant you. As the user admin, you can explore all objects supported by Object Explorer.</th>
</tr>
</thead>
</table>

- **Determine which objects are in which apps** – You can determine which objects are in the current app, in any app, in no apps, and in any app or no apps. When you navigate to an object, you can display the apps that the object is in.

- **View and navigate between related objects** – In Object Explorer, related objects are connected by lines. Clicking an object moves it to the center of the Object Explorer window and displays related objects around it.

- **View related objects** – When you select an object, Object Explorer centers and displays related objects around it, connected by lines, for example:

- **Perform some management tasks** – You can perform some management tasks inside Object Explorer. For example,
you can schedule jobs and tasks from Object Explorer. For other management tasks, Object Explorer takes you to
where you need to be.

• **Add and remove objects** (*Advanced feature*) – When you create an app, it consists of objects. When you create
additional apps, Fusion can *share* some objects across apps. Shared objects are in the _shared system group, and they
have multiple dots on them, one for each app they are in.

In some cases, you might want to remove an object from an app (undo sharing) or add an object to an app. For
example, after configuring a parser in one app, you could share the parser with other apps, *without having to share
things you don’t want to share*, for example, the collection.

### Object types

You can explore these types of objects with Object Explorer:

<table>
<thead>
<tr>
<th>App</th>
<th>Spark job or task</th>
<th>Collection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datasource</td>
<td>Index pipeline</td>
<td>Parser</td>
</tr>
<tr>
<td>Query pipeline</td>
<td>A generic icon that is currently used for query profiles, index profiles, blobs, and experiments</td>
<td></td>
</tr>
<tr>
<td>Recommendations group</td>
<td></td>
<td>System group</td>
</tr>
<tr>
<td>User-defined group or Fusion-defined group that isn’t a system group</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Get started

**How to open Object Explorer**

On a PC, press Ctrl+K. On a Mac, press Command +K. Alternatively, click System > **Object Explorer**.

Object Explorer opens and displays either a list of all objects in the current app, or the list of objects that match your
most recent search in Object Explorer, whether during this Fusion session or a previous session.

### Search and browse

• **Specify the context** To restrict the scope of searching and browsing, click **In Current App**, **In Any App**, or **In No Apps**.

• **Browse all objects in the context** – Specify the context. Object Explorer displays a list of all objects. If you don’t see
a list of objects, press Enter in the **Filter** box.

• **Search for objects** – Specify the context. Type any part of an object’s name or type (the gray, uppercase text below
the name, for example, **PARSER**) in the **Filter** box. You can enter **group** to search for groups. Enter the search term in
any case. Object Explorer displays a list of matching objects.

- **Display related objects** (with a mouse) – With a list of objects present, click an object to select it. Object Explorer centers the object and displays all related objects.

- **Display related objects** (with Tab keys) – With a list of objects present, tab from object to object. With an object highlighted, press Enter to select it. Object Explorer centers the object and displays all related objects.

- **Move backward and forward through a history of results** – You can move backward and forward through a history of results, that is, the objects you’ve browsed to, searched for, and clicked on. To the right of the **Filter** box, click Back ← and Forward →, or use the left arrow ◀ and right arrow ▶ keys.

- **Determine which apps an object is in** – Click **In any App**. You’ll see dots on the objects that represent the apps that the objects are in. To view a list of the apps for an object, hover over the object, and then click the App 🌐 icon.

![Object Explorer screenshot](image)

**Manage objects**

- **Edit an object’s name** – For some objects, for example, Group objects, clicking 🖋 lets you edit the object’s name.

- **Manage an object** – For most objects that display the 🛠 icon when you hover over the object, 🛠 takes you to the part of the Fusion UI where you can manage the object.

- **Schedule a job or task** – Hover over an object that displays the Schedule 🕒 icon when you hover. Click Schedule 🕒. From the scheduling dialog, you can click **Open Schedule** to open the Scheduler.

**Group objects**

Group objects to easily browse related objects. Fusion groups some objects for you. This is an example of viewing a group, in this case, the _shared group:
There are four kinds of groups:

- **System groups** – Groups created by Fusion to group objects such as signals and items that are shared among apps. You can’t modify the membership of a system group.

- **Recommendations groups** – Groups related to recommendations. These are a subtype of system groups. You can’t modify the membership of a recommendations group.

- **User-defined groups** – Groups you create and manage yourself. You can put whatever objects you like in them.

- **Fusion-created groups that aren’t system groups** – These look and behave like user-defined groups, but you don’t create them. Fusion does. You can put objects in them.

Procedures for managing groups are:

- **View existing groups** – A group is a means of tagging objects with a shared, arbitrary identifier.

- **Create a user-defined group** – Hover over an object that you want to add to the group you’re creating, and that displays the Add To Group icon when you hover. Click Add To Group. Enter the name of a new group.

  A group name can only contain the characters A-Z, a-z, 0-9, - (hyphen), and _ (underscore).

- **Add an object to a group** – Hover over an object that displays the Add To Group icon when you hover. Click Add To Group. Choose an existing group, or enter the name of a new group. You can’t add objects to the group _shared. Its members are calculated for you.

- **Remove an object from a group** – Select the group from which you want to remove an object. Object Explorer centers the group. Hover over the object you want to remove, and then click Remove From Group.

- **Delete a user-defined group** – When you remove the last object from a user-defined group, the group remains. To delete it, hover over the group, and then click.

**Share objects among apps**

Sharing objects among apps lets you configure something once and use it multiple times where needed. When you share an object, the object and any objects that the objects depends on are shared.
Caution

Fusion makes intelligent choices regarding what is shared among apps. Exercise caution when adding and removing objects from apps.

• **Add an object to an app** – While in the Fusion workspace for the app to which you want to add an object, open Object Explorer and click In Any App. Search for or browse to the object you want to add. Hover over the object, click the App icon, and then click **Add to this app**.

![Image of adding an object to an app](image)

• **Add an object to an app directly** – In cases when an object is linked to an app, but is not linked directly to the app (it is linked via some dependency), you can add the object to an app directly.

While in the Fusion workspace for the app to which you want to add an object directly, open Object Explorer and click In Any App. Search for or browse to the object you want to add. Hover over the object, click the App icon, and then click **Add to this app directly**.

![Image of adding an object to an app directly](image)

• **Remove an object from an app** – While in the Fusion workspace for the app from which you want to remove an object, open Object Explorer and click In Any App. Search for or browse to the object you want to remove. Hover over the object, click the App icon, and then click **Remove**.

![Image of removing an object from an app](image)
Close Object Explorer

On a PC or Mac, press Esc or click Close ×. Alternatively on a Mac, you can press Command + K.
Quickstart

The Quickstart is a wizard that lets you explore some of Fusion’s core capabilities:

• Creating or selecting an app
• Selecting a collection (where data is stored)
• Indexing data
• Searching
Setup

These steps explain how to get a temporary instance of Fusion in the cloud so you can try the Quickstart right away. Follow these steps if you don’t already have a running instance of Fusion.

1. Go to Download and Try Lucidworks Fusion.

2. Click Try In The Cloud.

   The Instance Details page appears.

![Instance Details](image)

3. Scroll down and notice the Instance Links and the Fusion Credentials; you’ll need these in the next step.

4. Under Instance Links, click the link next to Fusion Admin.

   The Fusion login page appears.

5. Enter the username and password from your Fusion Credentials.

   The first time you log in, Fusion automatically opens the Quickstart wizard.
Step through the Quickstart wizard

If it isn’t already open, open the Quickstart wizard. In the Fusion launcher, click **New here? Get started**.

1. Click **Continue**.

   On the **Select an App** screen, you can select or create an app for your quickstart data.

   If you click **Create new app**, Fusion prompts you to enter an app name (and optionally a description), then click **Create App** to return to the Quickstart. Your new app is selected by default.

2. After selecting an app, click **Continue**.
On the Select a Collection screen, you can select or create a collection for your quickstart data. A collection with the same name as your app is created automatically:

3. Click Continue.

On the Index Data screen, you can either select one of the built-in sample datasets or click Use my data to upload your own:

4. Click Continue.

On the Query Data screen, you can see all search results and enter your own search queries to test the indexed dataset. You can also select the display fields or view the parsed documents:
5. Click Continue.

From here, you can do the following:

- Open the Index Workbench to modify the index pipeline.
- Open the Query Workbench to modify the query pipeline.

The workbenches are essentials tools in the Fusion workflow.
Tutorials

These tutorials guide you through some basic Fusion Server tasks with step-by-step instructions using sample datasets.
Get Started with Fusion Server

The Get Started with Fusion Server tutorial takes you from installation to a user-ready data collection in five easy parts. The tutorial acquaints you with:

- The Fusion UI
- The Index Workbench (to get data into Fusion)
- The Query Workbench (to get data out of Fusion)
- Signals and boosting (to make search results more relevant)
Get Started with Fusion Server

This tutorial takes you from installation to application-ready search data in four easy parts, using a MovieLens dataset.

- Part 1: Run Fusion and Create an App
  
  Download, install Fusion, and run Fusion, then create a Movie Search app.

- Part 2: Get Data In
  
  Use the Index Workbench to configure an index pipeline, preview the results, and get data into the Movie Search app in a format that is useful for search.

- Part 3: Get Data Out
  
  Use Query Workbench to get data out of the Movie Search app, explore the role of query pipeline stages, configure faceting, and preview search results.

- Part 4: Improve Relevancy
  
  Use signals and boosting to make search results more relevant.

Get Started with Fusion Server

1. Setup

These steps explain how to get a temporary instance of Fusion in the cloud so you can try the Quickstart right away. Follow these steps if you don’t already have a running instance of Fusion.

1. Go to Download and Try Lucidworks Fusion.
2. Click Try In The Cloud.

   The Instance Details page appears.
3. Scroll down and notice the **Instance Links** and the **Fusion Credentials**; you’ll need these in the next step.

4. Under **Instance Links**, click the link next to **Fusion Admin**.

   The Fusion login page appears.

5. Enter the username and password from your **Fusion Credentials**.

   The first time you log in, Fusion automatically opens the Quickstart wizard.

---

2. **Create an app**

Create a Movie Search app. An app is a set of Fusion objects that performs a specific searching task (such as searching for movies).

1. In the Fusion launcher, click **Create new app**.

2. In the **App Name** field, enter **Movie Search**.

3. In the **App Description** field, enter **App to search for movies**.

4. Click **Create App**.
The Movie Search app now appears in the Fusion launcher:

What’s next

You now have Fusion installed, configured, and running. You also have the MovieLens dataset from which you will use a CSV file that contains data about movies. And you have a Fusion app that you will transform into a movie search app.

In Part 2, you’ll use Index Workbench to get the MovieLens dataset into Fusion.

Additional reading

• *Fusion Server Deployment Guide*
Get Started with Fusion Server

Fusion’s Index Workbench provides the tools to configure datasources, parsers, and index pipelines. It lets you preview the results of indexing before you load your data into the actual index.

Index Workbench first sets up the necessary data extraction configuration, and then retrieves a small number of documents as sample data. You can use the sample documents to test and refine your index pipeline. All processing is simulated processing of the test data. No actual data ingestion takes place.

After you have a complete configuration, Index Workbench saves this as a Fusion datasource. To load your data into Fusion, use the Fusion Datasource tool to run the resulting configuration.

Part 2 takes you through configuring a datasource using Index Workbench. In Part 3, you’ll load the data into Fusion and view it using Query Workbench.

Before you begin

To proceed with this part of the tutorial, you must first complete Part 1, which gives you a running instance of Fusion and a Fusion app.

1. Download the MovieLens dataset

   1. Download the dataset.

   This is a MovieLens dataset created by the GroupLens research lab.


   Fusion can parse .zip files, but for simplicity we’ll index just one file from the archive (movies.csv).

The movies.csv file contains a list of 9,125 movie titles, plus a header row. Here is a truncated listing:

<table>
<thead>
<tr>
<th>movieId,title,genres</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,Toy Story (1995),Adventure</td>
</tr>
<tr>
<td>2,Jumanji (1995),Adventure</td>
</tr>
<tr>
<td>3,Grumpier Old Men (1995),Comedy</td>
</tr>
<tr>
<td>4,Waiting to Exhale (1995),Comedy</td>
</tr>
<tr>
<td>5,Father of the Bride Part II (1995),Comedy</td>
</tr>
<tr>
<td>6,Heat (1995),Action</td>
</tr>
<tr>
<td>7,Sabrina (1995),Comedy</td>
</tr>
<tr>
<td>8,Tom and Huck (1995),Adventure</td>
</tr>
<tr>
<td>9,Sudden Death (1995),Action</td>
</tr>
<tr>
<td>10,GoldenEye (1995),Action</td>
</tr>
</tbody>
</table>

2. Open the Movie Search app

   1. If the Fusion UI isn’t already open, then open it.

   2. Enter the password for the user admin, and then click Log in.
The Fusion launcher appears. You see the Movie Search that app you created in Part 1:

3. In the Fusion launcher, click the **Movie Search** app.

The Fusion workspace appears. It has controls along the left and top sides.
4. In the upper right, hover over Apps. You can see that Movie Search is the currently selected app.

Also, the user collection Movie_Search is selected in the collection picker. This is the default collection for the Movie Search app, and where Fusion will place index data.

3. **Configure the datasource**

A collection includes one or more datasources. A datasource is a configuration that manages the import, parsing, and indexing of data into a collection. You'll use Index Workbench to configure a datasource for the movie data.

1. In the collection picker, verify that the collection Movie_Search is selected.

2. Open Index Workbench. Navigate to Indexing > Index Workbench.

Initially, no data preview appears because no datasource has been configured. When you configure a datasource, Fusion samples the data and displays a preview of how it would be formatted in the index using the default parsing and index pipeline configurations.

3. In the upper right, click New.

4. Select Or, upload a file.

5. Click Choose File.

6. Navigate to the movies.csv file, select it, and then click Open.
7. Click **Add New Datasource**.

The Datasource (File Upload) configuration panel appears, with the default datasource ID `movies_csv-Movie_Search` and the default file ID `movies.csv`. These default values are fine.

8. Enter the **Description** Movies CSV file.

9. Click **Apply**.

Index Workbench reads up to 20 documents into memory from the `movies.csv` file, and then displays a preview of how they would be indexed.

You have finished configuring the datasource. At the bottom of the page, click **Cancel**.
4. Analyze the default output

1. Notice that Fusion made some assumptions about your original fields:
   - genres became genres_t (the text_general field type) and genres_s (the string field type). String fields are useful for faceting and sorting, while text fields are for full-text search. At this point, Fusion doesn't know whether you intend to use this field for faceting and sorting, for full-text search, or for both.
   - title became title_t and title_s for the same reason.
   - movieId became movieId_t and movieId_s for the same reason. This might seem odd, because the original field contains numbers. But, at this stage, Fusion creates text_general and string fields. To use the contents of this field as an integer, you would map the field to an integer field.

You also see fields that begin with _lw. These fields contain data that Fusion creates for its own housekeeping. You can ignore them.

These fields are created by the Solr Dynamic Field Mapping stage in the default index pipeline. This stage attempts to automatically detect field types, and renames fields accordingly. For this tutorial, you'll manually configure the fields instead.

2. Turn off the Solr Dynamic Field Mapping stage by clicking the green circle next to it.

+ Your data's original fields reappear: genres, movieId, and title.
5. Configure the index pipeline

First you’ll configure the field mappings in the index pipeline so each field has the correct data type. Then you’ll split the genres field into multiple values so each value can be used as a facet in Part 3 of this tutorial.

5.1. Configure field mappings

Configure field mappings to control the field types of Fusion documents. Fusion uses field name suffixes to determine field types. When a field name has no suffix, Fusion stores it as a string field and treat’s it as an unanalyzed whole. For precise analysis and search, most fields need suffixes to indicate their specific types. You’ll see how this relates to the fields in the dataset.

1. In the list of index pipeline stages on the left, click Field Mapping to open the Field Mapping stage configuration panel.
2. Click Add + to create a new field mapping rule.
3. Under Source Field, enter genres.
4. Under Target Field, enter genres_ss.

The field suffix _ss means that this field is a multi-valued string field.

<table>
<thead>
<tr>
<th>Note</th>
<th>Fusion currently interprets this field as having a single value. You can see that the field actually contains a pipe-delimited array of values. You’ll fix this after you finish configuring field mappings.</th>
</tr>
</thead>
</table>

5. Under Operation, select move.

The move operation means that the resulting document no longer has a genres field; it only has genres_ss.
6. Click **Apply**.

Applying the new configuration re-runs the simulation and updates the contents of the preview panel. Notice the change in the field name from **genres** to **genres_ss**:
7. Click Add [+] to add more field mapping rules as follows:

- The **movieId** field is a unique document identifier. It should be *copied* into the document's **id** field.

- The **title** should be searchable as a text field, so you *move* it to the field **title_txt**.

Your field mappings should look like this:
### Field Mappings

List of mapping rules

<table>
<thead>
<tr>
<th>Source Field</th>
<th>Target Field</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>genres</td>
<td>genres_ss</td>
<td>move</td>
</tr>
<tr>
<td>movield</td>
<td>id</td>
<td>copy</td>
</tr>
<tr>
<td>title</td>
<td>title_txt</td>
<td>move</td>
</tr>
</tbody>
</table>

8. Click **Apply**.

* After you have specified these explicit field mapping rules, you can browse the resulting documents in the preview panel to check your work.

*
1. In the upper right, click **Save**. This saves your modified index pipeline. Get in the habit of saving your work as you work.

Now your document ID is more useful, and your movie titles are full-text searchable.
Tip

Because the input documents in this tutorial are simple documents with a fixed number of known fields, it’s easy to configure the Field Mapping stage to ensure the correct document structure for Fusion. When documents have large numbers of fields, the Solr Dynamic Field Mapping stage can reduce the work required to configure the index pipeline.

5.2. Split a multi-value field

The genres_ss field has been parsed as a single-value field, but you can see that it’s really a pipe-delimited array of values. To split this field into its constituent values, you’ll add a Regex Field Extraction stage to your index pipeline. This stage uses regular expressions to extract data from specific fields. It can append or overwrite existing fields with the extracted data, or use the data to populate new fields.

1. Click Add a stage.
2. Scroll down and select Regex Field Extraction.

The Regex Field Extraction stage configuration panel appears.

3. Under Regex Rules, click Add [+].
4. On the new line, hover over the […] under Source Fields, and then click Edit [Edit].

The Source Fields window opens.

5. Click Add [+].
6. Enter genres_ss, and then click Apply.
7. Under Target Field, enter genres_ss.
8. In the Write Mode field, select overwrite.
9. In the Regex Pattern field, enter this expression:
10. In the **Return If No Match** field, select `input_string`.

11. Click **Apply**.

Initially, your data doesn't change.

12. In the list of index pipeline stages, drag the **Regex Field Extraction** stage down so that it comes after the Field Mapping stage:

Now the preview shows multiple values for the `genres_ss` field:
Before

Simulated Results [5]

```json
id: "1"
▶ context: {"solrParams":{"wait":"30"},"parserId":"_lw_tmp_15355_"}
▶ fields: [24]
  ▶ genres_ss: Adventure|Animation|Children|Comedy|Fantasy
  ▶ movield: 1
  ▶ title_txt: Toy Story (1995)
  ▶ _lw_data_source_s: movies_csv-Movie_Search_samplelw_tmp
  ▶ _lw_parser_line_number_l: ["java.lang.Long",2]
  ▶ _lw_parser_id_s: coco7e0c9-4424-40d1-aa8b-a2c95e0bd77b
  ▶ _lw_file_size: ["java.lang.Long",458390]
  ▶ _lw_parser_type_s: csv
```

Tip

If the preview panel doesn't update automatically, select a different number of documents to view. This forces the preview to update.

1. To view the values of the genres_ss field, expand it and values under it by clicking the ▶:
These field values are useful for faceting, which you'll explore in Part 3 of this tutorial.

2. In the upper right, click **Save**. This saves your modified index pipeline.

### 5.3. Create a new field from part of an existing one

Notice that the `title_txt` field also contains the year in which the movie was released. Instead of including the year in your full-text search field, it would be more useful as a separate field that you can use for faceting. This is another job for the Regex Field Extraction stage.

1. In the list of index pipeline stages, click **Regex Field Extraction**.
2. In the Regex Field Extraction configuration panel, under Regex Rules, click **Add**.
3. On the new line, hover over the `[...]` under **Source Fields**, and then click **Edit**.

   The Source Fields window appears.

4. Click **Add**.
5. Enter `title_txt`, and then click **Apply**.
6. Under Target Field, enter `year_i`.

   The `_i` suffix indicates an integer point field (specifically, that the field is a dynamic field with a `pint` field type). Fusion will create this new field whenever the regular expression matches the contents of the source field.
Tip

When you use the Regex Field Extraction stage to create a new field, the value of Write Mode makes no difference.

7. In the **Regex Pattern** field, enter this expression to match the digits inside the parentheses at the end of the `title_txt` value:

\(((0-9]+)\)$

8. In the **Regex Capture Group** field, enter 1. This lets the index pipeline stage transfer the year into the `year_i` field.

<table>
<thead>
<tr>
<th>Tip</th>
<th>Scroll all the way to the right to see this field.</th>
</tr>
</thead>
</table>

9. Click **Apply**.

Now the preview includes the new `year_i` field:
10. In the upper right, click **Save**. This saves your modified index pipeline.

### 5.4. Trim a field’s value

The **title_txt** field still includes the year of the film’s release, which you’ve extracted into its own field, **year_i**. Let’s trim that information from the **title_txt** values so that only the title text remains.

1. In the list of index pipeline stages, click **Regex Field Extraction**.
2. In the Regex Field Extraction configuration panel, under Regex Rules, click Add [+].
3. On the new line, hover over **Source Fields**, and then click Edit [✓].
The Source Fields window appears.

4. Click Add [+].

5. Enter title_txt, and then click Apply.

6. Under Target Field, enter title_txt.

7. In the Write Mode field, select overwrite.

8. In the Regex Pattern field, enter this expression to match the digits inside the parentheses at the end of the title_txt value:

   ^(.+)\s\(([0-9]+)\)$

9. In the Regex Capture Group field, enter 1.

10. Click Apply.

    Now the preview pane shows the title_txt field with only the title string:
11. In the upper right, click **Save**. This saves your modified index pipeline.

### 6. Run the datasource job

Now you have a correctly-configured index pipeline appropriate to your data. You're ready to index the data.

1. In the upper left, click **Start job**.
This launches a datasource job that imports and indexes the complete contents of your movies.csv file, using the configuration you just saved.

Your datasource job is finished when the Index Workbench displays Status: success in the upper left.

### 7. Reindexing

Documents are associated with a collection through the name of the datasource. The datasource name is stored as a value in the _lw_data_source_s field. For various reasons, you may wish to remove all documents associated with a datasource from a collection before using CrawlDB to add relevant documents back to the collection. This process is known as **reindexing**.

To accomplish this, navigate to **Indexing** > **Datasources**, select the datasource name, and then click **Clear Datasource**. This will remove all documents with the selected datasource name in the _lw_data_source_s field. After the documents are removed from the collection, you can repeat the steps above to reindex the data.

**Warning**

Use caution when creating or changing a datasource name. If you create a new datasource name that is identical to an existing datasource name, all document associations will be shared between the datasource names. Changing the name to match an existing name will have the same result.

### 8. Close panels you no longer need open

Fusion opens panels beside already open panels. Close all of the panels that are open by clicking Close X.
What’s next

Now you have 9,125 movie listings from the MovieLens database in Fusion’s index, customized to indicate the data type for each field. You also split a multi-valued field so that its values can be treated individually, created a new field to contain partial contents of a different field, and trimmed that content from the original field.

Let’s compare the initial indexing of your data with the indexing after field mappings and extractions:

<table>
<thead>
<tr>
<th>Before</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="before.png" alt="Simulated Results" /></td>
<td><img src="after.png" alt="Simulated Results" /></td>
</tr>
</tbody>
</table>

In Part 3, you’ll use Query Workbench to get search results from your collection and configure the query pipeline that customizes those results. You’ll add faceting using the `genres_ss` and `year_i` fields so that users can easily filter their search results.
Additional reading

- Collections
- Index Workbench
- File Upload Connector and Datasource Configuration
- Field Mapping Index Stage
- Solr Dynamic Field Mapping Index Stage

Get Started with Fusion Server

In Part 2, you used Index Workbench to get data into Fusion by previewing the dataset, configuring the index pipeline, and then indexing the data.

In Part 3 (this part), you’ll explore Query Workbench and learn how to configure Fusion’s output (search results), including faceting. Facets are the ubiquitous, dynamic lists of categories or features offered as filters within a search results page.

Before you begin

To proceed with this part of the tutorial, you must first complete Part 1 and Part 2, which give you an indexed dataset for Query Workbench to read.

The dataset has three fields that users of your search application might find relevant:

- **genres_ss** - A list of one or more genre labels
- **title_txt** - The name of the movie
- **year_i** - The movie’s year of release

The field suffixes indicate the type of data stored in each field:

- Fields with the suffix **_ss** (multi-valued string fields) contain one or more strings values.
  
  String fields require an exact match between the query string and the string value stored in that field.

- Fields with the suffix **_txt** (text fields) contain text.
  
  Text fields allow for free text search over the field contents. For example, because the movie titles are stored in a text field, a search on the word "Star" will match movies titled "Star", "A Star is Born", all movies in the Star Wars and Star Trek franchises, as well as "Dark Star", "Lone Star", and "Star Kid".

- Fields with the suffix **_i** (point integer fields) contain integer values.
  
  Numeric fields allow range matches as well as exact matches, and point integer fields allow efficient comparisons between the field’s values and the search criteria.

The different field types allow for different kinds of searches. Configuration of the query pipeline determines how fields are searched.
1. **Explore search results**

Query Workbench lets you interactively configure a query pipeline while previewing the search results it produces. A query pipeline converts a free text query submitted to your search application into a structured query for Solr.

1. Log in to Fusion.
2. Click the **Movie Search** app.

   The Fusion workspace appears.

3. Open Query Workbench. Navigate to **Querying > Query Workbench**.

![Query Workbench interface](image)

1.1. **Perform a search**

The default search is the wildcard search (\_), which returns all documents in the collection. You’ll enter a different search query to get started with facet configuration.

1. In the search box, enter the query string **star**, and then press Enter or click Search.

   This search returns all movies that have the word “star” in the title.
The output is configured by the default query pipeline, which has the same name as the collection (in this case, movie-search).

To see more of the default output, you can perform other searches if you like.

2. In the search box, enter the query string \texttt{to return all documents}, and then press Enter or click Search.

Tip

If you wish to refine your results, you can change your index pipeline configuration and reindex your data.

1.2. Explore the role of query pipeline stages

A default query pipeline consists of the stages below. Some of these support Fusion AI features for advanced relevancy tuning, which you can read about separately. In this tutorial, we'll configure a couple of the basic stages.

- **Text Tagger** - Look for known phrases, synonyms, misspellings, and so on, that can be used to improve the query with query rewriting.

- **Boost with Signals** - Use signals data to boost relevant documents using collaborative filtering.

- **Query Fields** - Specify the set of fields over which to search.

  We'll configure this stage later in this tutorial.

- **Field Facet** - Specify the fields to use for faceting.

  We'll configure this stage, too, for basic faceting and range faceting.

- **Apply Rules** - Modify the query using business rules, if any are triggered.

- **Solr Query** - Perform the query and return the results.

  This is the only stage that is always required to perform a query and receive results.

- **Modify Response with Rules** - Modify Solr's response using business rules, if any are triggered.
1. Turn off the Solr Query stage.

All search results disappear from the preview pane because Fusion doesn’t send a query to Solr.

2. Turn on the Solr Query stage and turn all other stages off.

Now the search results look much like they did before. At this point, the disabled stages do not affect the output because they are not yet configured.

3. Turn all stages on.
2. Configure faceting

Facets are the ubiquitous, dynamic lists of categories or features offered as filters within a search results page. Facets provide a simple way for users to explore and filter their search results without having to construct complicated queries. You configure facets as a part of configuring a query pipeline.

The data you indexed in Part 1 has two fields that are natural choices for faceting: `genres_ss` and `year_i`. For example, a user could search for science fiction of the 1950s in just a few clicks.

2.1. Configure basic faceting for genres

The `genres_ss` field is ready for faceting as-is.

1. Click **Add a field facet** and select the `genres_ss` field.
2. Click Sci-Fi to select movies that have the value Sci-Fi for genres_ss:

2.2. Configure range faceting for years

If you were to just configure faceting for the year_i field as you did above for the genres_ss field, you would get one
facet per year, which isn’t very useful.

But the `year_i` field will be more usable if you configure range faceting. Range faceting is a way of grouping values together so that the user can select a value range instead of one specific value. For example, range facets are commonly used with pricing ($50-$100) or ratings (4 stars or higher). Here, we group years by decade.

Range faceting requires sending an additional query parameter to Fusion’s Solr core. You can configure this with the Additional Query Parameters stage. In this case, you’ll use several of Solr’s range facet query parameters.

Use the Additional Query Parameters stage to configure range faceting for the `year_i` field:

1. Click Add a stage.
2. Scroll down and select Additional Query Parameters.

![Additional Query Parameters stage](image)

The Additional Query Parameters configuration panel appears.

3. Under Parameters and Values, add the following parameter names and values:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Parameter Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>facet.range</td>
<td><code>year_i</code></td>
</tr>
<tr>
<td>facet.range.start</td>
<td>1900</td>
</tr>
<tr>
<td>facet.range.end</td>
<td>2020</td>
</tr>
<tr>
<td>facet.range.gap</td>
<td>10</td>
</tr>
<tr>
<td>facet.range.include</td>
<td><code>outer</code></td>
</tr>
</tbody>
</table>

In this case, you do not need to modify the Update Policy field; the default value of append is fine.

4. Click Apply, and then Cancel (which just closes the configuration panel for the Additional Query Parameters stage).

The year facets are now grouped by decade:
Tip

Facets are not the only way for users to find items by year. In your user application, you can let users search for specific values in the `year_i` field, for example, by using a text field or dropdown list.

3. Configure query fields

In this section, you’ll see why it’s useful to specify which fields Fusion should use to match a query.

1. Search for “2001”.

The results are not what an end user might expect:
"2001: A Space Odyssey" is not the top search result.

2. Under "Lethal Weapon 2", click **show fields**.

Here's the reason: your search query matches the **id** field, but users don’t care about this field.

You'll use the Query Fields stage to specify the fields that users really care about.

3. Under "Lethal Weapon 2", click **hide fields**.

4. Click the **Query Fields** stage of the query pipeline.

The Query Fields configuration panel appears.

5. Under **Search Fields**, click **Add**.

6. Enter **title_txt**.

7. Click **Add** again.

8. Enter **year_i**.

9. Click **Apply**, and then **Cancel** (which just closes the configuration panel for the Query Fields stage).

Now "2001: A Space Odyssey" rises to the top of your search results, followed by films made in the year 2001:
4. Save the query pipeline configuration

1. In the upper right, click **Save**.

   The **Save Pipeline** window appears. By default, you'll overwrite the default query pipeline for this datasource.

2. Click **Save pipeline**.

**What’s next**

With just two facet fields combined with keyword search, this prototype is already beginning to feel like a real search application.

In Part 4, you'll enable signals, generate some signal data, aggregate it, and search it to see what it looks like. Signals can be used for recommendations or boosting.

**Additional reading**

- Query Workbench
- Query Pipelines
- Field Facet query stage
- Additional Query Parameters query stage
- Query Fields query stage

**Get Started with Fusion Server**

Signals are events that can be aggregated and used for automatic boosting or recommendations, which are ways of making search results more relevant.

As an example of boosting, the most popular search results for certain queries can be boosted so that they appear first.
(or at least nearer the top) when other users make similar queries.

Similarly for recommendations, click events or purchase events can be collected as signals and used to display "Customers who viewed this also viewed" or "Best-selling holiday items".

In a production environment, users' actions generate signals. For the purposes of this tutorial, you'll use Query Workbench to generate click signals.

**Before you begin**

To proceed with this part of the tutorial, you must first complete Part 1, Part 2, and Part 3, which give you an indexed dataset that's configured for faceted search.

If the Fusion UI isn't already open, then open it.

1. In a browser window, open `localhost:8764`.
2. Enter the password for the user `admin`, and then click **Login**.

   The Fusion launcher appears.

3. Click the **Movie Search** app.

   The Fusion workspace appears.

1. **Format display fields**

To help you understand the continuity in the next steps, first make sure that some relevant fields are displayed.

1. Open Query Workbench. Navigate to Querying > **Query Workbench**.
2. At the top right of the page, select **Display Fields**.
3. In the **Name** field, click the row to reveal a dropdown of possible values.
4. Select `title_txt`. You can filter the list of possible values to help you.
5. In the **Description** field, select `id`.
6. Click **Display Fields** to close the Display Fields window.
2. Enable synthetic signals

You need some signal data beyond the few signals you generated while completing Part 3. Because this is a prototype app, users aren’t generating signals. Instead, you’ll enable synthetic signals in Query Workbench.

1. At the bottom of the Query Workbench page, click **Format Results**.

2. Select **Show signal generators** and **Send click signals**.

   Control the number of results displayed, if highlighting information is utilized, and signals.

   **Set results per page**

   **Display highlighting?**

   **Signals**

   Caution: Sending click signals will modify the signals collection, potentially affecting real search results.

   **Show signal generators**

   **Send click signals**

3. Click **Save**.
4. Hover over one of the search results.

Now when you hover over a search result, Query Workbench displays controls that include a **Simulate** button next to a field that lets you specify the number of signals to simulate:

3. **Generate signals**

With synthetic signals enabled, you’ll generate a simple set of signal data that you can use to generate meaningful recommendations.

For this tutorial, you’ll generate signals that you can use to boost your favorite sci-fi titles so that they appear first.

1. Search for **star wars**.

The top results are not your favorite titles:

   ![Image of search results]

Next you’ll generate signals that you can use to boost certain titles. Signals are tied to the search query, so your boosted titles will appear first in the search results only when users search for **star wars**.

2. Hover over “Star Wars: Episode IV - A New Hope”.
3. Set the number of signals to 4000 and click **Simulate**.
4. Hover over "Star Wars: Episode V - The Empire Strikes Back".
5. Set the number of signals to 3000 and click **Simulate**.
6. Hover over "Star Wars: Episode VI - Return of the Jedi".
7. Set the number of signals to 2000 and click **Simulate**.

   **Tip**

   With synthetic signals enabled, you can also send a single signal by clicking the underlined movie title.

8. In the upper right, click **Save**.

   The **Save Pipeline** window appears. By default, you’ll overwrite the existing query pipeline for the selected collection (in this case, **Movie_Search_signals**).

9. Click **Save pipeline**.

### 4. Explore the raw signals

Whenever you create a collection, two corresponding collections are also created automatically: `<collection>_signals` for raw signals and `<collection>_signals_aggr` for aggregated signals (in this case, **Movie_Search_signals** and **Movie_Search_signals_aggr**). Just as you did with your primary collection, you can use Query Workbench to explore the data in the _signals collection.

1. In the collection picker in the upper left, select **Movie_Search_signals**.

2. Open Query Workbench. Navigate to Querying > **Query Workbench**.

   Your signal data appears.

   ![Query Workbench](image)

   Your signals collection contains several types of signals, such as click signals and search result signals. Let’s investigate the click signals.

3. Search for **type:click**.

4. For any of the results, click **show fields**.

   - The **count_i** field shows the number of click signals you generated for this event. For example, for 260 (the **doc_id_s** for Star Wars: Episode IV - A New Hope), **count_i** equals 4000.
5. Click **hide fields**.

### 4.1. Optional: Format signals fields

You can configure your search view to display fields that are meaningful for your investigation. For example, you can display the document ID and the number of click signals.

1. Click **Display Fields**.
2. For the **Name** field, select `doc_id_s`.
3. For the **Description** field, select `count_i`.
5. Explore the aggregated signal data

Aggregation jobs are enabled automatically. Aggregation jobs for click signals run every fifteen minutes by default to aggregate any new raw signals. Aggregations jobs for other types of signals run at other frequencies by default.

Check whether the aggregated data has arrived in the Movie_Search_signals_aggr collection.

1. In the collections picker in the upper left, select Movie_Search_signals_aggr.
2. Open Query Workbench. Navigate to Querying > Query Workbench.

Your aggregated signal data should appear. If not, wait a minute and then reload your browser, or click Search in Query Workbench.

3. Click Display Fields.
4. For the Name field, select doc_id_s.
5. For the Description field, select aggr_count_i.
6. For the result 1210, click show fields.
The fields for aggregated signals are very similar to the fields for raw signals, with additional fields to describe the aggregation:

- `aggr_count_i` – Number of signals that have been aggregated (in this case, 3000)
- `aggr_id_s` – Name of the aggregation job
- `aggr_job_id_s` – Job ID
- `aggr_type_s` – Aggregation type (in this case, sql)

7. In the upper right, click **Save**.

   The **Save Pipeline** window appears. By default, you'll overwrite the existing query pipeline for the selected collection (in this case, **Movie_Search**).

8. Click **Save pipeline**.

### 6. View the search results with and without default boosting

1. In the collections picker in the upper left, select **Movie_Search**.
2. Open Query Workbench. Navigate to Querying > Query Workbench.
3. Search for **star wars**.

   Now, "Star Wars: Episode IV - A New Hope" is the first search result, followed by Episode V and then VI. These search results are automatically boosted by the default configuration of the Boost with Signals query pipeline stage, which boosts on the `id` field.
Note: The Boost with Signals stage requires a Fusion AI license. Your Fusion Server trial license enables Fusion AI features.

4. Click **Compare**. Another preview panel opens. Now the working pipeline is on the right and a static snapshot of that same pipeline is on the left.

In this view, you can compare results from one query pipeline side by side with another query pipeline. In this case, you’ll compare results for the same pipeline (**Movie_Search**) with and without the Boost with Signals stage enabled.

5. Turn off the Boost with Signals stage.
Now the search results on the right appear as they did before you generated synthetic click signals. Rank-change indicators indicate which results moved up or down as a result of turning off boosting.

6. Turn on the Boost with Signals stage again to restore the boosted results.

7. Close the comparison preview panel by clicking the close icon.
Resources and Support

In addition to this documentation, these resources are available to assist you.
Self-help resources

• **Blog**
  
  Learn from the experts in these in-depth topics.

• **Videos and webinars**
  
  View demos, webinars, and tutorials.

• **Answers**
  
  Browse questions and answers from the Lucidworks community, or ask a new question.

• **Knowledge base**
  
  Find solutions in these articles from the Lucidworks support team.

• **Downloads***
  
  Download Fusion products.

• **Download Fusion connectors***
  
  Download Fusion connectors.

* Customer login required.
Technical support

When you need personalized, expert assistance, the Lucidworks technical support team is available 24/7.

Before you open a support case, see the instructions below to learn how to gather data that the technical support team can use to evaluate your issue and provide a solution.

Once you have obtained the relevant data from Fusion, you can open a support case here.

Exporting log files from the DevOps Center

In Fusion 4.2 and above, the DevOps Center provides an easy way to export logs for the timeframe that pertains to your issue.

How to export data from the DevOps Center

1. In the Fusion UI, navigate to System > DevOps Center > Log Viewer.
2. Select the time period that is relevant to the incident you are interested in.

   Log files will be exported according to the tab you are viewing.
Click **Export service logs** in the Service Logs tab or **Export request logs** in the Request Logs tab to download a .csv file containing the logs (shown here with example timestamps):


### Finding log files in the filesystem

Log files are located in the `fusion/5.0.x/var/log` directory. Generally, the log files below are the most useful in diagnosing an issue:

- API logs

  `fusion/5.0.x/var/log/api/api.log` and `fusion/5.0.x/var/log/api/gc_<timestamp>.log.0.current`

- Connectors logs

  `fusion/5.0.x/var/log/connectors-classic/connectors-classic.log` and `fusion/5.0.x/var/log/connectors-rpc/connectors-rpc.log`
If it's apparent that additional components are involved, collect their logs and associated gc logs from the fusion/5.0.x/var/log directory.

### Opening a support case

Lucidworks provides technical support to customers 24 hours a day, 7 days a week, with unlimited incidents. Once you have obtained the relevant data from Fusion as described above, you can open a support case.

For the most efficient response, follow the instructions below.

1. Go to [https://support.lucidworks.com/s/contactsupport](https://support.lucidworks.com/s/contactsupport).

   You can also navigate to **Contact Support** from any page on the Lucidworks support site:

2. Enter a subject and description of your issue.

3. In the **Severity** field, select the appropriate severity value for your issue according to the guidelines below.

<table>
<thead>
<tr>
<th>Severity</th>
<th>Time</th>
<th>Definition</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>1 Hour</td>
<td>Severe commercial impact on Customer’s business which either makes the system inoperable or prevents content discovery.</td>
<td>Production search is down.</td>
</tr>
<tr>
<td>S2</td>
<td>12 Hours</td>
<td>Significant commercial impact on Customer’s business which makes one or more critical areas of functionality inoperable.</td>
<td>Search performance has slowed down to a crawl.</td>
</tr>
<tr>
<td>S3</td>
<td>24 Hours</td>
<td>Moderate commercial impact on Customer’s business which makes one or more non-critical areas of functionality inoperable.</td>
<td>Slight increase in query time.</td>
</tr>
<tr>
<td>S4</td>
<td>48 Hours</td>
<td>No commercial impact on Customer’s business.</td>
<td>How to implement a certain search feature.</td>
</tr>
</tbody>
</table>
4. Select your **Platform**, **Product Line**, and **Solr Version**.

5. Select either your **Lucidworks Search Version** (legacy customers only) or your **Fusion Version** (3.0 and up are currently supported).

6. Select your **Environment**, one of the following:
   - Production
   - QA
   - Sandbox
   - Proof of Concept

7. Click **Add Attachment** and select your log files.

| Tip | You can attach only one file. Upload all of your files as one .tar or .zip archive. |

8. When you have attached all pertinent data, click **Submit**.
You’ll receive an email message containing your case number. A Lucidworks support engineer will contact you within the time period corresponding to your issue’s severity level.