



IBM Cloud
Discovery v2

Product guide

Edition notices

This PDF was created on 2023-09-29 as a supplement to *Discovery v2* in the IBM Cloud docs. It might not be a complete set of information or the latest version. For the latest information, see the IBM Cloud documentation at <https://cloud.ibm.com/docs/discovery-data>.

© IBM Corp. 2023

Getting started with Watson Discovery

In this tutorial, we introduce IBM Watson® Discovery and walk you through the Discovery sample project. Exploring the sample project is a great way to tour and try out some of the product's features.

Before you begin

Choose the appropriate step to complete for your deployment:

- IBM Cloud Pak for Data Install Discovery. See [Installing Discovery for Cloud Pak for Data](#).
- IBM Cloud Complete the following steps:
 1. Sign up for a IBM Cloud account or log in.
 2. You can use a Plus plan for 30 days at no cost. However, to create a Plus plan instance of the service, you must have a paid account.

For more information about creating a paid account, see [Upgrading your account](#).

 **Important:** If you decide to discontinue use of the Plus plan and don't want to pay for it, delete the service instance before the 30-day trial period ends.

3. Go to the [Discovery resource](#) page in the IBM Cloud catalog and create a Plus plan service instance.

Step 1: Open Watson Discovery

IBM Cloud

These instructions apply to all managed deployments, including IBM Cloud Pak for Data as a Service instances.

1. Click the Discovery instance that you created to go to the service dashboard.
2. On the **Manage** page, click **Launch Watson Discovery**.

If you're prompted to log in, provide your IBM Cloud credentials.

IBM Cloud Pak for Data

These instructions apply to Discovery deployments:

1. From the IBM Cloud Pak for Data web client main menu, expand **Services**, and then click **Instances**.
2. Find your instance, and then click it to open its summary page.



Note: You can create a maximum of 10 instances per deployment. After you reach the maximum number, the **New instance** button is not displayed in IBM Cloud Pak for Data.

3. Click **Launch tool**.

Step 2: Open the sample project

A new browser tab or window opens and the **My Projects** page is displayed.

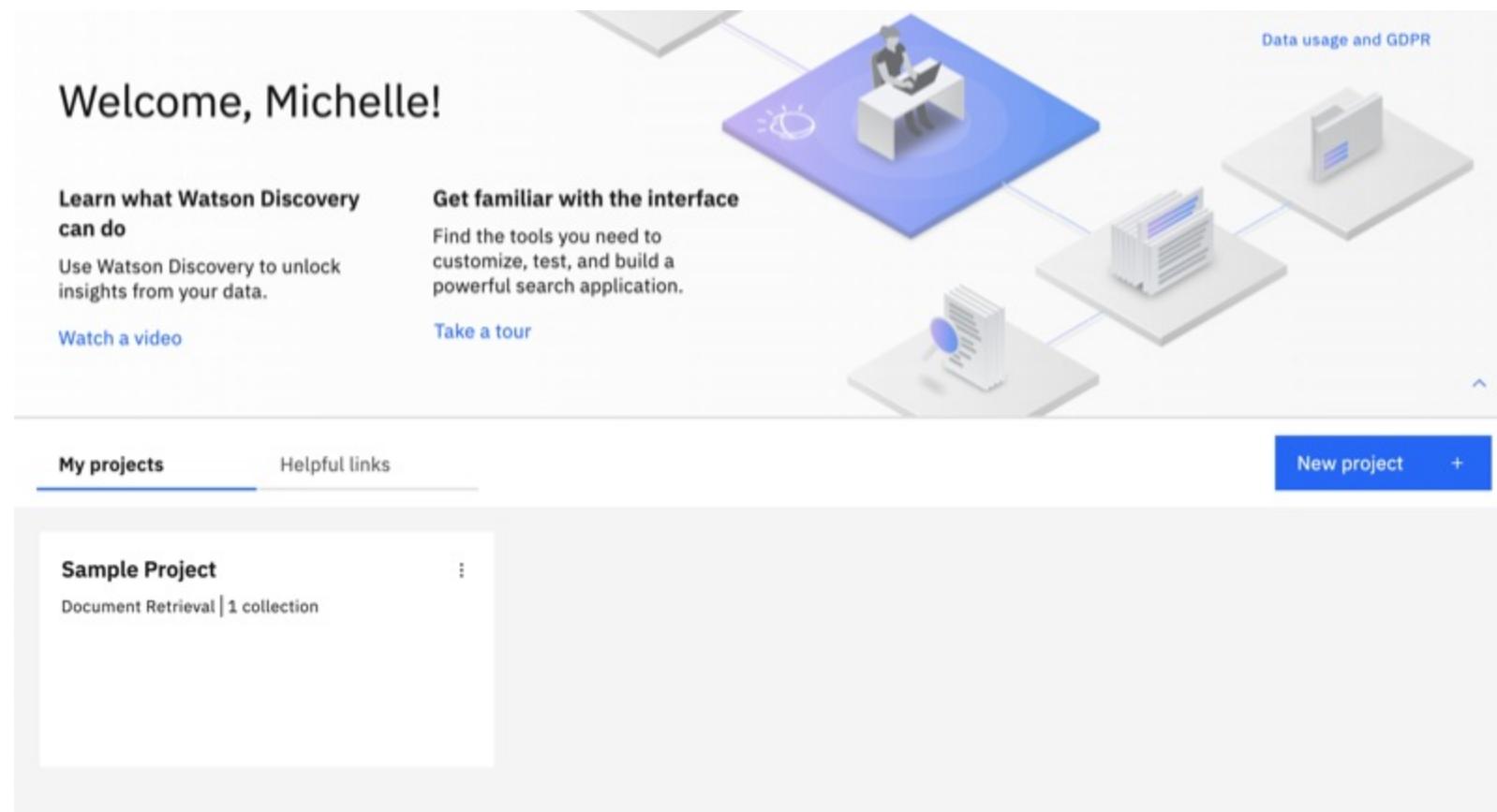


Figure 1. My projects page of the Sample project

Tip: To get familiar with the product, you can watch an under 3-minute overview video by clicking the [Watch a video](#) link from the product home page.

In this tutorial, you explore the sample project.

The sample project is a built-in project that is provided as a resource for you to initially explore the product. The sample project is a **Document Retrieval** project type. Document Retrieval projects are used to search and find the most relevant answers from your data.

1. Click **Sample Project**.

The **Improve and customize** page is displayed.

Note: If you just installed Discovery, the Sample Project needs time to finish processing documents. Wait for processing to finish before you start experimenting. You can check the status of data processing from the **Activity** page, which is described in the next step.

Figure 2. Sample project Improve and customize page

Step 3: Learn about the sample collection

Learn about ways you can manage and enhance a collection by exploring the sample collection that is available with the sample

project. The sample collection consists of a set of uploaded IBM Support PDF documents.

1. Click the **Manage collections** icon on the navigation panel.

Any collections in your project are displayed here. This project has only one collection.

The screenshot shows the 'Manage collections' page for a 'Sample Project'. On the left, there's a sidebar with icons for 'Sample Project', 'Manage collections', and 'Sample Collection'. The main area displays a card for 'Sample Collection' showing '40 documents' and 'Last updated: 9/20/2021'. In the top right, there's a blue button labeled 'New collection' with a '+' sign.

Figure 3. Collections page in the Sample project

2. Click **Sample Collection**.

The **Activity** page is displayed. This page shows the status of the collection. For example, it shows the total number of documents and when it was last updated. If Discovery encounters a problem when a document is uploaded or a data source is crawled, any associated messages are displayed here.

The screenshot shows the 'Sample Collection' activity page. At the top, there are tabs: 'Activity' (which is selected), 'Manage data', 'Identify fields', 'Manage fields', 'Enrichments', 'Processing settings', and 'CSV settings'. Below the tabs, it says 'Collection last updated: 21/08/2023, 16:36:48 GMT+5:30'. It shows '40 Documents available' and '0 Warnings / errors'. A section titled 'Warnings and errors at a glance' has a link 'View all'. To the right, there are two boxes: 'Upload data' (with a placeholder 'Add data by uploading more to this collection') and 'Try it out' (with a placeholder 'When your collection is ready, go to this page'). At the bottom, there's a note: 'Nothing to report' with a sun icon, followed by 'When there are warnings or errors, they'll appear here.'

Figure 4. Activities page in the Sample project

After you create a collection, you can come to this page to find information about the processing status of the data in the collection.

3. Click the **Enrichments** tab.

The **Enrichments** page shows you a list of available enrichments. Enrichments make meaningful information easier to find and return in searches. You can apply built-in enrichments to your collection to leverage powerful Natural Language Understanding models that tag terms, such as commonly known keywords.

Name	Fields to enrich	Type	Status
ml_en_wks_mah	Selected fields	Machine learning	Ready
dict_fq8	Selected fields	Dictionary	Ready
jpn_dict	Selected fields	Dictionary	Ready
adada	Selected fields	Dictionary	Ready
mah_4digits	Selected fields	Regular expression	Ready
Family Members	Selected fields	Entity extractor	Ready
mah_ssn	Selected fields	Regular expression	Ready
2023.08.23.23.04.32-new_format_4_cat_multi_train.csv	Selected fields	Sentence classifier	Ready
2023.08.23.23.04.26-new_format_4_cat_multi_train.csv	Selected fields	Sentence classifier	Ready
Sentiment of Document	Selected fields	System	Ready
2023.08.23.23.04.28-new_format_4_cat_multi_train.csv	Selected fields	Sentence classifier	Ready
Entities v2	1 x Selected fields	System	Ready

Figure 5. Enrichments page of the Sample project

The following enrichments are applied to the sample collection:

Entities

Recognizes proper nouns such as people, cities, and organizations that are mentioned in the content.

Part of Speech

Identifies the parts of speech (nouns and verbs, for example) in the content.

These enrichments are applied automatically to collections that are added to projects of the **Document Retrieval** type.

- For the **Entities v2** enrichment, click **1x Selected fields**.

A list of available fields is displayed and the **text** field is selected. This selection means that the **Entities** enrichment was applied to content that was indexed and added to a field named **text** when documents from the collection were processed.

Figure 6. Entities enrichment being applied to the text field

- For the **Part of Speech** enrichment, click **1x Selected fields**.

Again, you can see that the enrichment is applied to the **text** field.

From this page, you can apply new enrichments to your collection or change the fields where an enrichment is applied.

A powerful feature of Discovery is that you can add your own custom enrichments, such as dictionaries, patterns, and machine learning models. When you create custom enrichments, they are listed on this page also. You can manage where they are used from here.



Tip: For more information about custom enrichments, see [Adding domain-specific resources](#).

6. You are going to apply another enrichment to the collection. Find the **Keywords** enrichment in the list, and then click **Select fields**.

The Keywords enrichment recognizes significant commonly-known terms in your content.

7. Scroll through the list of fields until you find the **text** field, and select it.

A screenshot of a user interface showing a list of fields under the 'Selected fields' section. The list includes: question, subtitle, table, table_of_contents, and text. The 'text' field is highlighted with a blue border, indicating it is selected. The background shows other enrichment options like 'Keywords' and 'Entities v2'.

Figure 7. Fields to which you can apply the Keywords enrichment

8. Click **Apply changes and reprocess**.

While your documents are being reprocessed to look for and tag keywords, you can continue to explore the tools available for managing a collection.

9. Click **Identify fields**.

Most content from a document is indexed in the **text** field automatically. You might want to index certain types of content in different fields or split up large documents so that the **text** field contains fewer passages per document. To do so, you can teach Discovery to recognize important fields in your documents by applying a **Smart Document Understanding** model to your collection.

Smart Document Understanding (SDU) is a technology that learns about the content of a document based on the document's structure. You can apply a prebuilt SDU model or create a custom SDU model.

A screenshot of the 'Identify fields' page. At the top, there are tabs: Activity, Manage data, Identify fields (which is selected and highlighted in blue), Manage fields, Enrichments, Processing settings, and CSV settings. Below the tabs, there is a heading 'How would you like Smart Document Understanding to convert unstructured documents?'. Three options are listed: 'Text extraction only (default)' (selected), 'User-trained models', and 'Pre-trained models'. A note at the bottom states: 'Note: If OCR is enabled for the collection, text from images will also be extracted.'

Figure 8. Smart Document Understanding model options

To create a custom SDU model, you select the **User-trained model** option, and then annotate fields in your document. (You will not annotate documents as part of this tutorial.)

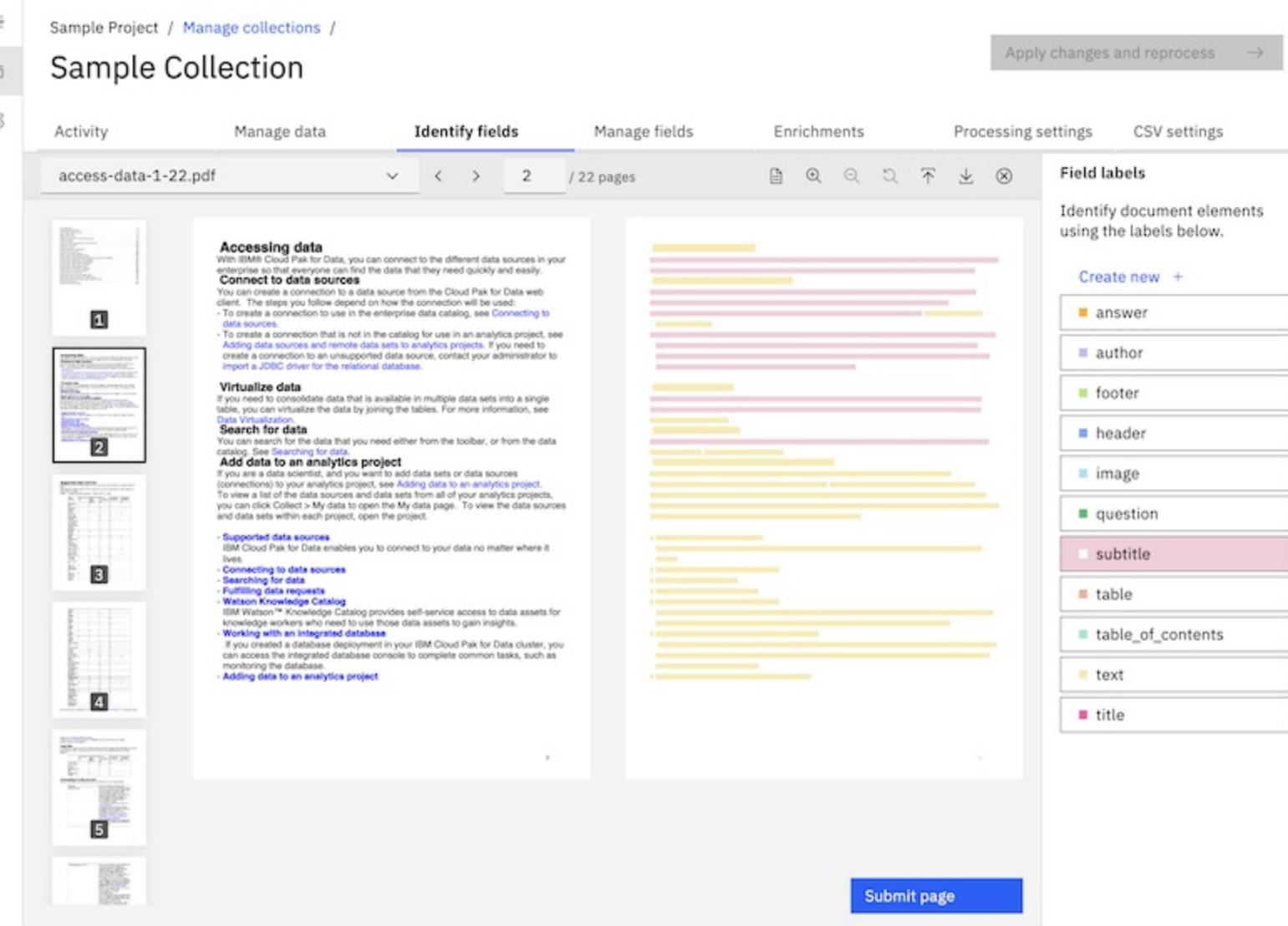


Figure 9. Smart Document Understanding annotation tool

Tip: For more information about SDU, see [Using Smart Document Understanding](#).

10. Click **Manage fields**.

The **Manage fields** page lists the indexed fields. From here, you can include or remove fields from the index. You can also split large documents into many smaller documents.

Field	Type ⓘ	Include in index
text	String	Yes
answer	—	Yes
author	—	Yes
footer	—	Yes
header	—	Yes
image	—	Yes
question	—	Yes
subtitle	—	Yes
table	—	Yes
table_of_contents	—	Yes
title	—	Yes

Improve query results by splitting your documents
You can split your documents into segments based on fields. Once split, each segment is a separate document that will be enriched, indexed, and returned as a query separately.

Split document +

Date format settings
Here are the date formats that are currently being supported. If your documents contain date formats that aren't in the list below, they won't be added to the index.

Date formats

- yyyy-MM-dd'T'HH:mm:ssZ
- yyyy-MM-dd'T'HH:mm:ssXXX
- yyyy-MM-dd'T'HH:mm:ss.SSSZ
- yyyy-MM-dd'T'HH:mm:ss.SSSX
- yyyy-MM-dd
- M/d/yy
- yyyyMMdd
- yyyy/MM/dd

Select a time zone ⓘ

Time zones

Select a date locale ⓘ

Locales

Figure 10. Fields in the collection index

Tip: For more information about splitting documents, see [Splitting documents to make query results more succinct](#).

Step 4: Search the sample project

1. Click the **Improve and customize** icon from the navigation panel.

The **Improve and customize** page is where you can try out queries, then add and test customizations to improve the query results for your project. A list of sample queries is displayed to help you get started with submitting test queries.

2. Click the **Run search** button for **IBM**.

Query results are displayed.

3. From one of the query results, click **View passages in document**.

A preview of the document where the result was found is shown.

4. Do one of the following things to explore the search result.

1. Click **Open advanced view**.

Useful summary information is displayed, such as the number of occurrences of any enrichments that are detected in the document.

2. Select the **URL** entity to highlight mentions of URLs within the text.

The screenshot shows the 'Advanced view' interface for the document 'add-ons-integrations-158-169.pdf'. On the left, the 'Identified elements' sidebar lists various entities: Organization (96), Number (71), TwitterHandle (5), URL (4), Person (2). The 'URL' checkbox is selected. Below this are 'Keywords' like Machine Learning add-on (1), top of IBM (5), following information (1), Parent topic (7), following TAR file (4). In the center, the document text is displayed with URLs highlighted in blue. On the right, the 'Matches found' section shows a list of URLs: <https://docs.shareinsights.com/docs/3.3/icp4d.htmlUsage>, <https://docs.shareinsights.com/docs/3.3/>. At the bottom, it says 'Cognos® Analytics Separately priced Self-service analytics,'.

Figure 11. Advanced view that shows entities that were recognized

3. To see how the information from the document is stored in JSON format, click the **View as** menu from the view header, and select **JSON**.

A JSON representation of the document is displayed.

The screenshot shows a JSON viewer interface with a dark theme. At the top right, there is a dropdown menu labeled "View as:" with options: "JSON" (selected), "PDF", and "Text". The main area displays a hierarchical JSON structure of the document. The root node contains fields like "document_id", "result_metadata", "enriched_text", "metadata", "extracted_metadata", and "text". The "text" field contains the document's content, which is a multi-line string starting with "1. 2. 3. - - 4. 5. 6. 7. 8. Installing the Watson Machine Learning add-on".

```

{
  "root": {
    "document_id": "51314522-fbf1-4ef8-8270-b6daf3668fed",
    "result_metadata": {
      "collection_id": "2b7bcb61-624a-9835-0000-017ebc024f96"
    },
    "enriched_text": [
      {
        "0": {
          "model_name": "natural_language_understanding",
          "mentions": [
            {
              "0": {
                "confidence": 0.7309633,
                "location": [
                  {
                    "text": "ibm"
                  }
                ],
                "text": "ibm"
              }
            }
          ],
          "text": "ibm",
          "type": "Organization"
        }
      ]
    }
  }
}

```

Figure 12. JSON representation of the document

You can explore the JSON representation to see information that Discovery captured from the document. For example, if you expand the `enriched_text` section, and then expand the `entities` section, you can see mentions of entities that were recognized and tagged by the Entities enrichment.

This screenshot shows the expanded `enriched_text.entities` section. It displays a list of entity mentions, each represented as an object with properties like `confidence`, `location`, and `text`. One specific mention for the word "ibm" is highlighted with a blue selection box, showing its confidence score of 0.7309633, its location within the text, and its text value.

```

{
  "10": [
    {
      "model_name": "natural_language_understanding",
      "mentions": [
        {
          "0": {
            "confidence": 0.7309633,
            "location": [
              {
                "text": "ibm"
              }
            ],
            "text": "ibm"
          }
        }
      ],
      "text": "ibm",
      "type": "Organization"
    }
  ]
}

```

Figure 13. Shows the enrichment_text.entities section of the JSON representation

Step 5: Customize the sample project

Now, let's customize the search result view a bit by adding a facet. A facet is a way to organize and classify documents that share similar patterns or content.

- From the **Improve and customize** page, submit the following natural language query:

\$ How do I install Discovery?

- Review the query results that are displayed.

The screenshot shows the Watson Discovery interface. At the top, there's a search bar with the query "How do I install Discovery?". Below it, a sidebar on the left has sections for "Top Entities" (with "Organization" selected), "Number", and "Collections" (with "Available collections"). The main area displays search results from three documents:

- "Installing the Watson Discovery add-on You can install the Watson™ Discovery add-on on top of IBM® Cloud Pak for Data." (Collection: Sample Collection)
- "Installing the Watson Assistant for Voice Interaction add-on You can install the add-ons that comprise Watson Assistant for Voice Interaction on top of IBM Cloud Pak for Data. Installing the Watson Discovery add-on You can install the Watson Discovery add-on on top of IBM Cloud Pak for Data. Installing the Watson Knowledge" (Collection: Sample Collection)
- "If you haven't run discovery on the data source, Run automated discovery on the data source where you want to audit assets. For details, see Using automated discovery . On the Discovery results page, click Review discovery results. Select the data sets that you want to audit." (Collection: Sample Collection)

On the right side, there's a panel titled "Improvement tools" with sections for "Customize display", "Extract meaning", "Teach domain concepts", "Define structure", and "Improve relevance".

Figure 14. Top Entities facet results

Notice that a **Top Entities** section is displayed. You can expand the entities and click one of them to filter the query results to show only those results in which the entity is mentioned. The **Top Entities** section is a built-in facet. It uses information that was added to the documents by the Entities enrichment.

You will add your own facet that uses the Keywords enrichment that you applied to the collection in a previous step.

- On the **Improvement tools** panel, expand **Customize display**, and then click **Facets**.

The screenshot shows the "Improvement tools" panel with the "Customize display" section expanded. It includes a "Facets" section with the sub-instruction "Add and manage filters used to refine a search or analysis.", and two other sections: "Search bar" and "Search results".

Figure 15. Customize display options

- Click **New facet**, and then click the **From existing fields in a collection** button.
- Choose `enriched_text.keywords.mentions.text`, change the label to **Keywords**, and then click **Apply**.

[Back](#)

Facets

New facet

Field

enriched_text.keywords.mentions.tex X V

Label

Keywords

Filtering options

- Multiple-choice checkboxes
 Single-choice radio buttons

Max number of values

4

Cancel Apply

Figure 16. Creating a Keywords-based facet

Remember the JSON representation of the document that you looked at earlier? Now that the Keywords enrichment is applied to the `text` field, and the documents are reprocessed, any keyword mentions found in the `text` field are included in the JSON representation of the document.

The field you picked to use for the facet (`enriched_text.keywords.mentions.text`) reflects where the keyword text is stored in JSON.

```
"enriched_{field_name}": [  
  "keywords" : [  
    "mentions" : [  
      "text": "Cloud Pak"  
    ]  
  ]  
]
```

6. The new facet is displayed. You can click a keyword to filter the documents to include only those results that mention the keyword.

The screenshot shows the Watson Discovery interface with a search bar at the top containing the query "How do I install Discovery?". Below the search bar, there are two main sections: "Top Entities" and "Keywords".

- Top Entities:**
 - Number: A dropdown menu showing "Installing the Watson Discovery add-on You can install the Watson™ Discovery add-on on top of IBM® Cloud Pak for Data."
 - Organization: A dropdown menu showing "Installing the Watson Assistant for Voice Interaction add-on You can install the add-ons that comprise Watson Assistant for Voice Interaction on top of IBM Cloud Pak for Data. Installing the Watson Discovery add-on You can install the Watson Discovery add-on on top of IBM Cloud Pak for Data. Installing the Watson Knowledge"
- Keywords:**
 - A list of checked keywords: Data, Cloud Pak, data, IBM.
 - A "Collections" section with a dropdown menu showing "Available collections".

On the right side, there are two document cards:

- add-ons-integrations-60-93-1-33.pdf** (Collection: Sample Collection)
- add-ons-integrations-1-27.pdf** (Collection: Sample Collection)

Figure 17. Keywords facet

You successfully added a built-in NLU enrichment that recognizes keywords in the sample collection documents. Then, you added a facet that uses the keywords enrichment to let you filter the documents by keyword.

Step 6: Share the sample project

1. Click **Integrate and deploy** from the navigation panel.

From here, you can share your project with colleagues and deploy it.

2. Follow the on-screen instructions to add a user, and then send login credentials and the provided link to your colleague.

The screenshot shows the "Integrate and deploy" page with the "Preview Link" tab selected. It includes sections for "Add user" and "Copy link".

Add user: This section contains four numbered steps with screenshots:

- In the IBM Cloud console, click to Manage and select Access (IAM).
- Click Invite users and enter user's information.
- Add one or more of the access groups you manage. You can assign the following types of access:
 - 1. Add users to access groups
 - 2. Manually assign access to IAM access policies or classic infrastructure permissions.
- Click Invite to add the user. Copy link below and send to the intended individuals.

Copy link: A text input field containing the URL: <https://us-south.discovery.watson.cloud.ibm.com/v2/instances/crn%3Av1%3Abluemix%3Apublic%3Adiscovery%3Aus-south%>

Figure 18. Integrate and deploy page

After you build your own search application and are ready to deploy it, you can use prebuilt user interface components or build a custom application.

- o Click **API Information**. From this page, you can get the project ID for your project. You need the project ID to use the Discovery API. You also need the service instance URL and API key. The credential details are available from the Manage page of your service instance in IBM Cloud.
- o Click **UI Components** to find links to ready-to-use code that you can use to create a full-featured search application faster.

Step 7: Add your own content

Now that you know more about some of the product features, you're ready to evaluate the data you want to search.

It's all about the data. Review the types of content you own that you want your search solution to be able to leverage.

Supported data sources

The following table shows the supported data sources for each deployment type.

Data source	IBM Cloud	IBM Cloud Pak for Data
Box	✓	✓
Database (IBM Data Virtualization, IBM Db2, Microsoft SQL, Oracle, Postgres)	✓	
FileNet P8	✓	
HCL Notes	✓	
IBM Cloud Object Storage	✓	
Local file system		✓
Salesforce	✓	✓
Microsoft SharePoint Online	✓	✓
Microsoft SharePoint On Premises	✓	✓
Website	✓	✓
Microsoft Windows file system	✓	

Supported data sources

Step 8: Not sure what you can build?

For more information about the types of search solutions you can build, see [Start getting value from your data](#).

 **Tip:** You can access the product documentation at any time by selecting the Help icon ⓘ from the page header of the product user interface. The help content is customized to provide information that is related to what you're doing in the product.

No matter what you build, step one is to create a project. Decide which project type best fits your needs.

If none of the existing types is quite right, you can choose **None of the above** to create a custom project instead.

Project descriptions

Need	Goal	Project type
<i>Which document contains the answer to my question?</i>	Find meaningful information in sources that contain a mix of structured and unstructured data, and surface it in a stand-alone enterprise search application or in the search field of a business application.	Document Retrieval
<i>Where is the part of the contract that I need for my task?</i>	Quickly extract critical information from contracts.	Document Retrieval for Contracts
<i>I want the chatbot I'm building to use knowledge that I own.</i>	Give a virtual assistant quick access to technical information that is stored in various external data sources and document formats to answer customer questions.	Conversational Search
<i>I want to uncover insights I didn't know to ask about.</i>	Gain insights from pattern analysis or perform root cause analysis.	Content Mining

For more information, see [Creating projects](#).

About Watson Discovery

IBM Watson® Discovery is an intelligent document processing engine that helps you to gain insights from complex business documents.

Use Discovery to visually train AI for deep understanding of your content, including tables and images, to help you find business value that is hidden in your enterprise data. Use natural language or structured queries to find relevant answers, surface insights, and build AI-enhanced business processes anywhere.

Start by connecting your data to Discovery. Next, teach Discovery to understand the language and concepts that are unique to your business and industry. Enrich your data with award-winning Watson Natural Language Processing (NLP) technologies so you can identify key information and patterns. Finally, build search solutions that find answers to queries, explore your data to uncover patterns and insights, and leverage search results in automated workflows.

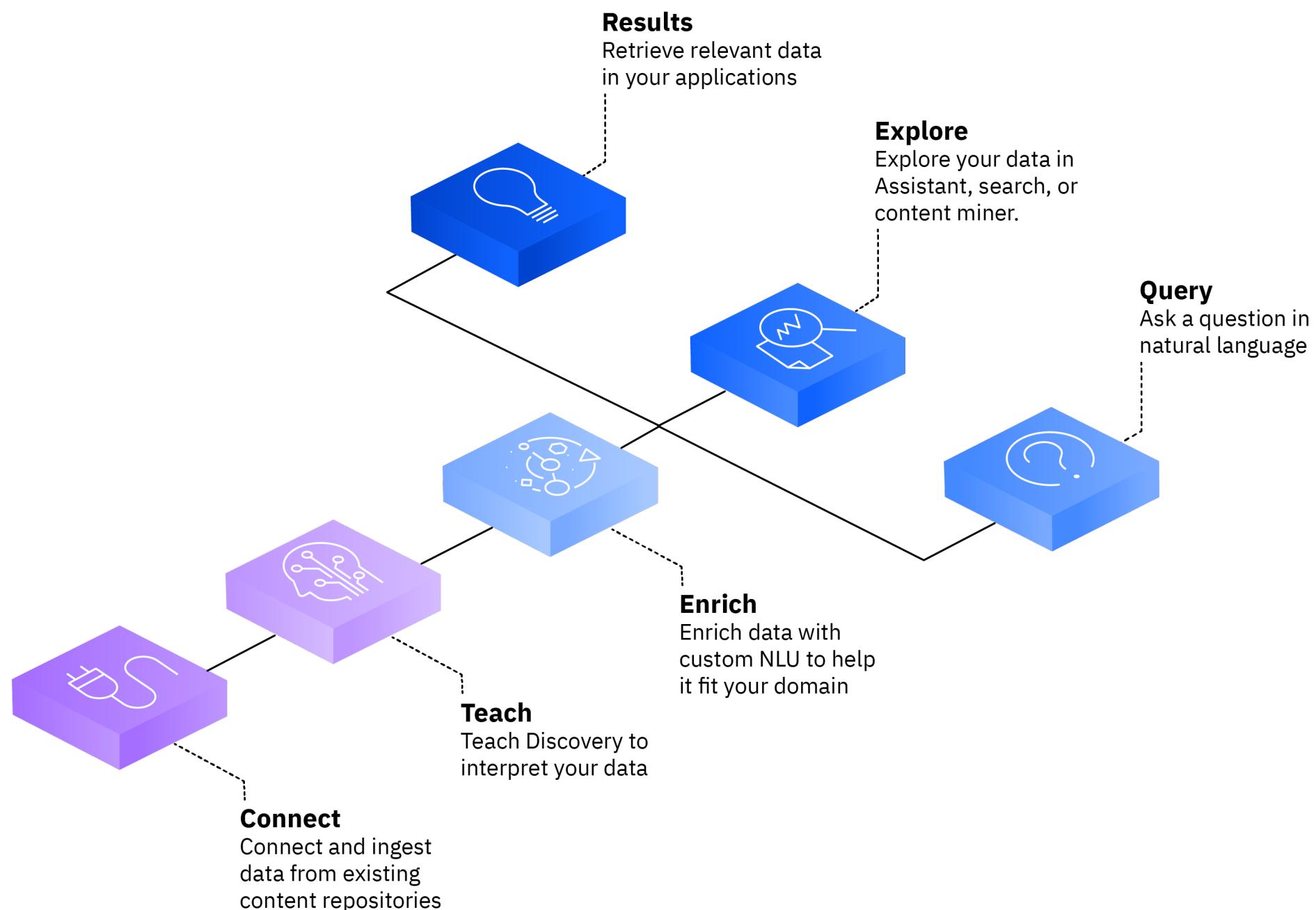


Figure 1. How to use Watson Discovery

Find out how Discovery is transforming data into artist insights at the 2023 GRAMMYs®. Read the [IBM Business Operations blog post](#) to learn more.

Overview video

Watch a video about how Discovery uses AI-powered search, retrieval, and content mining. This overview covers the key basics of projects, collections, fields, and enrichments. It explains how to upload your data and query for answers, find insights, and spot trends.



View video: [Get started with Watson Discovery](#)

Video transcript

Get started with Watson Discovery presented by David Williams - (Music intro) Welcome to Watson Discovery with AI.

In this video, we'll walk through some key concepts and show you how to get started.

Watson Discovery is made up of four main concepts, projects, collections, fields, and enrichments.

A project is a space where you can import different types of data from a variety of sources, and query for insights or answers.

A collection is a set of documents that you upload or crawl from a connected data source.

As documents are crawled, unstructured text is organized into fields such as author, file type, text, and more.

And enrichments are AI capabilities that you can apply to fields to identify and extract relevant information from your documents. This helps you find answers or insights from your data.

Let's dive in to the different project types.

A document retrieval project is used to build an AI-powered search function that finds answers in your business data.

A conversational project is used to enhance your chatbot's question and answer ability.

A content mining project helps you spot trends across large volumes of text-heavy business data.

Watson Discovery supports a wide selection of data sources you can crawl, like webpages, Cloud Object Storage, Microsoft SharePoint, and more. You can even upload your own data from any data source.

After connecting and processing your data, you can apply enrichments to bring your data to life. Some commonly used enrichments are entities, contracts, and table understanding. Entities enrichment can be used to recognize people, organizations, and more. Contracts enrichment can be used to decompose contracts to fields, clauses, and relationships. The table understanding enrichment can be used to identify tables and return them as an answer to a query.

You can also create custom enrichments, such as a dictionary, so Discovery can understand your industry-specific terminology and support intelligent queries.

Now, you know the basics.

To get started, take our step-by-step product tour to get familiar with the user interface and sample project.

Using Discovery

Discovery can be deployed as a managed cloud service or can be installed on premises. This documentation describes how to use the product regardless of how it is deployed. Information that applies exclusively to one deployment type is denoted by the appropriate icon:

- IBM Cloud Pak for Data for installed instances, such as IBM Watson® Discovery Cartridge for IBM Cloud®.
- IBM Cloud for managed instances, such as Discovery Plus, Enterprise, and Premium plan instances that are hosted by IBM Cloud or instances that are provisioned with [IBM Cloud Pak for Data as a Service](#).

 **Tip:** Click the Help icon ⓘ from the header of any page in the product user interface to open the Discovery documentation.

Browser support

IBM Cloud Pak for Data

- The minimum required browser software for the product user interface includes the following browsers:

Google Chrome

Latest version -1 for your operating system

Mozilla Firefox

Latest regular -1 and Extended Support Release (ESR) version for your operating system

Microsoft Edge

Latest version -1 for Windows

Apple Safari

Latest version -1 for Mac

- The IBM Cloud Pak for Data web client where you create service instances supports the IBM Cloud Pak for Data requirements. For more information, see [Supported web browsers](#)

IBM Cloud

- Deployments of Discovery that are managed by IBM Cloud follow the IBM Cloud requirements. For more information, see [Prerequisites](#)
- For more information about browser support for deployments that are provisioned with Cloud Pak for Data as a Service, see [Which web browsers are supported for Cloud Pak for Data as a Service](#).

Language support

Language support by feature is detailed in the [Supported languages](#) topic.

Beta features

IBM releases services, features, and language support for your evaluation that are classified as beta. These features might be unstable, might change frequently, and might be discontinued with short notice. Beta features also might not provide the same level of performance or compatibility that generally available features provide and are not intended for use in a production environment.

Terms and notices

IBM Cloud

- [IBM Cloud Terms of use](#)
- [Service terms \(Search for Watson Discovery\)](#)
- [Data Processing and Protection Datasheet](#)

IBM Cloud Pak for Data

- [Security on Cloud Pak for Data](#)

Trademarks are listed in the [Trademarks](#) page for all IBM Cloud services.

Getting the most from Discovery

Getting the most from Discovery

Discovery was redesigned to introduce new features and a simpler way to build solutions.

The redesigned product is referred to as Discovery v2. When you create an instance on IBM Cloud or install and provision an instance on IBM Cloud Pak for Data, you get the new and improved version of Discovery.

Advantages of using the latest version

Discovery v2 offers the following features and enhancements:

- A project-based experience that supports many different use cases within a single environment.
- Built-in customization tools for adding dictionaries, patterns, and classifiers to help business users build projects that understand the language of their domain.
- Connectors to popular data sources that can quickly access valuable data where it resides.
- Smart Document Understanding that learns from the structure of human-readable documents, such as PDFs.
- Natural language query support across all document types, optimized with machine learning to find targeted answers.
- Advanced search capabilities, such as answer finding, curations, and table retrieval.
- An out-of-the-box contract understanding function that helps you search and interpret legal contracts.
- A full-featured Content Mining application that you can use to conduct in-depth analysis of unstructured text.
- Customizable user interface components that help you to deploy custom applications.

For more information, see [Migrating to Discovery v2](#).

Comparing v1 and v2 features

If you are already familiar with Discovery v1, learn more about how Discovery v2 compares.

Discovery v2 has new features that were previously unavailable. The following table describes feature support in both versions.

Feature	Product redesign (v2)	Earlier version (v1)
Use projects to organize your work	✓	
Use the Smart Document Understanding (SDU) to annotate your documents	✓	✓
Leverage intuitive user interface tools to add domain-specific artifacts, such as dictionaries and custom machine learning models	✓	
Create a content mining project type and then use the built-in Content Mining application to do in-depth data analysis (<i>IBM Cloud Pak for Data, Enterprise, and Premium plans only</i>)	✓	
Perform real-time NLP with the Analyze API (<i>IBM Cloud Pak for Data and Enterprise plans only</i>)	✓	
Apply a pretrained Smart Document Understanding model to your collection for similar benefits with less effort	✓	
Process text from scanned documents or other images	✓	✓
Extract meaning from tables	✓	
Get insights from contracts (<i>IBM Cloud Pak for Data, Enterprise, and Premium plans only</i>)	✓	
Apply the <i>Part of Speech</i> enrichment to your data	✓	
Use the Entity Extraction, Document and Phrase Sentiment Analysis, and Keyword Extraction enrichments	✓	✓

Use the Category classification, Concept tagging, Relation Extraction, Emotion Analysis, and Semantic Role Extraction, Sentiment of Keywords and Entities enrichments, which are available with the Natural Language Understanding service	✓
Build a custom entity type system	✓
Apply Watson Knowledge Studio NLP models to your data	✓ ✓
Support for more connectors from a IBM Cloud Pak for Data deployment, including databases, file systems, FileNet P8, and HCL Notes	✓
Some connectors support document-level security from a IBM Cloud Pak for Data deployment	✓
Programmatically configure external data source crawls	✓
Configure the normalization processes of document segmentation and HTML file inclusion or exclusion rules during ingestion	✓
Configure the JSON normalization process during ingestion and after enrichment	✓ ✓
Configure dictionary tokenization	✓
Advanced question-answering capabilities, such as returning the exact answer	✓
Discovery Query Language (DQL) API support	✓ ✓
Retrieve passages from documents	✓ ✓
Perform relevancy training to improve query results	✓ ✓
Configure continuous relevancy training	✓
Retrieve tables	✓
Query result deduplication	✓
Identify document similarity in query results	✓ ✓
Indicate a preference (bias) in queries	✓
Review query logging and metrics	✓

Feature support details

Limit details

For more information about artifact limits per plan, see the feature documentation:

- [Advanced rules model limits](#)
- [Classifier limits](#)
- [Collection limits](#)
- [Dictionary limits](#)
- [Document limits](#)
- [Entity extractor limits](#)
- [Machine Learning model limits](#)
- [Pattern limits](#)
- [Project limits](#)
- [Query limits](#)
- [Regular expression limits](#)
- [SDU limits](#)

The following limits apply only to Content Mining project types:

- [Document classifier limits](#)
- [Regular expression pattern limits](#)

To check the current status of the limits and usage for your plan type, you can open the [Plan limits and usage](#) page at any time.

1. From the product page header, click the user icon .

The **Usage** section shows a short summary.

2. Click **View all** to see usage information for all of the plan limit categories.

To leave the page, click the web browser back button or the **My Projects** tab.

Migrating to Discovery v2

A redesign of the product, Discovery v2, was introduced in November 2019. Discovery v2 offers significant advantages over Discovery v1.

Learn about how to migrate a v1 Discovery service instance to Discovery v2, including how to move data and update your applications.

The major structural differences between Discovery v1 and v2 include:

- There is no concept of an environment in v2. The deployment details such as size and index capacity are managed for you when you choose the appropriate service plan for your needs. For managed deployments, you can choose a Plus, Enterprise, or Premium plan, for example. For installed deployments, the sizing is managed by the deployment type that you specify when you install the service in Cloud Pak for Data.
- There is no single configuration object in v2. Control of the enrichments that are applied to documents is managed in the collections and project objects in v2. Other v1 configuration capabilities, such as the ability to customize the conversion step of ingestion, are not available in v2.
- Greater programmatic support is available for custom enrichments in v2. New enrichment API methods are available that you can use to create enrichments. v2 also introduces document classifier API methods that you can use to train document classifier models programmatically. You can apply these custom enrichments to a collection by using the API.
- The capabilities of a natural language query search are expanded in v2 to enable the return of the top passages per document and of succinct answers from passages. Other advanced search capabilities are introduced, including table retrieval. In v2, the deduplication parameter is not available and the continuous relevancy training and query logging functions are not available.
- For more information about feature differences, see [the feature comparison table](#).
- For more information about detailed API differences, see [API version comparison](#).

Discovery v2 is available for all users of Plus or Enterprise plan instances, or Premium plan instances that were created after 15 July 2020. v2 is also available for IBM Watson® Discovery Cartridge for IBM Cloud Pak® for Data users.

Migration overview

Migrating from Discovery v1 to v2 is a multistep process that you can do independently.

The two versions of the Discovery service have many differences, but you can adopt techniques and utilities that were applied to a v1 instance for use with your new v2 instance.

To migrate from v1 to v2, you must complete the following high-level steps:

1. [Plan the migration](#).
2. [Transfer your documents](#).
3. [Update your application to use the v2 API](#).
4. Regression test and deploy the updated application.
5. [Delete your v1 plan service instance](#).



Note: Some steps require you to make programmatic changes by using the API and others involve changes that you can make from the product user interface.

Plan the migration

Get familiar with what's new in v2 and learn about how it differs from v1 before you provision a v2 instance. Your first v2 Plus plan trial instance is available at no charge for 30 days. Learn about and plan for the migration before you provision the instance so that you can get the most from your trial.

When you're ready to start the migration, create a migration schedule that you and your team can follow as you complete the process. Be sure to set up the new v2 service instance and get projects and collections re-created in the new service instance before you switch over to using the v2 service and before you delete your v1 instance.

Learn about the Discovery v2 plan options, so you can choose the right plan for your long-term needs. The Plus plan that you use to get started might be sufficient. However, you might choose to use an Enterprise or Premium plan instead. From a Plus plan, you can do an in-place upgrade to an Enterprise plan, but not to a Premium plan.

Plan how to adapt your application

One of the main changes between versions is that Discovery v2 introduces projects. A project consists of one or more collections. The advantage of using projects is that one query can run against many collections at the same time. Each collection can contain documents that you upload or that you crawl from a single data source, such as a website, Microsoft SharePoint, and more.

Things to consider when you adapt your application to use projects:

- Although the concept of an environment does not exist in v2, data is still organized into collections. In v2, collections are grouped into projects. In most cases, you want to migrate a single v1 collection to a single v2 collection.

If you want to keep relevancy training information that is applied to a v1 collection, add the collection documents to a single collection in your v2 project.

- Decide how many collections you want to add to each v2 project. All project types, except Content Mining projects, can contain up to 5 collections. Choose the right type of project for your data.

To optimize search results, different enrichments and configuration options are applied automatically to collections that are added to different project types. For more information, see the following topics:

- [Project descriptions](#)
- [Default project settings](#)
- [Default query settings](#)

- The Discovery v2 API changed to account for projects and collections, among other enhancements. Some API calls changed to support actions at the project level instead of the collection level, such as submitting a query and running relevancy training. Many other API methods changed and some are not available in v2. For a detailed comparison of the v1 and v2 API methods, see [API version comparison](#).

Picking a service plan

Choose among the **Plus**, **Enterprise**, and **Premium** managed plans or opt for an on-premises installation by purchasing the Discovery Cartridge for IBM Cloud Pak for Data. Review the benefits and limits of each type of plan before you choose one.

- For more information about the plans, see [Discovery pricing plans](#).
- For more information about artifact limits, see [Limit details](#).

The following table shows plan types for managed deployments that are generally similar between v1 and v2.

Current v1 plan	Example v1 data usage	Similar v2 plan
Lite	Not applicable	Plus Trial (no charge for 30 days only)
Advanced (low usage)	10,000 documents, 10,000 queries per month	Plus
Advanced (high usage)	100,000 documents, 100,000 queries per month	Enterprise
Premium	Not applicable	Enterprise or Premium

Similar plans

 **Tip:** To get information about the current storage, documents, and collections used, click the **Environment details** icon from the product user interface header.

You cannot do an in-place upgrade from a v1 plan, such as Lite or Advanced, to a v2 plan. You must create a new v2 plan, and then move your data to the new service instance. While you migrate your data from v1 to v2, you will likely have both a v1 and v2 instance deployed at the same time. Consider using the 30-day no charge trial that is available with your first Plus plan instance during this time.

Collecting metrics

Make a note of the following information so you can compare it to your service instance data after the migration:

- Number of collections

To get the number of collections in an instance in v1, use the [List collections](#) API.

- Number of documents per collection

To get the number of documents in a collection in v1, use the [Get collection details](#) API.

```
GET {url}/v1/environments/{environment_id}/collections/{collection_id}`
```

The API returns information about the status of the documents in the collection, which includes the total number of available documents.

```
"document_counts": {  
    "available": 34,  
    "{other}": "{values...}"  
}
```

Transferring documents from v1 to v2

How you transfer your documents depends on the technique that was used to ingest the documents in v1.

Re-create one collection at a time. If you start multiple ingestion processes at the same time, you can tax the system resources and increase the overall time that it takes for the processing to be completed. You also want to keep an eye out for any informational messages that are generated by the ingestion process. It is easier to troubleshoot an ingestion issue, for example, when you ingest one collection at a time.

Uploaded data

If you used the API to upload documents into Discovery v1, a similar API is available in v2 to upload documents into collections. You must update any workflows that you use to automate the process to account for the new arrangement of projects and collections.

If the original documents that you ingested into Discovery v1 are no longer available, you can use the query API to extract the document text from Discovery v1. You can then add the text to a collection in Discovery v2. For more information, see [Recovering documents](#).

Crawled data

If you crawled data from an external data source in v1, you can continue to crawl data from the same external data source in v2. All of the same data sources are supported.

To use data from an external data source, you must re-create the collections within a v2 project, and configure how the data source is crawled. For more information, see [Overview of data sources](#).

The service needs time and resources to crawl and ingest documents from external data sources. Re-create the connectors one at a time. Factor the time it takes to recrawl the data into your migration plan schedule.

Prebuilt data collections

The following built-in data source collections are not available in v2:

Watson Discovery News

This pre-enriched data source is not offered in v2. For more information about an alternative way to get news data, see [Using a news service with v2](#).

COVID-19 kit

This pre-built collection was designed to help you fuel a dynamic chatbot that is built with IBM Watson® Assistant and Discovery to answer your customers' questions about COVID-19. In v2, you can build a similar solution. Create a **Conversational Search** project type with collections that crawl trusted websites for answers to COVID-19 questions.

Ingesting data

To ingest v1 data into a Discovery v2 instance, complete the following steps:

1. Create a v2 service instance.
2. Create a project.

3. Add a collection to the project.

- Uploaded data:

From the API, you create a collection and add documents to it with two separate methods. Use the [Create a collection](#) method to create the collection. Next, add the same source documents that you added to your v1 collection to the v2 collection. Use the [Add document](#) or [Update document](#) methods. To assign the same v1 document ID to the document as you add it to the v2 collection, append the document ID to the endpoint. For more information, see [Retaining document IDs](#).

From the v2 product user interface, upload the same source documents that you added to your v1 collection to the v2 collection.

- Crawled data: You cannot crawl data from an external data source programmatically in v2. From the product user interface, re-create the connection to the external data source, and then crawl the external data source from scratch.

4. From the product user interface, you can configure the Discovery v2 collection. For example, you can choose whether to enable optical character recognition. For an external data source, you can set the crawl schedule.

5. Apply enrichments to your data. You can apply pre-built Natural Language Processing enrichments or custom enrichments that you create.

In v1, enrichments are associated with the configuration that is generated when you create the environment. In v2, enrichments are associated with the collection configuration. Some enrichments are applied to your collection by default, depending on the type of project used. For more information, see [Default project settings](#). In v2, you can configure the collection to use any subset of available enrichments on the fields of your document.

Retaining document IDs

Document IDs are assigned to the documents that you add to a v2 collection when you upload them from the product user interface or add them by using the [Add a document](#) API method.

You might want to retain the IDs of your v1 documents in v2 if you are using processes that depend on these unique identifiers. For example, regression testing for the application might verify that specific documents are returned by checking the document IDs. Relevancy training uses the document IDs to track documents between training runs. These processes are easier to adapt if the document IDs are the same between your v1 and v2 instances. Otherwise, the processes that are used with the Discovery v1 instance must be remapped to the IDs that are assigned to the documents after they are added to the Discovery v2 instance.

If you specified your own documents IDs when you added documents to the v1 service instance, you can retain the IDs by using the [Update a document](#) method instead of the [Add a document](#) method. With the update method, you can assign a document ID to the document as you add it to the v2 collection. For more information, see [Update a document](#).



Note: If your data is stored in a JSON file, an array in the original document generates a document ID with a number appended to it. For example, `original_id_n`. To retain the original document ID without the number suffix, remove the array in the JSON file. Change `[{"name": "value"}]` to `{"name": "value"}`, for example.

If your v1 documents have system-generated IDs, you can submit an empty [search query](#) to retrieve a list of the documents and their IDs. You can then assign the same ID to each document as you add it to your new collection in v2.

Recovering documents

In some cases, the original documents that were ingested into Discovery V1 are no longer available. You can use the Discovery v1 instance to retrieve information from the document. Discovery creates a text copy of each document that it ingests. The copy is text only, so any documents in HTML, PDF, or other nontext formats are converted to a text-only version.



Important: You can recover only the first 10,000 documents in a collection by using this method. For more information about a way to recover more than 10,000 documents, see [Recovering more than 10,000 documents from a collection](#).

To transfer document information from v1 to v2, complete the following steps:

1. Extract the documents from v1 by using the API to [submit an empty query](#).

For example, `GET {url}/v1/environments/{environment_id}/collections/{collection_id}/query?q=`.

The API returns the results. The `matching_results` field specifies the total number of results. The results object returns the matching documents. Each document is returned as a separate JSON object. It returns a maximum of 10 documents by default.

```
{  
  "matching_results": 34,  
  "session_token": "nnn",  
  "results": [  
    {"result objects": "maximum of 10 by default"}]
```

```
    ]  
}
```

2. You can use the `count` and `offset` parameters to page through the query results and save all of the documents.

For example, to get 100 documents at a time, you can set the `count` to `100` and `offset` to `0` and submit the query.

```
GET {url}/v1/environments/{environment_id}/collections/{collection_id}/query?q=&count=100&offset=0
```

Next, you can again set the count to 100, but this time set the offset to 100 to get the next 100 documents.

```
GET {url}/v1/environments/{environment_id}/collections/{collection_id}/query?q=&count=100&offset=100`
```

Repeat this process, incrementing the offset by 100 until you retrieve all of the documents.

3. Prepare the exported documents to be ingested into v2.

Each resulting JSON file that you get from Discovery v1 contains data that is extracted from the original document, such as text, html, and other fields. If custom metadata was associated with the document when it was uploaded to v1, it is also present in the JSON file. In addition, the file contains several fields that were generated by the v1 analysis. Retain only a subset of this data as part of the document that you add to Discovery v2.

The following tips can help you decide which fields to keep:

- o Include the `text` field or any other field with textual content that you want to be able to enrich or search in Discovery v2.
- o Include any custom metadata that is stored in the document. This metadata is typically specific to the application that uses Discovery and is used to filter documents in a search. For example, `metadata.customer_id`.
- o Do not include enrichments from Discovery v1. For example, `enriched_text.entities`. Discovery v2 generates its own enrichments.
- o Exclude fields that are generated by Discovery unless they are used by your application and contain information that is unique to the v1 version of the document. In that case, rename the field so that it does not get replaced when the document is ingested into Discovery v2. For example, `extracted_metadata.publicationdate` is a field that is generated by Discovery when a document is ingested. Maybe you want to retain the `metadata.parent_document_id` information from v1 to understand how subdocuments were originally generated from a single source document.
- o Avoid fields that have reserved field names. For more information, see [How fields are handled](#).

4. Ingest each edited v1 JSON document into the Discovery v2 instance. The Discovery v1 document ID can be maintained in Discovery v2. For more information about how to retain the document ID, see [Retaining document IDs](#).

Recovering more than 10,000 documents from a collection

A query can only return up to 10,000 documents. However, if you want to recover more than 10,000 documents from your collection, you need a way to separate the documents into non-overlapping subgroups. Each subgroup should contain fewer than 10,000 documents that can be returned by a query. Then, you can paginate through the results to retrieve the documents.



Note: Pagination for results is restricted to the maximum of 10,000 documents that are returned by the query. Specifically, the combined use of the `count` and `offset` pagination parameters cannot exceed 10,000 documents.

One way to separate the documents into non-overlapping subgroups is to leverage a field that exists in every document and contains a unique value. For example, the SHA-1 field contains a hash of the original source file and is formatted as a hexadecimal string value. You can use the first character of the field as a way of dividing the collection into subgroups. Because SHA-1 contains a hexadecimal value, the first character can have up to 16 possible values (0-9 or a-f). If you filter by the `first_char_of (SHA-1) == 0`, it might return approximately 1/16 of the entire collection. You can then loop through each of the possible 16 values to get the rest of the documents. If optimum number of documents are not returned in one of the subgroups, you can use the first 2 characters of the SHA-1 field to divide the collection into 256 subgroups instead.

Transferring relevancy training

Relevancy training that was done in Discovery v1 can be transferred to Discovery v2. Transferring the training works best with a Discovery v2 project that has one collection that contains the same documents from the Discovery v1 collection.

Even if collections were added or documents changed, the relevancy training can be transferred. However, you must update the training to account for the changes.

To transfer relevancy training, complete the following steps:

1. Load the documents in Discovery v2.
2. Programmatically download the queries that were used for relevancy training in Discovery v1. For more information, see [List training data](#).

3. Programmatically re-create the relevancy training data in Discovery v2. Add each training query separately by using the [Create a query](#) method. For more information, see [Create a training query](#).

Be sure to specify the v2 collection ID. You must also specify the document ID also.



Note: If you did not [retain the document IDs](#) between the v1 and v2 collections, then you must find the v2 document ID that corresponds to the v1 document ID that is referenced in the downloaded query example.

Transferring models

You can reuse some of the models that you created in v1 with your v2 project.

Smart Document Understanding (SDU) models

You can import an SDU model that was built with Discovery v1 into Discovery v2. However, the performance of the model might differ between versions. Compare the results of the v1 SDU model in v2 to verify that the behavior is the same. You cannot edit the imported v1 SDU model. If the imported model can't recognize document elements that it recognized in v1 and that are important to your use case, you must re-create the SDU model in the Discovery v2 product user interface. For more information, see [Exporting SDU models](#) in the v1 documentation and [importing the SDU model](#) in the v2 documentation.

Machine learning models

You cannot deploy models directly to Discovery v2 service instances from Knowledge Studio. Instead, you must export the machine learning models from Knowledge Studio, and then import them into Discovery. The model must have been exported from Knowledge Studio after 16 July 2020. If you have a model that was exported before that date, you must reexport the model from Knowledge Studio. Only paid Knowledge Studio plans support exporting models.

For more information, see one of the following topics:

- IBM Cloud Pak® for Data: [Exporting a machine learning model](#)
- IBM Cloud: [Deploying a machine learning model to Watson Discovery](#)

For information about how to import a model to Discovery v2, see [Importing Machine Learning models](#).

Update your application to use the v2 API

The Watson Developer SDKs support both Discovery v1 and v2.

These instructions assume that your application is using the latest version of the v1 API (version [2019-04-30](#)).

When you port an application that currently uses the Discovery v1 API to use v2, you must plan how to address the following high-level differences between the two versions.

In addition to these high-level changes, review the differences at a per-method level to understand what else you might need to change. For more information, see [API version comparison](#).

- v2 organizes data by project and collections; there is no concept of an environment. For example, compare the following requests to get a collection:

v1 [Get collection](#)

```
GET {url}/v1/environments/{environment_id}/collections/{collection_id}
```

v2 [Get collection](#)

```
GET {url}/v2/projects/{project_id}/collections/{collection_id}
```

- In v1, relevancy training runs on a single collection. In v2, relevancy training runs on a project. The project might contain many collections. If so, relevancy training is applied across all of the collections. For information about how to transfer relevancy training, see [Transferring relevancy training](#).

For example, compare the following requests that return the status of relevancy training:

v1 [Get collection](#)

```
GET {url}/v1/environments/{environment_id}/collections/{collection_id}
```

v2 [Get project](#)

```
GET {url}/v2/projects/{project_id}
```

- Submitting a query is similar between the two versions. In v2, you can query all of the collections in a project or you can limit the query to one or more collections by specifying a `collection_ids` parameter. For example, compare the following requests to query data:

v1 [Query](#) request

```
POST {url}/v1/environments/{environment_id}/collections/{collection_id}/query
```

Data that is submitted with the request:

```
{  
  "query": "text:IBM"  
}
```

v2 [Query](#) request

```
POST {url}/v2/projects/{project_id}/query
```

Data that is submitted with the request:

```
{  
  "collection_ids": [  
    "{collection_id_1}",  
    "{collection_id_2}"  
  ],  
  "query": "text:IBM"  
}
```

You can optionally omit the `collection_ids` parameter to query across all of the collections in the project.

- The `passage` parameter for a query has a new `per_document` option that ranks the documents by document quality, and then returns the highest-ranked passages per document in a `document_passages` field for each document entry in the results list of the response. If false, ranks the passages from all of the documents by passage quality regardless of the document quality and returns them in a separate `passages` field in the response.
- When passages are returned for a query, you can also enable answer finding. When true, answer objects are returned as part of each passage in the query results. When `find_answers` and `per_document` are both set to true, the document search results and the passage search results within each document are reordered by using the answer confidences. The goal of this reordering is to place the best answer as the first answer of the first passage of the first document. Similarly, if the `find_answers` parameter is set to true and `per_document` parameter is set to false, then the passage search results are reordered in decreasing order of the highest confidence answer for each document and passage.
- Both v1 and v2 support custom stop words. However, there are a few differences in how custom stop words are used:
 - There is no default custom stop words list for Japanese collections in v2.
 - When you define custom stop words in v1, your stop words list replaces the existing stop words list. In v2, your list augments the default list. You cannot replace the list, which means you cannot remove stop words that are part of the default list in v2.

Update how your application handles query results

The way that your application shows query results might need to be updated due to the following differences between the query results document syntax between the v1 and v2 queries:

- At the entity enrichment level, the following information is not supported in v2:
 - Disambiguation
 - Emotion
 - Sentiment

The **Part of Speech** enrichment is applied automatically to documents in most project types in v2, but the index fields that are generated by the enrichment are not displayed in the JSON representation of the document.

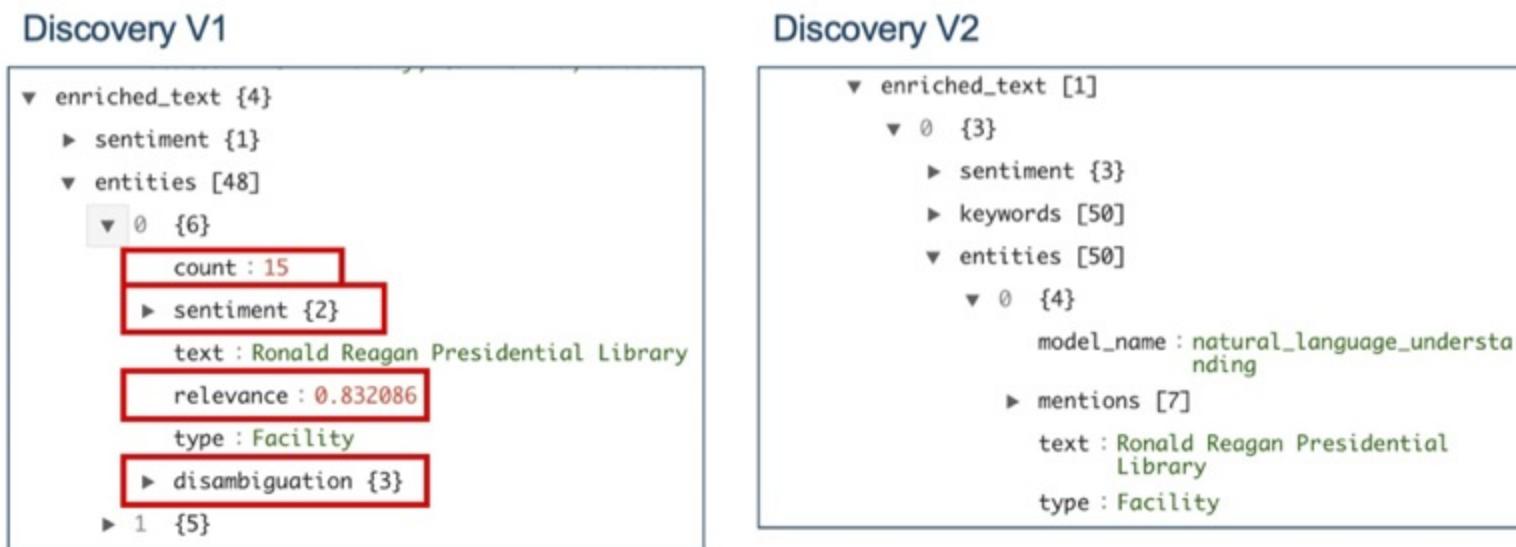


Figure 1. Entities data structure differences

- Instead of the `count` and `relevance` in v1, v2 includes the mentions.

Each entry in the mention corresponds to an occurrence of the entity in the document text. In the following example, seven occurrences are found. For each occurrence, a confidence score and the offsets of the mention text are displayed. You can use the offsets to highlight the mention in the document text when the result is displayed in a user interface.

```

▼ entities [50]
  ▼ 0 {4}
    model_name : natural_language_understanding
  ▼ mentions [7]
    ▼ 0 {3}
      confidence : 0.24154784
    ▼ location {2}
      end : 34
      begin : 0
    text : Ronald Reagan
          Presidential Library

```

Figure 2. Entity mentions in Discovery v2

- The JSON structure of query responses is rearranged slightly in v2.
- Deduplication information is not included in the v2 query response.
- In v2, `enriched_text` is an array instead of an object.
- In Discovery v2, the Entities v2 enrichment is used. Entity type names in v2 are specified in headline case, instead of all uppercase letters. If you use a query or aggregation that specifies an entity name, you must change the capitalization. For example, change `PERSON` to `Person`.
- Fields from JSON files that are added to a collection are converted differently during ingestion between v1 and v2. If your application manipulates these results, you might need to make adjustments.



Note: You can specify the `normalizations` and `conversions` objects in the [Update a collection](#) method of the API to move or merge JSON fields.

Original JSON	v1	v2	Notes
field content	representation	representation	
<code>"field": null</code>	<code>"field": null</code>	N/A	v1 retains the null value. v2 skips the null field altogether.

"field": ""	"field": ""	N/A	v1 retains the empty text value. v2 skips the empty text field altogether.
"field": "value2"	"field": "value2"	"field": "value2"	No difference.
"field": []	"field": []	N/A	v1 retains the empty array. v2 skips the field with the empty array altogether.
"field": ["value4"]	"field": ["value4"]	"field": "value4"	v1 retains the singleton array. v2 converts the singleton array into the value only; it is not stored as part of an array.
"field": [1, 2, 3]	"field": [1, 2, 3]	"field": [1, 2, 3]	No difference.
"field": ["v6", "v7", "v8"]	"field": ["v6", "v7", "v8"]	"field": ["v6", "v7", "v8"]	No difference.

How JSON source fields are handled

Verifying that your data was migrated successfully

To verify that the migration was successful, compare the following metrics to the [metrics that you noted before the migration](#).

- Number of collections

Be sure to re-create all of the collections that you used in v1 and want to keep. With the v2 [List collections](#) API method, you can get a list of collections, but you must submit a request per project. You cannot use one call to get the total number of collections per service instance.

- Number of documents per collection

For collections with uploaded data, check the number of documents in the collection by sending an empty query with the [Query a project](#) API method. Specify the collection ID parameter to limit the results to only documents in one collection. An empty query returns all documents. Therefore, you can get the total number of documents from the [matching_results](#) value in the response.

The number of documents per collection should be close to the number of documents that were stored in the same collection in v1. The numbers might not be the same.

For crawled data, do not be surprised if the v2 collection has fewer documents. The v1 connectors do not delete documents from a Discovery collection that are deleted from the external data source. Your v2 version of the collection has a fresher crawl of the data as it exists in the external data source today.



Tip: Do not expect the search results to be the same for queries that you submit in the v1 and v2 instances.

Using a news service with v2

If you used the Watson Discovery News data source in v1 and want to create a data source with equivalent function in v2, find a news and events data provider service. Look for a service that offers a News API that extracts news articles in JSON format. You can then upload the JSON files to create a News collection in your v2 project.

Delete your v1 service instance

After your data is migrated and your applications are updated to use the new v2 service instance, be sure to delete your v1 service instance. You are charged for the v1 service instance until you delete it. For more information, see [Deleting a managed service instance](#).

API version comparison

For most API methods, the request parameters and response bodies differ between v1 and v2. Learn about the equivalent or alternative v2 methods that you can use to do actions that are supported by the v1 API.

The comparison information assumes you are using the latest version of the v1 API (version [2019-04-30](#)) and compares it to the latest version of the v2 API (version [2020-08-30](#)).

Environments

There is no concept of an *environment* in v2. The deployment details such as size and index capacity are managed based on the service plan type. In v2, collections are organized in projects. You can create different types of projects to apply default configuration settings to the collections that you add to the projects.

There are no equivalent methods in v2 for the v1 environment methods. However, the following table shows v2 methods that serve similar functions to the corresponding v1 methods. The supported parameters and response bodies that are returned for each method differ also.

Action	v1 API	Related v2 API
Create an environment	POST /v1/environments	POST /v2/projects
List environments	GET /v1/environments	GET /v2/projects
Get environment info	GET /v1/environments/{environment_id}	GET /v2/projects/{project_id}
Update an environment	PUT /v1/environments/{environment_id}	POST /v2/projects/{project_id} v2 uses POST instead of PUT .
Delete an environment	DELETE /v1/environment/{environment_id}	DELETE /v2/projects/{project_id}
List fields across collections	GET /v1/environments/{environment_id}/fields	GET /v2/projects/{project_id}/fields

Environment API action support details

Configurations

The v2 API does not have an endpoint that is dedicated to configurations. Instead, configuration settings for projects, collections, and queries are specified directly in the API for those objects. Not all of the configuration parameters that are available in v1 are available or applicable in v2.

In the [v1 configuration API](#), the JSON object that is used to specify a configuration object contains several parameters that are either available in different formats from other v2 endpoints or are not available in v2. The following table describes how to find related parameters in v2.

You cannot customize the conversion of documents during the ingestion process in v2 as you can in v1.

v1 configuration parameter	v2 API
<code>"conversions.html": { ... }</code>	Not available
<code>"conversions.image_text_recognition": { ... }</code>	Not available from the API. However, you can enable optical character recognition (OCR) for a collection from the product user interface to extract text from images. OCR has other benefits, too. For example, if a page in a document can't be processed, OCR converts the page into an image and scans it to ensure that the document is uploaded successfully.
<code>"conversions.json_normalizations": { ... }</code>	Moved to the Collections API .
<code>"conversions.pdf": { ... }</code>	Not available. If you used special parameters to extract text from images in PDFs, enable optical character recognition (OCR) from the product user interface for the collection that contains the PDFs instead.
<code>"conversions.segment": { ... }</code>	Not available programmatically. You can split a document at each occurrence of an SDU-generated field such as <code>subtitle</code> from the product user interface. The <code>segment_metadata</code> object with <code>parent_id</code> , <code>id</code> , and <code>total_segments</code> information is not available in v2. You can use the <code>metadata.parent_document_id</code> field to find the common parent for many document segments.
<code>"conversions.word": { ... }</code>	Not available

<code>"enrichments": { ... }</code>	<p>/v2/projects/{project_id}/enrichments, /v2/projects/{project_id}/collections/{collection_id}</p> <p>Use the enrichments API to explore existing enrichments. Use the collections API to see and change the enrichments that are enabled on a field in a collection.</p> <p>Some enrichments are applied to the service by default based on the type of project that you create. For more details, see Default project settings.</p> <p>The version of the Entities enrichment that is available in v2 doesn't include the disambiguation field, which in v1 contains the disambiguation information for the entity and includes the entity subtype information.</p> <p>The following enrichments are not available in v2:</p> <ul style="list-style-type: none"> • Categories • Concepts • Emotion • Relations • Semantic roles • Sentiment of Entities • Sentiment of Keywords 	
<code>"normalizations": [...]</code>	Moved to the Collections API .	
<code>"source": { ... }</code>	Not available. Configure connections to external data sources through the user interface. For more information, see Creating collections .	
Configuration setting details		
<h2>Collections</h2>		
Action	v1 API	v2 API
Create a collection	POST /v1/environments/{environment_id}/collections	POST /v2/projects/{project_id}/collections The supported parameters and responses differ between two versions. See the collection notes .
List collections	GET /v1/environments/{environment_id}/collections	GET /v2/projects/{project_id}/collections In v2, only the collection ID and name of each collection are returned in the list. You must use the Get collection method to return more details about each collection.
Get collection details	GET /v1/environments/{environment_id}/collections/{collection_id}	GET /v2/projects/{project_id}/collections/{collection_id} See the collection notes .
Update a collection	PUT /v1/environments/{environment_id}/collections/{collection_id}	POST /v2/projects/{project_id}/collections/{collection_id}
Delete a collection	DELETE /v1/environments/{environment_id}/collections/{collection_id}	DELETE /v2/projects/{project_id}/collections/{collection_id} In v2, the status field is not returned in the response.
List collection fields	GET /v1/environments/{environment_id}/collections/{collection_id}/fields v1 lists the fields per collection.	GET /v2/projects/{project_id}/fields v2 lists fields per project instead. You can pass a single collection ID with the collection_ids parameter to get fields from a single collection.
Collections API support details		

Collections API notes

The following table shows the important differences between the v1 and v2 collection APIs.

Method	Notes

Create a collection	The v2 response doesn't include the <code>status</code> and <code>configuration_id</code> fields. You can get status information for a specific document by using the Get document details method. The objects <code>disk_usage</code> , <code>training_status</code> , and <code>crawl_status</code> are not present in the response body in v2. The <code>document_counts</code> object is not present in the response body in v2 currently. Training status is returned in the Get project method response. The other information is not available in v2. In v2, you can define the enrichments to apply to the documents in the collection by specifying an optional <code>enrichments</code> object.
---------------------	---

Get collection details	The v2 response doesn't include the <code>status</code> and <code>configuration_id</code> fields. You can get status information for a specific document by using the Get document details method. The objects <code>document_counts</code> , <code>disk_usage</code> , <code>training_status</code> , and <code>crawl_status</code> are not present in the response body in v2. Training status is returned in the Get project method response. The other information is not available in v2. For example, you cannot get the document count for a collection and cannot get the crawl status for a collection that connects to an external data source in v2. In v2, you can get information about the enrichments that are applied to the collection.
------------------------	--

Update a collection	v2 uses <code>POST</code> instead of <code>PUT</code> . In v2, you can update the enrichments that are applied to the documents in the collection by specifying an optional <code>enrichments</code> object. The v2 response doesn't include the <code>status</code> and <code>configuration_id</code> fields.
---------------------	--

Collections API notes

Query modifications

The method that was available in v1 for configuring tokenization programmatically is not supported in the v2 API.

v1 API	v2 API
Tokenization dictionary API	Not available.
Expansions v1 API	Expansions v2 API
Stopwords v1 API	Stopwords v2 API

Query modifications API support details

Documents

Action	v1 API	v2 API
List documents	Not available from the v1 API	GET /v2/projects/{project_id}/collections/
Create a document	POST /v1/environments/{environment_id}/collections/{collection_id}/documents	POST /v2/projects/{project_id}/collections/{collection_id}/documents Unlike v1, the v2 response does not include information by using the Get document details method.
Update a document	POST /v1/environments/{environment_id}/collections/{collection_id}/documents/{document_id}	POST /v2/projects/{project_id}/collections/{collection_id}/documents/{document_id} When you update a document that was split into multiple segments, the segments are merged back into a single document.
Get document details	GET /v1/environments/{environment_id}/collections/{collection_id}/documents/{document_id}	GET /v2/projects/{project_id}/collections/{collection_id}/documents/{document_id} In v2, there is no <code>statusDescription</code> . v2 returns notices that are associated with the document ingestion.
Delete a document	DELETE /v1/environments/{environment_id}/collections/{collection_id}/documents/{document_id}	DELETE /v2/projects/{project_id}/collections/{collection_id}/documents/{document_id} Segments of an uploaded document cannot be deleted with a DELETE request that includes the <code>purge</code> parameter.

Documents API support details

v2 introduces a custom header that is named `X-Watson-Discovery-Force` that is not available in v1. You must include the header when you perform an operation on data that is shared across many collections to indicate that you want to perform the operation in each collection. If you do not include the header, a `403` error is returned.

Fields from JSON files that are added to a collection are converted differently during ingestion between v1 and v2. For more information about how JSON files are stored in the v2 index, see [JSON files](#).

Queries

Action	v1 API	v2 API
Query a collection	Supports a GET or POST request. GET or POST /v1/environments/{environment_id}/collections/{collection_id}/query	Queries a project. To specify a single collection, use the <code>{collection_id}</code> parameter. Supports a POST POST /v2/projects/{project_id}/query
Query multiple collections	GET or POST /v1/environments/{environment_id}/query	POST /v2/projects/{project_id}/query
Query system notices	GET /v1/environments/{environment_id}/collections/{collection_id}/notices	GET /v2/projects/{project_id}/collections/{collection_id}/notices
Query multiple collection system notices	GET /v1/environments/{environment_id}/notices	GET /v2/projects/{project_id}/notices
Get Autocomplete suggestions	/v1/environments/{environment_id}/collections/{collection_id}/autocomplete	GET /v2/projects/{project_id}/autocomplete See the query notes .

Documents API support details

Some query result configurations are applied to the service by default based on the type of project that you create. For more details, see [Default project settings](#).

Query notes

- v2 queries return results from all of the collections in the project. To restrict the query to use only certain collections within the project, use the `collection_ids` query parameter. You cannot query multiple collections that are added to different projects with one v2 query request.
- v2 results include a `confidence` field, but not a `score` field.

The confidence score replaced the score information in v1, but score was retained for backward compatibility. In v2, only the confidence field is returned.

- Use POST calls (instead of GET calls) to submit queries with v2.
- v1 queries accept many parameters. The [Query parameters comparison](#) table maps v1 parameters to v2 parameters.

v1 parameter	v2 parameter	Notes
N/A	collection_ids	Use this parameter in v2 to specify collection ids.
filter	filter	Same expression language.
query	query	Same expression language.
natural_language_query	natural_language_query	No notes.
passages	passages	The passage format changed and was enhanced in v2. The <code>passages:true</code> parameter changed to <code>passages.enable:true</code> . In addition to the <code>count</code> , <code>characters</code> , and <code>fields</code> options, you can specify <code>per_document</code> , which ranks the documents by document quality, and then returns the highest-ranked passages per document. You can also specify <code>find_answers</code> to return an answer object per passage, which contains a succinct answer to the query.

aggregation	aggregation	Same expression language.
count	count	No notes.
offset	offset	No notes.
return	return	No notes.
sort	sort	No notes.
highlight	highlight	If <code>passages.enabled</code> and <code>passages.per_document</code> are <code>true</code> , then passages are returned for each document instead of highlights.
spellingSuggestions	spellingSuggestions	No notes.
deduplicate	N/A	Not supported in v2.
similar	similar	The format changed in v2. The <code>similar:true</code> parameter changed to <code>similar.enable:true</code> . The <code>document_ids</code> and <code>fields</code> parameters changed from strings to string arrays. The <code>document_ids</code> parameter now is required if <code>enabled</code> is true.
bias	N/A	Not supported in v2.

Query parameters comparison

Training data

You can use the v1 training data API to work with two related objects:

- trained queries
- examples that are used to train the queries

These two objects have separate API endpoints in v1. In v2, the examples that are used to train each query are provided together with the query and only one endpoint is used to work with the training data.

For example, to add a trained query and its training example documents in v2, you use the request **POST** `/v2/projects/{project_id}/training_data/queries` and pass the query and all examples in the payload of one call. Similarly, if you want to update one example in the training set in v2, you must pass the query and the modified example (along with all of the other examples) to the v2 update endpoint. In v1, to update the example information, you use the update example endpoint to modify one example only.

Another important difference between v1 and v2 is that in v1, the trained model is associated with a particular collection. In v2, the trained model is associated with a project. You can use the data from multiple collections within a project to train a relevancy model. When you create or update training examples in v2, the API requires the `collection_id` for the collection where the document is stored.

Action	v1 API
List training data	GET /v1/environments/{environment_id}/collections/{collection_id}/training_data
Add query to training data	POST /v1/environments/{environment_id}/collections/{collection_id}/training_data
Delete all training data	DELETE /v1/environments/{environment_id}/collections/{collection_id}/training_data

Get details about a query	GET /v1/environments/{environment_id}/collections/{collection_id}/training_data/{query_id}
Delete a training data query	DELETE /v1/environments/{environment_id}/collections/{collection_id}/training_data/{query_id}
List examples for a training data query	GET /v1/environments/{environment_id}/collections/{collection_id}/training_data/{query_id}/examples
Add example to training data query	POST /v1/environments/{environment_id}/collections/{collection_id}/training_data/{query_id}/examples
Delete example for training data query	DELETE /v1/environments/{environment_id}/collections/{collection_id}/training_data/{query_id}/examples/{example_id}
Change label or cross-reference for example	PUT /v1/environments/{environment_id}/collections/{collection_id}/training_data/{query_id}/examples/{example_id}
Get details for a training data example	GET /v1/environments/{environment_id}/collections/{collection_id}/training_data/{query_id}/examples/{example_id}

Training data API support details

User data

The user data API is the same in v2 and v1.

Action	v1 API	v2 API
Delete	DELETE /v1/user_data	DELETE /v2/user_data Similar to v1. Use customer_id to delete the data associated with that customer ID.

User data API support details

Events and feedback

The v1 events and feedback API (`/v1/events`) is not available in v2.

Credentials

The v1 credentials API (`/v1/environments/{environment_id}/credentials`) is not available in v2. The function is available from the v2 product user interface.

Gateway configuration

The v1 gateways API (`/v1/environments/{environment_id}/gateways`) is not available in v2. The function is available from the v2 product user interface. For more information, see [Installing IBM Secure Gateway for on-premises data](#).

Status codes

For almost every API method, the status codes that are returned for v2 requests are different from the status codes that are returned for v1 requests.

Migration FAQ

Find answers to questions that are commonly asked about migrating from Discovery v1 to v2.

Do the two versions have all the same features?

There are many feature differences between the two versions. For a full feature comparison, see [Getting the most from Discovery](#).

How do I know which version I'm using now?

When you open the product user interface in v2, the following page is displayed:

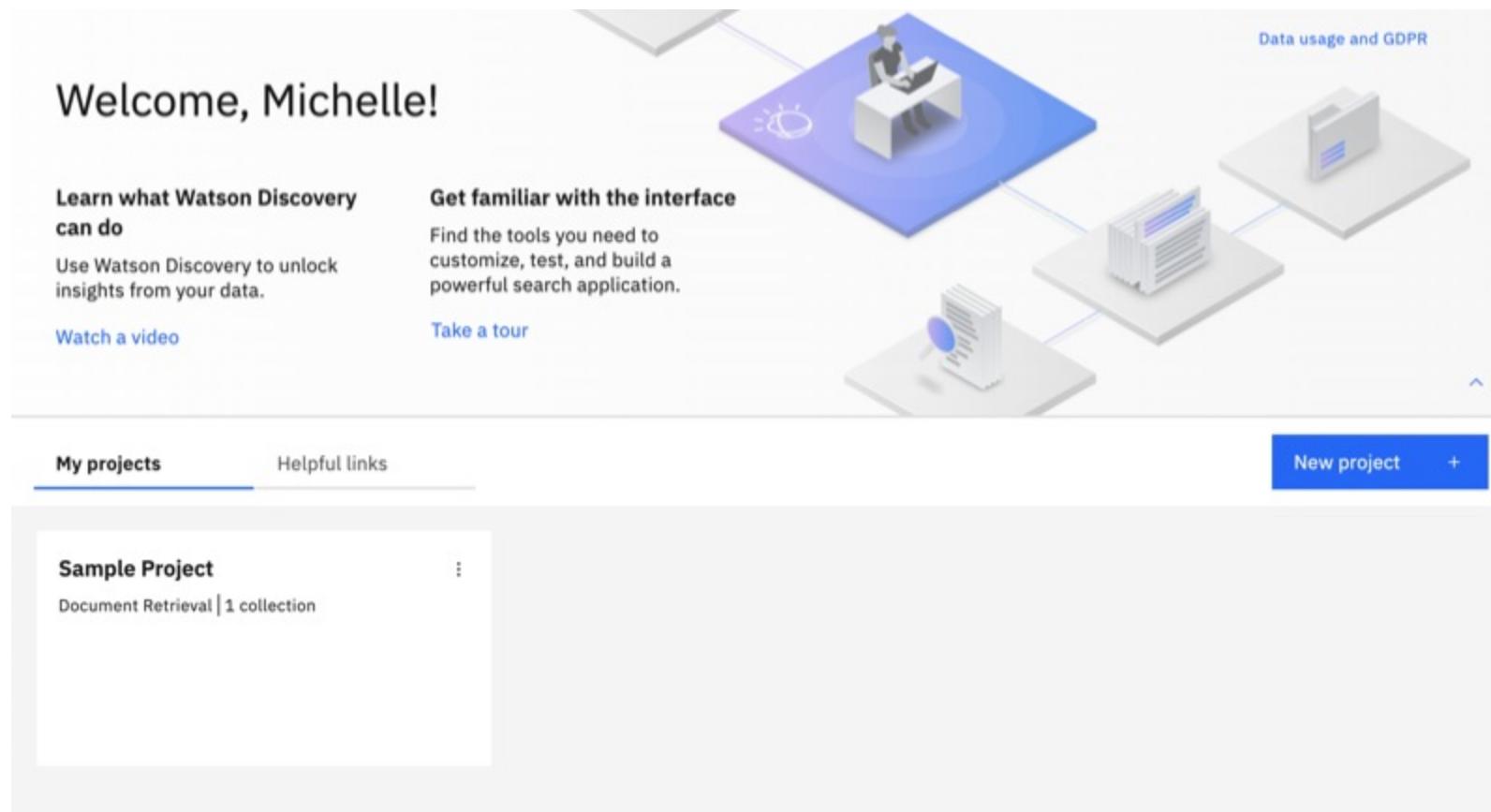


Figure 1. Home page from the Sample Project

How long will the migration take?

The time you need to set aside for the migration differs based on the amount of data you want to retain in your existing v1 service instance.

Do I need to update my existing applications for them to work with v2?

Yes. You will need to edit any existing applications to account for changes that are introduced with Discovery v2. For more information, see the [API version comparison](#).

To get started, see [Migrating to Discovery v2](#).

Migrating enrichments from Watson Explorer

If you have resources from IBM Watson Explorer, some of them can be migrated to IBM Watson® Discovery.

Types of resources that can be migrated

The following types of resources can be migrated from Watson Explorer to Discovery:

- From Watson Explorer Analytical Components: [User dictionaries](#)
- From Watson Explorer oneWEX: [Dictionaries](#), [character patterns](#), and [facets](#).

To analyze data with these migrated enrichments, you can use a Content Mining project. The tools in the associated Content Mining application are similar to tools that are available in Watson Explorer.

- For more information about how to create a Content Mining project, see [Creating projects](#).
- For more information about how to apply enrichments to a collection in the Content Mining application, see [Applying the annotator](#).

Importing dictionaries from Watson Explorer Analytical Components

You can import [user dictionaries](#) from IBM Watson Explorer Analytical Components.

 **Tip:** The default file location and name for dictionaries that are saved in Watson Explorer Analytical Components is `${primary_server_node}/{primary_configuration}/{collection_ID}/{dictionary_name}.fdic.xml`.

1. Download your user dictionaries from Watson Explorer Analytical Components.
2. From your Discovery Content mining project, open the Content Mining application.
3. From the analysis view of your collection, click the **Collections** link in the breadcrumb to open the *Create a collection for your analytics solutions* page of the Content Mining application.
4. To create an annotator, click **collection**, and then select **custom annotator** from the list.

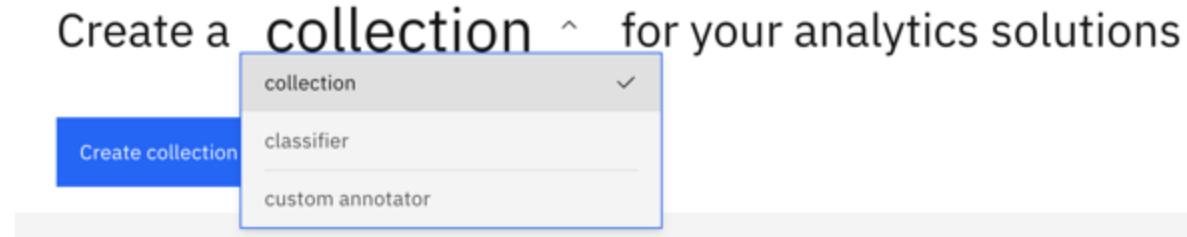


Figure 1. Collection menu

5. Click **Create custom annotator**.
6. Name your annotator, and then optionally add a description.
7. From the **Annotator Type** menu, select **Dictionary**, and then click **Next**.
8. Click the **Import** button, and then select the `{name}.fdic.xml` dictionary file that you want to import.
9. Click **Save**.

Uploading dictionaries from Watson Explorer oneWEX

You can import [dictionaries](#) from IBM Watson Explorer oneWEX.

1. From Watson Explorer oneWEX, Version 12.0.0 or later modifications or fix packs, download the dictionary CSV file.
 - Log in to the oneWEX administrator console.
 - Open the **Resource** tab.
 - Select dictionary enrichments, open the dictionary tab, and click the download icon. The dictionary is downloaded as a CSV file.
2. From your Discovery Content mining project, open the Content Mining application.
3. From the analysis view of your collection, click the **Collections** link in the breadcrumb to open the *Create a collection for your analytics solutions* page of the Content Mining application.
4. To create an annotator, click **collection**, and then select **custom annotator** from the list.

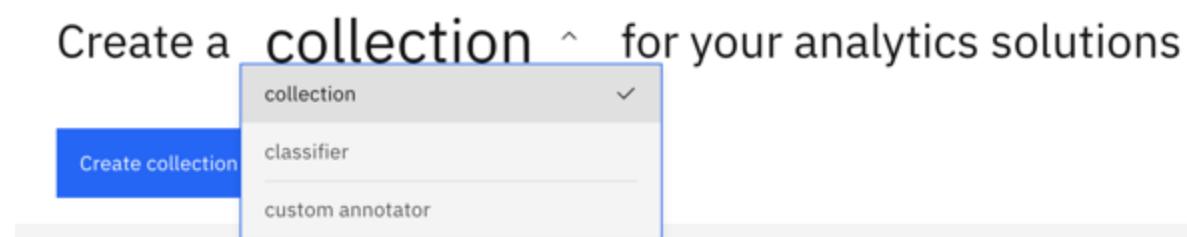


Figure 2. Collection menu

5. Click **Create custom annotator**.

6. Name your annotator, and then optionally add a description.
7. From the **Annotator Type** menu, select **Dictionary**, and then click **Next**.
8. Click the **Import** button, and then upload the CSV file of the dictionary that was downloaded from oneWEX.
9. Click **Import**, and then click **Save**.

Importing character patterns from Watson Explorer oneWEX

You can import [character patterns](#) from IBM Watson Explorer oneWEX.

1. From your Discovery Content mining project, open the Content Mining application.
2. From the analysis view of your collection, click the **Collections** link in the breadcrumb to open the *Create a collection for your analytics solutions* page of the Content Mining application.
3. To create an annotator, click **collection**, and then select **custom annotator** from the list.

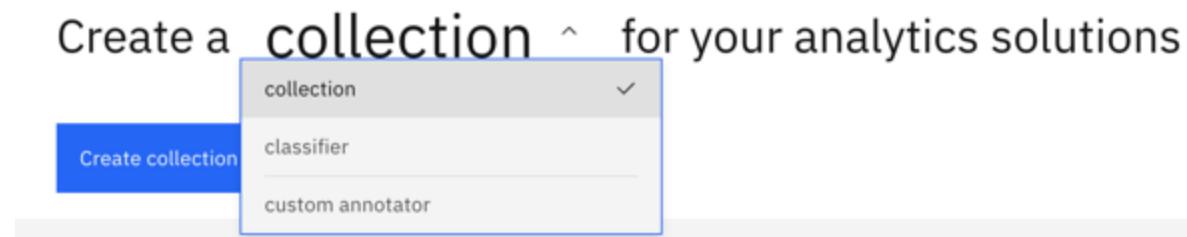


Figure 3. Collection menu

4. Click **Create custom annotator**.
5. Name your annotator, and then optionally add a description.
6. From the **Annotator Type** menu, select **Regular expression**, and then click **Next**.
7. Click the **Import** button.
8. Select the JSON file that you want to import, and then click **Save**.

Importing facets from Watson Explorer Content Analytics Studio IBM Cloud Pak for Data

 **Note:** You can import a PEAR file to use as the machine learning source file from IBM Cloud Pak for Data deployments only.

You can show Content Analytics Studio facets in the Content Mining application. Only facets with a UIMA Feature of type **Literal Value** are displayed.

For more information about how to import Content Analytics Studio machine learning models for use in other project types, see [Use imported ML models to find custom terms](#).

1. From the Watson Explorer Content Analytics Studio, export the machine learning model that defines the facets that you want to use. The model file must have a **.pear** extension.
 2. In the export configuration, remove the facet path, but keep the subfacet value. Set the Index Field name to the Facet Tree Path in Content Analytics Studio.
- For more information, see [Creating Custom PEAR Files for Use with Lexical Analysis Streams](#).
3. From your Discovery Content mining project, open the Content Mining application.
 4. From the analysis view of your collection, click the **Collections** link in the breadcrumb to open the *Create a collection for your analytics solutions* page of the Content Mining application.
 5. To create an annotator, click **collection**, and then select **custom annotator** from the list.

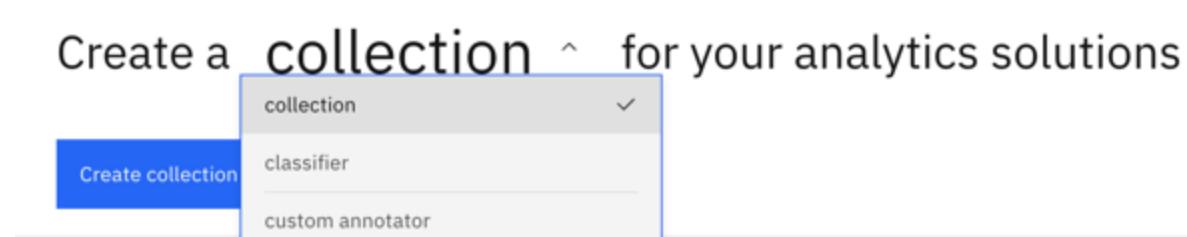


Figure 4. Collection menu

6. Click **Create custom annotator**.

7. Name your annotator, and then optionally add a description.
8. From the **Annotator Type** menu, select **PEAR File**, and then click **Next**.
9. Click **Select file** to find the .pear file that you exported.
10. Specify a facet path, and then click **Save**.

Migrating Knowledge Studio solutions

Use custom models and other resources that you created in Knowledge Studio by migrating them to Discovery.

Using a model as is

To start using your Knowledge Studio model immediately, export the model from Knowledge Studio and import it to Discovery as a machine learning enrichment.

When you import a Knowledge Studio model to use as is in Discovery, root-level entity types that were defined in the model can be recognized when they occur in your documents. Any mentions of entity subtypes that occur are identified as mentions of the parent entity type. The subtype entities themselves are not preserved. If you want the model to continue to distinguish between different subtypes of an entity, you must take extra steps. For more information, see [Retaining subtype information](#).



Note: You cannot continue to update a model that you import as an ML enrichment.

The following types of models can be imported and used as is:

- Rule-based models created in Knowledge Studio that find entities in documents based on rules that you define. (File format: .pear)
- Machine learning models created in Knowledge Studio that understand the linguistic nuances, meaning, and relationships specific to your industry (file format: .zip)

The models that you can add depend on your deployment type:

- IBM Cloud You can add models that were created with a IBM Watson® Knowledge Studio instance that is hosted in IBM Cloud only.
- IBM Cloud Pak for Data You can add models that were created with an instance of IBM Watson® Knowledge Studio that is hosted on IBM Cloud Pak® for Data or IBM Cloud.

For more information, see [Using imported ML models to find custom terms](#).

Using a corpus as training data

Discovery has an entity extractor tool that you can use to define a type system. The entity extractor user interface is similar to the Knowledge Studio user interface that is used to annotate documents that you add to corpus for a machine learning model. However, in Knowledge Studio, you define root-level entities only, not subtypes or relationships.

As an alternative to importing a Knowledge Studio model as is and applying it as an enrichment, you can also import a Knowledge Studio corpus. When you add a Knowledge Studio corpus to the Discovery entity extractor tool, any root-level entities from the corpus are represented as new entities in the Discovery entity extractor workspace. Entity subtypes are not recognized. Although, you can take extra steps to [retain subtype information](#).

Relations and coreferences from the Knowledge Studio machine learning model are not represented, neither are any custom dictionaries that are associated with the model.

Things to consider when choosing whether to import a model or import a corpus:

- You can continue to edit the type system when you import the corpus. When you import a trained model, you cannot subsequently edit it in Discovery.
- An imported model that you apply to a collection as an enrichment can recognize any entity subtype, relation, and coreference information that the original model was trained to recognize in addition to root-level entities. An entity extractor enrichment can find and tag entities only.

For more information, see [Importing a Knowledge Studio corpus](#).

Retaining subtype information

When you import a Knowledge Studio model to Discovery, any subtypes that were defined in the model are identified as mentions of the parent entity type. The subtype entities themselves are not preserved. To retain the subtype information, you must **flatten** your type system by converting entity subtypes into new root-level entity types.

Follow these steps only if you are sure that the subtype distinctions add significant value to the model. In many use cases, using the root-level entity types is sufficient.

⚠️ Important: You cannot use this procedure to retain subtypes if any of the documents in your corpus were pre-annotated with the Natural Language Understanding service. Make sure that your flattened type system doesn't surpass the allowed number of entity types for your plan. For more information, see [Entity extractor limits](#).

For example, your model might have entity types with the following hierarchy:

```
APPLIANCES
FURNITURE
  PATIO
  LIVING
  DINING
```

A flattened version of the type system looks like this:

```
APPLIANCES
FURNITURE_NONE
FURNITURE_PATIO
FURNITURE_LIVING
FURNITURE_DINING
```

A useful approach for flattening the type system involves the following changes:

- Add the parent entity type label (**FURNITURE**) as a prefix to the label of each child subtype to produce a new root-level entity that preserves the hierarchical relationship in its label. For example, **FURNITURE_PATIO**, **FURNITURE_LIVING**, and **FURNITURE_DINING**.
- Append the word **NONE** to the parent root-level entity label to identify it as the parent. For example, **FURNITURE_NONE**.
- Leave the labels of entity types that don't have subtypes unchanged. For example, the label **APPLIANCES** doesn't change.

To retain entity subtype information, complete the following steps:

1. Ensure that the annotation and training of the Knowledge Studio model is completed and the model is ready to be deployed.
2. Export the type system that was used to annotate the documents in your corpus from Knowledge Studio as a .json file.

Follow the appropriate steps for exporting based on your Knowledge Studio deployment type:

- IBM Cloud [Uploading resources from another workspace](#)
- IBM Cloud Pak for Data [Uploading resources from another workspace](#)

3. Modify the type system JSON file. For each subtype, add a new root-level entity type.

For example, the original type system might contain the following types:

```
{
  "id": "b9d6caa2-90ac-47ff-91f6-2149b8ffcf20",
  "label": "FURNITURE",
  "sireProp": {
    "mentionType": null,
    "subtypes": ["PATIO", "LIVING", "DINING"],
    "roles": ["b9d6caa2-90ac-47ff-91f6-2149b8ffcf20", "93ba1f27-173f-4714-b31e-77bdd8cb9932"],
    "clazz": null,
    "color": "black",
    "hotkey": "m",
    "backGroundColor": "#00FFFF",
    "active": true,
    "roleOnly": false},
    "creationDate": 1610611788484,
    "source": null,
    "modifiedDate": 0,
    "typeType": null,
    "typeClass": null,
    "typeVersion": null,
    "typeDesc": null,
    "typeSuperType": null,
    "typeSuperTypeID": null,
    "typeCreateDate": null,
    "typeUpdateDate": null,
    "typeProvenance": null,
    "alchemyAPITypes": null,
    "nluAPITypes": null},
```

To convert the subtypes to new root-level types, make the following change:

```
{
  "id": "b9d6caa2-90ac-47ff-91f6-2149b8ffcf20",
  "label": "FURNITURE_NONE",
```

```

"sireProp": {
    "mentionType":null,
    "subtypes":null,
    "roles":["b9d6caa2-90ac-47ff-91f6-2149b8ffcf20","93ba1f27-173f-4714-b31e-77bdd8cb9932"],
    "clazz":null,
    "and so on"
}
},
{
    "id":"b9d6caa2-90ac-47ff-91f6-2149b8ffcf20",
    "label":"FURNITURE_PATIO",
    "sireProp": {
        "mentionType":null,
        "subtypes":null,
        "roles":["b9d6caa2-90ac-47ff-91f6-2149b8ffcf20","93ba1f27-173f-4714-b31e-77bdd8cb9932"],
        "clazz":null,
        "and so on"
    }
},
{
    "id":"b9d6caa2-90ac-47ff-91f6-2149b8ffcf20",
    "label":"FURNITURE_LIVING",
    "sireProp": {
        "mentionType":null,
        "subtypes":null,
        "roles":["b9d6caa2-90ac-47ff-91f6-2149b8ffcf20","93ba1f27-173f-4714-b31e-77bdd8cb9932"],
        "clazz":null,
        "and so on"
    }
},
{
    "id":"b9d6caa2-90ac-47ff-91f6-2149b8ffcf20",
    "label":"FURNITURE_DINING",
    "sireProp": {
        "mentionType":null,
        "subtypes":null,
        "roles":["b9d6caa2-90ac-47ff-91f6-2149b8ffcf20","93ba1f27-173f-4714-b31e-77bdd8cb9932"],
        "clazz":null,
        "and so on"
    }
},

```

4. Assign a unique ID to each new root-level entity type.

5. Export the corpus for your machine learning model from Knowledge Studio as a compressed file.

Follow the appropriate steps for exporting based on your Knowledge Studio deployment type:

- IBM Cloud [Uploading resources from another workspace](#)
- IBM Cloud Pak for Data [Uploading resources from another workspace](#)

6. In the downloaded corpus, for all mentions with a subtype defined, update the type information for the mention to specify the new root-level entity type.

For example, the original type system might include the **PATIO** subtype mention:

```

{
    "id" : "Blogs_shopper.com_dc5cf4764d91f87575b17ac8a5268462.en-M92",
    "source" : "IMPORT",
    "properties" : {
        "SIRE_ENTITY_CLASS" : "SPC",
        "SIRE_MENTION_CLASS" : "SPC",
        "SIRE_ENTITY_LEVEL" : "NONE",
        "SIRE_ENTITY_SUBTYPE" : "PATIO",
        "SIRE_MENTION_ROLE" : "FURNITURE",
        "SIRE_MENTION_TYPE" : "NONE"
    },
    "type" : "FURNITURE",
    "begin" : 3221,
    "end" : 3234,
    "inCoref" : false
},

```

Replace the value of the **SIRE_MENTION_ROLE** and **type** for the mention with the new root-level entity label, such as **FURNITURE_PATIO**. Specify **NONE** as the **SIRE_ENTITY_SUBTYPE** value.

```

{
    "id" : "Blogs_shopper.com_dc5cf4764d91f87575b17ac8a5268462.en-M92",
    "source" : "IMPORT",

```

```

"properties" : {
    "SIRE_ENTITY_CLASS" : "SPC",
    "SIRE_MENTION_CLASS" : "SPC",
    "SIRE_ENTITY_LEVEL" : "NONE",
    "SIRE_ENTITY_SUBTYPE" : "NONE",
    "SIRE_MENTION_ROLE" : "FURNITURE_PATIO",
    "SIRE_MENTION_TYPE" : "NONE"
},
"type" : "FURNITURE_PATIO",
"begin" : 3221,
"end" : 3234,
"inCoref" : false
},

```

Don't forget to rename the parent mention labels.

For example, find mentions that specify `"SIRE_ENTITY_SUBTYPE" : "OTHER"`, and then change the value from `OTHER` to `NONE`.

Change the value of the `SIRE_MENTION_ROLE` and `type` for the mention to the new parent entity type label.

For example, change the `SIRE_MENTION_ROLE` and `type` values for these mentions from `FURNITURE` to `FURNITURE_NONE`, and the `SIRE_ENTITY_SUBTYPE` to `NONE`.

```

{
    "id" : "Sports_herald.com_be99aca94a7cff5abb74476b844a11b6.en-M75",
    "source" : "IMPORT",
    "properties" : {
        "SIRE_MENTION_CLASS" : "SPC",
        "SIRE_ENTITY_LEVEL" : "NONE",
        "SIRE_ENTITY_SUBTYPE" : "NONE",
        "SIRE_ENTITY_CLASS" : "SPC",
        "SIRE_MENTION_TYPE" : "NONE",
        "SIRE_MENTION_ROLE" : "FURNITURE_NONE"
    },
    "type" : "FURNITURE_NONE",
    "begin" : 2063,
    "end" : 2071,
    "inCoref" : false
},

```

7. Add annotations for relationships that are missing based on the new flattened entity types.

8. Create a Knowledge Studio workspace, and then upload the converted type system.

Follow the appropriate steps for uploading a type system based on your Knowledge Studio deployment type:

- IBM Cloud [Adding a type system to the workspace](#)
- IBM Cloud Pak for Data [Adding a type system to the workspace](#)

9. Upload the annotated documents to the workspace. Retain the original file structure of the exported data. Ensure that the compressed file has the same root-level directory as the original exported file, for example.

Follow the appropriate steps for uploading documents based on your Knowledge Studio deployment type:

- IBM Cloud [Adding documents to a workspace](#)
- IBM Cloud Pak for Data [Adding documents to a workspace](#)

10. From Knowledge Studio, click **Train** to retrain the model.

For more information, see the appropriate topic for your deployment type:

- IBM Cloud [Training the machine learning model](#)
- IBM Cloud Pak for Data [Training the machine learning model](#)

11. Now, you're ready to export the model from Knowledge Studio and import it to Discovery to use the model as a machine learning enrichment.

For more information, see [Using imported ML models to find custom terms](#).

Guided tours

You can explore the Discovery interface by taking an interactive tour. Click **Guided tours** from the header of the *My Projects* page to select a tour.

The following tours are available:

- Learn the essentials of Watson Discovery
- Extract meaning from text
- Create and apply dictionaries
- Create a Content Mining project
- Learn the keys to a successful mining setup

Release notes

Release notes for Discovery for IBM Cloud

Learn about features and changes that were included for each release and update of the product software.

IBM Cloud



Note: This information applies only to managed instances of IBM Watson® Discovery that are hosted on IBM Cloud or that were provisioned with [IBM Cloud Pak for Data as a Service](#). For information about releases and updates for installed deployments, see [Release notes for IBM Watson® Discovery Cartridge for IBM Cloud Pak® for Data](#).

21 September 2023

Updated the tokenizer for all languages

The updated tokenizer might affect the ranking order of results for certain queries. If you observe any ranking differences in your query results, you can reindex the documents in the collection. Discovery tokenizes words both when it ingests and stores data in the index, and at run time when it analyzes queries that are submitted by users. By reindexing the collection, you ensure that your documents are indexed with the same tokenizer that is used for matching queries.

To reindex documents, open the **Manage collection** page, choose a collection, and navigate to the **Enrichments** tab. Select a field to enrich, and then clear the field. Next, click **Apply changes and reprocess** and wait for the documents in the collection to be reprocessed.

15 August 2023

Option to apply or remove a crawl schedule

This option is helpful for easily applying or removing a crawl schedule, and also for stopping a crawl. For more information, see [Crawl schedule options](#).

9 August 2023

You can now specify fields from which to extract content when querying data from the UI

The ability to specify fields allows you to improve the search results when content is not indexed in the default fields. Content might not be indexed in the default fields when you ingest structured files or when you apply a Smart Document Understanding model. For more information, see [Excerpt unavailable](#).

Enrichments in the advanced document view for PDFs are highlighted in distinct colors

When you select multiple enrichments in the advanced document view for PDFs, each enrichment type is highlighted in the document with distinct colors. Overlapping enrichments are also highlighted in a distinct color.

26 July 2023

You can now specify a custom date and time for the crawl schedule

This option is helpful if you want to avoid heavy load on a target system during business hours. For more information, see [Crawl schedule options](#).

10 June 2023

All Entities enrichments use the Entities v2 type system

Natural Language Understanding Entities v1 is no longer supported. IBM Cloud instances that were created before 2 June 2021 and Discovery for IBM Cloud Pak for Data 2.x deployments used version 1 of the Natural Language Understanding Entities type system for English and Korean collections. Now, all collections use only version 2 of the Natural Language Understanding Entities type system.

Classifiers are identified more clearly

The [Enrichments](#) page lists classifier enrichments as either **text classifier** or **document classifier** enrichments.

16 May 2023

Improved tool for creating Smart Document Understanding (SDU) user-trained models

The SDU tool that you use to annotate documents when you create a user-trained SDU model now uses the React UI framework. This update does not change the behavior of the tool, but does make it more responsive.

You can now define JSON normalizations by using the Collections API

The **Create a collection** and **Update a collection** methods now support the addition of **conversions** and **normalizations** objects that you can specify to apply normalization operations to the documents in the collection. For example, you can define an operation to copy or merge one field to another in the JSON representation of the documents. The **conversions** object defines normalization operations that occur during ingestion and the **normalizations** object defines normalization operations that occur after enrichments are applied. For more information, see the [Collections API reference](#).

31 March 2023

Update to API version

The current API version (v2) is now 2023-03-31. One change was made with this version.

Changed how fields named **document_id** are handled

If you add a JSON file that contains a field named **document_id** to a collection, the field is ignored. The system assigns a new unique document ID to the document when it is added to the index. To assign a document ID to a document regardless of its file type, use the **Update document** method from the API.

Previously, when you uploaded a JSON file with a field named **document_id** from the product user interface or by using the **Add document** API method, the document ID from the file was shown as the **document_id** value in query results. However, a different document ID was assigned to the document, and the assigned ID had to be used for certain other tasks, such as deleting the document. If your application relies on the previous behavior, specify a version number earlier than 2023-03-31, such as **2020-08-30**, in your API calls.

2 March 2023

Now you can specify the types of files to add to a collection

When you connect to an external data source, you can limit the types of files to add to the collection from the external data source. For example, you can choose to add only PDF files from a Box data source.

21 February 2023

Optical character recognition v2 technology is used

The latest version (OCR v2) is used automatically when you enable OCR for English, German, French, Spanish, Dutch, Brazilian Portuguese, and Hebrew collections in all IBM Cloud service plans.

The new optical character recognition model was developed by IBM Research to be better at extracting text from scanned documents and other images that have the following limitations:

- Low quality images due to incorrect scanner settings, insufficient resolution, bad lighting (such as with mobile capture), loss of focus, unaligned pages, and badly printed documents
- Documents with irregular fonts or a variety of colors, font sizes, and backgrounds

The entity extractor limits changed

The number of documents that are allowed in the training data for the Plus plan increased from 100 to 200.

The number of entity types that you can create per plan decreased.

- For Premium plans, the limit changed from 75 to 18.
- For Enterprise plans, the limit changed from 50 to 18.
- For Plus plans, the limit changed from 20 to 12.

The string variation operator now works with phrases

When you include the string variation operator with query input that contains a phrase, the variation is applied to each word in the phrase. For example, `"tom cat"~1` matches `top hat` in addition to `tom cat`. For more information about Discovery Query Language operators, see [Query operators](#).

10 February 2023

Entity extractor is generally available

The **Extract entities** enrichment brings the powerful ability to build a custom type system into Discovery. Use the tool to label entity examples within your industry data to build a machine learning model that Discovery can use to recognize meaningful terms for your business. Already built an entity type system in Knowledge Studio? You can use the corpus from Knowledge Studio as a starting point for your Discovery entity extractor training data. For more information, see [Entity extractor](#).

If you created an entity extractor enrichment for testing purposes when the feature was in beta release, now that it is generally available, it will count toward your custom model limit. The entity extractor enrichment incurs charges whether or not it is applied to a collection.

7 February 2023

Support for hourly crawls was removed

You can no longer choose to crawl a data source every hour. If an existing collection is configured to crawl hourly, you will be prompted to change the scheduled crawl the next time you edit the connector settings.

You can no longer enable FAQ extraction for a collection

The checkbox to enable or disable the beta FAQ extraction feature was removed. FAQ extraction was a beta feature that captured question-and-answer pairs from the data source as it was crawled. FAQ extraction generated a new subdocument for each pair and stored the question in the `title` field and the answer in the `text` field.

You cannot apply FAQ extraction to new collections.

Any existing collections with FAQ extraction enabled retain FAQ documents in their indexes until the collection is reprocessed. At that time, most of the question-and-answer pair subdocuments are deleted. However, any FAQ subdocuments that were generated from HTML or TXT source files remain. If you want to remove these subdocuments, go to the **Manage data** page to delete them. Subdocuments that are generated from one parent document all have the same `metadata.parent_document_id` value.

If you need a way to extract question-and-answer pairs from source documents that use a consistent style and formatting for questions and answers, you can use the Smart Document Understanding tool to annotate the pairs instead. For more information, see [Using Smart Document Understanding](#).

25 January 2023

Set up a Microsoft SharePoint Online data store connector that has **Read** permission

When you create a Microsoft SharePoint Online connector to crawl a SharePoint data source by using Open Authentication v2, the enterprise application that is created by Discovery to make the connection requires **Read** permission only. The enterprise application that was configured for you previously required **Write** permission.

If you want to update an existing connector so that you can use the new Read permission configuration, you must delete your existing enterprise application first.

For more information, see [Microsoft SharePoint Online connector](#).

FAQ extraction deprecation announcement

The beta FAQ extraction feature that detects and extracts question-and-answer pairs from documents is being removed. Support for the feature will end in 1Q 2023.

6 December 2022

Now you can stop a data source crawl

You can stop a crawl that is in progress or that is scheduled to occur in the future. For more information, see [Stopping a crawl](#).

The following item is a known issue:

Box data source scheduled crawls are not updating documents

Due to a problem in the Box Events API, changes that occur between crawls in documents that are stored in Box are not detected and picked up by the Discovery collection during scheduled recrawls. To ensure that your collection is up-to-date, stop and restart the crawl.

1 December 2022

Plus plan supports fewer entity extractors

The maximum number of entity extractors that you can create with a Plus plan decreased from 6 to 3.

12 November 2022

Discovery users might experience issues with documents in collections where OCR is enabled that were added or processed between Nov 1 and Nov 11

Between 1 November and 11 November 2022, some projects with optical character recognition (OCR) enabled, including Document Retrieval for Contracts projects, experienced problems. The problems were related to a new version of the optical character recognition (OCR v2) feature that was enabled automatically for English, German, French, Spanish, Dutch, Brazilian Portuguese, and Hebrew collections during that timeframe. The new version changes sentence boundaries in ways that can negatively impact other functions, including element identification in contracts and the document labeling view in the entity extractor tool.

If you experience any of these issues with documents that were added or processed during this period, revert the version of OCR that is applied to the documents. Starting on 12 November 2022, OCR v1 is applied to all collections where OCR is enabled. To go back to using OCR v1, make a change that will reprocess the affected documents. For example, you can re-add documents that were added during the timeframe to reprocess them. Or you can reprocess an entire collection.

To reprocess a collection, from the **Manage collections** page, open the collection, and then go to the **Processing settings** tab.

Expand the **More processing settings** section, set the OCR switch to **Off**, and then set it back to **On**. Click **Apply changes and reprocess** to reprocess your collection.

2 November 2022

A new and improved optical character recognition technology is available

A new version of optical character recognition technology is now available. This latest version (OCR v2) is used automatically when you enable OCR for English, German, French, Spanish, Dutch, Brazilian Portuguese, and Hebrew collections in all IBM Cloud service plans. The new optical character recognition model was developed by IBM Research to be better at extracting text from scanned documents and other images that have the following limitations:

- Low quality images due to incorrect scanner settings, insufficient resolution, bad lighting (such as with mobile capture), loss of focus, unaligned pages, and badly printed documents
- Documents with irregular fonts or a variety of colors, font sizes, and backgrounds

1 November 2022

Entity extractor loads the first 40,000 characters from training data documents

Even extra long documents from the collection that you use to define custom entity examples are loaded into the document view of the tool. However, only the first 40,000 characters, which is approximately 15-20 pages, are displayed. The rest of the file content is truncated. You'll know if your document is truncated because a notification is displayed in the document view. For more information, see [Entity extractor](#).

You can set the passages per document setting to be higher than one

A bug was fixed that prevented you from using the search bar settings in the product user interface to increase the maximum number of passages to return per document. For more information, see [How passages are derived](#).

Improved query aggregation documentation

The documentation that describes the aggregation types that you can specify in the query aggregation parameter was updated. For more information, see [Query aggregations](#).

30 September 2022

Lite plans are no longer available from the London data center

Lite plans are discontinued. You cannot create **new** service instances that use the Lite plan type in any location, including London. Use the new Plus plan and its associated 30-day free trial to explore new features and a simpler way to build that is available with the latest version of the product.

22 September 2022

Plus plan supports more entity extractors

The maximum number of entity extractors that you can create with a Plus plan increased from 3 to 6.

You cannot apply a Smart Document Understanding model to Microsoft Excel files

The quality of structural analysis that can be produced for Excel files is not sufficient. Starting on 22 September 2022, you cannot apply an SDU model to Excel files. This change does not impact Excel files in collections where an SDU model was applied before 22 September 2022.

16 September 2022

In-context document preview is now available for PDF files that are crawled

When you click to view a passage from a search result that is extracted from a PDF document, a document preview page is displayed that shows the returned passage in the context of the original PDF page. The in-context view is available for PDF files to which a Smart Document Understanding model is applied.

15 August 2022

SDKs were updated to reflect the latest API changes.

The following [Discovery v2 API](#) changes are now reflected in the SDKs:

- Use the new document classifier API to get, add, update, or delete a document classifier.
- A new document status API is available. You can use it to get a list of the documents in a collection and to get details about a single document.
- You can now get, add, and remove a stop words or expansion list for a collection.
- A `smart_document_understanding` field is returned with the **Get collection** method. This new field specifies whether an SDU model is enabled for the collection and indicates the model type.
- A `similar` parameter is available from the **Query** method. Use it to find documents that are similar to documents of interest to you.
- The `suggested_refinements` parameter of the **Query** method is deprecated. The `suggested_refinements` parameter was used to identify dynamic facets from Premium plan data.

8 August 2022

Larger documents can be crawled

The maximum file sizes that are allowed for crawled documents increased for Premium plans. It also increased for the Box, IBM Cloud Object Storage, and Salesforce connectors. For more information, see [File size limits](#).

2 August 2022

IAM authentication support was added to the IBM Cloud Object Storage connector

You can now choose to authenticate with the IBM Cloud Identity and Access Management (IAM) service. For more information, see [IBM Cloud Object Storage](#).

28 July 2022

API updates

The following changes were made to the [Discovery v2 API](#).

New fields are available:

- A `smart_document_understanding` field is returned with the **Get collection** method. This new field specifies whether an SDU model is enabled for the collection and indicates the model type.
- A `similar` parameter is available from the **Query** method. Use it to find documents that are similar to documents of interest to you.

The `suggested_refinements` parameter of the `Query` method is deprecated. The `suggested_refinements` parameter was used to identify dynamic facets from Premium plan data.

Discovery v1 deprecation announcement

- ⊖ **Deprecated:** Watson Discovery v1 is being deprecated. Existing clients who use Watson Discovery v1 are asked to migrate to Watson Discovery v2 before the end-of-support date of **11 July 2023**. End of Support means that no v1 instance will work on or after 11 July 2023. For more information about migration, see [Getting the most from Discovery](#).

11 July 2022

The advanced document view highlights even more enrichments

In addition to the built-in **Entities** and **Keywords** enrichments that are recognized by Watson Natural Language Processing models, the advanced document view now highlights the following types of enrichments:

- Custom dictionary terms
- Terms or numbers that match regular expression patterns that you define
- Custom entities and relationships that are defined by Watson Knowledge Studio machine learning and rules-based models
- Custom entities that are defined by using the entity extractor tool that is available as a beta feature

For more information about enrichments that you can add to your documents, see [Adding domain-specific resources](#).

30 June 2022

Watson SDK support change

Support for the following SDKs is provided by the Watson community of developers instead of IBM:

- Go
- Ruby
- Swift
- Unity

For more information, see [Watson SDKs](#).

1 June 2022

The entity extractor tool is now easier to use

The user interface was redesigned to better support the workflow of adding entity types and labeling examples of them. As part of the new design, the bulk labeling feature now is enabled by default, the documents view is easier to find and use, the suggestions pane is more responsive, and you can track metrics scores across multiple training runs. For more information about the entity extractor, see [Customizing the terms that Discovery can recognize](#).

The entity extractor is now available in more plans and languages

The entity extractor beta feature is now available to users of Plus and Enterprise plans in addition to Premium plans. The extractor enrichment is supported for collections in languages other than English.

When you remove a starting URL from a Web crawl connector its associated documents are deleted

The Web crawl connector was updated. Starting with collections that you create after April 2022, if you remove a starting URL from the Web crawl configuration, any indexed documents that were derived from the content of the web page at that URL are deleted with the next crawl. For more information, see [Web crawl](#).

16 May 2022

Added API methods for working with stop words and expansion lists

You can now get, add, and remove a stop words or expansion list for a collection programmatically. For more information, see the [Query modifications](#) methods.

13 May 2022

An improved JSON view is available

You can now use keyboard keys to tab through elements in the view. The new JSON view also numbers the occurrences of elements in each JSON object, which makes it easier to keep track of information and to read totals at a glance.

20 April 2022

Analyze API is supported in Enterprise plan deployments

Use the Analyze API to process a JSON file according to a collection's configuration settings, and then return the file for realtime use without storing it in the collection. The Analyze API was supported only in installed deployments previously. For more information, see [Analyze API](#).

A new document status API is available

Use the new document status API to programmatically get a list of the documents in a collection and to get details about a single document. The following notes apply to this release:

- The API is supported for collections that are created after 23 March 2022.
If you want to get status information about a collection that was created earlier, trigger a process that runs the conversion step of ingestion on the documents. For example, you can enable the API by making changes in the **Identify fields**, **Manage fields**, **CSV settings**, or **Processing settings** (such as OCR or FAQ extraction settings) pages, or by applying a Smart Document Understanding model to the older collection.
- The API is available only from Plus and Enterprise plan instances.

For more information about the new API, see the [API reference documentation](#).

More messages are shown to keep you informed about the status of document processing

An issue was fixed which previously prevented informative messages from being displayed about the status of document conversion and indexing during the ingestion process. Now that the issue is fixed, you might see more messages than usual when you add or reprocess documents. This increase is expected. Nothing you did caused the increase in messages.

6 April 2022

Project tile has a more intuitive menu

The project tile was updated to include an overflow menu that you can use to perform actions such as deleting or renaming a project.

30 March 2022

A new document classifier API is available

Use the new document classifier to programmatically get, add, update, or delete a document classifier. Document classifier methods are supported on installed instances (IBM Cloud Pak for Data) or IBM Cloud-managed Premium or Enterprise plan

instances.

For more information about the new API, see the [API reference documentation](#). For more information about adding a document classifier by using the product user interface, see [Classifying documents](#).

21 March 2022

Visualize enrichments found in your documents

When you click to view the passage from a search result, a document preview page is displayed that shows a representation of the original document where the search result was found. For most document types, you can open a new **advanced view** of the document to see useful summary information, such as the number of occurrences of any enrichments that are detected in the document. You also can select one of the enrichments to highlight every occurrence of the element within the document text.



Note: Currently, only the **Entities** and **Keywords** enrichments are listed.

Improved format of search results from PDF documents

When you click to view a passage from a search result that is extracted from a PDF document, a document preview page is displayed that shows the returned passage in the context of the original PDF page.



Note: The in-context view is available for PDF files to which a Smart Document Understanding model is applied. The rich preview does not work on images, meaning it doesn't work on scanned PDF documents. The in-context view is available for PDFs in all languages; however, the enrichment highlighting might be misaligned in some languages.

Tell us what you think

Share your opinions and ideas with us at any time by clicking the **Share feedback** button from the page header of the product user interface.

10 March 2022

Manage the data in a collection from the new **Manage data** page

You can now access the **Manage data** page for a collection from the **Manage collections** navigation pane. Go there to see a list of the documents in your collection and get a quick view of information about the documents. You can also delete documents from a collection with just a few clicks. For more information, see [Excluding content from query results](#).

15 February 2022

An alternative authentication mechanism is available for Microsoft Sharepoint Online connectors

You can now use Open Authentication to sign in to Microsoft SharePoint directly when you configure a new IBM Cloud connector. The **Sign in with Microsoft** option that uses Open Authentication to authenticate with the external data source is a beta feature. For more information, see [Microsoft SharePoint Online](#).

7 January 2022

Upgrade from Plus to Enterprise without help

You can perform an in-place upgrade from a Plus plan to an Enterprise plan. For more information, see [Upgrading](#).

6 December 2021

Crawling web pages with dynamic content is now generally available

The **Execute JavaScript during crawl** feature was introduced as a beta feature, but is now generally available. For more information, see [Web crawl](#).

Capturing the SharePoint ACL information from crawled documents

You can now configure the data source crawl to store ACL information as metadata in the documents that are added to your SharePoint Online collection. For more information, see [Microsoft SharePoint Online](#).

You can add more documents to the training data of the beta entity extractor model

If you added and labeled 20 documents to train a model, and now want to continue to improve the model's performance, you can add more documents. Add the additional documents to the collection that you are using to train the model. After you label the first 20 documents, and the model is up to date with any changes, you can choose to continue labeling documents. The new documents that you added to the collection are loaded. You can label them to augment the training data, and then retrain your model. For more information, see [Customizing the terms that Discovery can recognize](#).

Log out of Discovery

You can log out of the Discovery service instance at any time by clicking **Log out** from the user profile menu that is available from the page header of the product user interface.

18 November 2021

Enterprise plan is now available everywhere

The Enterprise plan is available from all data center locations. Scale and secure your Discovery application with enterprise-grade support and performance, and address more use cases including contract analysis and content mining to explore insights across documents. For more information, see [Discovery pricing plans](#).

11 November 2021

New locations for Enterprise plan now available

The Enterprise plan is available from the Frankfurt, London, Sydney, and Tokyo locations in addition to the Dallas location.

3 November 2021

New Enterprise plan

Scale and secure your Discovery application with enterprise-grade support and performance and address more use cases, including contract analysis and content mining to explore insights across documents. Currently, the Enterprise plan is available only from the Dallas location. For more information, see [Discovery pricing plans](#).

New beta entity extractor enrichment

The **Extract entities** enrichment brings the powerful ability to build a custom type system into Discovery. Use the tool to label entity examples within your industry data to build a machine learning model that Discovery can use to recognize meaningful terms for your business. Currently, this beta feature is available for English-language projects that are created in Premium plan service instances only. For more information, see [Customizing the terms that Discovery can recognize](#).

New **Helpful links** tab

The home page includes a **Helpful links** tab that has quick links to documentation, a community site, and other resources.

Improved field selection choices

When you apply an enrichment to a field or choose a field to use as the source for a facet, the fields that are displayed for you to choose from now include only fields that are valid choices. Previously, the list included fields that were not valid choices.

14 October 2021

New Discovery home page

A new home page is displayed when you start Discovery and gives you quick access to a product overview video, and tours. You can collapse the home page welcome banner to see more projects.

New plan usage section

Stay informed about plan usage and check your usage against the limits for your plan type from the [Plan limits and usage](#) page. From the product page header, click the user icon . The **Usage** section shows a short summary. Click **View all** to see usage information for all of the plan limit categories.

Change to spelling settings in Search

The spelling correction setting changed from being enabled automatically in new projects to being disabled by default. If you want to alert users when they misspell a term in their query, turn on **Spelling suggestions**. For more information, see [Customizing the search bar](#).

Improved **Guided tours** availability

The **Guided tours** button is now available from the product page header, which make them accessible from anywhere. Previously, it was available from the **My Projects** page only.

1 October 2021

Change to Lite and Advanced plans in all locations

Lite and Advanced plans are discontinued. You cannot create **new** service instances that use the Lite or Advanced plan types in the Dallas, Frankfurt, London, Sydney, Tokyo, and Washington DC locations. Any existing Lite and Advanced plans continue to function properly and continue to be supported. You can upgrade from a Lite plan to an Advanced plan. Use the new Plus plan and its associated 30-day free trial to explore new features and a simpler way to build that is available with the latest version of the product.

24 September 2021

New scoring for NLU enrichments

Relevance and confidence scores are displayed for NLU enrichments that are returned by search. For example, when you open the JSON view of the document preview from a query result, you can see confidence scores for Entities mentions and relevance scores for Keyword mentions.

9 September 2021

New location for Plus plan

The Plus plan is now available from the Sydney location. Use the new Plus plan and its associated 30-day free trial to explore new features and a simpler way to build that is available with the latest version of the product. For more information, see [Getting the most from Discovery](#).

Change to Lite and Advanced plans in most locations

Lite and Advanced plans are discontinued. You cannot create **new** service instances that use the Lite or Advanced plan types in the Dallas, Frankfurt, London, Sydney, Tokyo, or Washington DC locations. Any existing Lite and Advanced plans continue to function properly and continue to be supported. You can upgrade from a Lite plan to an Advanced plan.

26 August 2021

New locations for the Plus plan

The Plus plan is now available from the London and Washington DC locations, in addition to Dallas, Frankfurt, and Tokyo.

Change to Lite and Advanced plans in some locations

You cannot create **new** service instances that use the Lite or Advanced plan types in the Dallas, Frankfurt, London, Tokyo, or Washington DC locations. Any existing Lite and Advanced plans continue to function properly and continue to be supported. You can upgrade from a Lite plan to an Advanced plan.

New answer finding feature

Answer finding is now generally available for managed deployments. Use answer finding when you want to return a concise answer to a question. For more information, see [Answer finding](#).

16 August 2021

New locations for the Plus plan

The Plus plan is now available from the Frankfurt and Tokyo locations, in addition to Dallas.

Change to Lite and Advanced plans in some locations

Lite and Advanced plans are no longer offered. You cannot create **new** service instances that use the Lite or Advanced plan types in the Dallas, Frankfurt, or Tokyo locations. Any existing Lite and Advanced plans continue to function properly and continue to be supported. You can upgrade from a Lite plan to an Advanced plan.

27 July 2021

Improved document size limit

Document size limit is increased. For Premium plan collections, you can now upload files that are up to 50 MB in size instead of 32 MB. For more information, see [Document limits](#).

23 July 2021

Improved SharePoint Online connector

The Microsoft SharePoint Online data source connector now accepts any valid Azure Active Directory user ID syntax; the format of the user ID doesn't need to match the `<admin_user>@.onmicrosoft.com` syntax. For more information, see [Microsoft SharePoint Online](#).

16 July 2021

New beta dynamic website web crawl

The Web crawler can now crawl dynamic websites that use JavaScript to render content. If you enable this beta feature, the time it takes to crawl the site increases. For more information, see [Web crawl](#).

23 June 2021

New Plus plan

Use the new Plus plan and its associated 30-day free trial to explore new features and a simpler way to build that is available with the latest version of the product. Currently, the Plus plan is available from the Dallas location. For more information, see [Getting the most from Discovery](#).

Change to Lite and Advanced plans

Lite and Advanced plans are no longer offered. You cannot create **new** service instances that use the Lite or Advanced plan types in the Dallas location. Any existing Lite and Advanced plans continue to function properly and continue to be supported. You can upgrade from a Lite plan to an Advanced plan.

Endpoint deprecation reminder

Change to Discovery API endpoint

As part of work done to fully support Identity and Access Management (IAM) authentication, the endpoint that you use to access your Discovery service programmatically is changing. The old endpoint URLs are deprecated and **will be retired on 26 May 2021**. Update your API calls to use the new URLs.

The pattern for the endpoint URL changed from `gateway-{location}.watsonplatform.net/discovery/api/` to `api.{location}.discovery.watson.cloud.ibm.com/`. The domain, location, and offering identifier are different in the new endpoint. For more information, see [Updating endpoint URLs from watsonplatform.net](#).

If your service instance API credentials use the old endpoint, create a new credential and start using it today. After you update your custom applications to use the new credential, you can delete the old one.

19 March 2021

Improved Web crawl connector

You can use the Web crawl collection type to connect to content that is stored on an internal company website. For more information, see [Web crawl](#).

4 March 2021

New drag and drop feature when uploading

Upload collections now support dragging and dropping documents before and during document upload. For more information, see [Uploading data](#).

New list view for collections

You can view a list of collections that are connected to a particular gateway. For more information, see [Viewing collections connected to a gateway](#).

17 December 2020

Improved date and time display on Activity tab

Each collection now displays the **Next sync scheduled for** date and time on the **Activity** tab of the **Manage collections** page.

New beta FAQ extraction

Released the beta feature FAQ extraction. FAQ extraction automatically extracts question-and-answer pairs from FAQ (frequently asked questions) documents and web pages so that your application returns more precise answers. For more information, see [FAQ extraction](#). For a statement explaining beta features, see [Beta features](#).

3 December 2020

New Content Intelligence

You can now apply the **Contracts** enrichment to a **Document Retrieval** project when you create it. The Contracts enrichment can be used to classify contract terms, parties, effective dates and more within your documents. For more information, see [Document Retrieval for Contracts](#).

10 November 2020

New Box connector

Crawl Box systems. For more information, see [Box](#).

New SharePoint 2016 On-Premises connector

Crawl SharePoint 2016 On-Premises systems. For more information, see [SharePoint 2016 On-Premises](#).

The Box connector does not run on Safari

For more information, see [Box connector](#).

Metadata conversion

If the **metadata** property is converted to an array in the index, the document cannot be deleted by using the **Delete labeled data** API method. For more information, see the [API reference](#).

30 October 2020

New language support for Bosnian, Croatian, Hindi, and Serbian

Basic language support now available for Bosnian, Croatian, Hindi, and Serbian. For more information, see [Language support](#).

New beta Patterns enrichment

The beta release of Patterns enrichment uses pattern induction to help you teach Discovery to recognize patterns in your data. Pattern induction generates extraction patterns from the examples you specify. After you specify a small number of examples, Discovery will suggest additional rules that you verify to complete the pattern. You can use pattern induction as an enrichment or to create a facet. For more information, see [Patterns](#) and [Creating a facet by identifying a pattern](#). For a statement explaining beta features, see [Beta features](#).

Change to Document Retrieval projects

In new **Document Retrieval** projects, the **suggested refinements** query setting is now set to **false** by default. It was previously set to **true**.

14 September 2020

New pre-trained model for SDU

A new pre-trained model is available in Smart Document Understanding for Document Retrieval projects. This model is ideal if you need to extract data from documents that include a large number of tables. For more information, see [Identifying fields](#).

30 August 2020

Update to API version

The current API version (v2) is now 2020-08-30. The following change was made with this version:

Change to 'options' object

The List enrichments method no longer returns the **options** object per enrichment. Use the Get enrichment method to return the **options** object for a single enrichment.

16 July 2020

New release for Premium instances

This release is available for Premium instances of Discovery on IBM Cloud created after 16 July 2020. For Premium instances created before that date and for all Lite and Advanced plans, see [Getting started with Discovery](#).

Change to IBM Cloud Premium

The Premium plan is now generally available.

New Project-based interface

The project-based UI includes configurations optimized for three common use cases: Document Retrieval, Conversational Search, and Content Mining. For more information, see [Creating projects](#).

New Content Mining app

This entirely new capability of Watson Discovery allows you to find insights in your data when you may not even know the question to ask. The powerful correlation tooling will help you unlock value from large unstructured data sets. For details, see [Analyzing your data with the Content Mining application](#).

New tables as answers

Snippets of text aren't helpful if they are found in a table, so Discovery instead returns a formatted table as an answer if your question is best answered by a table. For more information, see [Table retrieval](#).

New dynamic faceted search feature

Underspecified queries are common. Dynamic Faceted Search automatically categorizes your search results into intelligence facets without training by understanding how they are used in the sentences. See [Facets in Document retrieval projects](#).

New reusable components

You no longer need to build a Discovery application from scratch. We now ship out of the box with reusable, open source, React components. As you configure your Discovery application, you are using the real components. From there you simply deploy to get a custom Discovery application. See [Building and deploying components](#).

New Domain Vocabulary feature

You can build a facet for your users without a Dictionary. Use Domain Vocabulary to build a powerful facet with our understanding of how the data is used in as little as 5 minutes. See [Facets](#).

New relevancy training

You can train at a project level. Discovery ranks the best answer regardless of the data source/collection. See [Improving result relevance with training](#).

New built-in spelling corrector

Discovery has spelling suggestions built in. See [Parameters descriptions](#).

Improved Autocomplete

Discovery includes autocomplete (type-ahead) for searches, as well as a reusable component for providing this feature to your end users.

New support for 12 languages

Language support for Discovery is now available in 12 additional languages. For the complete list, see [Language support](#).

Cloud Object Storage connector limitation

When connecting to an IBM Cloud® Object Storage data source, only the first 75 buckets for a given credential are displayed.

Current API version

The API version (v2) is **2019-11-29**.

Change to features in this release

Deduplication is not available in this release.

Anomaly Detection is not offered.

IBM Watson® Discovery News is no longer included.

Several Watson Natural Language Understanding enrichments are not available at this time (Entity extraction, Relation extraction, Keyword extraction, Category classification, Concept tagging, Semantic Role extraction, Sentiment analysis, Emotion analysis)

The SharePoint 2016 On-Premises and Box data sources are not available at this time.

Release notes for IBM Watson® Discovery Cartridge for IBM Cloud Pak® for Data

Learn about features and changes that were included for each release and update of the product software.

IBM Cloud Pak for Data



Note: This information applies only to instances of IBM Watson® Discovery that are installed on IBM Cloud Pak® for Data. For information about releases and updates for managed deployments, see [Release notes for Watson Discovery for IBM Cloud](#).

For the list of Discovery known issues, see [Limitations and known issues in Watson Discovery](#).

Knowledge Studio for IBM Cloud Pak for Data deprecation announcement

After version 4.7, the operator for IBM Knowledge Studio will no longer be supported and will be removed from the IBM Watson Discovery Cartridge for IBM Cloud Pak for Data and from github.com. The service will not be displayed in the Cloud Pak for Data catalog. This change will not impact existing deployments of the operator.

Migrate your solutions to Watson Discovery, which has powerful custom natural language processing capabilities. Any existing Watson Knowledge Studio for Cloud Pak for Data rules-based or machine learning models can be imported to Watson Discovery and applied to your data as custom enrichments. And the recent release of the custom entities extraction feature brings equivalent function to label and train custom entity models into Watson Discovery. For more information about these features, see [Choose enrichments](#).

For more information about migrating your solutions, see [Migrating Knowledge Studio solutions](#).

4.7.1 release, 26 July 2023

IBM Watson® Discovery Cartridge for IBM Cloud Pak® for Data 4.7.1 is available.

For a list of new features and bug fixes, see [What's new and changed in Watson Discovery](#)

Features that are not available in this release

The following features are generally available from managed IBM Cloud deployments at the time of this release, but not from installed deployments:

- Answer finding

4.7.0 release, 28 June 2023

IBM Watson® Discovery Cartridge for IBM Cloud Pak® for Data 4.7.0 is available.

For a list of new features and bug fixes, see [What's new and changed in Watson Discovery](#)

Features that are not available in this release

The following features are generally available from managed IBM Cloud deployments at the time of this release, but not from installed deployments:

- Answer finding
- Optical Character Recognition v2

4.6.6 release, 18 May 2023

IBM Watson® Discovery Cartridge for IBM Cloud Pak® for Data was not refreshed as part of 4.6.6. You can use Discovery 4.6.5 with IBM Cloud Pak for Data 4.6.6.

4.6.5 release, 2 May 2023

IBM Watson® Discovery Cartridge for IBM Cloud Pak® for Data 4.6.5 is available.

For a list of new features and bug fixes, see [What's new and changed in Watson Discovery](#)

Manage the data in a collection from the new **Manage data** page

You can now access a **Manage data** page for a collection. From the new page, you can see a list of the documents in your collection and get a quick view of information about the documents. You can also delete documents from a collection with just a few clicks. For more information, see [Excluding content from query results](#).

You have more control over the data that is crawled by the database connector

When you connect to a database as an external data source, you can now specify the column from which to extract data. If you don't specify the column, a column with text or with a single large object is chosen to be crawled. You can also specify the MIME type of the data in the column that you want to crawl.

Features that are not available in this release

The following features are generally available from managed IBM Cloud deployments at the time of this release, but not from installed deployments:

- Answer finding
- Optical Character Recognition v2

4.6.4 release, 29 March 2023

IBM Watson® Discovery Cartridge for IBM Cloud Pak® for Data was not refreshed as part of 4.6.4. You can use Discovery 4.6.3 with IBM Cloud Pak for Data 4.6.4 on Red Hat OpenShift Container Platform versions 4.10 or 4.12.

4.6.3 release, 23 February 2023

IBM Watson® Discovery Cartridge for IBM Cloud Pak® for Data 4.6.3 is available.

For a list of new features and bug fixes, see [What's new and changed in Watson Discovery](#)

Features that are not available in this release

The following features are generally available from managed IBM Cloud deployments at the time of this release, but not from installed deployments:

- Answer finding
- **Manage data** page

Important: Back up your data before upgrading to version 4.6.3

Before upgrading to version 4.6.3, you must make a backup of your data. Preserve the backup in a safe location. For more information about backing up your data, see [Backing up and restoring data in IBM Cloud Pak for Data](#). That topic also includes information about restoring your data if that becomes necessary.

4.6.2 release, 30 January 2023

IBM Watson® Discovery Cartridge for IBM Cloud Pak® for Data 4.6.2 is available.

For a list of new features and bug fixes, see [What's new and changed in Watson Discovery](#)

Features that are not available in this release

The following features are generally available from managed IBM Cloud deployments at the time of this release, but not from

installed deployments:

- Answer finding
- **Manage data** page

4.6.1 release, December 2022

IBM Watson® Discovery Cartridge for IBM Cloud Pak® for Data was not refreshed as part of 4.6.1. However, the product documentation was updated with fixes and enhancements.

4.6 release, 30 November 2022

IBM Watson® Discovery Cartridge for IBM Cloud Pak® for Data 4.6 is available.

For a list of new features and bug fixes, see [What's new and changed in Watson Discovery](#)

Features that are not available in this release

The following features are generally available from managed IBM Cloud deployments at the time of this release, but not from installed deployments:

- Answer finding
- **Manage data** page

4.5.3 release, 13 October 2022

IBM Watson® Discovery Cartridge for IBM Cloud Pak® for Data 4.5.3 is available.

There are no new features in this release. For a list of bug fixes, see [What's new and changed in Watson Discovery](#)

Features that are not available in this release

The following features are generally available from managed IBM Cloud deployments at the time of this release, but not from installed deployments:

- Answer finding
- **Manage data** page
- Advanced document view for search results
- The **similar** parameter of the **Query** method
- The **smart_document_understanding** field in the **Get collection** method response

15 August 2022

SDKs were updated to reflect the latest API changes.

The following [Discovery v2 API](#) changes are now reflected in the SDKs:

- Use the new document classifier API to get, add, update, or delete a document classifier.
- A new document status API is available. You can use it to get a list of the documents in a collection and to get details about a single document.
- You can now get, add, and remove a stop words or expansion list for a collection.
- The **suggested_refinements** parameter of the **Query** method is deprecated.

4.5.1 release, 3 August 2022

IBM Watson® Discovery Cartridge for IBM Cloud Pak® for Data 4.5.1 is available.

There are no new features in this release. For a list of bug fixes, see [What's new and changed in Watson Discovery](#)

Features that are not available in this release

The following features are generally available from managed IBM Cloud deployments at the time of this release, but not from installed deployments:

- Answer finding
- **Manage data** page
- Advanced document view for search results
- The **similar** parameter of the **Query** method
- The **smart_document_understanding** field in the **Get collection** method response

4.5 release, 29 June 2022

IBM Watson® Discovery Cartridge for IBM Cloud Pak® for Data 4.5 is available.

For a list of new features and bug fixes, see [What's new and changed in Watson Discovery](#)

Features that are not available in this release

The following features are generally available from managed IBM Cloud deployments at the time of this release, but not from installed deployments:

- Answer finding
- **Manage data** page
- Advanced document view for search results

4.0.9 release, 25 May 2022

IBM Watson® Discovery Cartridge for IBM Cloud Pak® for Data 4.0.9 is available.

API usage information is now available from the user interface

You can now get information about analyze API usage from the **Data usage>API usage** page in the product user interface. For more information about the analyze API, see [Analyze API](#).

A new document status API is supported in IBM Cloud Pak® for Data instances

Use the new document status API to programmatically get a list of the documents in a collection and to get details about a single document.

- The API is supported for collections that are created after 23 March 2022.

If you want to get status information about a collection that was created earlier, trigger a process that runs the conversion step of ingestion on the documents. For example, from the **Activity** page for the collection, click **Recrawl**.

- The API is not supported from the SDKs currently.

For more information about the new API, see the [API reference documentation](#).

Security vulnerabilities were addressed

The following security patches were applied:

- [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data affected by vulnerability in Node.js](#)
- [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data affected by vulnerability in Apache Xerces](#)
- [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data affected by vulnerability in OpenSSL](#)

- [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data affected by vulnerability in Google Protocol Buffers](#)

4.0.8 release, 27 April 2022

IBM Watson® Discovery Cartridge for IBM Cloud Pak® for Data 4.0.8 is available.

The **Development** deployment type was changed to **Starter**

When you install Watson Discovery, you can optionally specify the type of deployment by including the `deploymentType` parameter in your custom resource. The **Development** option is now called the **Starter** option.

The **Development** and **Starter** options are functionally the same, and both values are accepted by the service.

Security vulnerabilities were addressed

The following security patches were applied:

- [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data affected by vulnerability in Google Protocol Buffers](#)
- [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data affected by vulnerability in Node.js](#)
- [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data affected by vulnerability in Java](#)
- [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data affected by vulnerability in PostgreSQL](#)
- [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data affected by vulnerability in Kotlin](#)
- [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data affected by vulnerability in Apache POI](#)
- [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data is affected by a remote code execution in Spring Framework \(CVE-2022-22965\)](#)

IBM Watson® Discovery for IBM Cloud Private (ICP) for Data 2.2.x End Of Support

Effective 30 April 2022, IBM will withdraw support for the following programs:

- IBM Watson Discovery for ICP for Data 2.2.x
- IBM Watson Discovery for ICP for Data Add-on 2.2.x

For more information, see announcement [ENUS921-134.PDF](#).

4.0.7 release, 30 March 2022

IBM Watson® Discovery Cartridge for IBM Cloud Pak® for Data 4.0.7 is available.

IBM Cloud Block Storage is now supported

When you install Discovery, you can specify IBM Cloud Block Storage Gold tier (ibmc-block-gold) as your storage class. For more information about the storage class, see [Storing data on classic IBM Cloud Block Storage](#).

Security vulnerabilities were addressed

The following security patches were applied:

- [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data affected by vulnerability in NumPy](#)
- [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data affected by vulnerability in Spring](#)
- [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data affected by vulnerability in FasterXML jackson-databind](#)
- [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data affected by vulnerability in TensorFlow](#)
- [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data affected by vulnerability in XStream](#)
- [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data affected by vulnerability in Go](#)

Features that are not available in this release

The following features are generally available from managed IBM Cloud deployments at the time of this release, but not from installed deployments:

- Home page updates

- Answer finding
- **Manage data** page
- Advanced document view for search results

30 March 2020

A new document classifier API is available

Use the new document classifier to programmatically get, add, update, or delete a document classifier. The following notes apply to this release:

- The **enrichments** property of the Document Classifier object is documented as being optional. However, the property is required currently.
- The **field** property in the **federated_classification** object is documented as a string. However, it is currently an array.

For more information about the new API, see the [API reference documentation](#). For more information about adding a document classifier by using the product user interface, see [Using the Content Mining application](#).

The document classifier endpoints are not supported in the SDKs currently.

4.0.6 release, 1 March 2022

IBM Watson® Discovery Cartridge for IBM Cloud Pak® for Data 4.0.6 is available.

Multitenancy is now supported

An administrator can now create up to 10 instances of the Discovery service per deployment, which means that more teams can work on discrete Discovery projects at the same time.

Simpler installation and management of custom connectors

The `manage_custom_crawler.sh` script was improved to make it easier for you to install and manage your custom connectors in a multitenant environment. For more information, see [Installing a custom crawler](#).

Security vulnerabilities were addressed

The following security patches were applied:

- [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data affected by vulnerability in Java](#)
- [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data affected by vulnerability in Logback](#)
- [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data affected by vulnerability in Apache Log4j](#)

Features that are not available in this release

The following features are generally available from managed IBM Cloud deployments at the time of this release, but not from installed deployments:

- Home page updates
- Answer finding
- Access to guided tours from the page header

4.0.5 release, 26 January 2022

IBM Watson® Discovery Cartridge for IBM Cloud Pak® for Data 4.0.5 is available.

A security vulnerability was addressed

The following security patch was applied: [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data affected by vulnerability in Apache Log4j](#)

Features that are not available in this release

The following features are generally available from IBM Cloud deployments at the time of this release, but not from installed deployments:

- Home page updates
- Answer finding
- Guided tours

4.0.4 release, 20 December 2021

IBM Watson® Discovery Cartridge for IBM Cloud Pak® for Data 4.0.4 is available.

Guided tours are available

Access guided tours from anywhere in the product user interface by clicking the **Guided tours** button in the page header.

Security vulnerabilities were addressed

The following security patches were applied:

- [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data affected by vulnerability in LibTIFF](#)
- [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data affected by vulnerability in TensorFlow](#)
- [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data affected by vulnerability in Node.js](#)
- [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data affected by vulnerability in Netty](#)
- [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data affected by vulnerability in Apache Log4j](#)
- [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data affected by vulnerability in Apache Log4j 1.2](#)

Features that are not available in this release

The following features are generally available from IBM Cloud deployments at the time of this release, but not from installed deployments:

- Home page updates
- Answer finding

4.0.3 release, 30 November 2021

IBM Watson® Discovery Cartridge for IBM Cloud Pak® for Data 4.0.3 is available.

Another storage option is supported

IBM Spectrum Scale Container Native storage is now supported in addition to Red Hat OpenShift Container Storage and Portworx.

Microsoft SharePoint Online data source improvement

The **Sharepoint Online** data source now supports crawling your data as a service principal, which means you can access your data without disabling multifactor authentication. For more information, see [Microsoft Sharepoint Online](#).

Microsoft Windows File System improvements

Extra configuration options mean you can specify the following information:

- The types of files (by file extension) to include or exclude from a crawl of a Windows directory.
- The character encoding of the data to be crawled. Typically, the encoding is detected automatically. However, you can choose to specify the character encoding as a Java character set yourself.

For more information, see [Windows File System](#).

Field selection is improved

When you apply an enrichment to a field or choose a field to use as the source for a facet, the fields that are displayed for you to choose from now shows only fields that are valid choices.

Search settings change

The spelling correction setting changed from being enabled automatically in new projects to being disabled by default. If you want to alert users when they misspell a term in their query, turn on *Spelling suggestions*. For more information, see [Customizing the search bar](#).

A Salesforce crawling issue was fixed

Previously, Discovery had an issue where it timed out before it crawled some of the object types in a Salesforce collection. If your collection is configured to crawl the following object types, run a full data source crawl to make sure that your collection contains the most up-to-date data from all of the objects in your Salesforce data source:

- Attachment
- ContentVersion
- Document

Security vulnerabilities were addressed

The following security patches were applied:

- [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data affected by vulnerability in Node.js](#)
- [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data affected by vulnerability in Axios](#)
- [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data affected by vulnerability in Python Pillow](#)
- [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data affected by vulnerability in Apache Commons Compress](#)
- [Security Bulletin: IBM Watson Discovery for IBM Cloud Pak for Data affected by vulnerability in Java](#)

Features that are not available in this release

The following features are available from IBM Cloud deployments at the time of this release, but not from installed deployments:

- Home page updates
- Guided tours
- Answer finding

4.0.2 release, 5 October 2021

IBM Watson® Discovery Cartridge for IBM Cloud Pak® for Data 4.0.2 is available.

Support for newer platform software

IBM Cloud Pak® for Data 4.0.2 can be installed on Red Hat® OpenShift® on IBM Cloud® 4.8.

New scoring for NLU enrichments

Relevance and confidence scores are displayed for NLU enrichments that are returned by search. For example, when you open the JSON view of the document preview from a query result, you can see confidence scores for Entities mentions and relevance scores for Keyword mentions.

Improved Web crawl

The *Web crawl* data source supports more customization options, including the ability to ignore a site's robots.txt file. For more information, see [Web crawl](#).

New upgrade support

The 4.0.2 release supports in-place upgrade from IBM Watson® Discovery Cartridge for IBM Cloud Pak® for Data 4.0.0. For more information, see [Upgrading Watson Discovery to a newer 4.0 refresh](#)

IBM Cloud Private End Of Support

Effective 30 September 2021, IBM withdrew support for the following programs:

- IBM Watson Assistant Discovery Extension for IBM Cloud Private 2.1.0–2.1.4
- IBM Watson Discovery for ICP for Data 2.1.0–2.1.4
- IBM Watson Discovery for ICP for Data Add-on 2.1.0–2.1.4

For more information, see announcements [ENUS921-005.PDF](#) and [ENUSLP21-0099.PDF](#).

4 release, 13 July 2021

New version now available

Discovery for Cloud Pak for Data 4 is available

This release is supported on IBM Cloud Pak® for Data 4.0.0.

Change to service name

The new name is IBM Watson® Discovery Cartridge for IBM Cloud Pak® for Data.

New Smart Document Understanding (SDU) predefined model

When you identify fields, instead of annotating documents with the SDU tool, you can choose to use a pretrained model. The pretrained model applies a non-customizable model that automatically extracts text and identifies tables, lists, and sections.

Improved contract analysis

To enable the Contracts enrichment that recognizes and tags contract-related concepts in your data, you can choose to create a Document Retrieval project type, and then select **Apply contracts enrichment**. You no longer need to use an installation override YAML file to enable it. This change also means that you can choose which Document Retrieval projects use the Contracts enrichment; it is not applied to all Document Retrieval projects automatically.

New LDAP directory data source

Connect to data that is stored in an external directory that supports the Lightweight Directory Access Protocol (LDAP), such as a corporate email directory. As the directory data is added to your collection, Discovery interprets and stores key attributes of each record, such as department and location information. Later, you can find relevant records by filtering on these attribute categories. For more information, see [LDAP directory](#).

Improved SharePoint OnPrem connection process

The steps you follow to connect to a SharePoint instance that is hosted on-premises were simplified. You no longer need to deploy a web services package on the SharePoint server before you can connect to the SharePoint OnPrem data source. For more information, see [SharePoint OnPrem](#).

New Salesforce proxy support

You can now connect to a Salesforce data source when using a proxy server. For more information, see [Salesforce](#).

Improved custom connector improvements

Support was added for Optical character recognition (OCR)

Support was added for Document-level security

For more information about the custom connector, see [Building a Cloud Pak for Data custom connector](#).

Change to Dynamic Faceted Search

Support for **Dynamic Faceted Search** and its associated `suggested_refinements` API query parameter was removed.

2.2.1 release, 26 February 2021

New release now available

IBM Watson™ Discovery for IBM Cloud Pak for Data version 2.2.1 is available.

Support for upgrade

Discovery for Cloud Pak for Data supports an in-place upgrade from version 2.2.0 to 2.2.1 so that you do not need to manually uninstall an earlier version and then install the latest version of the service. For more information, see [Upgrading Discovery for Cloud Pak for Data](#).

New SDK download support

You can now download the custom connector SDK package from your Discovery for Cloud Pak for Data cluster, instead of

retrieving the images and the SDK package from the Docker registry. For more information, see [Downloading the custom-crawler-docs.zip file in Discovery 2.2.1 and later](#).

Change to Invoices and Purchase orders

Invoices and **Purchase orders** models can no longer be enabled in the tooling. If you need these models, please contact [IBM Cloud Support](#) to obtain instructions for enabling these models.

Change to Contracts enrichment tables

In a **Document Retrieval** project that has the **Contracts** enrichment applied, tables are not included inside the **contracts** field, as they were previously in projects that had the **Contracts** enrichment enabled. Tables will continue to be included in a separate **tables** field when the **Table Understanding** enrichment is applied.

Change to support for Oracle Database 11g and Postgres 9.5

Support for connecting to Oracle Database 11g was removed because the vendor ended version support on 31 December 2020.

Support for connecting to Postgres 9.5 was removed because the vendor ended version support on 11 February 2021.

2.2.0 release, 8 December 2020

New release now available

IBM Watson™ Discovery for IBM Cloud Pak for Data version 2.2 is available.

Discovery for Cloud Pak for Data now works with IBM Cloud Pak® for Data 3.5.

New support for Notes attachments

Added support for attachments in the Notes data source. For more information, see [Notes](#)

New web crawl scheduling option

You can specify the exact time that you would like your crawls to run for any data source, giving you the flexibility to run them at the times you prefer. For more information, see [Configuring Cloud Pak for Data data sources](#).

New Facet creation in Content Miner

You can now create Facet groups in a Content Miner application.

New custom crawler creation

Added the option to create your own custom crawler plug-in. For more information, see [Building a Cloud Pak for Data crawler plug-in](#). **Note:** Any custom code used with Watson Discovery is the responsibility of the developer and is not covered by IBM support.

Change to Dynamic Facets

Dynamic Facets are no longer enabled by default in Document Retrieval projects.

2.1.4 release, 2 September 2020

New release now available

IBM Watson® Discovery for IBM Cloud Pak® for Data version 2.1.4 is available.

New Notes connector

Crawl Notes version 9.0.1 systems. For more information, see [Notes connector](#).

New **Enable proxy settings** in multiple connectors

You can now select the option to enable proxy settings in [Box](#), [Microsoft SharePoint Online](#), and [Microsoft SharePoint OnPrem](#) connectors.

New options for Database connector

Added support for multiple tables and the Row filter option to the [Database connector](#).

New authentication types for Web crawler

You can select from three new authentication types in [Web crawler](#): Basic authentication, NTLM authentication, and FORM

authentication.

New Analyze API usage monitoring

You can now monitor the usage of the Analyze API using the tooling. For more information, see [Monitoring usage](#).

30 August 2020

Update to API version

The current API version (v2) is now 2020-08-30. The following change was made with this version:

Change to 'options' object

The List enrichments method no longer returns the **options** object per enrichment. Use the Get enrichment method to return the **options** object for a single enrichment.

2.1.3 release, 19 June 2020

New release now available

IBM Watson® Discovery for IBM Cloud Pak® for Data version 2.1.3 is available.

Discovery for Cloud Pak for Data now works with IBM Cloud Pak® for Data 3.0.1.

New Finnish and Hebrew language support

Added basic support for Finnish and Hebrew. For more information, see [Language support](#).

Change to Analyze endpoint

The Analyze endpoint, which supports stateless document ingestion workflows. For details, see the [Analyze API](#). The Analyze API supports JSON documents only. Use of the Analyze API affects license usage.

New options for Content Miner

The content mining application includes two new options: Cyclic time scale on the **Time series** dashboard, and the **Contextual view** tab.

New shortcut for Content Mining projects

For **Content Mining** projects only, the **Improve and customize** page includes a shortcut: the **Launch application** button. Previously, you were required to open the **Integrate and deploy** page, select the **Launch application** tab, and click the **Launch** button.

Improved segment limit

The segment limit when splitting documents has been increased to 1,000. For details, see [Split documents to make query results more succinct](#).

Improved Filenet connector

The [Filenet connector](#) has document level security.

New beta Curations feature

You can specify up to 1,000 curations. For details about this beta feature, see [Curations](#).

Fixed defects in the 2.1.3 release

In versions 2.1.2, 2.1.1, and 2.1.0, PNG, TIFF, and JPG individual image files are not scanned, and no text is extracted from those files. PNG, TIFF, and JPEG images embedded in PDF, Word, PowerPoint, and Excel files are also not scanned, and no text is extracted from those image files.

2.1.2 release, 31 March 2020

New release now available

IBM Watson® Discovery for IBM Cloud Pak® for Data version 2.1.2 is available.

New IBM FileNet connector

You can now crawl IBM FileNet systems. For more information, see [FileNet connector](#).

New Swedish, Norwegian, and Danish language support

Added basic support for Swedish, Norwegian (Bokmål and Nynorsk), and Danish. For more information, see [Language support](#).

Change to Advanced rules models enrichment

The [Advanced rules models enrichment](#) is now GA.

New document preview for search results

You can now view your search results in a document preview for the following source documents: PDF, Word, PowerPoint, Excel, and all image files. See [supported file types](#) for the list of image files. This view makes it easier for you to see search results as highlighted passages within the text of the original document, making the context clearer.

New proxy support for Web Crawl

Support was added to the [Web Crawl connector](#) for proxy support.

Change to empty aggregations parameter

Running a query with an empty **aggregations** parameter returns zero aggregations in the response.

Change to Postgres support

Support for connecting to Postgres 9.4 was removed because the vendor ended version support was ended by the vendor on 13 February 2020.

Fixed the following defects in the 2.1.2 release

When installing Discovery for Cloud Pak for Data on OpenShift, the **ranker-rest** service might intermittently fail to startup, due to an incompatible jar in the **classpath**.

When you upload documents to a collection with existing documents, a **Documents uploaded!** message displays on the **Activity** page, but no further processing status displays until the number of documents increases.

Running a query with an empty **aggregations** parameter returns an empty aggregations array.

De-provisioning a IBM Watson® Discovery for IBM Cloud Pak® for Data Instance will not delete the underlying data. Delete the collections and documents manually.

2.1.1 release, 24 January 2020

New release now available

IBM Watson® Discovery for IBM Cloud Pak® for Data version 2.1.1 is available.

Fixed the following defects in the 2.1.1 release:

In Document Retrieval project types, when you perform an empty search, and the search results source is set to **passages**, the query results will display **excerpt unavailable** in the Project workspace.

When visiting the Storybook links on the Integrate and deploy page, the links do not go to the correct location. Please visit [Storybook](#) instead to view documentation.

If you are using Smart Document Understanding, two variables no longer need to be set during installation or reinstallation. For more information, see [Environment variable settings for Smart Document Understanding](#).

Discovery for Content Intelligence and Table Understanding enrichments are configured out of the box to be applied on a field named **html**. When a user uploads a JSON document without a root-level field named **html**, these enrichments will not yield results in the index. To run the enrichments on this kind of JSON documents, users must re-configure the enrichments to run on an existing field (or fields) in the JSON document.

2.1.0 release, 27 November 2019

New release now available

IBM Watson® Discovery for IBM Cloud Pak® for Data version 2.1.0 is available.

Discovery for Cloud Pak for Data now works with IBM Cloud Pak® for Data 2.5.0.0.

New Project-based interface

Test your application like an end-user would with the **Document retrieval**, **Conversational Search**, and **Content Mining** project types. For more information, see [Creating projects](#).

New Content Mining app

Build an end user interface for extracting insights proactively from your entire corpus. For more information, see [Analyzing your data with the Content Mining application](#).

New Content Intelligence add-on

Option to enrich your documents with pre-built domain knowledge for Contracts. For more information, see [Document Retrieval for Contracts](#).

New reusable components

Use reusable components to quickly build your application with Discovery. We ship an autocomplete, rich preview, results and facets component. For more information, see [Building and deploying components](#).

New Czech, Polish, Romanian, Russian, and Slovak language support

Basic support for Czech, Slovak, Russian, Polish and Romanian is added. For more information, see [Language support](#).

New built-in table understanding

Extract tables from your documents without training, and optionally return tables as answers to natural language queries. For more information, see [Understanding tables](#).

New SDK connector

Build custom connectors your Discovery users can use to build their own applications. For more information, see [Building and implementing a custom connector](#).

New pre-built sample project

The sample project is preloaded with data, so you can learn about Discovery. For more information, see [Getting started with Watson Discovery](#).

New passage retrieval

Will return the most relevant passages from your documents, plus you can specify the number of passages returned per document. See [Passages](#).

New project-level querying and relevancy training

Query multiple collections at once including relevance training.

Improved Web crawl connector

Additional options now available for the **Web crawl connector** - For more information, see [Web crawl](#).

New Local File System connector

Crawl Linux or other file systems. For more information, see [Local file system](#)

New dynamic Facets

Automatically generate facets based on the understanding of your data. For more information, see [Facets](#).

New Dictionary suggestions

Dictionary terms are suggested based on your content. For more information, see [Dictionary](#).

New beta Curations

Specify a particular result for a given query. For more information, see the [API reference](#).

2.0.1 release, 30 August 2019

New release now available

IBM Watson® Discovery for IBM Cloud Pak® for Data version 2.0.1 is available.

Discovery for Cloud Pak for Data now works with IBM Cloud Pak® for Data 2.1.0.1.

New Windows File System and Database connectors

Added the Windows File System and Database connectors. For more information, see [Database connector](#) and [Windows File System connector](#).

New Chinese language support

Added support for Traditional Chinese. For more information, see [Language support](#).

New FISMA support

Federal Information Security Management Act (FISMA) support is available for IBM Watson® Discovery for IBM Cloud Pak® for Data offerings purchased on or after August 30, 2019. FISMA support is also available to those who purchased the June 28, 2019 version and upgrade to the August 30, 2019 version. IBM Watson® Discovery for IBM Cloud Pak® for Data is FISMA High Ready.

New Classifier enrichment

Released the Classifier enrichment. For more information, see [Classifier](#).

New Red Hat OpenShift support

Added support for installing IBM Cloud Pak® for Data on Red Hat OpenShift.

Fixed the following defects in Discovery for Cloud Pak for Data offerings purchased on or after August 30, 2019

During an active web crawl, if you add an enrichment, then click the **Recrawl collection** button on the **Activity** page, the collection will stop processing. If the collection does not return to a Syncing state on its own, clicking the **Recrawl collection** button an additional time might be required.

While training a collection in the tooling , if you rate the relevancy of a result (for example, as **Relevant**), then switch to the opposite rating (**Not relevant**), the page may go blank. To restore the page, refresh the browser. Your updated rating will be retained.

Chinese, Japanese, and Korean language Microsoft Word, Excel, and PowerPoint documents will not display correctly in the index or the Smart Document Understanding editor.

If you upload a zip, gzip, or tar file to your collection, and that file contains multiple files/file types supported by Smart Document Understanding (PDF, Word, Excel, PowerPoint, PNG, TIFF, JPEG), only one of the files in that zip, gzip, or tar file will be available for training in the SDU editor (unless the SDU document limit has already been met). All of the documents will be available in the index. Unzip the file before uploading to avoid this issue.

Query expansion and autocomplete return the wrong error code when the **collection_id** is invalid. Query expansion will return a **500** error code instead of a **404**. Autocomplete will return a **400** when the **collection_id** is invalid and the **prefix** parameter isn't set. It should also return a **404**.

When crawling Microsoft SharePoint 2019 collections, only HTML documents will be crawled and indexed. This is a SharePoint issue with how it processes mime-types. See this Microsoft [blog post](#) for a workaround.

If you delete an installation of the Discovery for Cloud Pak for Data add-on, the instance will not uninstall completely and your re-installation will fail. See the Discovery for Cloud Pak for Data Readme for post-cleanup steps.

If a JSON document that contains nested JSON objects is ingested, the nested JSON will be indexed as a JSON string.

2.0.0, General Availability (GA) release, 28 June 2019

Discovery for Cloud Pak for Data now available

The IBM Watson® Discovery for IBM Cloud Pak® for Data service brings the cognitive capabilities of IBM Watson® Discovery to the IBM Cloud Pak® for Data platform.

Power your assistant with answers from web resources

In this tutorial, you will use the Watson Discovery and Watson Assistant services to create a virtual assistant that can answer questions about the latest research from the US Federal Reserve. The assistant will answer questions by using up-to-date, existing research publications from the Federal Reserve Economic Data (FRED) website.

IBM Cloud



Note: Follow this tutorial only if you are using a managed deployment.

Learning objectives

By the time you finish the tutorial, you will understand how to:

- Create an action in Watson Assistant that can recognize questions about a particular subject.
- Create a Conversational Search project in Discovery.
- Add a web crawl data source to your project.
- Connect your Watson Assistant action to a search extension that gets answers from your Discovery project.
- Use your assistant to return answers that it retrieves from the website.

Duration

This tutorial will take approximately 2 to 3 hours to complete.

Prerequisite

1. Before you begin, you must set up a paid account with IBM Cloud.

You can complete this tutorial at no cost by using a Plus plan, which offers a 30-day trial at no cost. However, to create a Plus plan instance of the service, you must have a paid account (where you provide credit card details). For more information about creating a paid account, see [Upgrading your account](#).

2. Create a Plus plan Discovery service instance.

Go to the [Discovery resource](#) page in the IBM Cloud catalog and create a Plus plan service instance.



Important: If you decide to stop using the Plus plan and don't want to pay for it, delete the Plus plan service instance before the 30-day trial period ends.

Step 1: Create an assistant

For this tutorial, you will create an assistant with a single action. First, you must create a Watson Assistant service instance.

Both Lite and Trial plan Watson Assistant service instances are available at no cost. You will create a Trial plan because a Plus or higher plan is required to add a search skill to an assistant and the Trial plan includes all Plus plan features. The Lite plan does not.

1. Create a Trial plan Watson Assistant service instance in the same data location where the Discovery service instance is hosted, such as Dallas.
2. From the Watson Assistant plan service page in IBM Cloud, click **Launch Watson Assistant**.

The Watson Assistant product user interface is displayed where you can create your first assistant.

3. Add **FRED research** as the assistant name, and then click **Next**.

Welcome to the new Watson Assistant

[Next](#)

[Create](#) [Personalize](#) [Customize](#) [Preview](#)

Create your first assistant

Let's get your assistant up and running. Name your assistant, add a description, and choose a language. In following steps we'll gather more information, show you basic customizations, and give you a preview of what your assistant will look like.

Assistant name
FRED research

Your assistant name will be kept internally and not visible to your customers

Description (optional)
Add a description for this assistant

0/128

Assistant language
English (US)

This is the language your assistant will speak.

Figure 1. Watson Assistant welcome page

- Fill out the fields to share information about you and your assistant, and then click **Next**.

In the **Which statement describes your needs best** field, choose I'm using Watson Assistant to complete a course or certification..

IBM Watson Assistant Trial | 28 days left | Extend trial | FRED research | Learning center | ? | @

Welcome to the new Watson Assistant

[Back](#) [Next](#)

Personalize your assistant

Tell us where your assistant will live
We will create your first channel integration for you, which will be visible on your dashboard. You can always add more or change later.

Where do you plan on deploying your assistant?
Web

Tell us about yourself
This information will be used to personalize your onboarding experience.

Which industry do you work in?
Banking and financial services

What is your role on the team building the assistant?
Content strategist or writer

Which statement describes your needs best?
I'm using Watson Assistant to complete a course or certification

Figure 2. Assistant details page

- When you create an assistant, a web chat application is created for you automatically.

Welcome to the new Watson Assistant

[Back](#)

[Next](#)



Customize your chat UI

Update the style to match your brand and your website. A developer can also add more advanced styling changes with code. [Learn more](#)

Assistant's name as known by customers

Watson Assistant

Primary color

#FFFFFF



Secondary color

#3D3D3D



Chat header

Accent color

#0354E9



Significant and interactive objects

IBM Watermark

Displays a link to the Watson Assistant website

On



Add an avatar image [?](#)

Figure 3. Web chat settings

6. Click **Next** to accept the default style for the web chat.

Welcome to the new Watson Assistant

[Back](#)

[Create](#)



Preview your assistant

See what your assistant will look like as a chatbot on your website.

[Copy link to share](#)

[Change background](#)

Sample website

Hi! I'm a virtual assistant.

How can I help you today?

Example: Find nearby location

Example: Check account balance

Example: See how I can help

Type something...

Built with IBM Watson®

Figure 4. Web chat preview

A preview of the web chat as it would be displayed in a web page is shown.

7. Click **Create** to create the assistant and the corresponding web chat app.

After a congratulatory message, the home page for your new assistant is displayed.

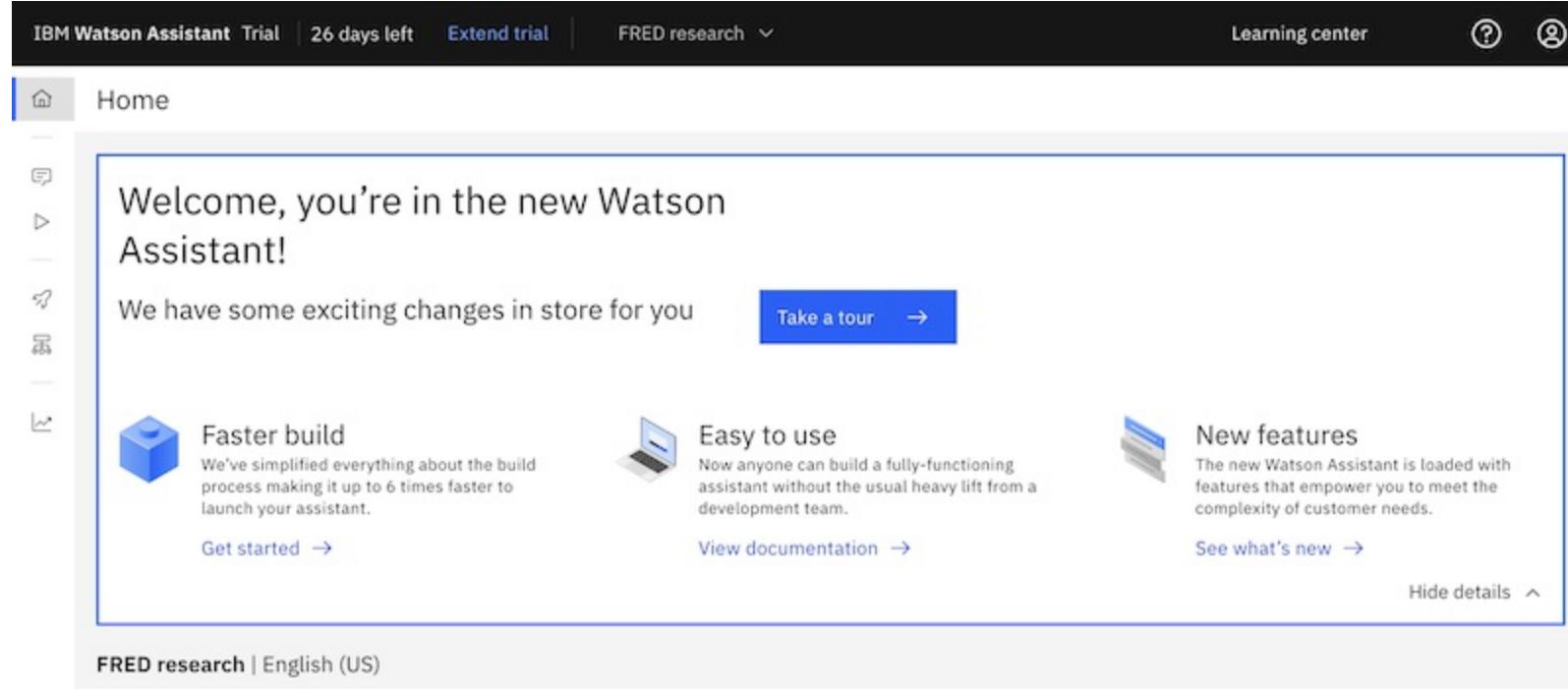


Figure 5. Assistant home page

Step 2: Create an action

Create a single action that can recognize questions about the latest research papers from the US Federal Reserve Economic Data (FRED) website.

In a real world scenario, you might want your assistant to answer questions about the products in your catalog or about insurance plan options or anything else. You can complete similar steps to teach the assistant to recognize when a customer is asking about a particular subject.

1. From the navigation panel, click **Actions**.

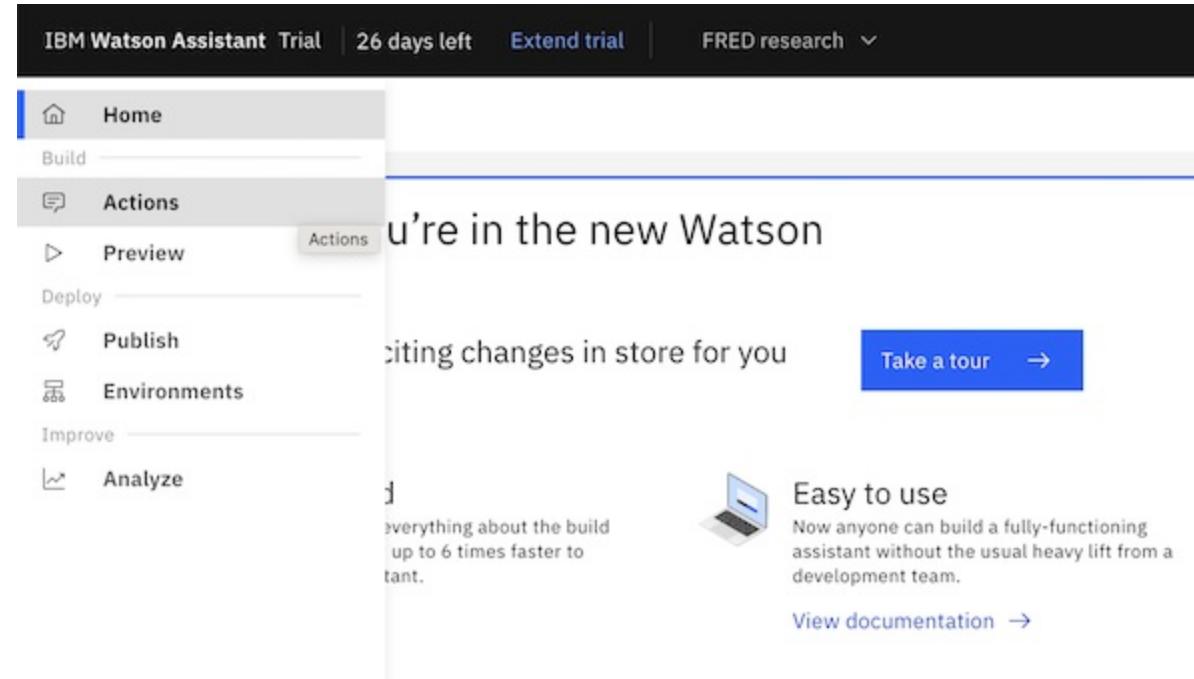


Figure 6. Actions menu

The Actions page is displayed.

Actions

Actions

Created by you

Set by assistant

Variables

Created by you

Set by assistant

Set by integration

Saved responses

Create your first action

With actions, you can help your customers accomplish their goals.

Create action +

Figure 7. Actions page

2. Click **Create action**, and then choose to start from scratch.

How would you like to build your action?

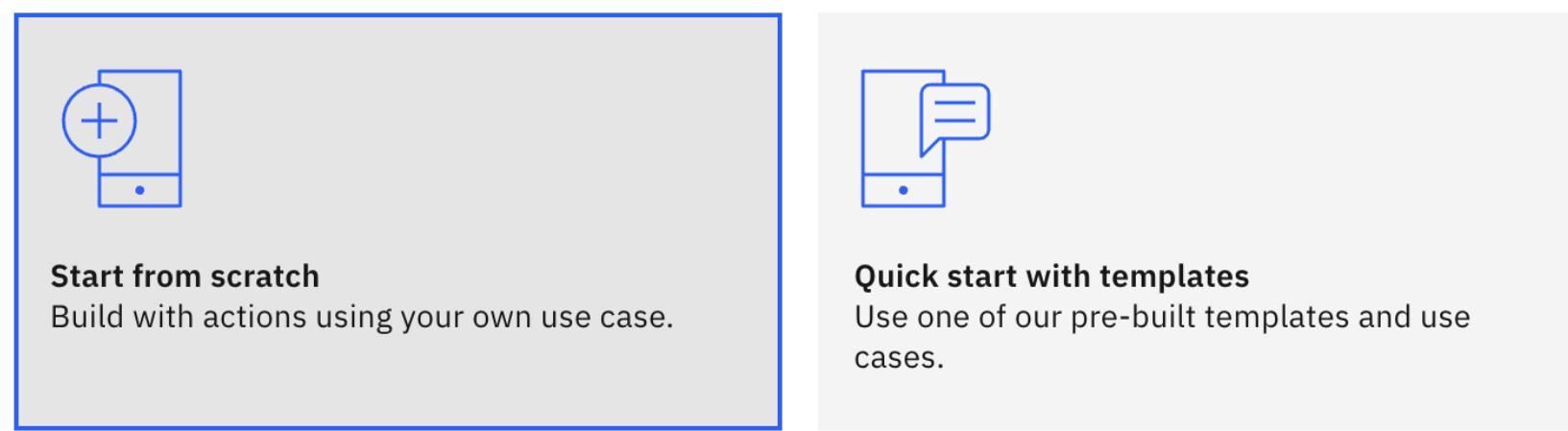


Figure 8. Action creation method options

3. Because you want the assistant to recognize when customers ask about economic research, add the following sample user question, and then click **Save**:

What are the latest working papers about?

The editor closes. We want to add a few more examples.

4. Click the **Customer starts with** tile to continue adding examples.

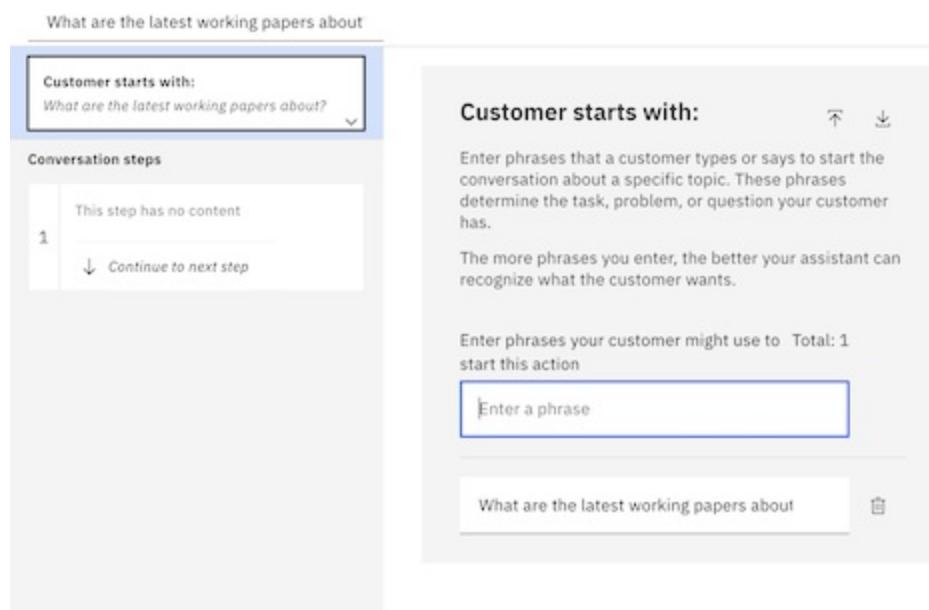


Figure 9. User question examples

5. Add the following questions:

Are there any working papers on the shipping industry?

Are there any papers that focus on inflation?

Are there papers about how trade policy affects pricing?

What's the latest research on municipal bond markets?

Figure 10. User examples list

6. Click the first step in the ***Conversation steps*** section.

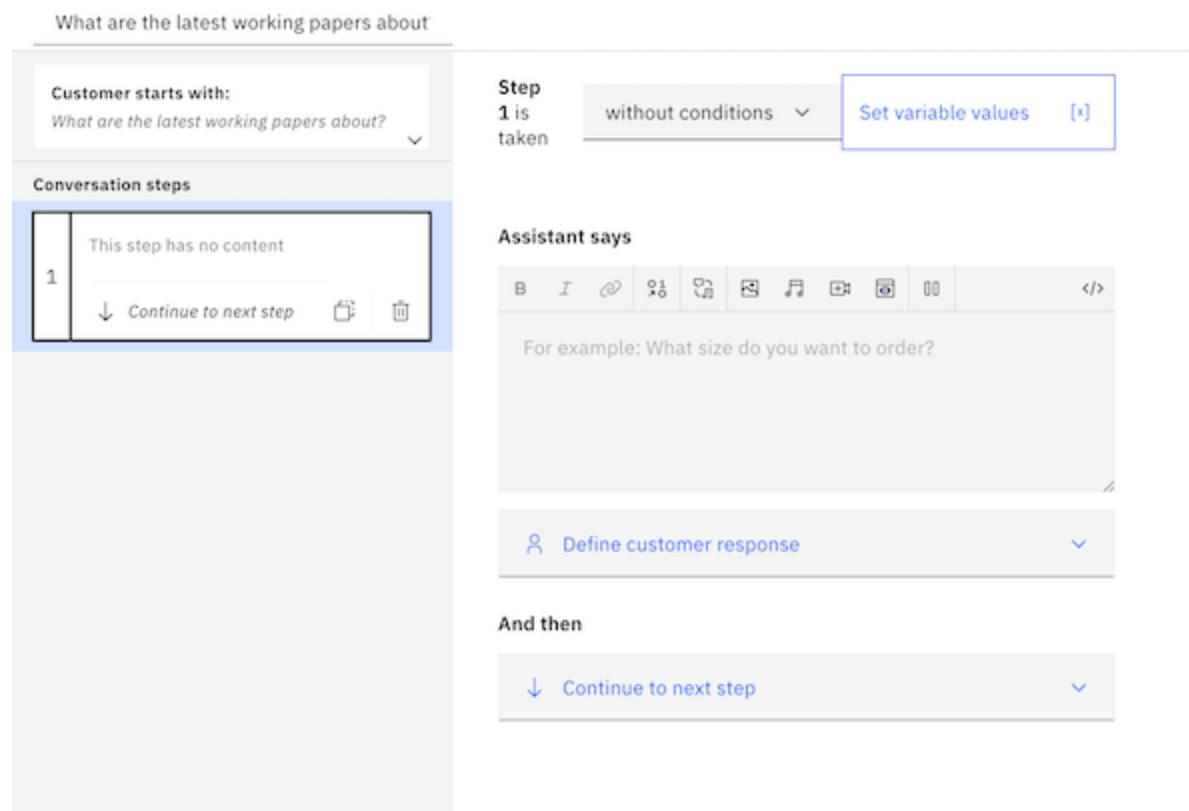


Figure 11. First step

7. Add the following text to the **Assistant says** field:

I'll check the Federal Reserve Economic Data website.

8. Do not add a customer response. Instead, in the **And then** section, click **Continue to next step**, and then choose **Search for the answer**.

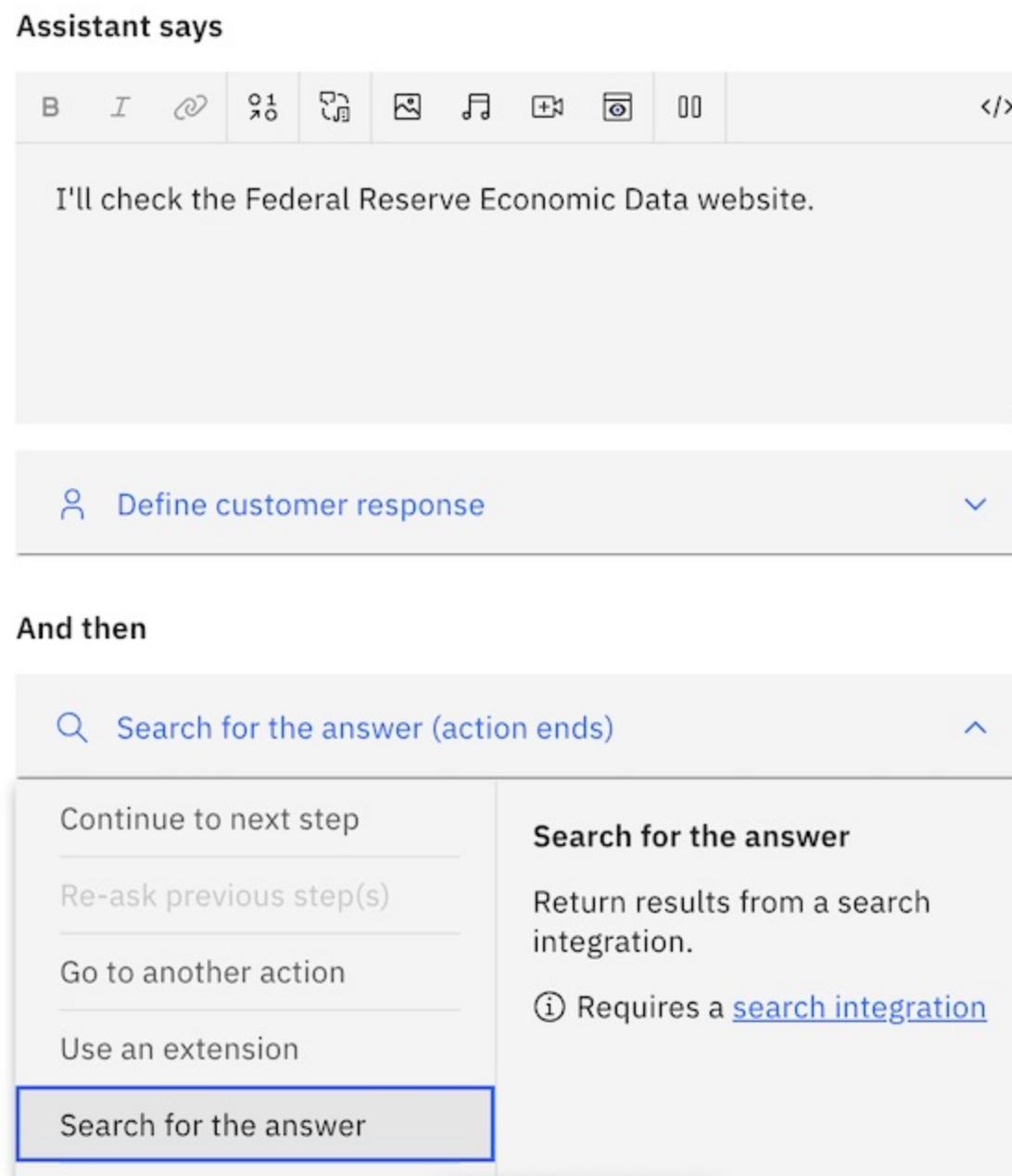


Figure 12. And then options

9. Click **Edit settings**.

And then

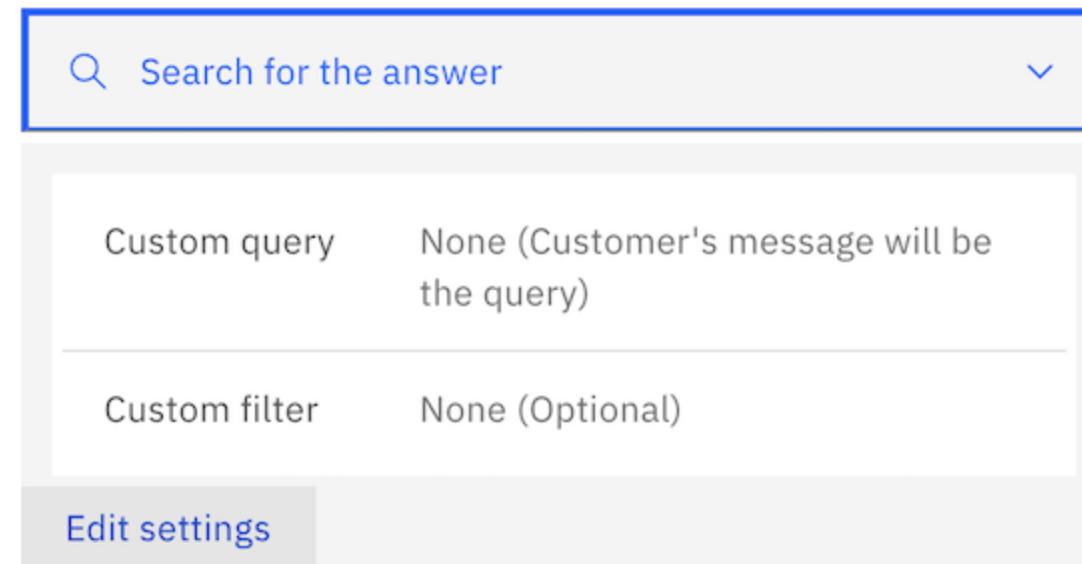


Figure 13. Search step

10. Select **End the action after returning results**, and then click **Apply**.

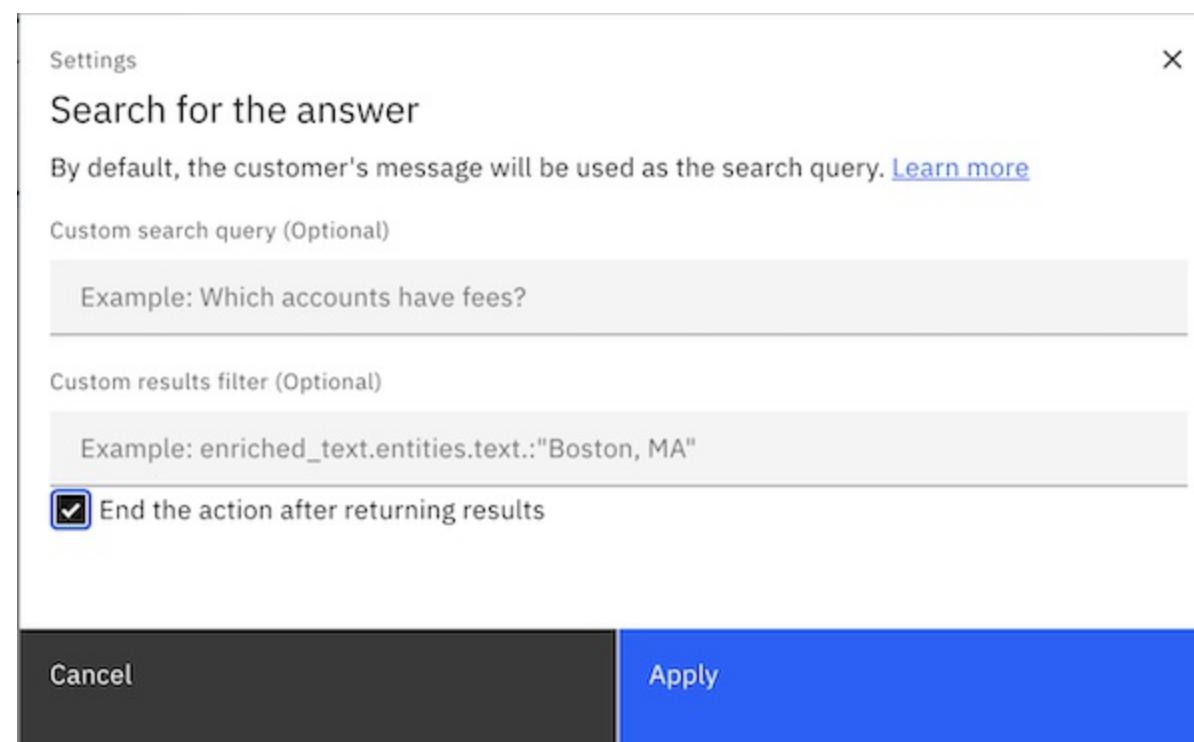


Figure 14. Search step settings

11. Save your changes, and then click the X to close the step.

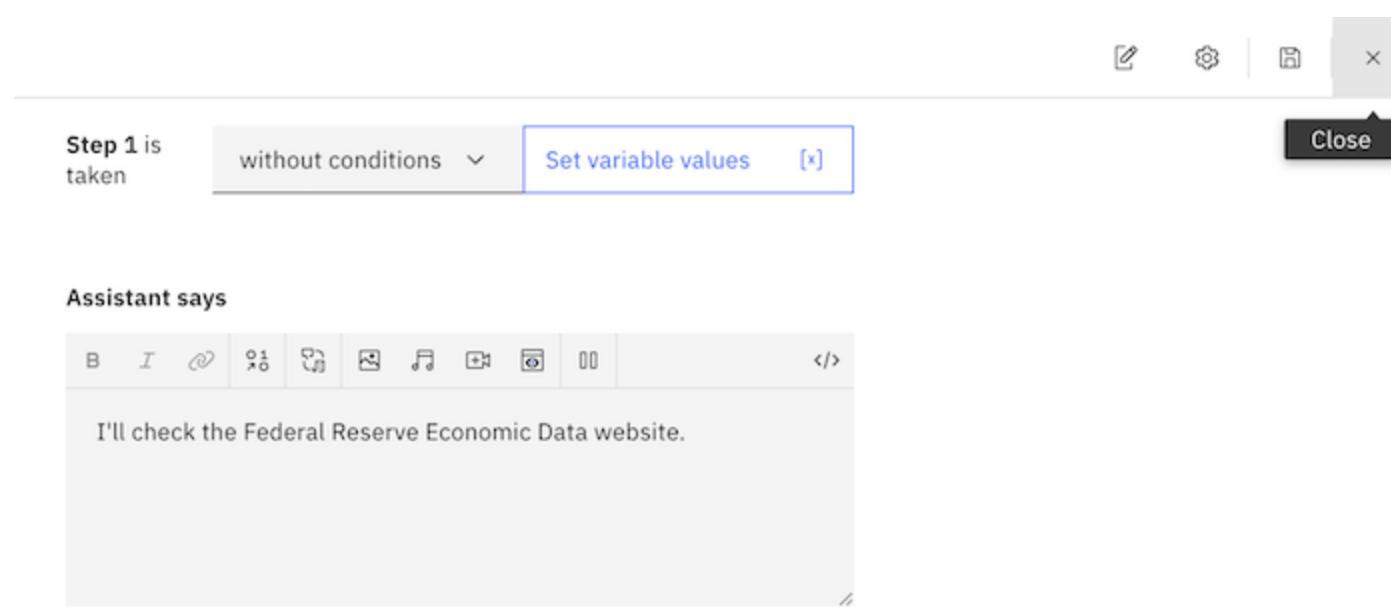


Figure 15. Close action

Congratulations! You successfully created an action that recognizes questions about FRED research papers and returns a search response.

The screenshot shows the IBM Watson Assistant interface. At the top, it displays a trial period of '27 days left' and an option to 'Extend trial'. Below this is a navigation bar with 'Learning center', a help icon, and a user profile icon. The main area is titled 'Actions' and contains a sidebar with categories like 'Actions', 'Variables', and 'Saved responses'. A list of actions is shown in a table with columns for 'Name', 'Last edited', 'Examples Count', and 'Status'. One action, 'What are the latest working papers about?', is highlighted.

Figure 16. Created action

In a later step, we will connect the search response in this action to a search extension that is configured for the assistant.

Step 3: Create a Conversational Search project

Now that the assistant can recognize questions about a subject, let's give it access to data from which it can retrieve accurate answers.

In Discovery, create a Conversational Search project type. This project type is optimized for retrieving answers during dialog-driven interactions. For example, unlike other project types, it does not apply prebuilt enrichments that aren't needed.

1. Open a new web browser page.

Tip: Keep the Watson Assistant page open in a separate tab so you can switch between the two applications.

2. From the Discovery Plus plan service page in IBM Cloud, click **Launch Discovery**.
3. From the **My Projects** page, click **New Project**.
4. Name your project **Federal Reserve research**, and then click the **Conversational Search** tile.

The screenshot shows the 'IBM Watson Discovery Plus' interface for creating a new project. At the top, there are tabs for 'IBM Watson Discovery Plus', 'Upgrade', and 'My projects', along with 'Share feedback' and 'Guided tours' buttons. Below this, there are four project type selection options: 'Select project type' (radio button selected), 'Select data source', 'Connect to data', and 'Configure collection'. The main area asks 'What type of project are you working on?'. It shows a 'Project name' input field containing 'Federal Reserve research'. Under 'Project type', three options are listed: 'Document Retrieval' (described as searching for relevant answers), 'Conversational Search' (selected, described as supplying answers to a virtual agent), and 'Content Mining' (Enterprise) (described as discovering hidden insights). A 'Next' button is at the bottom right.

Figure 17. Project type options

5. Click **Next**.

You'll configure the data source for the project in the next step.

Step 4: Connect to a website

We want the virtual assistant to be able to answer questions about the latest working papers from the US Federal Reserve, so we will connect our project to the Federal Reserve Economic Data website that hosts the working papers.

- From the **Select data source** page, click **Web crawl**, and then click **Next**.

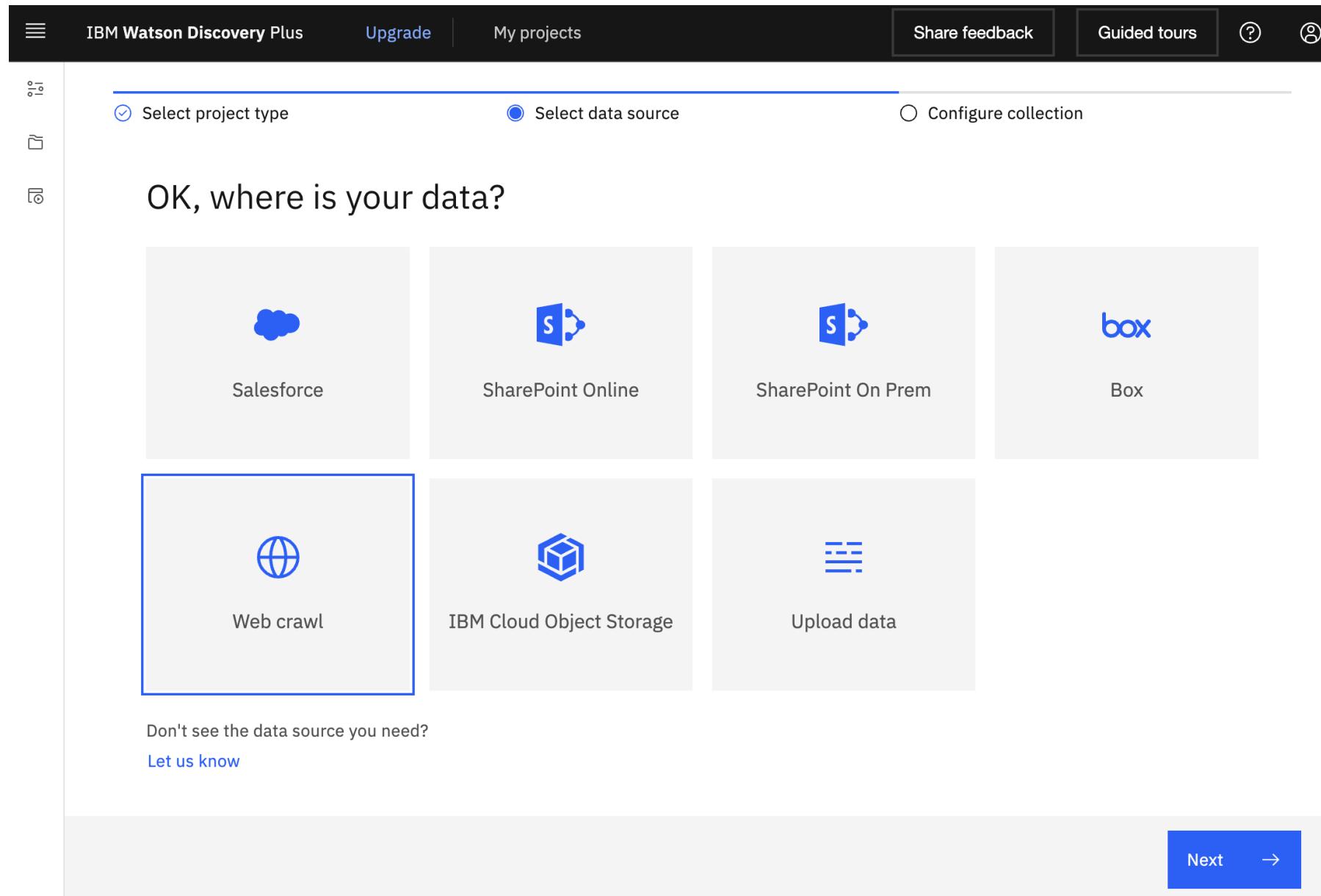


Figure 18. Data source options

- In the **Collection name** field, add **FRED papers**.

The screenshot shows the 'Configure collection' page for the Web crawl connector. The 'General' section includes the following fields:

- Data source: Web crawl
- Collection name: FRED papers (highlighted with a blue border)
- Collection language: English
- Crawl schedule: Weekly

Figure 19. Web crawl connector

- In the **Starting URLs** field, add the following URL:

```
https://research.stlouisfed.org/wp
```

You will add only one starting URL. In a real scenario, you might add multiple URLs that go to other pages with information about the same topic. By adding more URLs, you can expand the breadth of the expertise of your assistant.

4. Click **Add**.
5. Click the Edit icon for the URL that you just added.
6. In the **Maximum number of links to follow** field, change the value to 5.

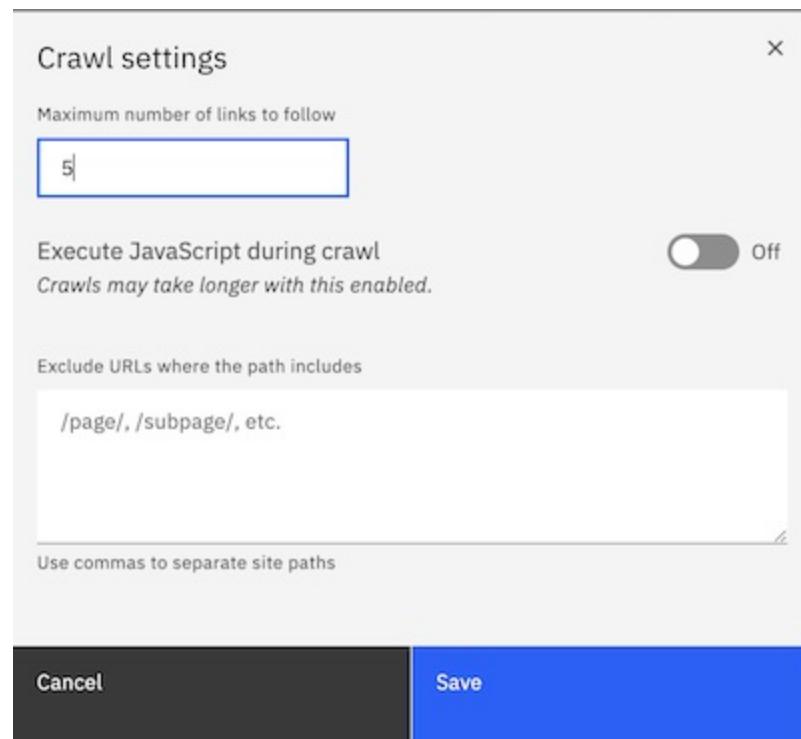


Figure 20. Starting URL settings

By changing the value to 5, you indicate that you want the service to process the page that you specified plus you want it to follow up to 5 links from the starting page.

7. Click **Save**, and then click **Finish**.

The Discovery service crawls the web page that you specified starting with the page that you specified as the starting URL.

While the website is being crawled and the data indexed, let's go back to our Watson Assistant service instance. It's time to connect the action that we created to this Discovery project.

Step 5: Add a search extension

Let's connect your assistant to your Discovery data.

1. From the navigation panel in Watson Assistant, click **Environments**.

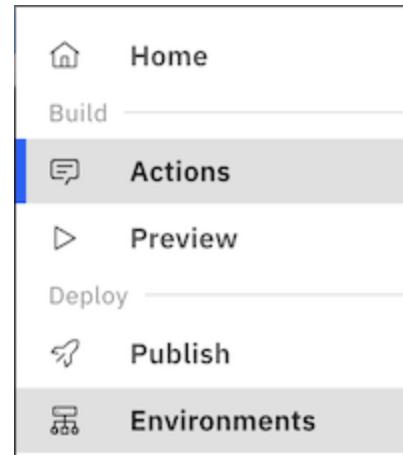


Figure 21. Environment menu

The draft environment is displayed. It shows that a web chat is connected to your assistant.

Environments

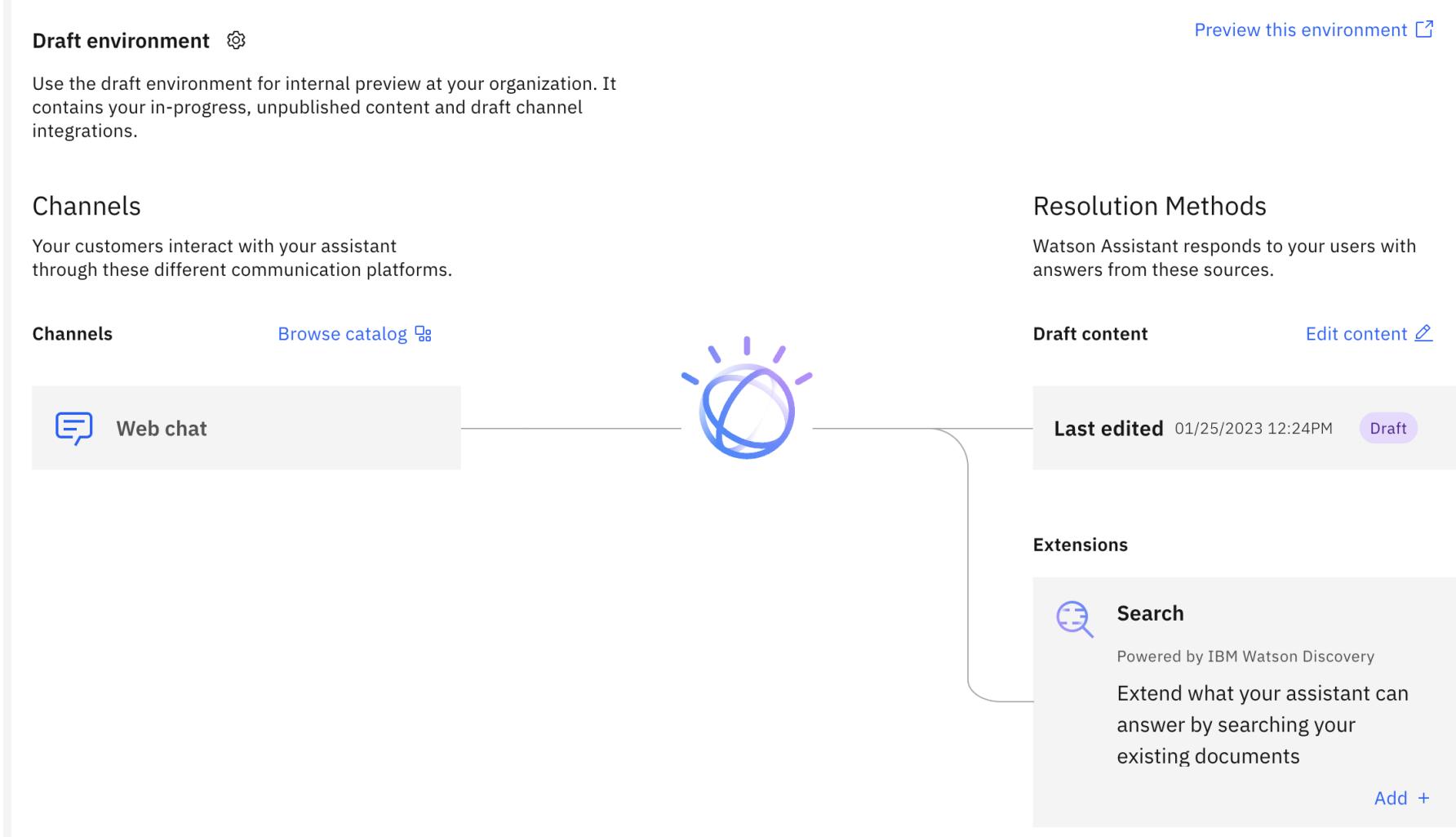


Figure 22. Draft environment diagram

2. Click the Web chat tile to edit the web chat.

We don't want to add multiple starter questions, so we are going to turn off the home screen for the web chat. Click the **Home screen** tab.

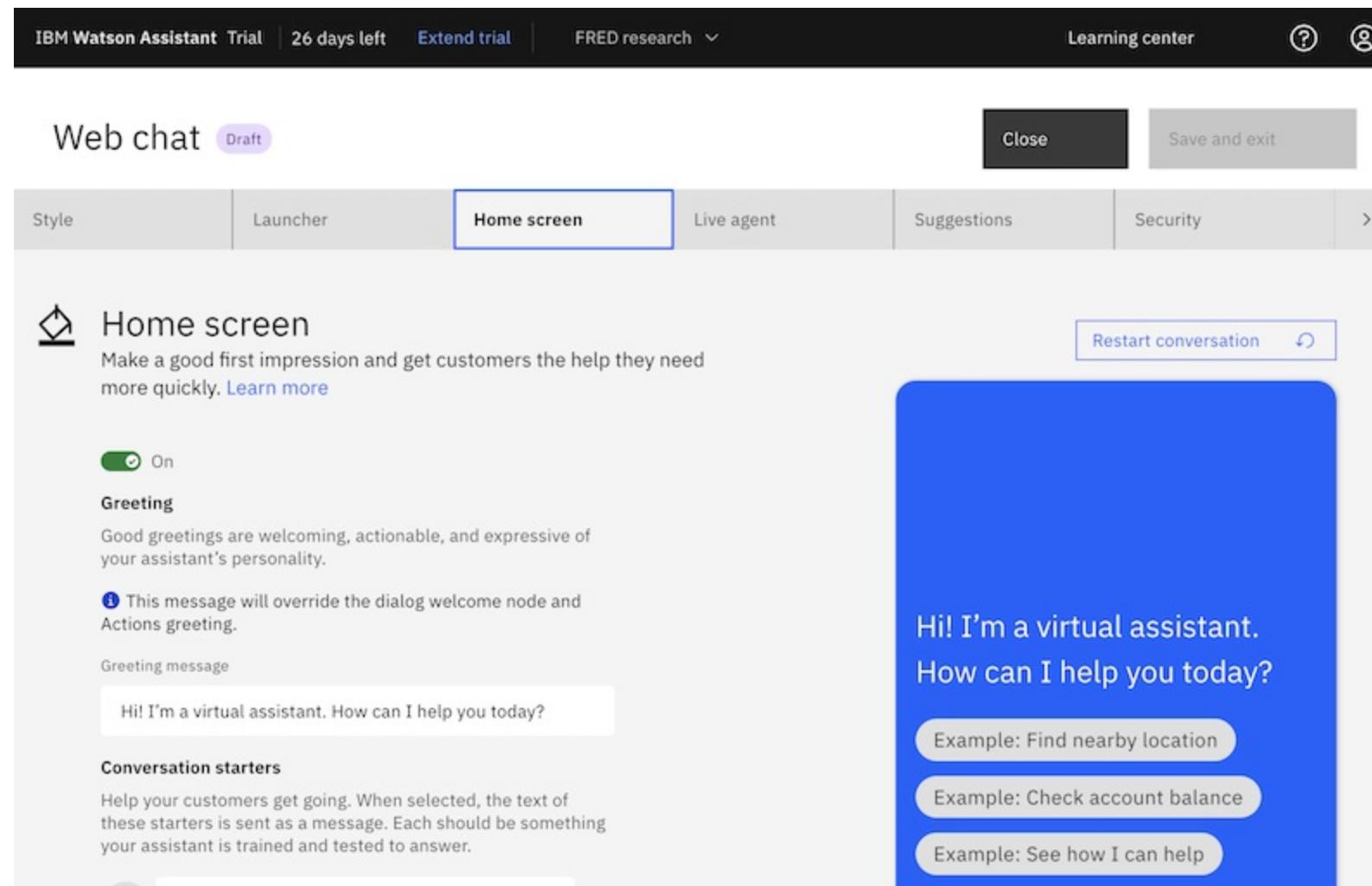


Figure 23. Web chat home screen configuration

3. Set the switcher to **Off**, and then click **Save and exit**.

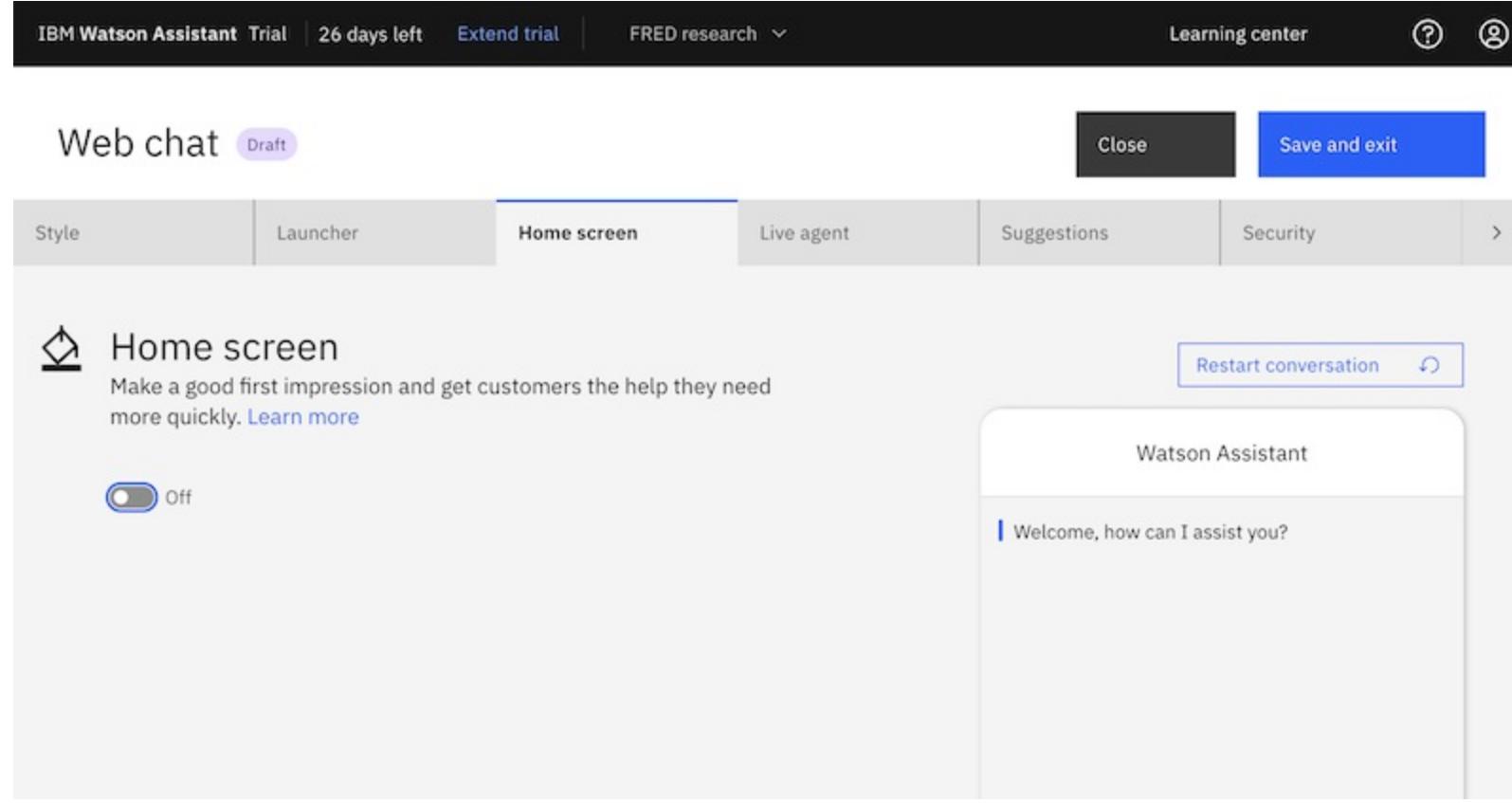


Figure 24. Web chat home screen disabled

4. We want to connect the web chat to a search extension. Click the **Add** button in the Search extension tile.

Environments

Draft environment ⊗ Preview this environment ↗

Use the draft environment for internal preview at your organization. It contains your in-progress, unpublished content and draft channel integrations.

Channels
Your customers interact with your assistant through these different communication platforms.

Resolution Methods
Watson Assistant responds to your users with answers from these sources.

Channels	Browse catalog ↗	Draft content	Edit content ↗
Web chat		Last edited 01/25/2023 12:24PM	Draft

Extensions

Search
Powered by IBM Watson Discovery
Extend what your assistant can answer by searching your existing documents Add +

Figure 25. Search extension in draft environment

The Search Integration page is displayed.

5. Select the Discovery instance where your project is stored, and then select the **Federal Reserve research** project that you created earlier. Click **Next**.

Project name	Collection Name
<input type="radio"/> Discovery docs	PDF1
<input checked="" type="radio"/> Federal Reserve research	Federal Reserve publications
	FRED papers

Figure 26. Search extension configuration

6. The default result content configuration uses the best fields; you don't need to change them.
7. In the *Define the text your search will display to the end user* section, edit the content to show the following message:

The Federal Reserve Economic Data website has this information:

Verify that the *Emphasize the answer* switch is set to **On**. This setting adds the `find_answers:true` parameter to the query request. As a result, a succinct answer to the query is shown in bold in the response that is returned by the assistant.

Figure 27. Search extension settings configured

8. Click **Create**.

Step 6: Preview the assistant

To preview an assistant that connects to data that is stored in Discovery, you must preview the assistant from the Environments page. When you test it separately, the assistant is not able to retrieve data from Discovery.

1. From the Environments page, click **Preview this environment**.

The screenshot shows the IBM Watson Assistant Trial interface. At the top, it displays "IBM Watson Assistant Trial | 26 days left | Extend trial | FRED research". On the right, there are links for "Learning center", a question mark icon, and a user profile icon. The main area is titled "Environments" and has tabs for "Draft" and "Live". A button "Add Environment" with a plus sign is visible. Below the tabs, a section titled "Draft environment" contains a brief description: "Use the draft environment for internal preview at your organization. It contains your in-progress, unpublished content and draft channel integrations." To the right of this is a "Preview this environment" button. On the left, there's a sidebar with icons for environments, channels, and other settings. The main content area includes sections for "Channels" (with a "Web chat" option), "Resolution Methods", "Draft content" (last edited 03/20/2023 12:04P...), and "Extensions" (Search).

Figure 28. Search extension enabled

A sample web page is displayed that includes a chat icon.

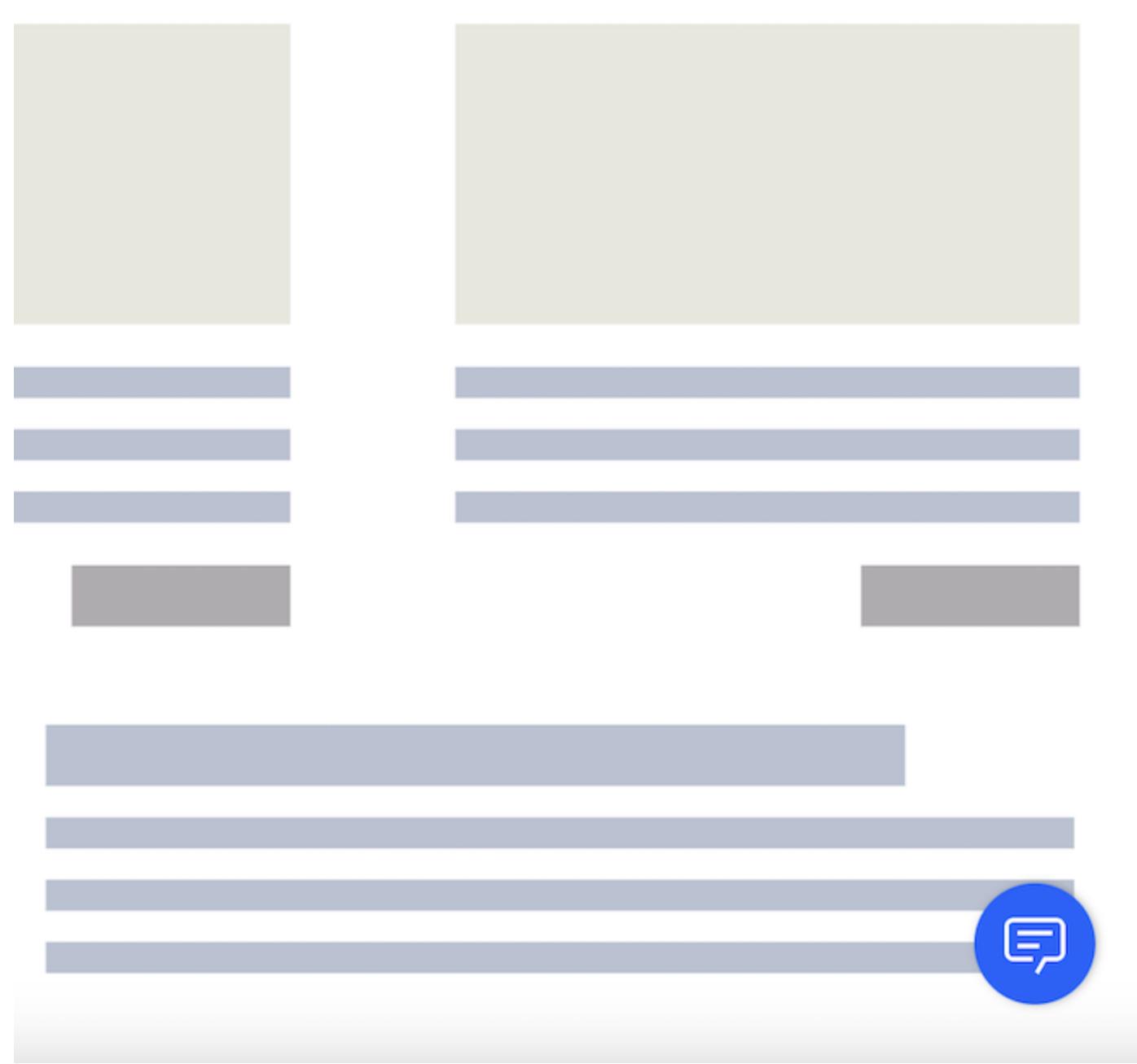


Figure 29. Web chat icon

2. Click the chat icon to open the web chat window.

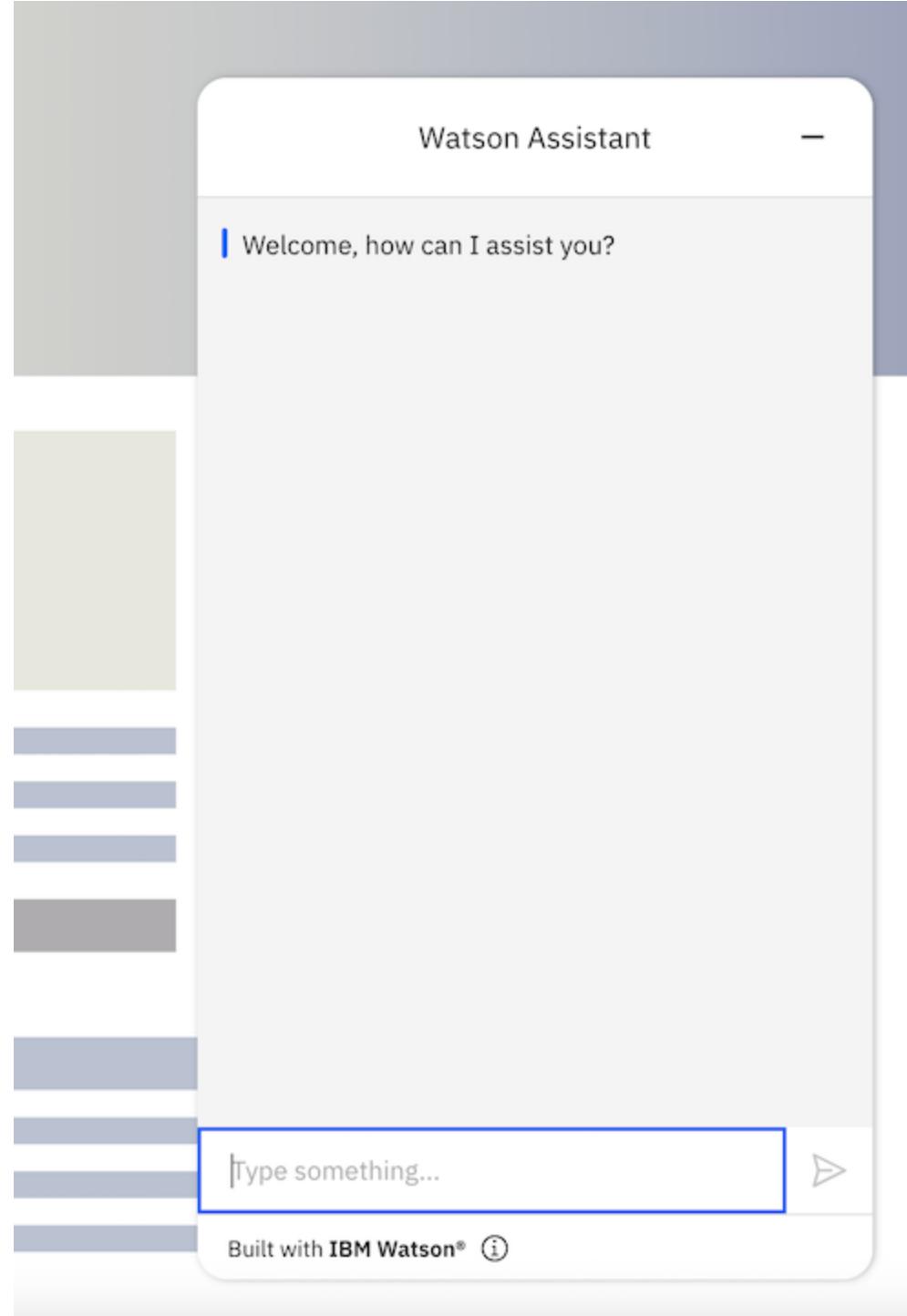


Figure 30. Web chat welcome message

3. Enter the following text question:

What impact is inflation having on the real estate market?



Note: This test question is not one of the questions that we used to train the assistant.

The correct answer is returned and it includes a link to the source documentation page.

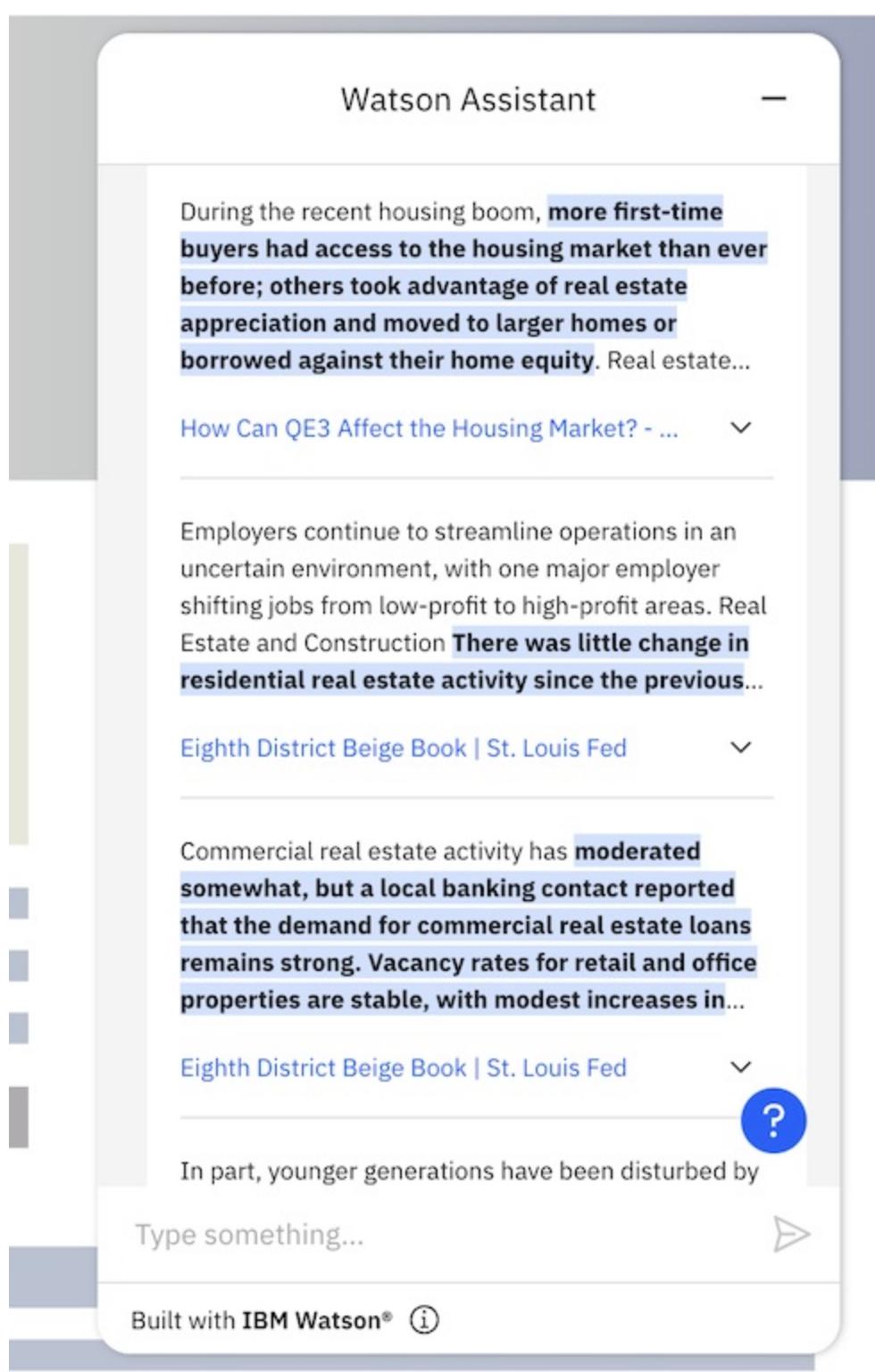


Figure 31. Web chat returns search response

Congratulations! You successfully created an assistant that can answer questions about economic topics by retrieving information from working papers that are available from the US Federal Reserve Economic Data website.

Summary

In this tutorial, you created a Watson Discovery Conversational Search project with a web crawl connector that collects information about working papers from the US Federal Reserve Economic Data website. Separately, you created a Watson Assistant virtual assistant with a single action that can recognize user questions about economic subjects. You added a Search extension to your assistant that connects the action's search response to the Discovery project where economic data is stored. Finally, you tested your virtual assistant by asking a question and getting a useful response that featured data from relevant economic research papers.

Next steps

The assistant that you created and connected to a search extension is available from the Draft environment. Next, you can publish your assistant to a production environment and deploy it. There are a variety of methods you can use to deploy the assistant. For more information, see [Overview: Previewing and publishing](#).

Use Smart Document Understanding (SDU) to improve search results

In this tutorial, you use the Smart Document Understanding feature of the Discovery service to create a user-trained Smart Document Understanding (SDU) model. You then split a single document into many smaller documents so that some types of answers are easier to find.



Note: This tutorial works with both managed and installed deployments.

Learning objectives

By the time you finish the tutorial, you will understand how to:

- Create a Document Retrieval project in Discovery.
- Upload a PDF document to your Discovery project.
- Use the Smart Document Understanding (SDU) tool to create a user-trained SDU model.
- Split a document into smaller, more consumable chunks.

Duration

This tutorial takes approximately 3 hours to complete.

Prerequisite

1. Before you begin, you must set up a paid account with IBM Cloud.

You can complete this tutorial at no cost by using a Plus plan, which offers a 30-day trial at no cost. However, to create a Plus plan instance of the service, you must have a paid account (where you provide credit card details). For more information about creating a paid account, see [Upgrading your account](#).

2. Create a Plus plan Discovery service instance.

Go to the [Discovery resource](#) page in the IBM Cloud catalog and create a Plus plan service instance.



Important: If you decide to stop using the Plus plan and don't want to pay for it, delete the Plus plan service instance before the 30-day trial period ends.

Step 1: Create the Document Retrieval project

Create a project. Choose to create a Document Retrieval project type. This type is optimized for finding answers that are returned as passages from large documents.

For more information about project types, see [Creating projects](#).

1. From the Discovery Plus plan service page in IBM Cloud, click **Launch Discovery**.
2. From the **My Projects** page, click **New Project**.
3. Name your project **Finance tutorial project**, and then click the **Document Retrieval** tile.

The screenshot shows the 'Select project type' step of the Discovery service setup. At the top, there are four tabs: 'Select project type' (which is selected), 'Select data source', 'Connect to data', and 'Configure collection'. Below the tabs, the question 'What type of project are you working on?' is displayed. A project name 'Finance tutorial project' is entered in the 'Project name' field. Under 'Project type', three options are shown: 'Document Retrieval' (selected, with a description: 'Search and find the most relevant answers from your data.'), 'Conversational Search' (with a description: 'Supply answers to a virtual agent built with IBM Watson Assistant.'), and 'Content Mining' (with a description: 'Discover hidden insights, trends, and relationships in your data.'). At the bottom, there is an unselected option 'None of the above – I'm working on a custom project'. A 'Next' button is located at the bottom right.

Figure 1. Project type options

4. Click **Next**.

You'll configure the data source for the project in the next step.

Step 2: Upload a PDF file

We want the search application to be able to answer questions about algorithmic trading. Therefore, we are adding the "Staff Report on Algorithmic Trading in US Capital Markets" PDF that was created on 5 August 2020 as a data source for the project.

1. Get a copy of the PDF so that you can upload it to your project. You can download the file from the [US Securities and Exchange Commission](#) website.
2. From the **Select data source** page, click **Upload data**, and then click **Next**.

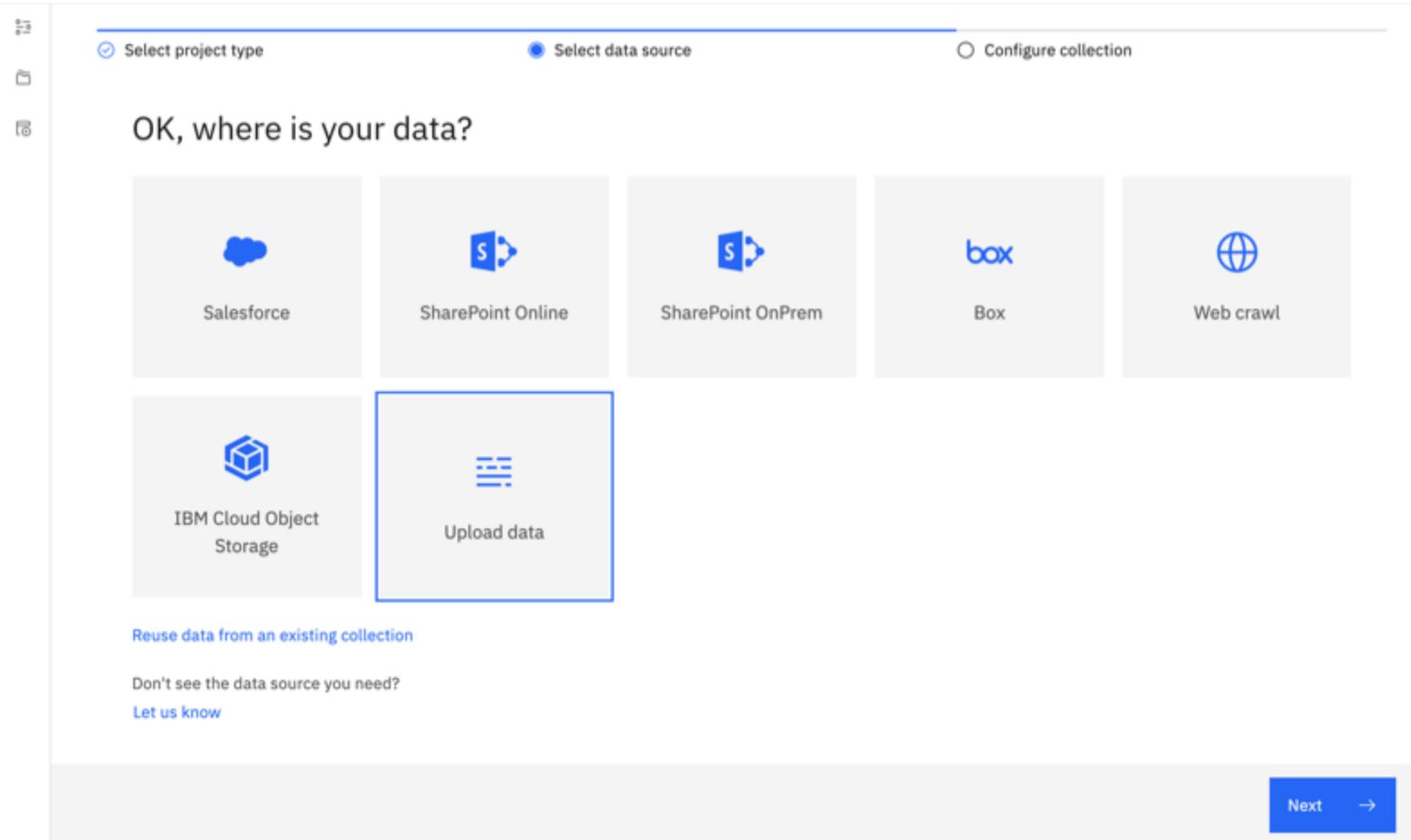


Figure 2. Data source options

3. In the **Collection name** field, add **Algorithmic Trading PDF**, and then click **Next**.

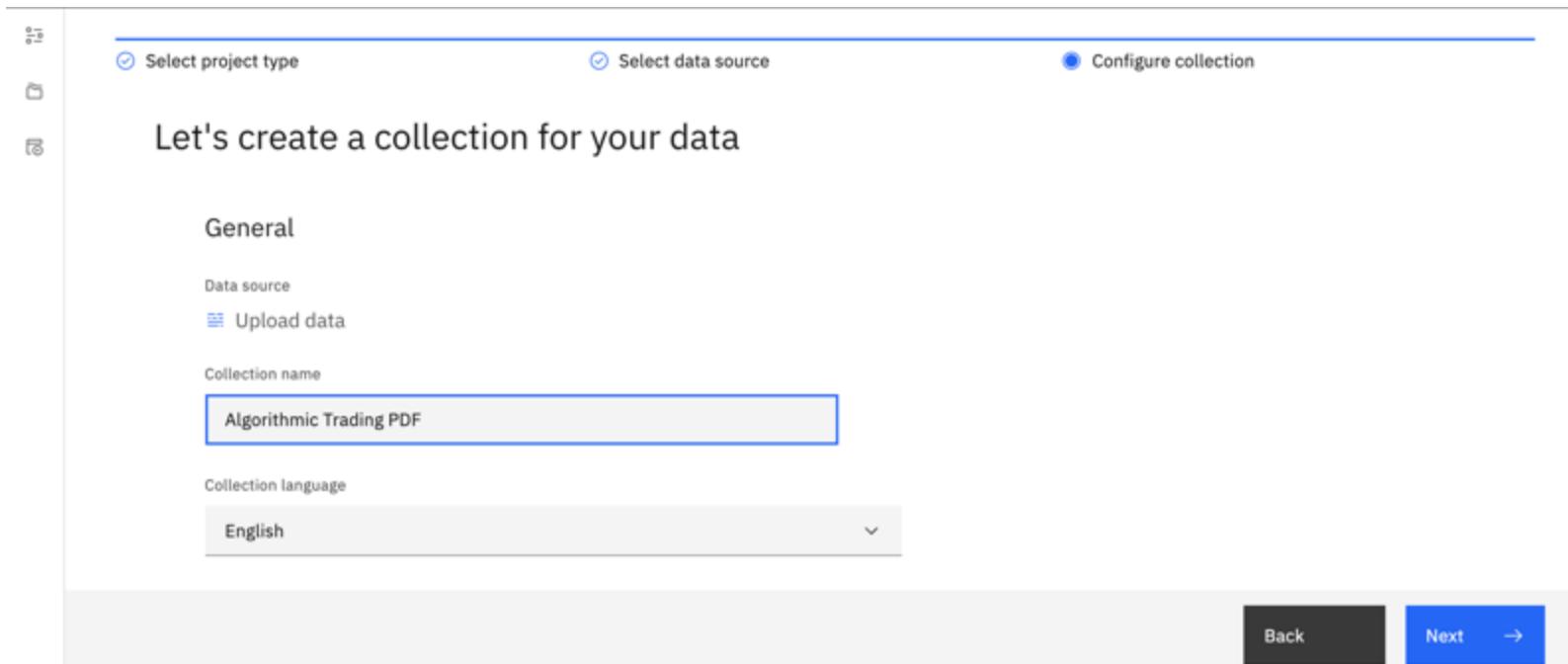


Figure 3. Uploaded data collection name field

4. Drag the file that you downloaded to the page and drop it into the tile with the **Drag and drop files here or upload** link.

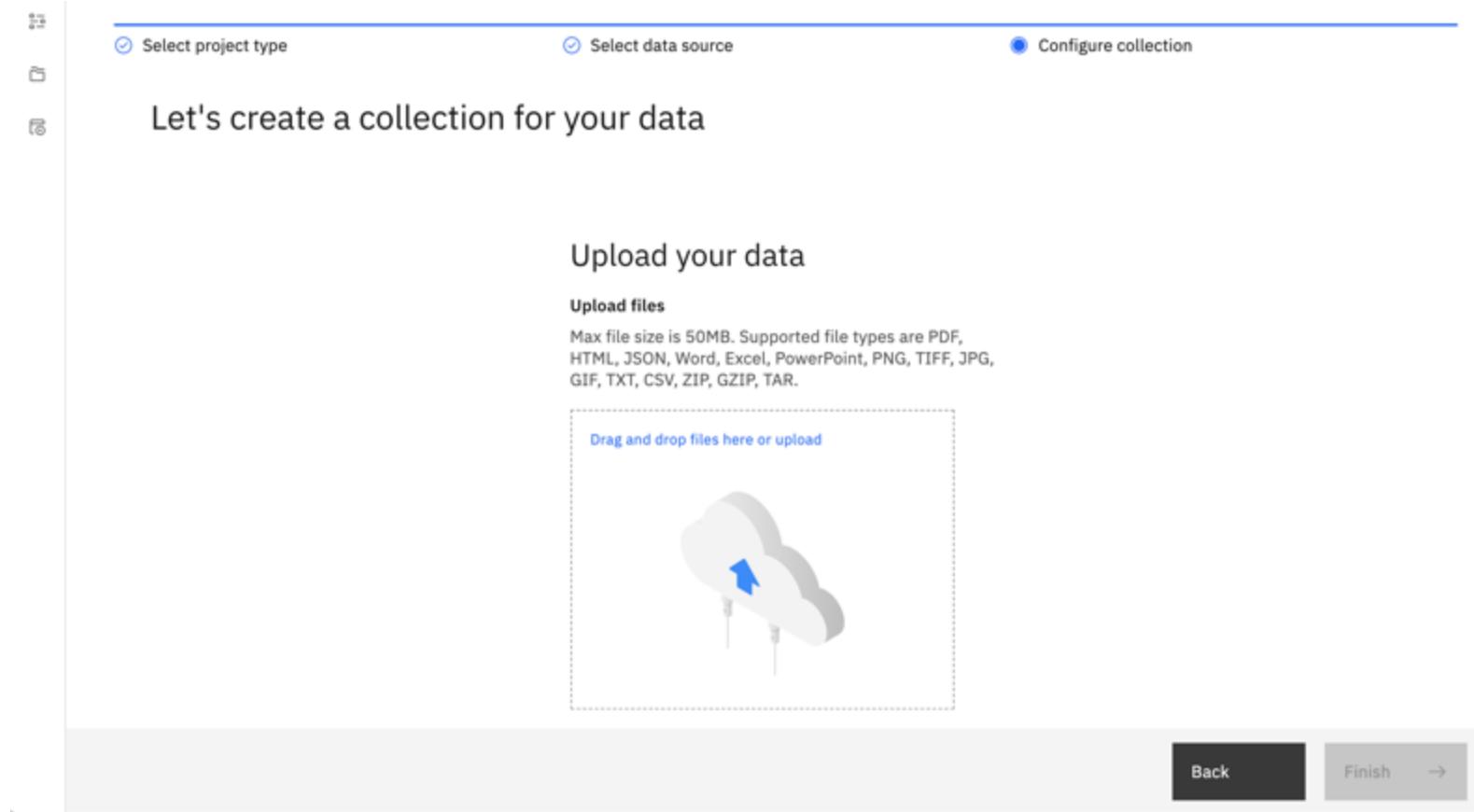


Figure 4. File upload dialog

5. Click **Finish**.

You add only one file. In a real scenario, you might upload multiple files with information about the same topic. By adding more files, you can expand the breadth of the information that your search application can leverage.

The service uploads the document. As it uploads the document, Discovery crawls the data and indexes key information. Because you created a Document Retrieval project type, Discovery makes a note of the **Parts of speech** and **Entities** information that it finds and recognizes as it crawls the document.

Step 3: Review the document

Analyzing and indexing the document can take a few minutes. While the processing is under way, review the source document to get a feel for its content. It is a good idea to understand the structure of your own documents before you use the Smart Document Understanding tool to annotate them.

Smart Document Understanding (SDU) uses visual imaging technologies to understand the structure of a document by analyzing the format and positioning of the text. You label sections of the document, such as subtitles or tables, to teach Discovery to recognize the sections. You can also label sections that you want the search function to ignore. For example, you might not want to search page footers or the table of contents information. After you teach the SDU tool to recognize footers, for example, you can exclude the footer field from the index.

1. Monitor the progress of collection processing by opening the **Activity** tab.

Click **Manage collections** from the navigation panel.

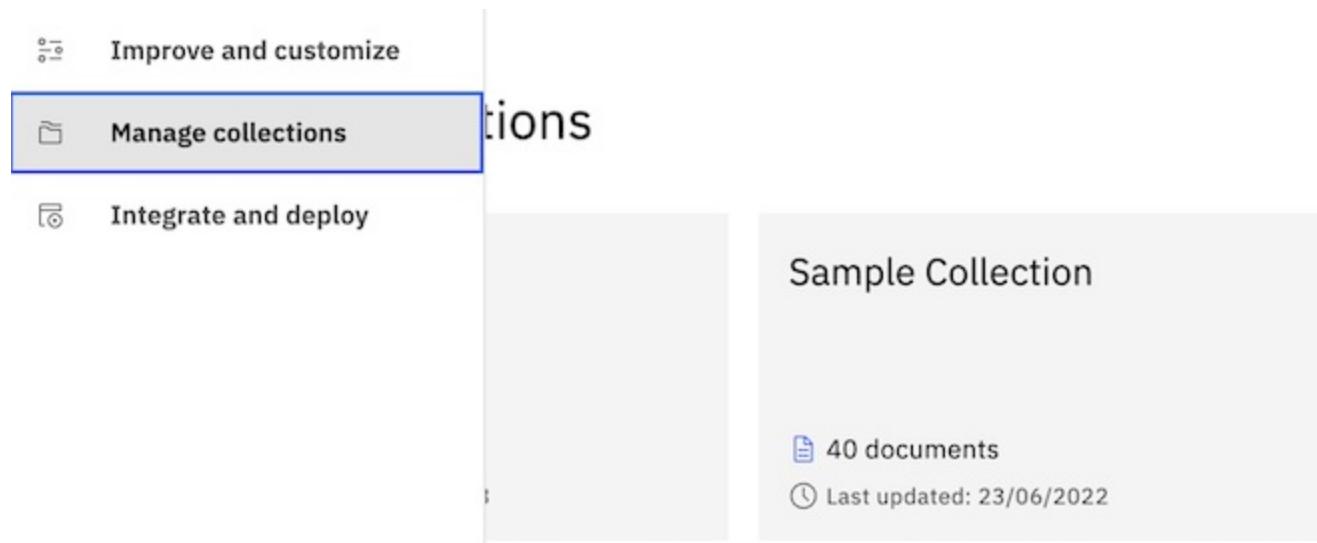


Figure 5. Manage collections menu option from the navigation panel

Click the **Algorithmic Trading PDF** collection tile. The collection opens to the Activity tab.

2. While you wait for the collection to be ready, open the **Algo_Trading_Report_2020.pdf** file that you downloaded previously.
3. Review the structure of the document.

Notice that the document consists primarily of the following structures:

- Title

- Table of contents
- Subtitles
- Text
- Footnotes
- Bibliography

4. The SDU tool has predefined labels for all but the **footnotes** and **bibliography**. You will create new field labels for these two document structures in a later procedure.

Processing is finished when the page shows that one document is available.

The screenshot shows the SDU Activity page for a project titled "Algorithmic Trading". The navigation bar includes "Manage collections" and tabs for "Activity", "Identify fields", "Manage fields", and "Enrichments". A status message indicates the collection was last updated on 12/28/2021, 11:21:32 AM EST. Below this, it shows 1 document available and 0 warnings or errors. A section titled "Warnings and errors at a glance" contains a sun icon and the text "Nothing to report", with a note that errors will appear here if any occur.

Figure 6. Activity page that shows the data upload is finished

Step 4: Test your project

1. After the crawl is completed, go to the **Improve and customize** page. From the navigation panel, click **Improve and customize**.
2. In the **Search** field, enter **When did the Flash Crash occur and why?**

The following passage is returned as the response:

These could in turn generate systemic destabilizing market events, such as the May 2010 “Flash Crash.” The “Flash Crash” occurred on May 6, 2010, when an algorithm rapidly sold 75,000 S&P500 e-mini futures contracts.

The returned passage contains an accurate answer to the question.

Figure 7. Search results

3. Ask another question, **What is the purpose of Rule 15c3-5?**

The following passage is returned as the response:

mechanism.306 b. 15c3-5 In November 2011, the SEC implemented the final provision of Rule 15c3-5 curbing unfiltered market access. The provision mandated that brokers verify their clients' order flow for compliance with credit and capital thresholds before routing to market centers

Again, the answer is accurate (despite there being some extraneous text at the beginning of the passage).

In both examples, a somewhat complex question is asked and the passage that is returned provides a valid answer.

However, not every question returns as clear an answer. Next, we try some queries that generate answers we might want to improve.

4. Enter **Where do muni bond trades get reported to?**

In this case, the response does not answer the question completely.

Post-trade transparency, in the form of transaction reports, generally is available for corporate and municipal bonds. 1. Transaction Reports in Corporate Bonds: TRACE Transactions in corporate bonds must be reported to the Trade Reporting

5. Similarly, the search query, **What are PTFs?**, does not return a direct answer.

Despite the surge in trading volume during the event window, there was no noticeable change in net positions of PTFs or bank-dealers. However, the report also finds evidence that some PTFs and bank-dealers may have contributed to the volatility

Your project is answering some of the questions successfully. Only one passage is being returned for each query. Let's see whether we can improve the responses that are given to these simpler search queries.

Step 5: Create a user-trained Smart Document Understanding (SDU) model

To improve the quality of the search results, build a Smart Document Understanding model for this document. The model helps Discovery understand the document structure. You can then instruct Discovery about which sections of the document to search and which sections to ignore.

1. From the **Improvement tools** panel of the **Improve and customize** page, expand **Define structure**, and then click **New fields**.

The screenshot shows a search interface with a list of results. To the right, the 'Improvement tools' panel is displayed, specifically the 'Define structure' section. The 'New fields' option is circled in yellow.

Figure 8. New fields tool in the Improvement tools panel

2. The *Identify fields* tab is displayed, where you can choose the type of Smart Document Understanding model that you want to use.

The screenshot shows the 'Identify fields' tab in the collection settings. It lists three options: 'Text extraction only (default)', 'User-trained models', and 'Pre-trained models'. The 'Text extraction only' option is selected. A note at the bottom states: 'Note: If OCR is enabled for the collection, text from images will also be extracted.'

Figure 9. Identify fields tab

- The *pretrained model* applies a noncustomizable model that extracts text and identifies tables, lists, and sections. The pretrained model is a great choice to save time.
- For the purposes of this tutorial, where we want to explore how the Smart Document Understanding tool works, we'll choose to use the *user-trained model*.

If you don't choose a model, the *text extraction* model is applied automatically. With the text extraction model, most of the document content is treated as standard text and is indexed in the `text` field.

3. Click **User-trained models**, and then click **Submit**.

The screenshot shows a confirmation dialog box asking if the user wants to switch to a User-trained model. The dialog includes a note about the change being irreversible and a 'Submit' button.

Figure 10. Confirmation dialog for user-trained model

4. Click **Apply changes and reprocess**.

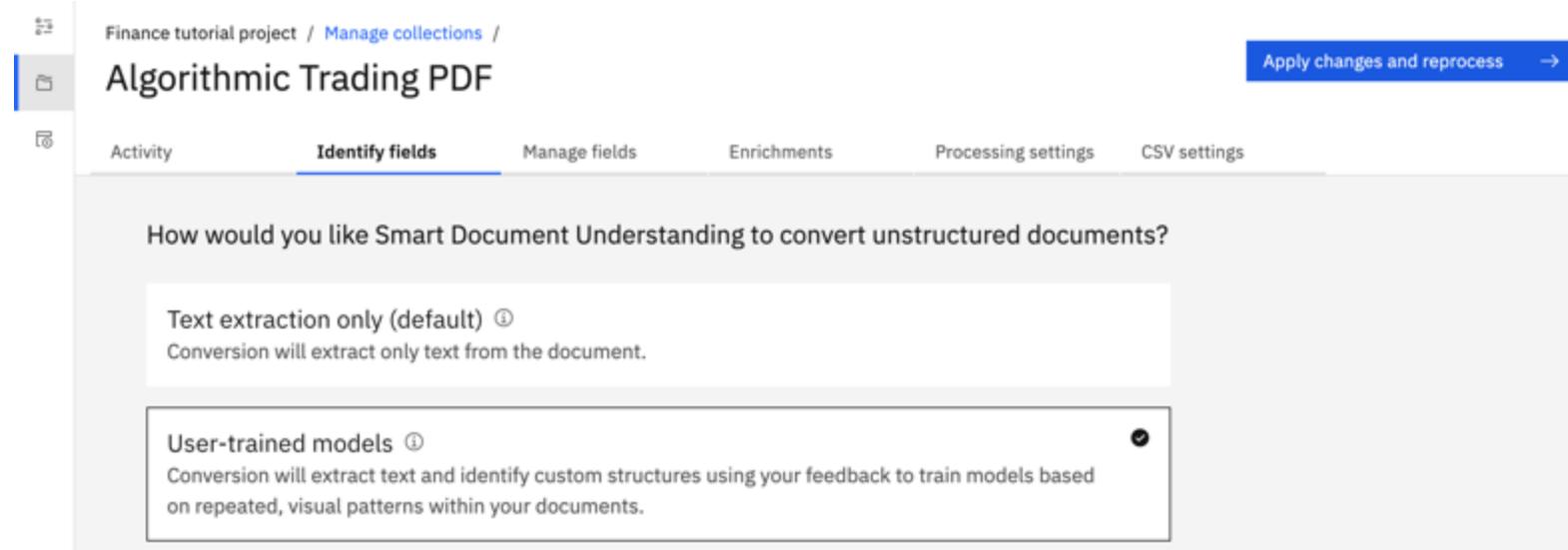


Figure 11. Apply changes and reprocess button

After the evaluation process is complete, a representation of the document is displayed in the Smart Document Understanding tool.

Field labels

Identify document elements using the labels below.

- + Create new
- answer
- author
- footer
- header
- question
- subtitle
- table_of_contents
- text
- title
- table
- image

Figure 12. PDF is displayed in the SDU tool

The tool shows you a view of the original document along with a representation of the document, where the text is replaced by blocks. The blocks represent field types.

Initially, the blocks are all the color of the **text** field label because all of the document content is considered to be standard text and will be indexed in the **text** field.

A **Field labels** list shows the predefined field labels that are available.

We are going to label blocks that represent specific types of information, such as titles and subtitles, with corresponding field labels. (The process of using labels to identify different parts of the document's structure is called **annotating** the document.)

- To annotate the document, click the label first. Then, click the block of text that you want to label.

Click **title** from the **Field labels** list, and then, in the document representation, click the yellow block that is situated in the location of the document title.

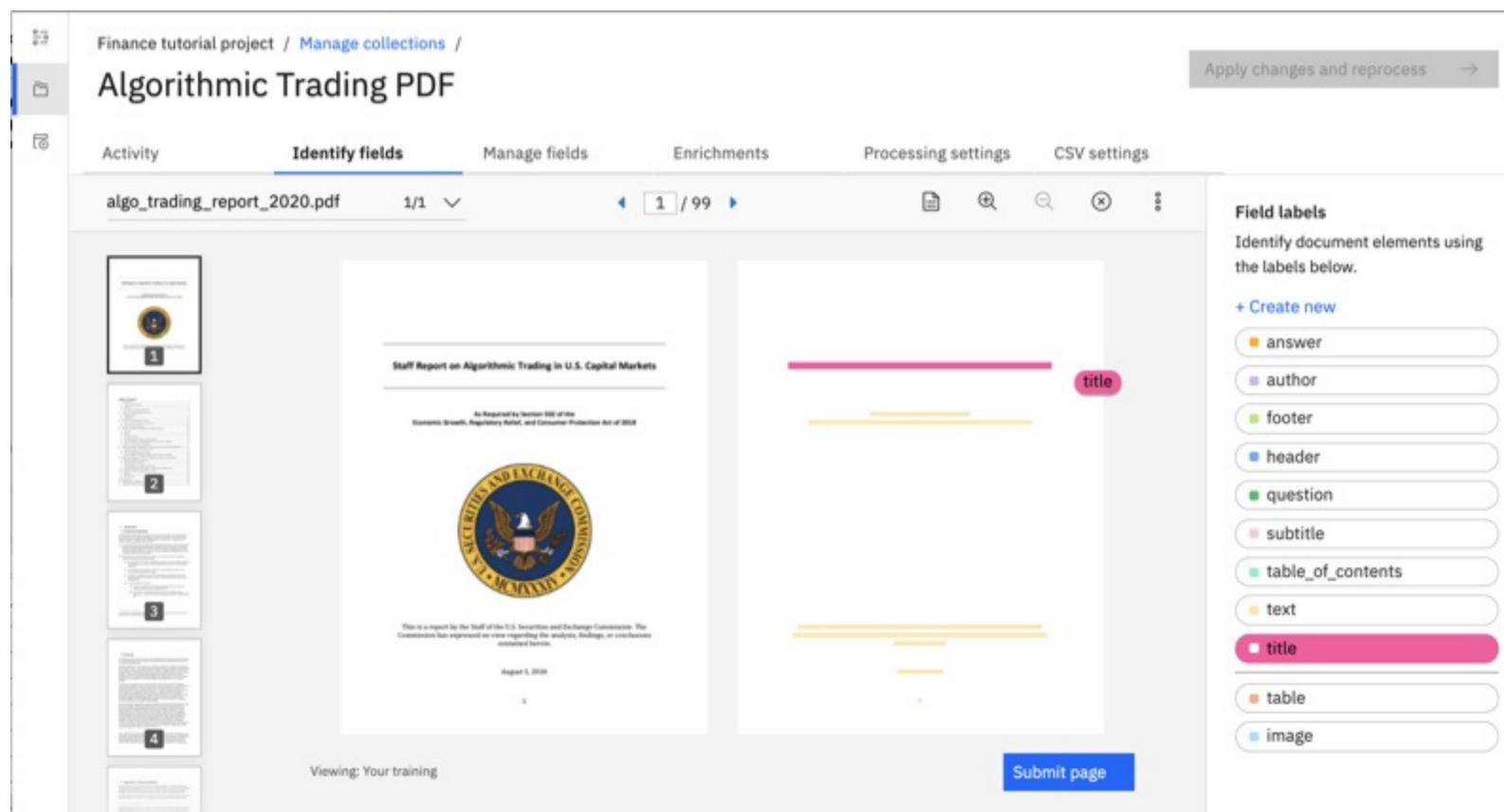


Figure 13. A title is being labeled in the Smart Document Understanding tool

You labeled the title of the document successfully!

6. The rest of the text on the page can be indexed as part of the **text** field. Therefore, click **Submit page**.
7. The next page is the **Table of contents** for the document. Click the **table_of_contents** label, and then select all of the text on the page to label it. (You can click and drag the mouse to select all.) Click **Submit page** to move to the next page.

Figure 14. A table of contents is being labeled in the Smart Document Understanding tool

8. The two headings on the page are subtitles. Click the **subtitle** label, and then select the headings.

This page has a footnote. As we noted earlier, the document has many footnotes where some important information is provided. Let's label the footnotes so we can include or exclude this type of information later. There is no footnote label, so we must add one.

9. From the **Field labels** list, click **Create new**. Add the name **footnote** as the label name. Click the color block repeatedly until you find a unique color to use for the label, and then click **Create**.

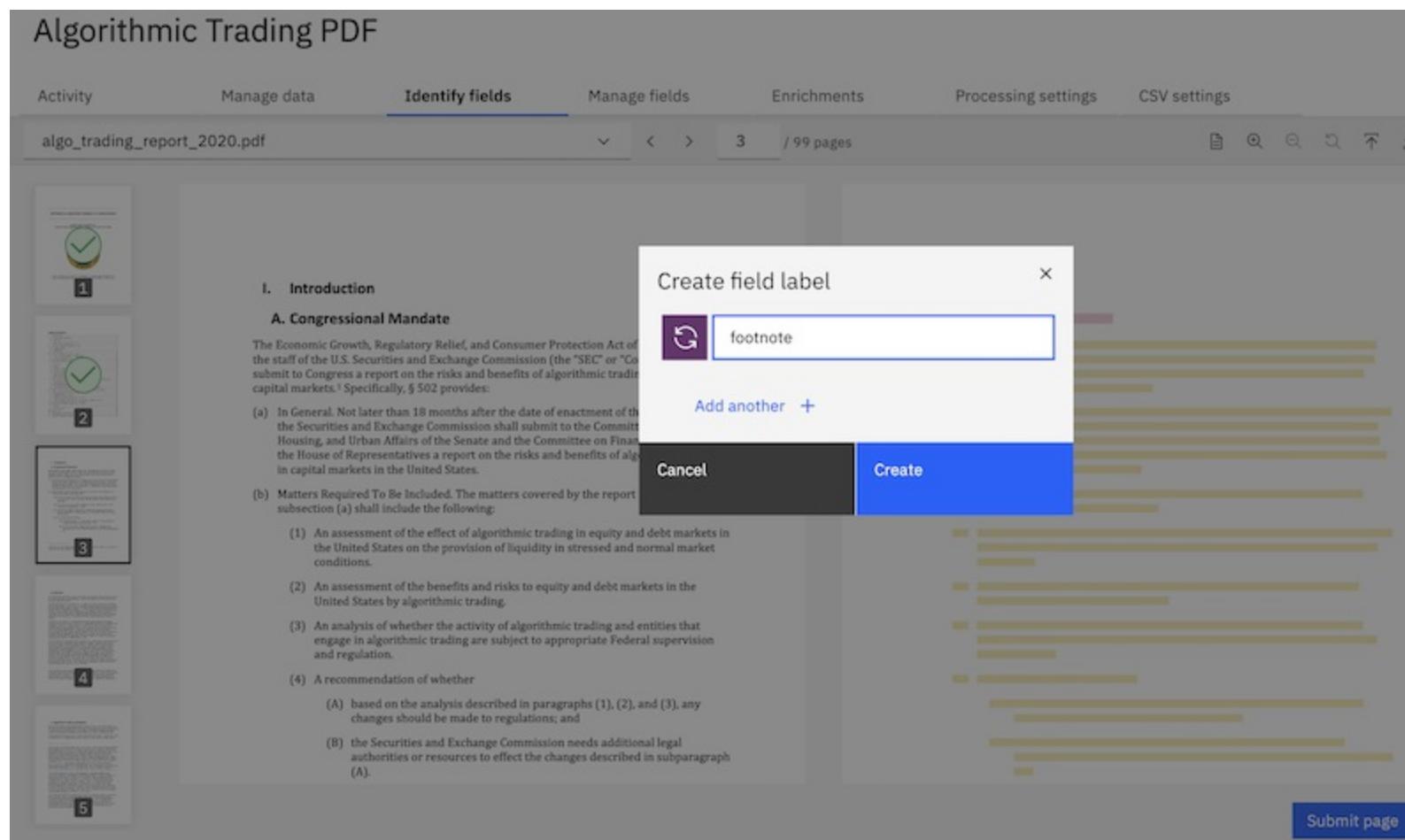


Figure 15. New label dialog

- Click the new footnote label that you added, and then label the footnote on the page with the label. Click **Submit page** to move to the next page.

Figure 16. A footnote is being labeled in the Smart Document Understanding tool

- Repeat this process to label and submit multiple pages.

For most pages, the content includes a **subtitle**, a **footnote**, and the bulk of the content on the page remains labeled as **text**.

The screenshot shows the 'Identify fields' interface for a PDF titled 'Algorithmic Trading PDF'. The interface includes a sidebar for 'Field labels' where the 'footnote' label is highlighted. The main area shows a preview of page 11, which contains several footnotes. A purple bar highlights the 'footnote' label in the sidebar.

Figure 17. Subtitle, footnote, and text labels are being applied

As you label and submit pages, the model learns from the annotations that you make. Gradually, the labels that are applied automatically become accurate and don't require any adjustments.

If the tool becomes overzealous in the application of labels, apply the **text** label to sections of standard text to correct it.

12. For tables, select the table caption and the entire table and label them with the **table** label.

The screenshot shows the 'Identify fields' interface for a PDF titled 'Algorithmic Trading PDF'. The interface includes a sidebar for 'Field labels' where the 'table' label is highlighted with a red box. The main area shows a preview of page 11, which contains a table. A red box highlights the 'table' label in the sidebar.

Figure 18. A table is being labeled

13. When a page contains an image, the image is not displayed in the representation of the page.

Images are never replicated. However, you can capture the text from an image so that the image text can be searched. To do so, enable the Optical Character Recognition (OCR) feature when you create a collection. OCR is helpful in cases where you want to extract text from images, such as from a scanned PDF, where the text is embedded in an image. For more information, see [Optical character recognition](#).

After you enable OCR, if you want to remove annotated image text from the collection index, you can label the image so that you can exclude the associated text. You will learn about how to configure the index in the next procedure.

The screenshot shows the AlgoPDF interface for managing PDF collections. The main area displays a document page titled "Figure 1 - # of Trades, Money, and Dollar Value in 2019". The sidebar on the right is titled "Identify fields" and contains a list of field labels with corresponding color-coded boxes: answer (orange), author (purple), footer (green), footnote (dark purple), header (blue), question (green), subtitle (pink), table_of_contents (light green), text (yellow), title (pink), table (orange), and image (light blue). A "Submit page" button is located at the bottom of the sidebar.

Figure 19. Shows an image in the page

14. When you reach the *Bibliography*, create a new label named `bibliography`.

The screenshot shows the AlgoPDF interface with a document page containing a bibliography section. A modal window titled "Create field label" is open, showing the text "bibliography" in a search input field. Below the input field are "Cancel" and "Create" buttons. The background of the interface shows a list of references and a "Submit page" button at the bottom right.

Figure 20. Creating a bibliography label

Apply the new label to each page.

The screenshot shows the AlgoPDF interface with a document page containing a bibliography section. The "Field labels" sidebar on the right has the "bibliography" label selected, indicated by a teal background. The background of the interface shows a list of references and a "Submit page" button at the bottom right.

Figure 21. A bibliography label is being applied

15. After you annotate and submit all the pages, click **Apply changes and reprocess**.

A notification is displayed to indicate that the collection was updated. You remain on the SDU tool page, but the **Apply changes and reprocess** button is disabled.

An SDU model is generated based on the structures that you labeled in this document.

For more information about the Smart Document Understanding feature, see [Using Smart Document Understanding](#).

Step 6: Streamline the searchable data

Now that you have an SDU model that can recognize the different types of sections in the document, you can instruct it to include some sections in searches and to exclude others. To control what data gets searched, you include or exclude fields from the search index.

1. Click **Manage fields**.

The screenshot shows the Microsoft Power Automate interface for managing fields in a document. The top navigation bar includes 'Finance tutorial project / Manage collections / Algorithmic Trading PDF'. Below the navigation is a toolbar with tabs: 'Activity', 'Identify fields' (which is active and highlighted in blue), 'Manage fields' (which is also highlighted with a yellow circle), 'Enrichments', 'Processing settings', and 'CSV settings'. The main area displays a PDF titled 'Staff Report on Algorithmic Trading in U.S. Capital Markets' from the 'SECURE ACT SECURITIES EXCHANGE ACT OF 2020'. On the left, there are three small thumbnail previews of the document pages. On the right, there is a 'Field labels' section with a list of document elements and their corresponding color-coded squares:

Field labels
Identify document elements using the labels below.
+ Create new
answer
author
bibliography
footer
footnote
header
question

Figure 22. The Manage fields tab

2. From the list of fields to index, set the switcher to **No** for all fields except these ones:

- o `footnote`
- o `html`
- o `subtitle`
- o `table`
- o `text`

Field	Type ⓘ	Include in index
bibliography	String	<input type="checkbox"/> No
footer	String	<input type="checkbox"/> No
footnote	String	<input checked="" type="checkbox"/> Yes
html	String	<input checked="" type="checkbox"/> Yes
subtitle	String	<input checked="" type="checkbox"/> Yes
table	Json	<input checked="" type="checkbox"/> Yes
table_of_contents	String	<input type="checkbox"/> No
text	String	<input checked="" type="checkbox"/> Yes
title	String	<input type="checkbox"/> No
answer	—	<input type="checkbox"/> No
author	—	<input type="checkbox"/> No

Figure 23. Fields in the index list

3. Click **Apply changes and reprocess**.

A notification is displayed to indicate that the collection was updated. You remain on the **Manage fields** page, but the **Apply changes and reprocess** button is disabled.

You successfully configured the index to control the content that is available to searches! You excluded fields that might contain popular search terms, but do not also include meaningful content.

For more information about managing fields, see [Excluding content from query results](#).

Step 7: Split the document

Now that Discovery knows more about the structure of the document, we can split the single 99-page document into more documents. Remember, only one passage was returned for each query that you submitted before. If we split the document into multiple segments, Discovery can return the best passages from across all of the document segments.



Note: When you split a document, you turn one document into many documents. Be aware of the document limits for your plan type. Each document segment that is generated by splitting a document counts toward the plan's document limit.

When you annotated the document, you identified the **subtitle** field. These subtitles are a good marker from which each new document segment can begin.

1. From the **Improve query results by splitting your documents** section of the **Manage fields** page, click **Split document**.
2. Select **subtitle** from the **Split document on each occurrence of** field.

The screenshot shows the 'Manage fields' page for the 'Algorithmic Trading PDF' collection. In the 'Fields to index' section, there are three fields listed:

Field	Type	Include in index
bibliography	String	No
footer	String	No
frontnote	String	Vac

A sidebar on the right provides information about improving query results by splitting documents based on fields like subtitle.

Figure 24. Choosing to split documents on the subtitle field

3. Click **Apply changes and reprocess**.

A notification is displayed to indicate that the collection was updated. You remain on the **Manage fields** page, but the **Apply changes and reprocess** button is disabled.

4. Click **Activity** from the page header to return to the **Activity** page where you can monitor the progress of the change you made.

When no documents are processing, document splitting is finished.

For more information about splitting documents, see [Split documents to make query results more succinct](#).

Step 8: Test the project again

Let's find out whether we improved the search function by adding a user-trained SDU model for the document. To do so, let's retest the project.

- From the navigation panel, click **Improve and customize** to open the **Improve and customize** page.
- First, to make sure that we didn't degrade the quality of the search, let's ask one of the questions that returned a good response when we tested earlier.

In the **Search** field, enter **What is the purpose of Rule 15c3-5?**

The screenshot shows the 'Improve and customize' page. In the search bar, the query 'What is the purpose of Rule 15c3-5?' is entered. Below the search bar, there are five search results:

- Commission
- one
- HFTs
- U.S.
- investors

Each result has a 'Run search' button to its right. On the right side of the page, there is a sidebar titled 'Improvement tools' with the following options:

- Customize display
- Extract meaning
- Teach domain concepts
- Define structure
- Improve relevance

Figure 25. Query added to the Improve and customize page

Multiple responses are returned this time. The following response contains the exact answer to the question without any extraneous text:

In November 2011, the SEC implemented the final provision of Rule 15c3-5 curbing unfiltered market access. The provision mandated that brokers verify their clients' order flow for compliance with credit and capital thresholds before routing to market centers.

The screenshot shows a search interface with a sidebar on the left containing icons for file management. The main area has a header 'Finance tutorial project / Improve and customize'. A search bar at the top contains the query 'What is the purpose of Rule 15c3-5?'. Below the search bar, there are two sections: 'Top Entities' and 'Collections'. Under 'Top Entities', there is a dropdown menu set to 'Organization', with options 'Number', 'Location', and 'Date' below it. To the right of the dropdown, a toggle switch is set to 'Off' for 'Show table results only'. Under 'Collections', there is a dropdown menu set to 'Available collections'. Two search results are displayed:

- Rule 15c3 - 5**: "In November 2011, the SEC implemented the final provision of Rule 15c3 - 5 curbing unfiltered market access. The provision mandated that brokers verify their clients' order flow for compliance with credit and capital thresholds before routing to market centers." [View passage in document](#)
- Report to Congress on Algorithmic Trading**: Collection: Algorithmic Trading PDF

Below these results, another section for 'Available collections' shows:

- "FINRA has proposed publishing aggregate trade count and volume statistics for each corporate bond ATS, by CUSIP.90 The stated **purpose** of this proposal is to provide the market with more readily available information about potential sources of liquidity." [View passage in document](#)
- Report to Congress on Algorithmic Trading**: Collection: Algorithmic Trading PDF

Figure 26. Multiple responses are returned for the query

Our updates only improved the quality of the accurate responses that were returned before.

- Now, let's ask a question that returned poor results previously. Enter **What are PTFs?** as the search query.

The same response that was returned as the only response last time is returned again. However, this time we get more than one response. And we can see that the second response that is returned defines the acronym for us.

(“principal trading firms” or “PTFs”)

The screenshot shows a search interface with a sidebar on the left containing icons for file management. The main area has a header 'Finance tutorial project / Improve and customize'. A search bar at the top contains the query 'What are PTFs?'. Below the search bar, there are two sections: 'Top Entities' and 'Collections'. Under 'Top Entities', there is a dropdown menu set to 'Organization', with options 'Number' and 'Date' below it. To the right of the dropdown, a toggle switch is set to 'Off' for 'Show table results only'. Under 'Collections', there is a dropdown menu set to 'Available collections'. Two search results are displayed:

- "Despite the surge in trading volume during the event window, there was no noticeable change in net positions of **PTFs** or bank-dealers. However, the report also finds evidence that some **PTFs** and bank-dealers may have contributed to the volatility." [View passage in document](#)
- Report to Congress on Algorithmic Trading**: Collection: Algorithmic Trading PDF

Below these results, another section for 'Available collections' shows:

- "Many participants in securities markets trade with their own principal (“principal trading firms” or “**PTFs**”).324 Principal trading firms trade in a wide variety of ways. They may, for example, act as liquidity-providing” [View passage in document](#)
- Report to Congress on Algorithmic Trading**: Collection: Algorithmic Trading PDF

Figure 27. Responses that answer the question about PTFs

- Let's try the other problematic search query. Enter **Where do muni bond trades get reported to?** as the search query.

This time it's the third response that provides an answer to the question. You must view the full passage to see the entire definition.

The screenshot shows a search interface with a sidebar for 'Top Entities' and 'Collections'. The search query is 'Where do muni bond trades get reported to?'. The results are as follows:

- Top Entities**
 - Organization: "low trading volume for most bonds .78 Largely because of their tax treatment, shorting of municipal bonds is difficult and rare.79 In recent years, several platforms have developed that facilitate the electronic trading of municipal bonds."
 - Number:
 - Location:
 - JobTitle:
 - Date:
- Collections**
 - Available collections: "Transactions in corporate bonds must be reported to the Trade Reporting and Compliance Engine (TRACE) operated by FINRA.85 TRACE data is disseminated by FINRA immediately 28 upon receipt.86 Each FINRA member that is party to a transaction in a TRACE-eligible security must report the trade as soon as practicable, but generally no later than within fifteen minutes of the exec"
 - Report to Congress on Algorithmic Trading Collection: Algorithmic Trading PDF
 - "Transactions in municipal bonds must be reported to the Municipal Securities Rulemaking Board's (MSRB) Real-time Transaction Reporting System (RTS)."
 - Report to Congress on Algorithmic Trading Collection: Algorithmic Trading PDF

Figure 28. Responses that answer the question about muni bonds

Click the *View passage in document* link to see the full definition highlighted in the document.

Transactions in municipal bonds must be reported to the Municipal Securities Rulemaking Board's (MSRB) Real-time Transaction Reporting System (RTS).

Congratulations! You successfully added a user-trained Smart Document Understanding (SDU) model that improves the quality of your search project.

Step 9: Filter results with a dictionary-based facet

Now that we are getting more passages returned per query, it might be useful to filter the results. To filter the results based on the types of financial instruments that are mentioned, we can add a search facet. One available source for a facet is a dictionary.

1. To create a dictionary, from the *Improvement tools* panel of the *Improve and customize* page, expand *Teach domain concepts*, and then click **Dictionaries**.
2. Click **New**.

Name	Used in
No dictionaries	If you add dictionaries, you'll be able to manage them here.

Figure 29. New button in the dictionary page

3. Enter **Financial instruments** as the dictionary name, add the term **municipal bond**, and then click the **Add term** button.

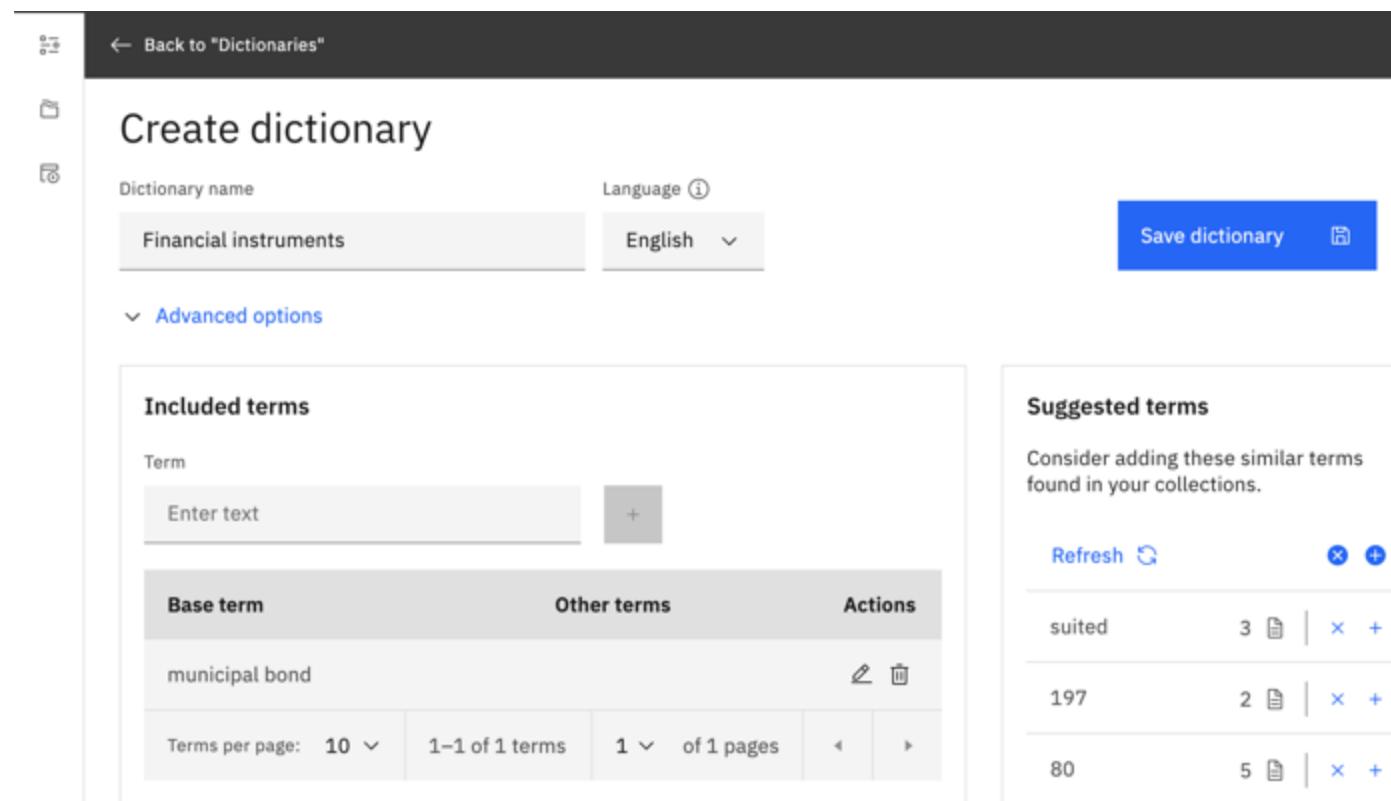


Figure 30. Financial instruments dictionary with one term

4. Add synonyms for the term by clicking the edit icon for the term.

Municipal Bonds, muni, munis, muni bonds

Add synonyms in a comma-separated list, and then click **Save term**.

5. Click **Save dictionary**.

You can choose a field in the document where you want the enrichment to be applied. Let's choose the **subtitle** field that was generated when we created the user-trained SDU model. From the **Fields to enrich** field, select **subtitle**. Click **Apply**.

The dictionary is created and each subtitle in the document is analyzed for mentions of terms or synonyms that are defined in the dictionary. Any mentions that are found are noted in the index.

6. Click **Improve and customize** from the navigation panel.
7. From the **Improvement tools** panel of the **Improve and customize** page, expand **Customize display**, and then click **Facets**.
8. Click **New facet**, and then select **From existing fields in a collection**.
9. Choose the index field that is associated with the dictionary enrichment that you applied to the **subtitle** field. From the **Field** field, select **enriched_subsection.entities.mentions.text**

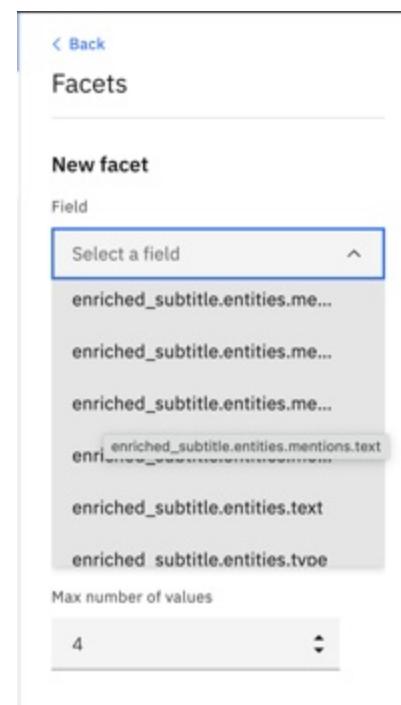


Figure 31. Fields from which you can create a facet

You might need to hover over the entries to see the full field names.

10. Add a label, such as **Dictionary terms** to the **Label** field, and then click **Apply**.

< Back

Facets

New facet

Field
enriched_subtitle.entities.

Label
Dictionary terms

Filtering options
 Multiple-choice checkboxes
 Single-choice radio buttons

Max number of values
4

Figure 32. Facet was created

11. Enter **Where do muni bond trades get reported to?** as the search query.

The **Dictionary terms** facet that you created is displayed along with the search results. A **Municipal Bonds** checkbox is shown, which indicates that at least one of the returned passages is extracted from a document segment with the term **Municipal Bonds** in its **subtitle** field.

Finance tutorial project /

Improve and customize

Where do muni bond trades get reported to?

Top Entities Off

- Organization
- Number
- Location
- Dictionary terms**
- Municipal Bonds**

Report to Congress on Algorithmic Trading Collection: Algorithmic Trading PDF

“low trading volume for most bonds.78 Largely because of their tax treatment, shorting of municipal bonds is difficult and rare.79 In recent years, several platforms have developed that facilitate the electronic trading of municipal bonds.”

Report to Congress on Algorithmic Trading Collection: Algorithmic Trading PDF

“Transactions in corporate bonds must be reported to the Trade Reporting and Compliance Engine (TRACE) operated by FINRA.85 TRACE data is disseminated by FINRA immediately 28 upon receipt.86 Each FINRA member that is party to a transaction in a TRACE-eligible security must report the trade as soon as

Report to Congress on Algorithmic Trading Collection: Algorithmic Trading PDF

“Transactions in municipal bonds must be reported to the Municipal Securities Rulemaking Board’s (MSRB) Real-time Transaction Reporting”

Items per page: 10 1–10 of 96 results 1 of 10 pages

Figure 33. Dictionary term facet with a Municipal Bonds option

12. To filter the results to show only passages from sections with **Municipal Bonds** in the subtitle, select the **Municipal Bonds** checkbox.

The best answer is now listed as the second response instead of the third.

Finance tutorial project /

Improve and customize

Where do muni bond trades get reported to?

[Clear all](#) [X](#)

Show table results only [Off](#)

Top Entities

- Organization
- Number
- Location

Dictionary terms [1 X](#)

- Municipal Bonds

Collections

Available collections [▼](#)

"low **trading** volume for most **bonds**.⁷⁸ Largely because of their tax treatment, shorting of municipal **bonds** is difficult and rare.⁷⁹ In recent years, several platforms have developed that facilitate the electronic **trading** of municipal **bonds**."

[View passage in document](#)

Report to Congress on Algorithmic Trading Collection: Algorithmic Trading PDF

"Transactions in municipal **bonds** must be **reported** to the Municipal Securities Rulemaking Board's (MSRB) Real-time Transaction **Reporting**."

[View passage in document](#)

Report to Congress on Algorithmic Trading Collection: Algorithmic Trading PDF

"Post- **trade** transparency, in the form of transaction **reports**, generally is available for corporate and municipal **bonds**. Transactions in

[View passage in document](#)

Table 2: Percentage of All NMS Stock Trades, Shares, and Dollar Volume in

[View table in document](#)

Figure 34. Best answer is the second result

Summary

In this tutorial, you created a Document Retrieval project, a Smart Document Understanding (SDU) model, a dictionary enrichment, and a search facet. You applied the facet that is based on your dictionary to the custom field that is generated by your SDU model to filter your query results for better answers.

Start getting value from your data

Learn what IBM Watson® Discovery can do to help you find answers, recognize patterns, and gain insights from your data.

Find checklists of the high-level steps to follow to achieve the following goals:

- [Pinpoint answers](#)
- [Extract meaning](#)
- [Enhance your chatbot](#)
- [Find trends](#)
- [Analyze contracts](#)

Pinpoint answers

Help customers find answers faster. Analyze content from various connected data sources, pinpoint the most relevant passage or phrase, and return the right information when someone asks for it.

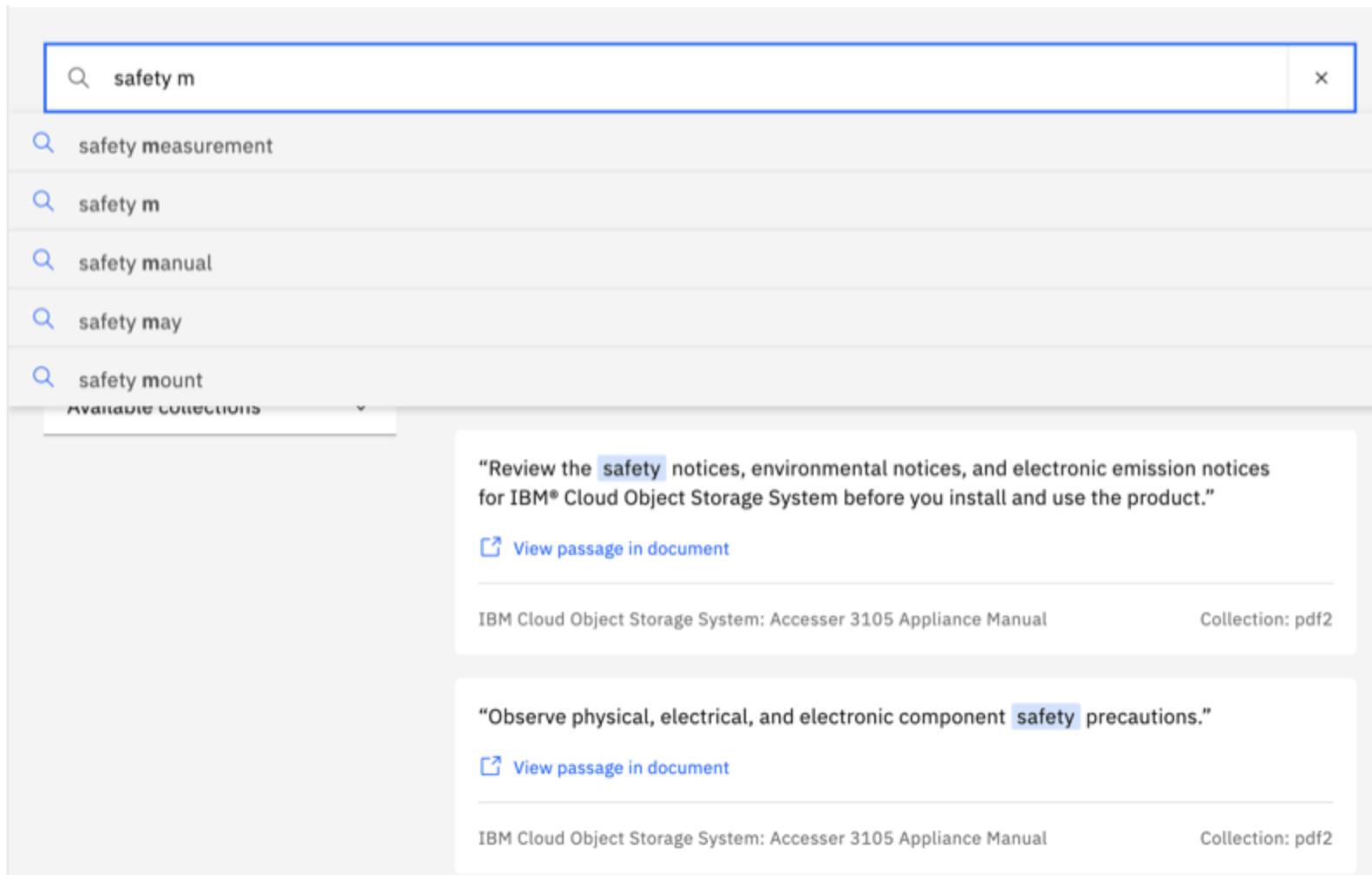


Figure 1. Search bar with search results

If pinpointing answers is your goal, complete the steps that are listed in the following table.

Step	Task	Related information
<input type="checkbox"/>	Create a <i>Document Retrieval</i> project.	Creating projects
<input type="checkbox"/>	Add up to 5 collections that connect to external data sources or contain uploaded files.	Creating collections
<input type="checkbox"/>	Run test queries to assess the quality of the initial results.	Previewing the default query results
<input type="checkbox"/>	Take actions to improve your results. For example, you can customize the search bar to enable autocomplete.	Improving your query results
<input type="checkbox"/>	Deploy your search solution.	Deploying your project

Checklist for getting answers

Extract meaning

Use award-winning natural language processing technology to enrich your data and ensure that the right information is found when someone searches for answers.

The screenshot shows a facet titled "Machine Learning Terms". It lists five items: "Neural network" (checked), "Reinforced learning", "CIFAR-10", "MNIST", and "Recommender systems". To the right of each item is a snippet of text and a "View passage in document" link.

Term	Description Snippet	Action
Neural network	"Neural-network training can be slow and energy weight data for the network between conventional... Analogue non-volatile memory can accelerate the backpropagation by performing parallelized multi...	View passage in document
Reinforced learning	Equivalent-accuracy accelerated neural.pdf	
CIFAR-10	"Neural-network training can be slow and energy weight data for the network between conventional... Analogue non-volatile memory can accelerate the backpropagation by performing parallelized multi...	View passage in document
MNIST		
Recommender systems		

Figure 2. Machine learning facet for filtering search results with custom enrichment values

If extracting meaning is your goal, complete the steps that are listed in the following table.

Step	Task	Related information
<input type="checkbox"/>	Create any project type.	Creating projects
<input type="checkbox"/>	Add collections that connect to external data sources or contain uploaded files.	Creating collections
<input type="checkbox"/>	Chunk large documents into many smaller documents so you can apply more targeted enrichments to the content.	Using Smart Document Understanding
<input type="checkbox"/>	Enhance your data by applying built-in NLU enrichments.	Applying prebuilt enrichments
<input type="checkbox"/>	Identify and promote terms and patterns from your data with special significance to your use case.	Adding domain-specific resources
<input type="checkbox"/>	Submit test queries to assess the results.	Testing your project
<input type="checkbox"/>	Create a facet that surfaces the enriched data from your documents.	Facets
<input type="checkbox"/>	Take actions to improve your results.	Improving your query results
<input type="checkbox"/>	Deploy your solution.	Deploying your project

Checklist for extracting meaning

Enhance your chatbot

Delight your customers by fortifying your chatbot with an answer to every question. Discovery is designed to work seamlessly with Watson Assistant to search and deliver answers from help content that you already own.

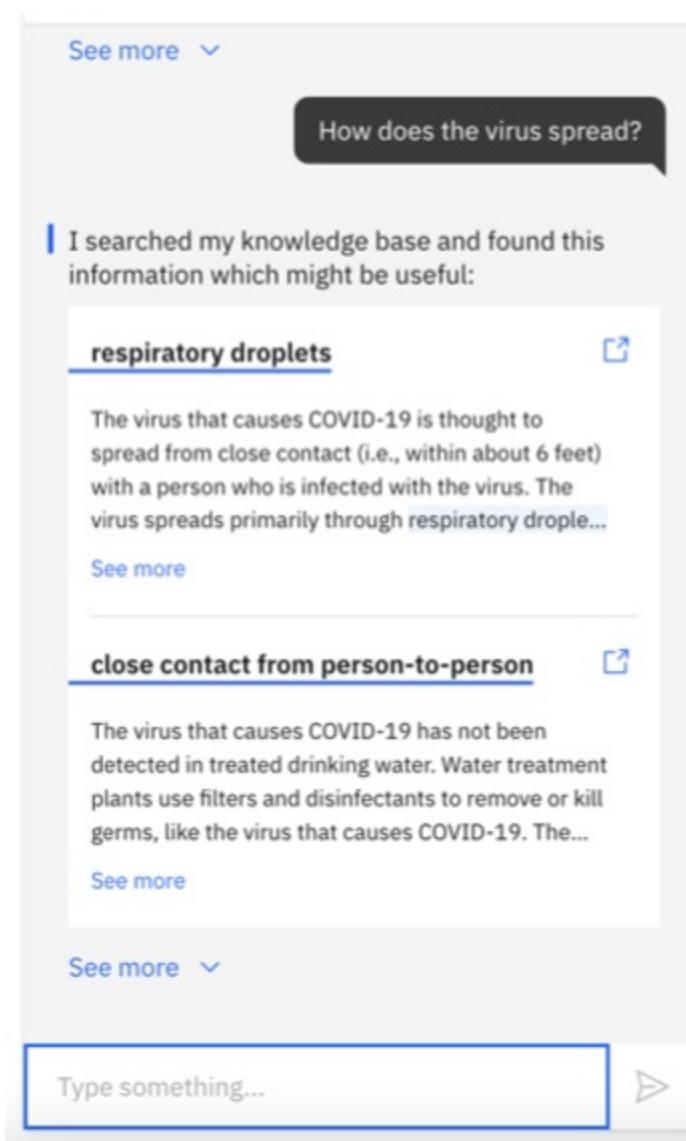


Figure 3. Answer finding enabled in the Watson Assistant web chat

If enhancing your chatbot is your goal, complete the steps that are listed in the following table.

Step	Task	Related information
<input type="checkbox"/>	Create a <i>Conversational Search</i> project.	Creating projects
<input type="checkbox"/>	Add a collection that connects to an external data source or contains uploaded files.	Creating collections
<input type="checkbox"/>	Run test queries to assess the quality of the initial results.	Previewing the default query results
<input type="checkbox"/>	Take actions to improve your results.	Improving your query results
<input type="checkbox"/>	Connect your project to a virtual assistant that is built with Watson Assistant.	Deploying your project
<input type="checkbox"/>	From the Watson Assistant user interface, deploy the web chat that is associated with your assistant.	Deploying your assistant

Checklist for enhancing your chatbot

For a more detailed look at these steps, take a tutorial that walks you through them. For more information, see [Power your assistant with answers from web resources](#).

Alternatively, you can add a generative language service named NeuralSeek between the Watson Discovery and Watson Assistant services. For more information, see [Use NeuralSeek to return polished answers from existing help content](#).

Find trends

Uncover patterns, trends, and relationships in structured and unstructured data. Use text analytics to gain insights into social media, e-commerce trends, and user behavior. Or start to address problems by finding their root cause.



Note: Only users of installed deployments (IBM Cloud Pak for Data) or Premium or Enterprise plan-managed deployments can create this type of project.

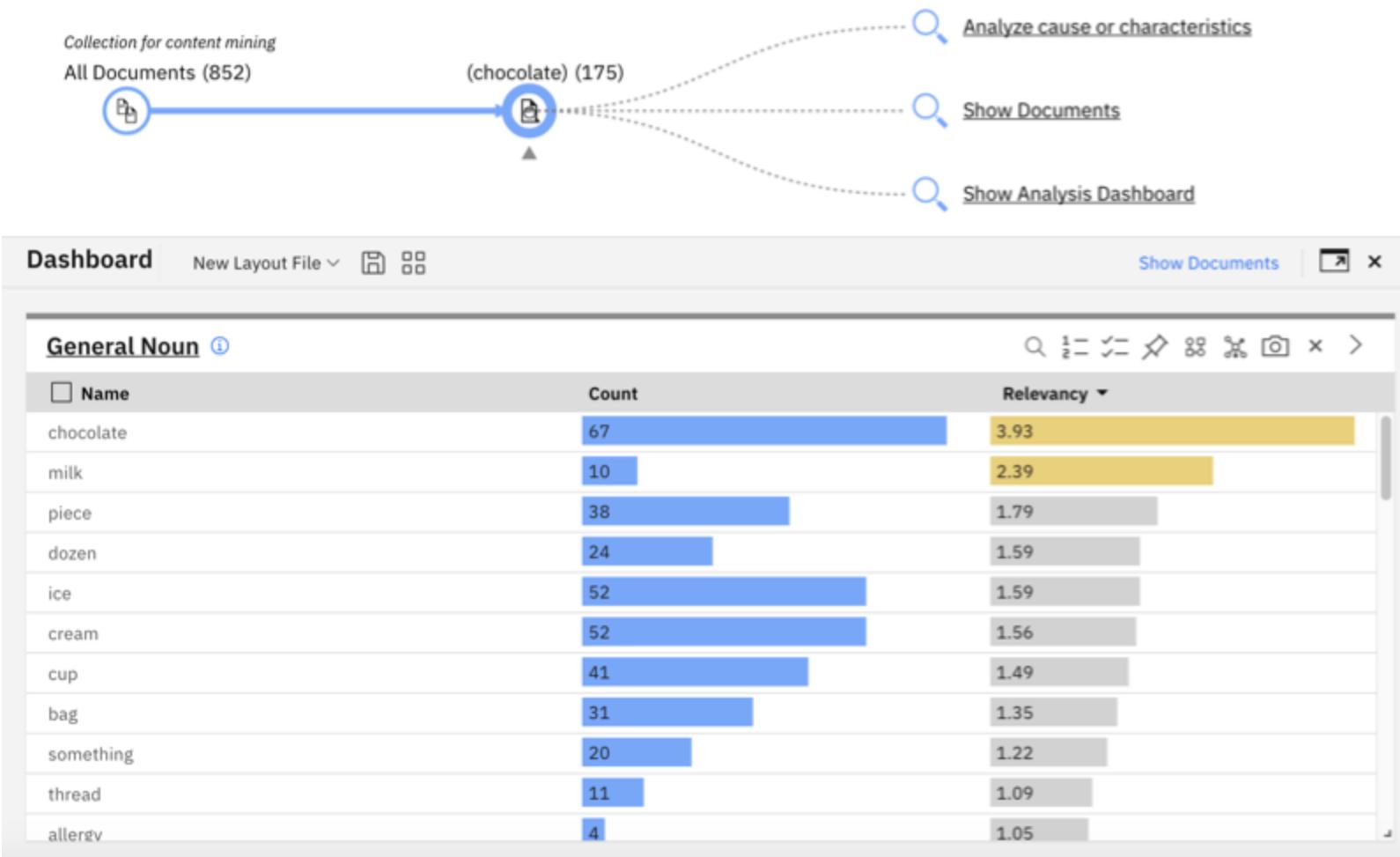


Figure 4. Analyzing data with the Content Mining application

If finding trends in your data is your goal, complete the steps that are listed in the following table.

Step	Task	Related information
<input type="checkbox"/>	Create a Content Mining project.	Creating projects
<input type="checkbox"/>	Add a collection that connects to an external data source or contains uploaded files.	Creating collections
<input type="checkbox"/>	Use the built-in Content mining application to analyze your data.	Analyzing your data with the content mining application

Checklist for finding trends

Analyze contracts

Accelerate the pace at which experts can analyze complex documents.



Note: Only users of installed deployments (IBM Cloud Pak for Data) or Premium or Enterprise plan-managed deployments can create this type of project.

The screenshot shows the 'Contract Data' tab selected in a browser window. The document title is '← Microsoft Word - 1-IBM Standard TSA w_Exhibits Final.doc'. The page number is '1 / 3'. The interface includes a 'Filters' sidebar with categories like 'Amendments', 'Asset Use (3)', 'Assignments', etc., where 'Asset Use' is checked. The main content area displays section 13.4 Asset Control with the following text:

In the event Supplier Personnel has access to information, information assets, supplies or other property, including property owned by third parties but provided to Supplier Personnel by Buyer ("Buyer Assets"), Supplier Personnel:

1. will not remove Buyer Assets from Buyer's premises without Buyer's authorization;
2. will use Buyer Assets only for purposes of this Agreement and reimburse Buyer for any unauthorized use;
3. will only connect with, interact with or use programs, tools or routines that Buyer agrees are needed to provide Services;
4. will not share or disclose user identifiers, passwords, cipher keys or computer dial port telephone numbers; and
5. in the event the Buyer Assets are confidential, will not copy, disclose or leave such assets unsecured or unattended. Buyer may periodically audit Supplier's data residing on Buyer Assets.

The right side of the interface shows 'Details' sections for 'Categories' (Asset Use), 'Types' (Nature: Right, Party: Supplier), and 'Attributes' (None).

Figure 5. Analyzing contracts

If analyzing contracts is your goal, complete the steps that are listed in the following table.

Step	Task	Related information
<input type="checkbox"/>	Create a <i>Document Retrieval for Contracts</i> project.	Creating projects
<input type="checkbox"/>	Add up to 5 collections that connect to external data sources or contain uploaded files.	Creating collections
<input type="checkbox"/>	Run test queries to assess the quality of the initial results.	Previewing the default query results
<input type="checkbox"/>	Take actions to improve your results.	Improving your query results
<input type="checkbox"/>	Analyze the data.	Understanding contracts

Checklist for analyzing contracts

Connecting to your data

Creating projects

A project is a convenient way to collect and manage the resources in your IBM Watson® Discovery application. You can assign a **Project type** and connect your data to the project by creating a collection.

Before you create a project, decide which project type best fits your needs.

Project descriptions

Need	Goal	Project type
<i>Which document contains the answer to my question?</i>	Find meaningful information in sources that contain a mix of structured and unstructured data, and surface it in a stand-alone enterprise search application or in the search field of a business application.	Document Retrieval
<i>Where is the part of the contract that I need for my task?</i>	Quickly extract critical information from contracts.	Document Retrieval for Contracts
<i>I want the chatbot I'm building to use knowledge that I own.</i>	Give a virtual assistant quick access to technical information that is stored in various external data sources and document formats to answer customer questions.	Conversational Search
<i>I want to uncover insights I didn't know to ask about.</i>	Gain insights from pattern analysis or perform root cause analysis.	Content Mining

Project type use cases



Note: If you created the Discovery service as part of a IBM Cloud Pak for Data as a Service deployment, the Discovery project is separate and distinct from the deployment project that is displayed in IBM Cloud.

To create a project, complete the following steps:

1. Open the **Projects** page by selecting **My Projects**.
2. Click **New project**. Name your project, and then choose the project type.

For more information about each type, see [Project types](#).

Otherwise, choose **None of the above** and a **Custom** project type is created for you.

3. If you choose a **Document Retrieval** project type and your data sources are in English, decide whether to enable the Content Intelligence feature.

If your data source contains contracts, enable the feature by selecting **Apply contracts enrichment**. Scroll to see the checkbox, if necessary.

4. Click **Next**.
5. Choose and configure a data source or connect to an existing collection.

For more information about supported data sources, see [Creating collections](#).

Take advantage of the following resources that are available from the page header:

- To open the product documentation, click the Help icon
- To see all of your projects, click **My projects**.

Project types

Choose a project type to get the correct set of enrichments applied to your documents automatically. The improvement tools that are available differ by project type, as do the deployment methods, which are optimized for each use case.

The following project types are available:

- [Document Retrieval](#)

- [Document Retrieval for Contracts](#)
- [Conversational Search](#)
- [Content Mining](#)
- [Custom](#)

For more information about the different settings that are applied to each project type, see [Default project settings](#).

Document Retrieval

Use this project type to search and find the most relevant answers from your data. Projects of this type are typically deployed as search field components that are added to websites or other applications.

Documents that you add to a project of this type are automatically enriched in the following ways:

- Entities, such as proper nouns, are identified and tagged.
- Parts of speech are identified and tagged.

This tagged information is used later when a natural language phrase is submitted as a search query to return a smarter response.



Tip: A sample Document Retrieval project is available for you to explore. For more information, see [Getting started with Watson Discovery](#).

Document Retrieval for Contracts

If you are working with English-language legal contracts, enable the Content Intelligence feature to apply a contracts enrichment that can recognize and tag contract-related concepts in your data. Use this project type to automate complex business processes, such as contract review and negotiation. This project type can help to increase productivity, minimize costs, and reduce your legal exposure.



Note: Only users of installed deployments (IBM Cloud Pak for Data) or Premium or Enterprise plan managed deployments can create this type of project.

In addition to the enrichments that are applied to a typical document retrieval project, the following enrichments are made automatically:

- Content from tables in the source document is tagged so that it can be found later.
- Contract details, such as payment terms or parties that are involved in the contract, are identified and tagged.

For any collection that you add to the project, optical character recognition (OCR) is enabled automatically so that text from scanned documents or other images is processed.



Note: When you apply the contracts enrichment, you cannot use Smart Document Understanding to annotate documents. A pretrained SDU model that can recognize contract-related information is applied automatically. The Table understanding enrichment is automatically applied.

For more information, see [Understanding contracts](#).

Conversational Search

The **Conversational Search** project returns information from a connected data collection as answers to questions that customers ask a chatbot, which is also known as an **assistant**.

Use IBM Watson® Assistant and Discovery together to give your assistant access to technical content and other knowledge base resources without having to relocate or copy your corporate data. The built-in synchronization capabilities mean that your assistant can share the most up-to-date information available. Use the integrations that are provided with Watson Assistant to deploy an assistant that connects to this project to various platforms, including your company website, in minutes.

The documents that you add to this type of project are not enriched automatically.

If you need to perform more complex searches from your virtual assistant, you might want to create a **Document Retrieval** project instead of **Conversational Search** project. For more information, see [Choosing the right project type for a chatbot](#).

IBM Cloud Another feature to consider enabling is the **Emphasize the answer** feature. When enabled, the answers that are returned to customers who interact with the assistant show the exact answer highlighted in bold font within the search response. For more information about how the exact answer is determined, see [Answer finding](#).

For more information about building a Watson Assistant search skill, see the appropriate documentation for your deployment:

- IBM Cloud Pak for Data [Adding a search integration](#).
- IBM Cloud [Embedding existing help content](#)



Note: From the classic Watson Assistant experience, see [Creating a search skill](#).

Content Mining

Use this project type to discover hidden insights, trends, and relationships in your data.



Note: Only users of installed deployments (IBM Cloud Pak for Data) or Premium or Enterprise plan managed deployments can create this type of project.

This project type is especially useful for analyzing structured data, such as data that you add by uploading a CSV file or by connecting to a database data source. You can add only one collection to a project of this type from the Discovery user interface.

Documents that you add as part of the initial collection are automatically enriched in the following way:

- Parts of speech are identified and tagged.

After you add a collection and optionally apply more enrichments to the data, a full-featured application is available for you to deploy. You can use the application to research your data in depth. For more information about using the application, see [Analyzing your data with the deployed Content Mining application](#).

From the Content Mining application, you can create the following enrichment types which are not available in other project types:

- [Document classifier](#)
- [Phrase sentiment](#)



Note: You can create a collection from the deployed Content Mining application. The collection that you create is not added to your existing Content Mining project. A new Content Mining project is created to store the collection. The collection can contain an uploaded CSV file only. The project that is generated is given the name that you specify for the collection.

Because the data that you add to this type of project is often structured, consider using the API to submit queries in the Discovery Query Language (DQL). With DQL queries, you can get information from specific fields or find specific enrichment type mentions. You cannot apply relevancy training to a **Content Mining** project.

Custom

Choose this type if you prefer not to use one of the other project types. No enrichments are applied automatically, so you can add only those enrichments that are necessary for your use case.

Basic project defaults

Some enrichments and query result settings are applied to each project type by default.

Project type	Default enrichments	Default query result settings
Document Retrieval	Entities, Part of Speech	Facets (by Entity), Passages
Document Retrieval for Contracts	Entities, Parts of speech, Table Understanding, and Contracts	Facets (by Category, Nature, Contract Term, Contract Payment Term, Contract Type, Contract Currency, Invoice Buyer, Invoice supplier, Invoice Currency, Purchase Order Buyer, Purchase Order Supplier, Purchase Order Payment Term) and Table Retrieval
Conversational Search	Part of Speech	Passages
Content Mining	Part of Speech	None
Custom	None	Passages

Basic project defaults

Project limits

The number of projects you can create depends on your Discovery plan type.

Plan	Projects per service instance
Cloud Pak for Data	Unlimited
Premium	100
Enterprise	100
Plus (includes Trial)	20

Plan details

The Sample project is excluded from the total number of projects.

Renaming a project

 **Note:** You cannot rename the *Sample Project*.

To rename a project after you create it, complete the following steps:

1. Go to the **My Projects** page.
2. Find the project that you want to rename, click the **Project actions** icon , and then choose **Rename**.
3. Edit the project name, and then click **Apply**.

Deleting a project

If you want to delete a project, but keep a collection from the project, share the collection with another project before you complete these steps. From another project (a type that allows multiple collections), open the **Manage collections** tab. Click **New collection**, and then click **Reuse data from an existing collection**. Select the collection that you want to keep, and then click **Finish**.

 **Note:** You cannot delete the *Sample Project*.

To delete a project, complete the following steps:

1. Go to the **My Projects** page.
2. Find the project that you want to delete, click the **Project actions** icon , and then choose **Delete**.
3. Click **Delete**.

Creating collections

A collection is a set of documents that you add to a project so that you can analyze, enrich, and extract useful information from it.

You can add data to your project in the following ways:

- Upload locally accessible files by using the product user interface. This method is the best way to get started and test your use case.
- Set up a scheduled crawl of documents that are stored on an external data source.

The product user interface offers several built-in data source connectors for you to choose from. The options differ depending on your deployment type. For more information, see [Supported data sources](#).

- Connect to an external data source for which no built-in support is available:

IBM Cloud

Use IBM App Connect to set up a scheduled crawl of documents that are stored on other external data sources.

IBM Cloud Pak for Data

Build a connector to crawl documents that are stored on other external data sources.

- To automate the process of adding data to your project, use the Discovery APIs to create a collection and upload documents to it.

When you add documents to Discovery, the original documents are crawled and information from the documents is stored in an index so that it can be enriched and analyzed or retrieved later. Not all rich content from the original document is retained. For example, images from .ppt or .doc files are not stored. For more information, see [How your data source is processed](#).

Choosing what to add to a collection

There are a few things to consider as you decide how to break up your source content into collections.

- Getting content from different data sources

If you store similar content in more than one type of data source (a website and Salesforce, for example), you can create one project with two separate collections. Each collection adds documents from a single data source. When they are built together into a single project, a user can search across both sources at the same time.

- Applying enrichments

Creating a collection is a good way to group documents that you want to enrich in a similar way. For example, maybe a subset of your documents contains industry jargon and you want to add a dictionary that recognizes the terms. You can create a separate collection and apply the Parts of Speech enrichment so you can use the term suggestions feature to speed up the process of creating the dictionary.

- Creating separate Smart Document Understanding (SDU) models

You can use the Smart Document Understanding tool to identify content based on the structure of a document. If you have 20 PDF files that were created by your Sales department and use one template and 20 PDF files that were created by your Research department and use a different template, group each set into its own collection. You can then use the SDU tool to build a model for each structure separately, a model that understands the unique structure. You can also use the tool to define custom fields that are unique to the source documents.

Creating a collection

Before you can create a collection, you must create a project. For more information, see [Creating projects](#).

Things to keep in mind:

- A collection can support only one external data source.
- Documents in the collection must be in one language only, the language that you specify for the collection.

To create a collection, complete the following steps:

1. Open a project, go to the **Manage collections** page, and then click **New collection**.
 - The Conversational Search, Document Retrieval, and Custom project types can contain up to 5 collections.
 - A Content Mining project can contain only 1 collection.
2. Choose how you want to add data to your collection.
 - [Uploading data](#)
 - [Reusing data from a collection](#)
 - Crawling an external data source.

For supported data sources, see the appropriate topic for your deployment type:

- IBM Cloud Pak for Data [IBM Cloud Pak for Data data sources](#)
- IBM Cloud [IBM Cloud data sources](#)

 **Tip:** These topics also describe how to connect to data sources that are not supported by default per deployment type.

For information about how to troubleshoot issues that you might encounter when you add documents to a collection, see [Troubleshooting ingestion](#).

For more information about how to create a collection programmatically, see the [API reference documentation](#).

Optical character recognition

One of the optional features that you can apply to a collection when you create it is optical character recognition. The optical character recognition (OCR) feature extracts text from images. This capability is useful for preserving information that is depicted in diagrams or graphs, or in text that is embedded in files such as scanned PDFs. By converting the visual information into text, it can later be searched.

A new version of the technology was introduced in cloud-managed instances. OCR v2 was developed by IBM Research to be better at extracting text from scanned documents and other images that have the following limitations:

- Low-quality images due to incorrect scanner settings, insufficient resolution, bad lighting (such as with mobile capture), loss of focus, misaligned pages, and badly printed documents
- Documents with irregular fonts or various colors, font sizes, and backgrounds

Things to keep in mind when you enable OCR:

- The time that it takes to ingest a document with images increases when OCR is enabled.
- OCR can read both clear and noisy images. It can convert noisy images to gray scale, and smooth and de-skew them. However, the image quality must meet the minimum requirement of **80 DPI** (dots per inch).
- OCR can recognize many languages, but the language of the text in the image must be the same as the language that is specified for the collection where the file is added.

For more information about languages for which OCR v1 and OCR v2 are supported, see [Language support](#).

For a list of files types where you can apply OCR, see the [Supported file types](#) table.

Enabling stemming for uncurated data IBM Cloud Pak for Data



Note: This feature is available from IBM Cloud Pak for Data deployments only. It was introduced with the 4.7.0 release.

You can configure Discovery to use stemming instead of lemmatization for normalization when you create a collection. This configuration is only occasionally useful when a collection contains data with many misspellings, missing accent marks, and grammatical errors.

Discovery normalizes words to enable faster recognition and matching of words and their various forms, such as plurals or alternative verb conjugations. By default, Discovery uses lemmatization to normalize words based on their meaning. Stemming normalizes words by using word stems only.

Lemmatization is more precise, but works best on curated data. If your data is not well curated, stemming might work better. The same word stem typically is detected whether or not a word is spelled correctly. However, lemmatization might not recognize a misspelled word or might misinterpret its meaning. As a result, the lemmatizer can add the wrong root word to represent the misspelled word in the index. A search against a stemmed version of a misspelled word is likely to return better results than a search against an incorrectly lemmatized word.

The following table shows examples of how some words are stemmed versus lemmatized.

Surface form	Lemmatized form	Stemmed form
running	run	run
ran	run	ran
instructor	instructor	instruct
instruction	instruction	instruct

Stemmer versus Lemmatizer comparison

As you can see from the examples, the lemmatizer captures the word meanings better than the stemmer. Both *running* and *ran* are recognized as different forms of the same root verb *run*. And the difference in meaning between the two nouns *instructor* and *instruction* is preserved. However, if the data contains misspellings such as *instructer* and *instructoin*, the normalized form that is generated by stemming (*instruct*) will return better matches.

Discovery normalizes words when it ingests and stores data in the index and at run time when it analyzes queries that are submitted by users. The same normalization method is used for both operations, even though one operation occurs at the collection-level and the other occurs at the project-level. When a query is submitted, it is federated to each collection within the project, where the query is normalized based on that collection's configuration. Collections that are configured to use the stemmer normalize the query by using stemming. The collections that are not, normalize the query by using lemmatization.

To enable the stemmer instead of the lemmatizer when you create the collection, expand **More processing options**, and then set the **Use stemming instead of lemmatization when indexing** switcher to **On**.

If you configure Discovery to use the stemmer, consider also designing the queries that extract information from the collection to allow for character differences during matching. For more information, see the [String variation operator](#).

For more information about the languages for which the stemmer is supported, see [Language support](#).

Collection limits

The number of collections that you can create per project differs by project type.

Project type	Collections per project
Document Retrieval	5
Document Retrieval for Contracts	5
Conversational Search	5
Content Mining	1
Custom	5

Collections per project limits

The number of collections you can create per service instance depends on your Discovery plan type.

Plan	Collections per service instance
Cloud Pak for Data	300
Premium	300
Enterprise	300
Plus (includes Trial)	40

Plan details

IBM Cloud Pak for Data The number of collections you can create depends on your hardware configuration. Discovery supports a maximum of 300 collections per instance and installation, but that number depends on many factors, including memory.

Supported file types

Discovery can ingest specific file types. For all other types of files, a warning message is displayed and the file is not ingested.

The following table shows the supported file types and information about feature support that varies by file type.

File type	Text extraction support	Smart Document Understanding (SDU) support	Optical Character Recognition (OCR) support
CSV	✓		
DOC, DOCX	✓	✓	✓
GIF	✓		
HTML	✓		
JPG	✓	✓	✓
JSON	✓		
PDF	✓	✓	✓
PNG	✓	✓	✓

PPT, PPTX	✓	✓	✓
TIFF	✓	✓	✓
TXT	✓		
XLS, XLSX	✓		✓

Supported file types

- PDF files that are secured with a password or certificate are not supported. Vector objects, including SVG images and vectorized text, are not supported. Only images of the supported image file types that occur in the PDF are rendered.
- Only single-page image files are supported.
- Files within compressed archive files (ZIP, GZIP, TAR) are extracted. Discovery ingests the supported file types within the archive; it ignores all other file types. The file names must be encoded in UTF-8. Files with names that include Japanese characters, for example, must be renamed before they are added to the ZIP file.
- Discovery supports MacOS ZIP files only if they are generated by using a command such as: `zip -r my-folder.zip my-folder -x "*.DS_Store"`. ZIP files that are created by right-clicking a folder and clicking **Compress** are not supported.
- PDF files that you upload as part of an archive file are not displayed in the advanced view for a query result that you open from the **Improve and customize** page. If you want the file to be viewable from the advanced view, reimport the PDF file separately from the archive file.

 **Note:** When you add files to a Document Retrieval for Contracts project type, any file types that support SDU and OCR are processed with a pretrained Smart Document Understanding model and Optical Character Recognition automatically.

Document limits

The number of documents that are allowed per service instance depends on your Discovery plan type.

The document limit applies to the number of documents in the index. Upload fewer documents at the start if the enrichments that you plan to apply might increase the number of documents later. For example, the following configurations generate more documents:

- When you split a document, the document is segmented into multiple documents
- CSV files that you upload generate one document per line
- Database data sources that you crawl produce one document per database row
- Each object that is defined in an array in a JSON file results in a separate document

Plan	Documents per service instance
Cloud Pak for Data	Unlimited
Premium	Unlimited
Enterprise	Unlimited
Plus (includes Trial)	500,000

Number of documents per service instance

For the Enterprise plan, you are charged after 100,000 documents per month. For more information about pricing, see [Discovery pricing plans](#).

 **Note:** The maximum allowed number can vary slightly depending on the size of the documents. Use these values as a general guideline.

File size limits

Crawled documents

The maximum size of each file that you can crawl by using a connector differs by deployment type.

IBM Cloud Managed deployments on IBM Cloud

- Premium plans only:
 - Box: 50 MB

- IBM Cloud Object Store: 50 MB
- Salesforce Files objects: 50 MB
- All other data sources: 10 MB
- All other plans: 10 MB

IBM Cloud Pak for Data Installed deployments on IBM Cloud Pak for Data

- All data sources: 32 MB

Uploaded documents

The size of each file that you can upload depends on your Discovery plan type. See the *Maximum document size table for details.

Plan	File size per document
Cloud Pak for Data	50 MB
Premium	50 MB
Enterprise	10 MB
Plus (includes Trial)	10 MB
Maximum document size	

Field limits

When a document is added to a collection, content from the document is evaluated and added to the appropriate fields in an internal index.

For structured data, such as uploaded CSV or JSON files, or data from crawled databases, each column or object is stored as a root-level field. For example, if you add a CSV file to collection, each column in the CSV file is stored as a separate field in the index.

A maximum of 1,000 fields can be added to the index.

You cannot assign the data type, such as Date or String, of a field. The data type is detected automatically and assigned to the field during document ingestion. The assignment is based on the data type that is detected from the first document that is indexed. Ingestion errors can occur in subsequent documents if a different data type is detected for the value in the same field. Therefore, if your documents have a mix of data types in a single field, first ingest the document that has a value with the most flexible data type, such as String, in the field.

When you crawl a website or upload an HTML file, the HTML content is added to the collection and indexed in an `html` field.

The following table shows the maximum size limit for fields per document.

Field type	Maximum allowed size per document
<code>html</code> field	5 MB
Sum of all other fields	1 MB
Maximum field sizes	

If the maximum size of the fields in the document exceeds the allowed limits, they are treated as follows:

- For a document with an oversized `html` field, all of the fields in the document are indexed except the `html` field.



Note: For IBM Cloud Pak for Data version 4.0 and earlier, the entire document is not indexed.

- For a document with oversized non-HTML fields, the document is not indexed.



Tip: If you are uploading a Microsoft Excel file and a message is displayed that indicates that the non-HTML field size limit is exceeded, consider converting the XLS file into a CSV file. When you upload a comma-separated value (CSV) file, each row is indexed as a separate document. As a result, no field size limits are exceeded.

For more information about how fields in uploaded files are handled, see [How fields are handled](#).

Supported data sources

The following table shows the supported data sources for each deployment type.

Data source	IBM Cloud	IBM Cloud Pak for Data
Box	✓	✓
Database (IBM Data Virtualization, IBM Db2, Microsoft SQL, Oracle, Postgres)	✓	
FileNet P8	✓	
HCL Notes	✓	
IBM Cloud Object Storage	✓	
Local file system	✓	
Salesforce	✓	✓
Microsoft SharePoint Online	✓	✓
Microsoft SharePoint On Premises	✓	✓
Website	✓	✓
Microsoft Windows file system	✓	

Supported data sources

Crawl schedule options

When you create a collection, the initial crawl starts immediately. The frequency that you choose for the crawl schedule determines when the next crawl will start.

To create a crawl schedule, complete the following steps:

1. In the **Crawl schedule** section, choose a frequency.

You can schedule the crawler to run at a specific day and time. This option is helpful if you want to avoid heavy load on a target system during business hours. If you specify an hour in the range 1 - 9, add a zero before the hour digit. For example, you can schedule the crawl for **01:00 AM** on Saturdays.

IBM Cloud When you schedule a crawl to run monthly, the day number options are limited to 1 through 28 because you must specify a day that occurs every month, including February which has 28 days.

IBM Cloud Pak for Data Installed deployments have more schedule options:

- If you want to crawl every 12 hours or every 10 days, choose **Custom intervals**. You can schedule the crawler to run on a custom number of days or hours.
- By default, the crawl is scheduled to start during off-peak hours.
- Do not set the interval to a frequency that is shorter than the time it takes for the crawl to finish.
- Do not configure multiple crawlers to run at short intervals.
- If you open a collection in a time zone other than the one in which the collection was created, the Coordinated Universal Time (UTC) offset information is displayed.

2. IBM Cloud Pak for Data Installed deployments have a **More scheduling settings** section where you can choose the type of schedule to use to crawl the data source.

The choices for all of the connectors (except the **Web crawl** connector) are as follows:

- **Full crawling**: Recrawls the external data source to update documents in the collection.
- **Crawling updates (look for new, modified, and deleted contents)**: Updates the collection only if data in the external data source was added, modified, or deleted since the last crawl.
- **Crawling new and modified contents**: Updates the collection only if data in the external data source that was added or modified since the last crawl.

Web crawl connector only: The **Web crawl** connector schedules crawls differently from the other connector types. For the **Web**

crawl connector only, choose among the following options:

- To control the frequency of the crawls yourself, choose this option:

Full crawling

When you choose a full crawl schedule type, the crawl occurs with the frequency that you specify in the *Crawl schedule* section of the page.

- To allow the system to manage the frequency of the crawls for you, choose one of the following options:

Crawling updates (look for new, modified, and deleted contents) or Crawling new and modified contents

When you choose a schedule type that crawls for updates or for new and modified contents, the frequency that you specify for the crawl schedule is ignored. The frequency with which each document is crawled is variable and is managed entirely by the service. And the frequency changes depending on how often changes are found in a document. For example, if 5 of the 10 documents in a collection changed by the end of the first crawl interval, then the frequency is automatically increased for those 5 documents. Currently, the highest frequency at which these self-managed refreshes can run is daily.

You cannot interrupt the automated management of frequency and you cannot trigger a one-off crawl when these types of scheduled crawls are configured.

If you want to change the flexible crawl schedule settings later, you can go to the *Processing settings* page, edit the settings, and then click **Apply changes and reprocess**.

IBM Cloud The next scheduled crawl is displayed on the Activity page.

If you change the schedule frequency, the next scheduled crawl time might not be what you expect. The crawls are set up to occur on a regular schedule at a specific time or day by default. For example, if you change the crawl schedule from weekly to monthly on 11 August, the next crawl might be scheduled for 31 August instead of 11 September. It is not scheduled for exactly a month from the day that you made the change. Instead, it is scheduled to run on the day that is designated as the default run day for the selected crawl frequency.

Stopping a crawl

You can stop a crawl without changing the crawl schedule frequency. This action is helpful if you want to perform a time-consuming task and do not want the crawl to start or run in between the task.

IBM Cloud To stop a crawl, complete the following steps:

1. Open the *Manage collections* page from the navigation panel.
2. Select the collection for which you want to stop the crawl.
3. On the *Activity* page, if the crawl is in progress, click **Stop**.
4. Go to the *Processing settings* page.
5. Set **Apply Schedule** to **No**, and then click **Apply changes and reprocess**.

The crawl is stopped and will not start again until you restart it.

IBM Cloud To restart the crawl, complete the following steps:

1. Open the *Manage collections* page from the navigation panel.
2. Select the collection for which you want to restart the crawl.
3. Go to the *Processing settings* page.
4. Set **Apply Schedule** to **Yes**, and then click **Apply changes and reprocess**.

The crawl starts immediately.

The next crawl will start based on the frequency that is selected in the crawl schedule options. If you want to start the crawl at any time before the scheduled frequency, click **Recrawl** on the *Activity* page.

IBM Cloud Pak for Data

You can temporarily stop a crawl that is in progress.

To stop a crawl temporarily, complete the following steps:

1. Open the *Manage collections* page from the navigation panel.
2. Select the collection for which you want to stop the crawl temporarily.

3. On the **Activity** page, click **Stop**.

The crawl starts again based on the frequency that is specified in the crawl schedule.

Uploading data

You can perform a one-time document upload from your local file system at any time to add data to a project.

You can upload up to 200 files at a time.

To process document sets that are larger than 200 files, you can add them to an external data source and use a data source crawler to upload them. For IBM Cloud Pak for Data deployments, you can use a **Local File System** data source for this purpose.

For more information about the maximum size allowed for each file, see [Document limits](#).

 **Tip:** Before you upload a CSV file to a Content Mining project, consider adding headers to the source file so that any fields that are generated from the file have meaningful names. Without headers, fields are given generic names, such as `column_0`, `column_1`, and so on.

To upload data, complete the following steps:

1. Open your project, go to the **Manage collections** page, and then click **New collection**.
2. Choose **Upload data** as your data source, and then click **Next**.
3. Name the collection.
4. If the language of the documents in storage is not English, select the appropriate language.
For a list of supported languages, see [Language support](#).
5. Optionally, click **More processing settings** to expand the menu, and then click **Apply optical character recognition (OCR)**.



Note: When OCR is enabled and your documents contain images, processing takes longer. For more information, see [Optical character recognition](#).

6. Click **Next**.
7. Browse for the files you want to crawl.

IBM Cloud You can drag documents that you want to add to your collection.

For more information about supported file types, see [Supported file types](#).

8. Click **Finish**.

The file upload is completed quickly. It takes more time for the data to be processed as it is added to the collection. After the files are uploaded and processed, the **Activity** page shows the upload results.

Unlike crawled data sources, you cannot schedule regular updates for uploaded files. If you want to add a later version of a file, delete the earlier version of the file, and then upload the latest version.

For information about how to troubleshoot issues that you might encounter when adding documents to a collection, see [Troubleshooting ingestion](#).

For more information about what happens next, see [How your data source is processed](#).

Configuring IBM Cloud data sources

Overview of IBM Cloud data sources

You can use IBM Watson® Discovery on the IBM Cloud® to connect to and crawl documents from remote sources.

IBM Cloud **IBM Cloud only**

 **Note:** This information applies only to managed deployments. For more information about IBM Cloud Pak for Data data sources, see [Overview of Cloud Pak for Data data sources](#).

Connect to an external data source so that you can pull documents into Discovery on a schedule. Discovery pulls documents from the data source by **crawling** the data source. Crawling is the process of systematically browsing and retrieving documents from a starting location that you specify. When the crawler first processes a data source, it performs a full crawl. Each time the crawler runs after the

initial crawl, it performs a refresh, where it checks for new and changed files only.

 **Important:** All Discovery data source connectors are read-only. Regardless of the permissions that are granted to the crawl account, Discovery never writes, updates, or deletes any content in the original data source.

You can use Discovery to crawl from the following data sources:

- [Box](#)
- [IBM Cloud Object Storage](#)
- [Microsoft SharePoint Online](#)
- [Microsoft SharePoint On Prem](#)
- [Salesforce](#)
- [Web crawl](#)

Your data source isn't listed? Check whether IBM® App Connect has a connector to the data source. You can use a default connector that is built for App Connect to send data from a data source to Discovery. For a list of the data sources supported by App Connect default connectors, see [Connectors A-Z](#). For more information about integrating App Connect with Discovery, see [How to use IBM App Connect with IBM Watson® Discovery](#).

 **Note:** To use an App Connect connector, you must create a separate App Connect instance. Costs that are incurred from a paid App Connect instance are not included with the cost of using Discovery. Except for indexing, Discovery does not support any integration with App Connect that you perform on your own.

Data source requirements

The following requirements and limitations are specific to Discovery on IBM Cloud:

- A collection can connect to only one data source.
- For more information about size limits, which can differ per plan, see the following topics:
 - [Collection limits](#)
 - [Document limits](#)

Installing IBM Secure Gateway for on-premises data

To connect to an on-premises data source, you first need to download, install, and configure IBM® Secure Gateway for IBM Cloud®.

After you install the client for one on-premises data source, you can reuse it for other data sources in the project.

The number of gateways that you can create is limited to 50.

For more information, see [About Secure Gateway](#).

You can use the IBM Secure Gateway with the following connectors only:

- [Web crawl](#)
- [Microsoft SharePoint On Prem](#)

To install IBM® Secure Gateway for IBM Cloud®, complete the following steps:

1. From the data source configuration page, click **Manage connection**.
2. On the **Download and install Secure Gateway client** page, download the appropriate version of IBM® Secure Gateway for IBM Cloud®.
3. After you complete the download, click **Download Secure Gateway and Continue**.
4. When prompted, enter the **Gateway ID** and **Token** that are displayed.

For more information, see [Installing the client](#).

5. On the machine where the Secure Gateway Client is running, open the Secure Gateway dashboard at <http://localhost:9003>.
6. Click **add ACL** on the dashboard, and add the URL of the data source that you want to access to the **Allow access** list.

For example, hostname: mycompany.sharepoint.com or mycompanywebsite.com and port: 80.

7. Return to Discovery, and click **Continue**.
 - If the connection is successful, a **Connection successful** message is displayed.
 - If the connection is unsuccessful, open the IBM® Secure Gateway for IBM Cloud® dashboard, and verify that the endpoints on the **Allow access** list are correct.

Data source connection and data isolation

When you connect to external data sources, you reduce the data isolation of your service instance because data in transit between the source and the service cannot be isolated. All other data isolation (at-rest, administration, query) remains in full. All in-flight communication among services and data sources is encrypted with TLS v1.2. The private keys for the TLS certificates are encrypted at rest with AES-256-GCM encryption. The service certificates expire every three years and the certificate revocation lists are updated monthly. All credentials are sent over an encrypted connection that uses TLS v1.2 and are encrypted at rest with AES-256 encryption. Connections to data sources use the secure protocols that are supported by the data sources.

Viewing collections that are connected to a gateway

You can view a list of collections that are connected to a particular gateway. Complete the following steps to view collections that share a particular gateway:

1. From the **My projects** page, click **Data usage and GDPR**.
2. Click **On premises**.

Collections that share a common gateway are displayed in the **Connected collections** list.

Connecting to data sources with IP restrictions

Some data sources allow crawlers from only a limited number of trusted network addresses or domains to access and process their data. If one of the data sources that you want to connect to limits access in this way, you can add IBM-managed IP addresses to the allowlist of the data source.

 **Tip:** Network addresses are subject to change from time to time. You can monitor for updates to these addresses by subscribing to the repo notifications for this page. Click **Edit Topic** and then select **Watching** in the Notifications dialog of the repo.

- For service instances that are hosted in a US-based data center and that were created on or after 1 May 2020, add the following IP addresses:

**150.238.21.0/28
169.48.255.224/28
174.36.69.128/28**

- For service instances that are hosted in non-US data centers and that were created on or after 21 February 2021, add the following IP addresses:

**159.122.203.64/28
158.175.114.128/28
158.176.107.48/28**

- For a list of IP addresses that you can add to an allowlist for services instances that were created before 1 May 2020 (US) and before 21 February 2021 (non-US), see the [network addresses](#) that are listed for Cloud Foundry.
 - Refer to the **Dallas** data center IP addresses for all US-hosted service instances.
 - Refer to the **London** data center IP addresses for all service instances that are hosted outside the US.

Box

Crawl documents that are stored in a Box data source.

IBM Cloud **IBM Cloud only**

 **Note:** This information applies only to managed deployments. For more information about connecting to Box from an installed deployment, see [Box](#).

What documents are crawled

During the initial crawl of the content, documents from all of the folders that can be accessed from your Box application are crawled and added to your collection. Box notes are stored in JSON format, so Discovery also ingests any Box notes in the specified folders.

The following table illustrates the objects that Discovery can crawl.

Data source	Supports scheduled document refreshes?	Objects that are crawled
Box (App access)	No	Files, folders that you share explicitly

Box (Enterprise access)	Yes (New and modified documents only)	Files, folders
-------------------------	---------------------------------------	----------------

Table 1. Data sources crawling support

When you configure Box with App access only, you must create App Users and share the files that you want to crawl with these users. You cannot crawl Box files that are shared only by the Service Account.

For more information about access, see these Box documentation help topics:

- [App Users](#)
- [Service Accounts](#)

Documents that are deleted from Box are not deleted from the collection.

All Discovery data source connectors are read-only. Regardless of the permissions that are granted to the crawl account, Discovery never writes, updates, or deletes any content in the original data source.

Data source requirements

In addition to the [data source requirements](#) for all managed deployments, your Box data source must meet the following requirement:

You must obtain any required service licenses for the data source that you want to connect to. For more information about licenses, contact the system administrator of the data source.

Prerequisite step

You must create a custom application in Box before you can connect to Box from Discovery.

1. In Box, create a custom app that uses **Server Authentication with JWT** as its authentication method.

For detailed steps, see [Setup with JWT](#) in the Box Developer Documentation.

Follow these guidelines when you create the app:

- During the setup procedure, choose to use the **Server Authentication with JWT** method to verify application identity with a key pair.
- When you configure the custom app, you can choose to use one of the application access levels:
 - App access only
 - App access plus Enterprise access

Refreshing documents on a schedule is supported only when you choose **App access plus Enterprise access**.



Important: If you set up the connection with **App access**, you must create App Users and share the files that you want to crawl with the App Users you define. With this configuration, new and modified documents are not crawled during a refresh.

- If you are an administrator, configure **App access plus Enterprise access**. Otherwise, you can configure the app to have **App access**. However, you must get application approval from a Box administrator.
- For both application access levels, specify the following settings:
 - **Read all folders stored in Box**
 - **Write all folders stored in Box**
 - **Manage Users**

For apps with Enterprise access only: Add this extra scope:

- **Manage Enterprise Properties**

- Enable the following advanced features:
 - **Make API calls using the as-user header**
 - **Generate User Access Tokens**

- Get the custom app authorized by an administrator.

For more information, see [App approval](#) in the Box Developer Documentation.

- After the app is created, authorized, and authentication is configured, download the app settings as a JSON file from the dev console.

You provide the following information from this file when it is requested later:

- `client_id`
- `enterprise_id`
- `client_secret`
- `public_key_id`
- `private_key`
- `passphrase`

Connecting to the Box data source

From your Discovery project, complete the following steps:

1. From the navigation pane, choose **Manage collections**.
2. Click **New collection**.
3. Click **Box**, and then click **Next**.
4. Refer to the values from the Box app settings JSON file that you downloaded during the previous procedure to complete the following fields:

Client ID

The private key that you specify when you configure your Box app.

Client Secret

The client secret that you specify when you configure your Box app.

Enterprise ID

The enterprise ID of the Box account.

Public Key ID

The public key ID that Box generates.

Private Key

A part of the key pair that is generated to interact with the Box website.

Passphrase

The passphrase that is required to decrypt the private key if the private key is an encrypted file.

5. Click **Next**.
6. Name the collection.
7. If the language of the documents in Box is not English, select the appropriate language.
For a list of supported languages, see [Language support](#).
8. **Optional:** Change the synchronization schedule.
For more information, see [Crawl schedule options](#).
9. Choose the folders that you want to crawl.
10. If you want to limit the types of files to add to the collection, you can list the file extensions for file types to either include or exclude.



Important: When you choose to list extensions for file types to exclude, you must add at least one file extension.

For a list of supported file types, see [Supported file types](#).

11. If you want the web crawl to extract text from images on the site, expand **More processing settings**, and set **Apply optical character recognition (OCR)** to **On**.



Note: When OCR is enabled and your documents contain images, processing takes longer. For more information, see [Optical character recognition](#).

12. Click **Finish**.

The collection is created quickly. It takes more time for the data to be processed as it is added to the collection.

If you want to check the progress, go to the Activity page. From the navigation pane, click **Manage collections**, and then click to open the collection.

Currently, not all documents are refreshed during scheduled recrawls. For more information, see the [release note](#).

IBM Cloud Object Storage

Crawl documents that are stored in an IBM Cloud® Object Storage data source.

IBM Cloud **IBM Cloud only**



Note: This information applies only to managed deployments.

What documents are crawled

During the initial crawl of the content, documents from all of the content that can be accessed from the storage endpoint are crawled and added to your collection. You cannot crawl private endpoints.

During subsequent scheduled recrawls, only new and modified documents are crawled and any changes are reflected in your collection. Documents that are deleted from the external data source are not deleted from the collection.

All Discovery data source connectors are read-only. Regardless of the permissions that are granted to the crawl account, Discovery never writes, updates, or deletes any content in the original data source.

The following table illustrates the objects that Discovery can crawl.

Data source	Objects that are crawled
IBM Cloud Object Storage	Buckets, files

Table 1. Data sources crawling support

What you need before you begin

Obtain any required service licenses for the content on the website that you want to connect to. For more information about licenses, contact the system administrator of the data source.

Endpoint

The **endpoint** for your IBM Cloud Object Storage data. For example, `s3.us-south.cloud-object-storage.appdomain.cloud`. Do not include `http://` or `https://` in the endpoint value. For more information, see [Regional Endpoints](#).

In addition to the endpoint, you must provide credentials to enable authentication with the object store. You can choose to use one of the following authentication methods:

HMAC

Uses a hash-based message authentication code to authenticate users. HMAC is a cryptographic authentication technique that uses a hash function and a secret key. The data is scrambled before it is sent over the internet. Then, the intended recipient uses the secret key to unscrambles the data. For more information, see [HMAC authentication](#).

IAM

Uses the IBM Cloud Identity and Access Management (IAM) service to authenticate users. The advantage of this authentication type is that the user can use the same process to access all of the resources in the IBM Cloud Platform. For more information, see [IAM authentication](#).

To access the credential information, go to the service credentials page of your IBM Cloud Object Storage service instance. Expand the service credential to see the credential details.

For more information, see [Service credentials](#) in the Object Storage product documentation.

HMAC authentication

If you want to use HMAC authentication, you must have the following information ready:

Access key id

The `access_key_id` that was generated when the IBM Cloud Object Storage instance was created. For example, `347aa3a4b34344f8bc7c7cccdf856e4c`.

Secret access key

The `secret_access_key` to use to sign requests. This key was generated when the IBM Cloud Object Storage instance was created. For example, `gvurfb82712ad14W7a7915h763a6i87155d30a1234364f61`.

IAM authentication

If you want to use IAM authentication, you must have the following information ready:

IAM API key

For example, `0viPH0Y7LbLNa9eLftrtHPpTjoGv6hbLD1QalRXikliJ`.

Resource instance ID

For example, `cloud-object-storage:global:a/3ag0e9402tyfd5d29761c3e97696b71n:d6f74k03-6k4f-4a82-b165-697354o63903::`.

Connecting to the data source

From your Discovery project, complete the following steps:

1. From the navigation pane, choose **Manage collections**.
2. Click **New collection**.
3. Click **IBM Cloud Object Storage**, and then click **Next**.
4. Choose a credential type, and then complete the fields with the information that you collected earlier.
 - IAM
 - HMAC

Click **Next**.

5. Name the collection.
6. If the language of the documents in storage is not English, select the appropriate language.

For a list of supported languages, see [Language support](#).

7. **Optional:** Change the synchronization schedule.

For more information, see [Crawl schedule options](#).

8. Choose the buckets that you want to crawl.

The more buckets that you select, the longer the processing of the documents takes.

9. If you want to limit the types of files to add to the collection, you can list the file extensions for file types to either include or exclude.



Important: When you choose to list extensions for file types to exclude, you must add at least one file extension.

For a list of supported file types, see [Supported file types](#).

10. If you want the crawler to extract text from images on the site, expand **More processing settings**, and set **Apply optical character recognition (OCR)** to **On**.



Note: When OCR is enabled and your documents contain images, processing takes longer. For more information, see

[Optical character recognition](#)

11. Click **Finish**.

The collection is created quickly. It takes more time for the data to be processed as it is added to the collection.

If you want to check the progress, go to the Activity page. From the navigation pane, click **Manage collections**, and then click to open the collection.

Microsoft SharePoint Online

Crawl documents that are stored in a Microsoft SharePoint Online data source.

IBM Cloud **IBM Cloud only**



Note: This information applies only to managed deployments. For more information about connecting to SharePoint Online from an installed deployment, see [SharePoint Online](#).

What documents are crawled

During the initial crawl of the content, documents from all of the objects that can be accessed from the site collection path that you specify are crawled and added to your collection. You cannot limit the crawl to one library within a site collection, for example. All objects in the specified Site collection path are crawled. Custom metadata that is associated with the SharePoint content is crawled also. You can crawl one site collection path per collection. You cannot crawl **Personal SiteCollections**.

During subsequent scheduled recrawls, only new and modified documents are crawled and any changes are reflected in your collection. Documents that are deleted from the external data source are not deleted from the collection.

All Discovery data source connectors are read-only. Regardless of the permissions that are granted to the crawl account, Discovery never writes, updates, or deletes any content in the original data source.

Discovery can crawl the following objects:

- SiteCollections
- Sites
- SubSites
- Lists
- List Items
- Document Libraries
- List Item Attachments

Data source requirements

In addition to the [data source requirements](#) for all managed deployments, your SharePoint Online data source must meet the following requirements:

- The Site Collection that you connect to must be one that was created with an Enterprise plan. It cannot be a collection that was created with a frontline worker plan.
- You must have an Azure Active Directory user ID with permission to read all of the objects that you want to crawl. For example, `<admin_user>@.onmicrosoft.com`. The user ID does not need **SiteCollection Administrator** permission.

You can choose how to authenticate with the external Microsoft SharePoint account from the following options:

Open Authentication (OAuth v2)

Authenticates with the external data source by using a token so that your user credentials do not need to be shared. With this authentication method, you can log in to your Microsoft account directly to generate a token that is used by Discovery to connect to your data.

The **Sign in with Microsoft** option that uses Open Authentication v2 to authenticate with the external data source is a beta feature.

Before anyone can create connectors that use this authentication method, a user with the **Global Administrator** role must complete a one-time [prerequisite steps](#) to authorize the connection for all projects in the Discovery service instance.

Security Assertion Markup Language (SAML)

An older mechanism for authentication and authorization that requires user credentials to be shared with the Discovery service.

If you choose to use this authentication method, your Microsoft SharePoint account must meet the following requirements:

- Unless you created your SharePoint Online account before January 2020, two-factor authentication is enabled for the account by default. You must disable two-factor authentication.

To view and change your multifactor authentication status, see [View the status for a user](#) or [Change the status for a user](#).

- The crawl user account must have legacy authentication and **Contribute** level permissions enabled.

To enable legacy authentication, go to the [Azure portal](#) or contact your SharePoint administrator.

- The connector supports the **Password hash synchronization (PHS)** method for enabling hybrid identity only. Use any other type (such as Pass-through authentication or Federation) at your own risk.

- You must know the following information:

Username

The username of the user account to use to connect to the SharePoint Online SiteCollection that you want to crawl.

For example, `<janedoe>@exampledomain.onmicrosoft.com`.

Password

The password to connect to the SharePoint Online SiteCollection that you want to crawl.

This value is never returned and is only used when credentials are created or modified.

What you need before you begin

You must have the following information ready. If you don't know it, ask your SharePoint administrator to provide the information or consult the [Microsoft SharePoint developer documentation](#):

Organization URL

The root URL of the source that you want to crawl. Specify the domain name of the URL, for example `https://<company><domain>.com`.

Site collection path

The `site_collection_path` to the section of the site where you want to start the crawl.

For example, if the content that you want to crawl is available from `https://<company>.<domain>.com/sites/test`, then you can specify `https://<company>.<domain>.com` as the Organization URL and `/sites/test` as the Site collection path.

- You cannot specify folder paths as input.
 - You cannot specify a path to an Active Server Page Extended (ASPx) file, such as URLs to document libraries, lists, and subsites.
 - If you don't specify a path, the default value of `/` is used, and the root site collection is crawled.
-
- Application ID:** ID of the data source that you want to crawl. This information is required only if you want to store ACL information that is associated with the source documents.

One-time prerequisite step for OAuth

Before anyone can configure the connector to use OAuth v2 authentication method, a user with the **Global Administrator** role in Microsoft Azure Directory where the data source is located must complete steps to register the Discovery enterprise application in Microsoft Azure. This step must be completed once per Discovery service instance.

The administrator does not need to create the application in Azure. When they choose SharePoint Online as the data source, the Discovery service generates the app automatically. As described in the procedure to follow, during the set up of the connector, the administrator must log in to Microsoft with credentials for a user with the **Global Administrator** role in Microsoft Azure Directory and allow the enterprise application to be registered.

The following steps must be completed by a global administrator one time only per service instance:

- Review the default user access settings that will be applied to the enterprise application in Microsoft Azure.

Enterprise applications can handle user access in many ways. Check the default settings to ensure that they are appropriate for your deployment by completing the following steps:

1. Log in to [Microsoft Azure](#).
2. From the *Enterprise applications* page in *Azure Active Directory*, click *Consent and permissions*.

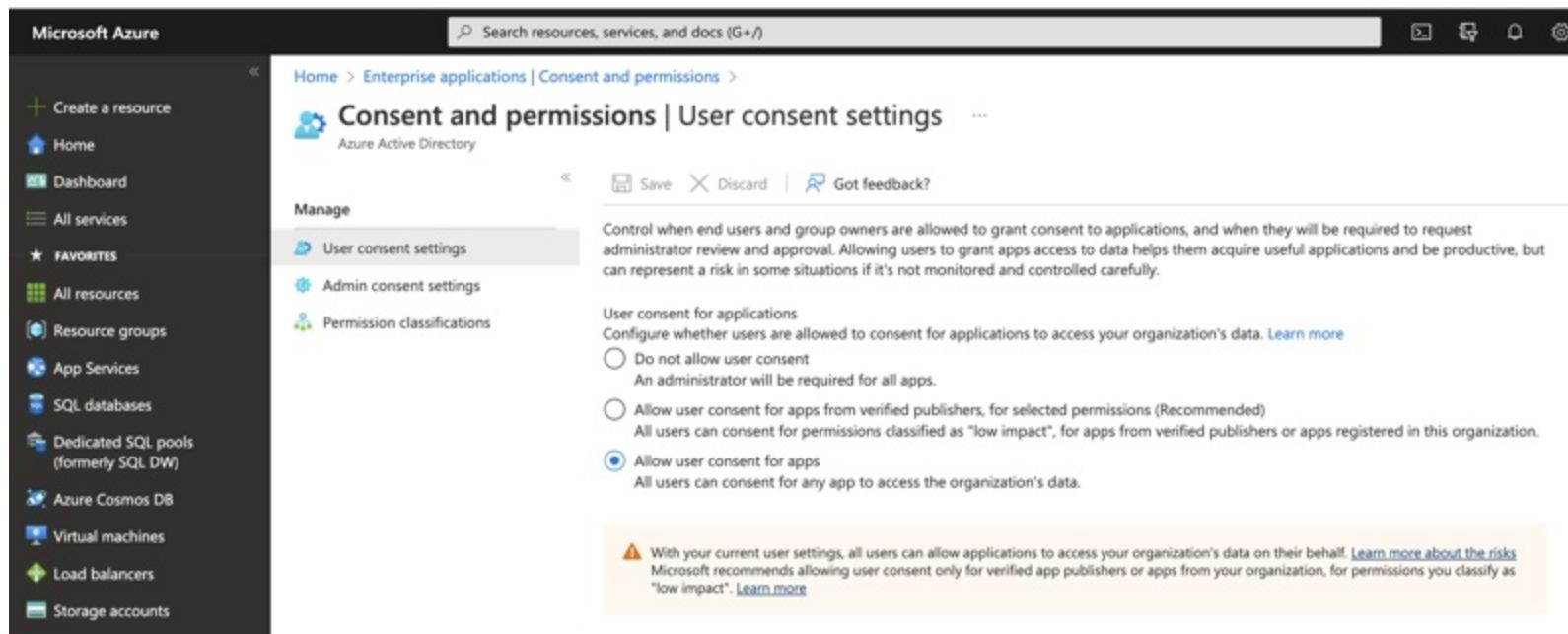


Figure 1. Microsoft Azure Enterprise application permissions user interface

1. Do one of the following things:

- If *Allow user consent for apps* is selected, no more action is needed.
- If *Allow user consent for apps from verified publishers, for selected permissions* is selected, then complete the following steps:

Click *Permissions classifications* link, and then ensure that the following permissions are configured at a minimum:

- Office 365 SharePoint Online: MyFiles.Read
- Office 365 SharePoint Online: AllSites.Read
- Microsoft Graph: offline_access
- Microsoft Graph: profile

The *Do not allow user consent option* is not supported.

The settings that you specify will be applied to the enterprise application that is created by Discovery in subsequent steps.

2. From the navigation pane of Discovery, choose **Manage collections**.

3. Click **New collection**.

4. Click **SharePoint Online**, and then click **Next**.

5. Add a URL to the **Organization URL** field.

6. Click **Sign in with Microsoft**.

Pop-ups must be enabled for this site in your web browser.

The *Sign in with Microsoft* option that uses Open Authentication to authenticate with the external data source is a beta feature.

Log in to your Microsoft SharePoint account with your user name and password, and then complete two-factor authentication, if necessary.

Important: Remember, the credentials you use must have the **Global Administrator** role in Microsoft Azure Directory. If you are not prompted for a user name and password, take note. You might be logged in to a Microsoft SharePoint account already. If you are logged in to an account that you don't want to use for this connector, stop here. (Any account where you are logged in will be used automatically. And you cannot change the account configuration later.) Open a web browser in incognito mode and start this procedure over from step 1.

Discovery generates an enterprise application that it will register with the SharePoint organization that you specify. The enterprise application name has the format **IBM App Connect_{unique name}**.

7. Review the permissions that are associated with the enterprise application that Discovery will register, and then select **Consent on behalf of your organization**.

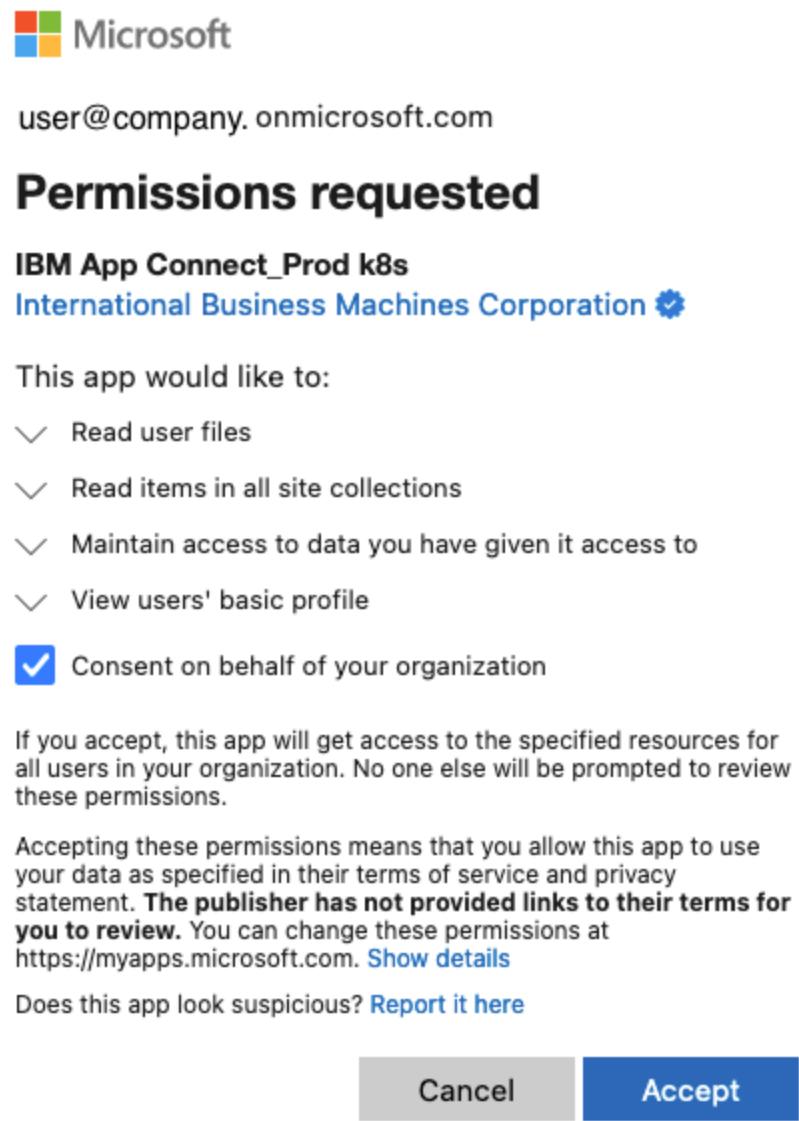


Figure 2. Discovery permission request dialog

8. Click **Accept**.
9. If you want to create a collection, you can name the collection, and then click **Finish**.

Otherwise, you can click **Back** to exit the collection creation process.

Now, anyone from your organization who works in a project that is hosted by the same Discovery service instance can create a collection by using the SharePoint Online connector.

OAuth support revisions

Support for the OAuth method of authentication was added with a software update in February 2022. If you want to update an existing connector to use OAuth instead of SAML, you must re-create the connector. You cannot change the authentication mechanism for an existing connector.

The OAuth method of authentication was updated in January 2023. The enterprise application that is registered with Microsoft Azure now requires **Read** access only. Previously, the enterprise application required **Write** access. If you want to take advantage of this change, delete your current enterprise application and recreate the connector. For more information about how to delete an enterprise application, see [the Microsoft documentation](#).

Connecting to the data source

To configure the Microsoft SharePoint Online data source, complete the following steps in Discovery:

1. From the navigation pane, choose **Manage collections**.
2. Click **New collection**.
3. Click **SharePoint Online**, and then click **Next**.
4. Add a URL to the **Organization URL** field.
5. To enable access to your external data source, choose the method that you want to use to authenticate with the data source from the following options:

Open Authentication (OAuth v2)

Click **Sign in with Microsoft**.

Pop-ups must be enabled for this site in your web browser.

The **Sign in with Microsoft** option that uses Open Authentication to authenticate with the external data source is a beta

feature.

Log in to your Microsoft SharePoint account with your user name and password, and then complete two-factor authentication, if necessary.

Security Assertion Markup Language (SAML)

Specify a username and password for a user that is authorized to access the site you want to crawl, and then click **Next**.

6. Specify the path you want to crawl in the **Site collection path** field.
7. Name the collection.
8. If the language of the documents on the site is not English, select the appropriate language.

For a list of supported languages, see [Language support](#).

9. **Optional:** Change the synchronization schedule.

For more information, see [Crawl schedule options](#).

10. **Optional:** If you want to store any access control information that exists in the SharePoint documents that you crawl, in the **Security** section, set the **Include Access Control List** switch to **On**.

When you enable this option, information about SharePoint access rules that is stored in SharePoint source documents is retained and stored as metadata in the documents that are added to your collection.

This feature is not the same as enabling document-level security for the collection. The access rules in the document metadata are not used by Discovery search. Enabling this feature merely stores the information so that you can leverage the access rules when you build a custom search solution.



Important: Use of this feature increases the size of the documents that are generated in the collection and increases the crawl time. Only enable the feature if your use case requires that you store the SharePoint document ACL information.

If you enable this feature, someone with the administrator role in Microsoft SharePoint must take extra steps to ensure that users who crawl the site have the right permissions to access ACL metadata.

An administrator must complete the following steps:

1. Log in to Microsoft SharePoint.
2. Open the page for your SharePoint site.
3. From the settings menu, choose **Site permissions**.
4. Click **Advanced permission settings**.
5. Make sure that people who want to collect access control information during a crawl have or are members of a group that has the **Full Control** permission for the site.

Type	Permission Levels
SharePoint Group	Edit
SharePoint Group	Full Control
SharePoint Group	Read

Figure 3. Microsoft SharePoint permissions user interface



Note: When access control list information is not extracted, **Read** permission is sufficient for all users who crawl the content.

11. If you want to limit the types of files to add to the collection, you can list the file extensions for file types to either include or exclude.



Important: When you choose to list extensions for file types to exclude, you must add at least one file extension.

For a list of supported file types, see [Supported file types](#).

12. If you want the crawler to extract text from images on the site, expand **More processing settings**, and set **Apply optical character recognition (OCR)** to **On**.



Note: When OCR is enabled and your documents contain images, processing takes longer. For more information, see [Optical character recognition](#).

13. Click **Finish**.

The collection is created quickly. It takes more time for the data to be processed as it is added to the collection.

If you want to check the progress, go to the Activity page. From the navigation pane, click **Manage collections**, and then click to open the collection.



Note: You cannot currently change the user account that is associated with the OAuth setup later, nor any of the details of the existing user account that the connector is configured to use. For example, you cannot update the password that was used to set up the connection after a password change in SharePoint.

Sample access control list information

The following screen capture illustrates the type of ACL information that is stored in the document when you include the access control list.

```
document_id": "sharepoint_filecollection_c088dd58-5a12-476a-847d-38030f1211eb",
  "result_metadata": {
    "collection_id": "0e36fdd2-7fb0-812b-0000-017edabfa1ab"
  },
  "enriched_text": [
    {...}
  ],
  "metadata": {
    "parent_document_id": "sharepoint_filecollection_c088dd58-5a12-476a-847d-38030f1211eb",
    "source": {
      "LinkingUrl": "",
      "Modified": "2020-07-07T03:18:14Z",
      "TimeLastModified": "2020-07-07T03:18:13Z",
      "ContentTypeId": "0x010100036B86C6B029AA42831269188B39583E",
      "acl": [
        "c:0o.c|federateddirectoryclaimprovider|",
        "i:0#.f|membership|",
        "SHAREPOINT\\system",
        "c:0t.c|tenant|",
        "c:0o.c|federateddirectoryclaimprovider|",
        "i:0#.f|membership|",
        "i:0#.f|membership|"
      ]
    }
  }
}
```

Figure 4. Representation of ACL information in document metadata

Microsoft SharePoint On Prem

Crawl documents that are stored in a Microsoft SharePoint data source that is hosted on premises.

IBM Cloud **IBM Cloud only**



Note: This information applies only to managed deployments. For more information about connecting to an on-premises SharePoint data source from an installed deployment, see [SharePoint On Prem](#).

What documents are crawled

During the initial crawl of the content, documents from all of the objects that can be accessed from the site collection path that you specify are crawled and added to your collection. Custom metadata that is associated with the SharePoint content is crawled also. You can crawl one site collection path per collection. You cannot crawl **Personal SiteCollections**.

During subsequent scheduled recrawls, only new and modified documents are crawled and any changes are reflected in your collection. Documents that are deleted from the external data source are not deleted from the collection.

All Discovery data source connectors are read-only. Regardless of the permissions that are granted to the crawl account, Discovery never writes, updates, or deletes any content in the original data source.

The following table illustrates the objects that Discovery can crawl.

Data source	Objects that are crawled
Microsoft SharePoint On Prem	SiteCollections, Sites, SubSites, Lists, List Items, Document Libraries, List Item Attachments

Table 1. Data sources crawling support

Data source requirements

In addition to the [data source requirements](#) for all managed deployments, your SharePoint On Prem data source must meet the following requirements:

- You can connect to a SharePoint 2013, 2016, or 2019 on-premises data source.
- The user ID must have **SiteCollection Administrator** permission and be able to access all of the sites and lists that they want to crawl.
- The crawler supports Windows New Technology LAN Manager (NTLM) v1 authentication only. It does not support NTLM v2 or Security Assertion Markup Language (SAML) authentication.

What you need before you begin

You must have the following information ready. If you don't know it, ask your SharePoint administrator to provide the information or consult the [Microsoft SharePoint developer documentation](#):

Username

The username to use to connect to the SharePoint On Prem web application that you want to crawl. For example, `siteadmin01`.

Password

The password to connect to the SharePoint On Prem web application that you want to crawl. This value is never returned and is only used when credentials are created or modified.

Web Application URL

The SharePoint web application URL. For example, `https://sharepointwebapp.com:8443`. If you do not enter a port number, the default value of **80** is used for an HTTP URL and **443** for HTTPS.

Domain

The domain name of the SharePoint On Prem account. For example, `sharepoint.mycointernal`.

Prerequisite step

Before you can connect to a SharePoint On Prem data source, you must install and configure IBM® Secure Gateway for IBM Cloud®.

For more information, see [Installing IBM Secure Gateway for on-premises data](#).

Connecting to the data source

To configure the Microsoft SharePoint On Prem data source, complete the following steps in Discovery:

1. From the navigation pane, choose **Manage collections**.
2. Click **New collection**.
3. Click **SharePoint On Prem**, and then click **Next**.
4. Add values to the following fields:
 - Username
 - Password
 - Web Application URL
 - Domain

Click **Next**.

5. Name the collection.

6. If the language of the documents on the site is not English, select the appropriate language.
For a list of supported languages, see [Language support](#).
7. **Optional:** Change the synchronization schedule.
For more information, see [Crawl schedule options](#).
8. If you want to limit the types of files to add to the collection, you can list the file extensions for file types to either include or exclude.

 **Important:** When you choose to list extensions for file types to exclude, you must add at least one file extension.

- For a list of supported file types, see [Supported file types](#).
9. If you want the crawler to extract text from images on the site, expand **More processing settings**, and set **Apply optical character recognition (OCR)** to **On**.

 **Note:** When OCR is enabled and your documents contain images, processing takes longer. For more information, see [Optical character recognition](#).

10. Click **Finish**.

The collection is created quickly. It takes more time for the data to be processed as it is added to the collection.

If you want to check the progress, go to the Activity page. From the navigation pane, click **Manage collections**, and then click to open the collection.

Salesforce

Crawl documents that are stored in a Salesforce data source.

IBM Cloud **IBM Cloud only**

 **Note:** This information applies only to managed deployments. For more information about connecting to Salesforce from an installed deployment, see [Salesforce](#).

What documents are crawled

During the initial crawl of the content, documents from all of the objects that can be accessed from the URL that you specify are crawled and added to your collection. Knowledge Articles are crawled only if their **version** is **published** and their languages is **en-us**.

During subsequent scheduled recrawls, only new and modified documents are crawled and any changes are reflected in your collection. Documents that are deleted from the external data source are not deleted from the collection.

All Discovery data source connectors are read-only. Regardless of the permissions that are granted to the crawl account, Discovery never writes, updates, or deletes any content in the original data source.

Discovery can crawl the following objects:

- Any default and custom objects that you have access to
- Accounts
- Contacts
- Cases
- Contracts
- Knowledge articles
- Attachments

Data source requirements

In addition to the [data source requirements](#) for all managed deployments, your Salesforce data source must meet the following requirements:

- The instance that you plan to connect to must be part of an Enterprise plan or higher.
- You must obtain any required service licenses for the data source that you want to connect to. For more information about licenses, contact the system administrator of the data source.

What you need before you begin

You must have the following information ready. If you don't know it, ask your Salesforce administrator to provide the information or

consult the [Salesforce developer documentation](#).

Username

The **username** of an account that has access to the Salesforce site. For example, `jdoe@example.com`

Password

The password associated with the username. For example, `myP@ssw0rd`.

Service token

A valid Salesforce security token. For example, `mna08jsRET5CiJww9JnURLNN`.

URL

The URL of the Salesforce site that you want to crawl. For example, `https://my.salesforce.com`

Connecting to the data source

To configure the Salesforce data source, complete the following steps in Discovery:

1. From the navigation pane, choose **Manage collections**.
2. Click **New collection**.
3. Click **Salesforce**, and then click **Next**.
4. Add values to the following fields:
 - Username
 - Password plus service token

To form the password, concatenate the Password and Service token values that you noted earlier. For example, `myP@ssw0rdmna08jsRET5CiJww9JnURLNN`. The password and token values are never returned and are used only when credentials are created or modified.

- URL

Click **Next**.

5. Name the collection.
6. If the language of the documents on the site is not English, select the appropriate language.

For a list of supported languages, see [Language support](#).

7. **Optional:** Change the synchronization schedule.

For more information, see [Crawl schedule options](#).

8. Select the objects that you want to crawl.

The more objects that you select, the longer the processing of the documents takes.

9. If you want to limit the types of files to add to the collection, you can list the file extensions for file types to either include or exclude.



Important: When you choose to list extensions for file types to exclude, you must add at least one file extension.

For a list of supported file types, see [Supported file types](#).

10. If you want the crawler to extract text from images on the site, expand **More processing settings**, and set **Apply optical character recognition (OCR)** to `On`.



Note: When OCR is enabled and your documents contain images, processing takes longer. For more information, see [Optical character recognition](#).

11. Click **Finish**.

The collection is created quickly. It takes more time for the data to be processed as it is added to the collection.

If you want to check the progress, go to the Activity page. From the navigation pane, click **Manage collections**, and then click to open the collection.

Web crawl

Add a web crawl collection to crawl a website, analyze its page content, and store meaningful information. Specify one or more base web page URLs and configure how many linked pages for the web crawl to follow. You can configure how often to synchronize with the website, so you control how up to date the data in your collection is.

IBM Cloud **IBM Cloud only**



Note: This information applies only to managed deployments. For more information about connecting to a website from an installed deployment, see [Web crawl](#).

What documents are crawled

You can connect to the following types of web content:

- Public websites
- Private company websites or other sites that require authentication
- Websites that are behind a corporate firewall

During the initial crawl of the content, all website pages that match your search settings are crawled and added to the document index of your collection. The crawl starts on the web page that you specify in the **Starting URLs** field. If your collection is configured to follow links, the crawl follows links on the starting page that share the same subtree as the starting page. For example, if you specify

`https://www.example.com/banking/faqs.html`, links with URLs that begin with `https://www.example.com/banking/` are crawled. If you specify `https://www.example.com/banking`, links with URLs that begin with `https://www.example.com/` are crawled.

The crawl cannot access secure subdirectories. For example, if a subdirectory that you expect the crawl to access, such as `https://www.example.com/banking/pdfs`, isn't being crawled, check whether you can access the subdirectory URL from a web browser directly. If you can't access it, the crawl can't access it.

During subsequent scheduled recrawls, a full recrawl is performed and any changes are reflected in your collection. Documents that were added to your collection from website pages that are later deleted from the external website are not deleted from the collection. However, starting with collections that were created after April 2022, when you remove a starting URL from the web crawl configuration, any associated documents are deleted. Deleted documents include indexed documents that were added to the collection based on the content of the web page at the starting URL and documents that were derived from web pages that the starting URL linked to. You cannot limit the number of indexed documents by changing other settings, such as changing the existing URL to include a path with a more limited scope than before or reducing the maximum number of links to follow to 0. Only by deleting the URL can you remove the indexed documents that are associated with it.

The web crawler can crawl web pages that use JavaScript to render content, but the crawler works best on individual pages, not entire websites. It cannot crawl sites that use dynamic URLs; if you can't see any content when you view the source code of a web page in your browser, then the service cannot crawl it.

If you want to crawl a group of URLs that includes some websites that require authentication and some that don't, consider creating a different collection for each authentication type. The connector does not support cookie-based crawling.

All Discovery data source connectors are read-only. Regardless of the permissions that are granted to the crawl account, Discovery never writes, updates, or deletes any content in the original data source.

The following table illustrates the objects that Discovery can crawl.

Objects that are crawled

Websites, website subdirectories

Table 1. Data sources crawling support

Prerequisite step

If you want to connect to a website that is hosted behind a firewall, set up an IBM® Secure Gateway for IBM Cloud® connection first.

Valuable content is often stored on your company's internal website. Typically, such intranet websites are accessible only from a computer that is connected to your office network or through a VPN connection. You can establish a persistent and more secure connection between the web crawler and this type of internal site by using Secure Gateway.

For more information about how to set up the connection, see [Installing IBM Secure Gateway for on-premises data](#).

Connecting to the data source

To configure the web crawl collection, complete the following steps:

1. From the navigation pane, choose **Manage collections**.
2. Click **New collection**.
3. Click **Web crawl**, and then click **Next**.
4. Name the collection.
5. If the language of the content on the website is not English, select the appropriate language.

For a list of supported languages, see [Language support](#).

6. **Optional:** You can change the synchronization schedule.

For more information, see [Crawl schedule options](#).

7. Specify the URL of the website that you want to crawl.

- o If the site you want to crawl requires a login, set **Basic authentication** to **On**, add the URL of the page to the **Starting URL** field, and then click **Add**.

Add a username and password with access to the site, and then click **Save credentials**. You can specify only one set of credentials per collection.

For example, you can specify <https://cloud.ibm.com> as the starting URL and add your IBMid as the credentials.

If you want to start the crawl from a specific section of the site, specify it in the **Starting URLs** field. The domain name of the subsection must match the domain in the URL you specified earlier.

For example, you might change the starting URL to <https://cloud.ibm.com/unifiedsupport/supportcenter>.

- o For any public web pages that you want to crawl, add the URL for the root page of the website to the **Starting URLs** field, and then click **Add**. You can add more than one starting page.

The final forward slash (/) in the URL determines the subtree to crawl. If you specify <https://www.example.com/banking/faqs.html>, all URLs that begin with <https://www.example.com/banking/> are crawled, for example. If you specify <https://www.example.com/banking> all URLs that begin with <https://www.example.com/> are crawled.

By default, the number of consecutive links that the crawl follows from the starting URL is **2**. To change the number of hops or to list website sections to exclude from the crawl, click the edit icon.

- The maximum number of hops allowed is **20**.
- To specify URL paths to exclude, add the site path. For example, if the starting URL is <https://example.com>, you can exclude <https://example.com/pricing> by entering </pricing/>.

Any section of the web address that contains the site path you specify is excluded. For example, if you specify </licenses/>, the page <https://example.com/products/licenses/europe> is excluded, among others.

- If you want to restrict the crawl to a single page, add the URL to the **Starting URLs** field. For example, <https://www.example.com/banking/faqs.html>. Click the edit icon to set the **Maximum number of links to follow** to **0**.

- o If the website that you want to crawl uses JavaScript to customize the page content before it is displayed, you must take an extra step.

After you enter the starting URL and click **Add**, edit the URL by clicking the edit icon . Set the **Execute JavaScript during crawl** switcher to **On**, and then click **Save**.

Note: When JavaScript processing is enabled, it takes 3 to 4 times longer to crawl a page. Use it only on individual web pages where you know it is necessary because the page renders its content dynamically. If you see timeout messages or the crawl ends without adding content to the collection, decrease the number of web pages that are included in the crawl. For example, you can specify the exact page to crawl in the **Starting URLs** field, and set **Maximum number of links to follow** to 0.

- o To connect to a website that is hosted behind a firewall, [set up an IBM® Secure Gateway for IBM Cloud® connection first](#).

Expand **More connection settings**, and then set **Connect to on-premises network** to **On**. Provide details about your Secure Gateway connection.

8. Optional: Add another web address to the **Starting URLs** field.

 **Important:** The number of starting URLs for a single collection must be less than 100. If you have a requirement to crawl a large number of websites, see [I need to crawl lots of sites. What's my limit?](#).

The number of web pages that are crawled is limited to 250,000, so the web crawler might not crawl all the specified websites.

The number of child URLs per URL that are crawled is limited to 10,000. If the number of child URLs within any crawled URL exceeds 10,000, the crawler cannot process any of the content in the child URLs.

9. If you want to limit the types of files to add to the collection, you can list the file extensions for file types to either include or exclude.

 **Important:** If the URLs for your website pages do not end in `.html`, use the exclude filter instead of the include filter. You must add at least one file extension to exclude.

For a list of supported file types, see [Supported file types](#).

10. If you want the web crawl to extract text from images on the site, expand **More processing settings**, and set **Apply optical character recognition (OCR)** to **On**.

 **Note:** When OCR is enabled and your documents contain images, processing takes longer. For more information, see [Optical character recognition](#).

11. Click **Finish**.

The collection is created quickly. It takes more time for the data to be processed as it is added to the collection.

If you want to check the progress, go to the Activity page. From the navigation pane, click **Manage collections**, and then click to open the collection.

I need to crawl lots of sites. What's my limit?

The service can support a total of 500 crawler connections per Discovery service instance. All of the data sources except Web crawl use one crawler connection each. For Web crawl, one connection is required for every 5 starting URLs. If you add 10 starting URLs, for example, Discovery generates the extra crawler connection that is needed to support the extra 5 URLs. Therefore, the maximum number of starting URLs that you can use depends on the other data collections that are configured in your service instance. You can calculate the limit yourself.

To calculate the starting URL limit, complete the following steps:

1. Calculate the number of other data source collections in the service instance, meaning this project and any other projects in the same Discovery instance.

For example, you might have 2 IBM Cloud Object Store collections in one project and 2 Salesforce collections and 1 SharePoint Online collection in another project. In this example, the total number of other data source collections is 5.

2. Subtract the number of other data source collections from the maximum allowed number of crawler connections, which is 500.

For example, $500 - 5 = 495$.

3. Multiply the remainder by 5 to determine the total number of starting URLs that you can use.

For example, $495 \times 5 = 2,475$.

 **Note:** To use the maximum-allowed number of starting URLs in the example, you would need 25 web crawl collections because each collection allows a maximum of 100 starting URLs to be configured. However, don't configure your instance to use the absolute maximum number allowed. If one or more additional data sources are added subsequently to a project in this service instance, it will impact the number of starting URLs that the instance can crawl successfully.

Troubleshooting crawler issues

A 403 Forbidden error is returned

The website that you want to crawl might block requests from all but a specific set of named entities. If possible, add the crawler to the allowlist for the site. The identifying header for the crawler is **User-Agent: IBM-AppConnect/V1**.

Configuring IBM Cloud Pak for Data data sources

Overview of Cloud Pak for Data data sources

In Discovery for Cloud Pak for Data, you can crawl documents from a local source that you upload or from a remote data source that you connect to. Learn more about the supported data sources and how to configure them.

IBM Cloud Pak for Data [IBM Cloud Pak for Data only](#)

 **Note:** This information applies only to installed deployments. For more information about IBM Cloud data sources, see [Overview of the IBM Cloud data sources](#).

 **Important:** All Discovery data source connectors are read-only. Regardless of the permissions that are granted to the crawl account, Discovery never writes, updates, or deletes any content in the original data source.

You can use Discovery for Cloud Pak for Data to crawl from the following data sources:

- [Box](#)
- [Database](#)
- [FileNet P8](#)
- [LDAP directory](#)
- [Local File System](#)
- [Notes](#)
- [Salesforce](#)
- [SharePoint Online](#)
- [SharePoint On Prem](#)
- [Web crawl](#)
- [Windows File System](#)

Your data source isn't listed? You can work with a developer to create a custom connector. For more information, see [Building a Cloud Pak for Data custom connector](#).

If you have special requirements when you add source documents, such as a need to exclude certain files, you can work with a developer to create a custom crawler plug-in. The crawler plug-in can apply more nuanced rules to what documents and what fields in the documents get added. For more information, see [Building a Cloud Pak for Data custom crawler plug-in](#).

Data source requirements

The following requirements and limitations are specific to IBM Watson® Discovery:

- The individual file size limit is 32 MB per file, which includes compressed archive files (ZIP, CZIP, TAR). When decompressed, the individual files within compressed files cannot exceed 32 MB per file. This limit is the same for collections in which you upload your own data.
- Depending on the type of installation (starter or production mode), the number of collections you can ingest simultaneously varies. A starter installation includes one `crawler` pod, which allows three collections to be processed simultaneously. A production installation includes two `crawler` pods, which can process six collections simultaneously.

If you are running a starter installation and you want to process more than three collections simultaneously, you must increase the number of `crawler` pods by running the following commands:

```
$ oc patch wd wd --type=merge --patch='{"spec": {"ingestion": {"crawler": {"replicas": <number-of-replicas>} } } }'
```

 **Note:** In a starter installation, the maximum number of simultaneous collections that can crawl an external data source is 3. If you start a fourth, that collection does not start to process until the prior three crawls finish.

Each `number-of-replicas` allows 3 simultaneous crawls, so `number-of-replicas=2` increases the replicas to 6, and `number-of-replicas=3` increases them to 9.

Crawler plug-in settings

When you deploy one or more crawler plug-ins, you can configure your collection to use one of the plug-ins.

These settings are only available when crawler plug-ins are deployed.

- For more information about building a plug-in, see [Building a Cloud Pak for Data crawler plug-in](#).
- For more information about deploying a crawler plug-in, see [Commands and options for managing your crawler plug-ins](#).

When you are ready to configure a collection to use a crawler plug-in that was created by using the `scripts/manage_crawler_plugin.sh` script, you can see a **Plug-in settings** section with the following options:

- **Enable plug-in:** The switch is set to **Off**. Enable this option if you want to use a crawler plug-in to process documents.
- **Plug-in:** Lists the names of available crawler plug-ins. Select a plug-in to use.

Supporting document-level security

If document-level security is activated, you can use the security settings from your source documents to control the search results that are returned to different users.

Discovery supports prefiltering only. To prefilter, Discovery replicates the document's source access control list (ACL) at crawl time into the index. The search engine must compare user credentials to the replicated document ACLs. Discovery is faster when documents are prefiltered and when you control which documents you add to the index. However, it is difficult to model all of the security policies of the various data sources in the index and implement comparison logic uniformly. Also, prefiltering is not as responsive to changes that occur in the source ACLs after the most recent crawl.

Document-level security is supported by the following data source types:

- Box
- FileNet P8
- HCL Notes
- Microsoft SharePoint Online
- Microsoft SharePoint On Prem
- Microsoft Windows File System

 **Important:** When you query collections where document-level security is enabled, no results are returned if the users associated with your Discovery instance are not present in the source system. For more information about querying these collections, see [Querying with document-level security enabled](#).

To enable document-level security, you must complete the following steps:

1. [Create Discovery users that match the users available on the source system](#).
2. Associate users with your Discovery instance. For more information, see [Giving users access to a Watson Discovery instance](#).
3. Enable document-level security for the data source when you connect to it.

Creating users for document-level security

You must create users that match the users available on the source system that Discovery is connecting to so that they can query with document-level security enabled.

1. Log in to Discovery as an administrator.
2. Create users who match the users available on your source or who are connected to the identity provider that your source system uses. If you create users for document-level security, keep the following points in mind:
 - Optional: For each user that you want to have access to query results, you must add users. The username must match the username that the source uses. This option is only for development and testing purposes. To create users individually, see [Managing users](#).
 - To connect to an identity provider that the source is using, see [Connecting to your identity provider](#).

 **Note:** Discovery does not synchronize changes that are made to the users in the identity provider with the user list for the service. Discovery administrators must ensure that the user list is current and remove any noncurrent users.

Box

Crawl documents that are stored in a Box data source.

IBM Cloud **IBM Cloud only**

 **Note:** This information applies only to managed deployments. For more information about connecting to Box from an installed deployment, see [Box](#).

What documents are crawled

During the initial crawl of the content, documents from all of the folders that can be accessed from your Box application are crawled and added to your collection. Box notes are stored in JSON format, so Discovery also ingests any Box notes in the specified folders.

The following table illustrates the objects that Discovery can crawl.

Data source	Supports scheduled document refreshes?	Objects that are crawled
-------------	--	--------------------------

Box (App access)	No	Files, folders that you share explicitly
Box (Enterprise access)	Yes (New and modified documents only)	Files, folders

Table 1. Data sources crawling support

When you configure Box with App access only, you must create App Users and share the files that you want to crawl with these users. You cannot crawl Box files that are shared only by the Service Account.

For more information about access, see these Box documentation help topics:

- [App Users](#)
- [Service Accounts](#)

Documents that are deleted from Box are not deleted from the collection.

All Discovery data source connectors are read-only. Regardless of the permissions that are granted to the crawl account, Discovery never writes, updates, or deletes any content in the original data source.

Data source requirements

In addition to the [data source requirements](#) for all managed deployments, your Box data source must meet the following requirement:

You must obtain any required service licenses for the data source that you want to connect to. For more information about licenses, contact the system administrator of the data source.

Prerequisite step

You must create a custom application in Box before you can connect to Box from Discovery.

1. In Box, create a custom app that uses **Server Authentication with JWT** as its authentication method.

For detailed steps, see [Setup with JWT](#) in the Box Developer Documentation.

Follow these guidelines when you create the app:

- During the setup procedure, choose to use the **Server Authentication with JWT** method to verify application identity with a key pair.
- When you configure the custom app, you can choose to use one of the application access levels:
 - App access only
 - App access plus Enterprise access

Refreshing documents on a schedule is supported only when you choose **App access plus Enterprise access**.



Important: If you set up the connection with **App access**, you must create App Users and share the files that you want to crawl with the App Users you define. With this configuration, new and modified documents are not crawled during a refresh.

- If you are an administrator, configure **App access plus Enterprise access**. Otherwise, you can configure the app to have **App access**. However, you must get application approval from a Box administrator.
- For both application access levels, specify the following settings:
 - Choose the following scopes:
 - **Read all folders stored in Box**
 - **Write all folders stored in Box**
 - **Manage Users**

For apps with Enterprise access only: Add this extra scope:

- **Manage Enterprise Properties**

- Enable the following advanced features:
 - **Make API calls using the as-user header**
 - **Generate User Access Tokens**

- Get the custom app authorized by an administrator.

For more information, see [App approval](#) in the Box Developer Documentation.

- After the app is created, authorized, and authentication is configured, download the app settings as a JSON file from the dev

console.

You provide the following information from this file when it is requested later:

- `client_id`
- `enterprise_id`
- `client_secret`
- `public_key_id`
- `private_key`
- `passphrase`

Connecting to the Box data source

From your Discovery project, complete the following steps:

1. From the navigation pane, choose **Manage collections**.
2. Click **New collection**.
3. Click **Box**, and then click **Next**.
4. Refer to the values from the Box app settings JSON file that you downloaded during the previous procedure to complete the following fields:

Client ID

The private key that you specify when you configure your Box app.

Client Secret

The client secret that you specify when you configure your Box app.

Enterprise ID

The enterprise ID of the Box account.

Public Key ID

The public key ID that Box generates.

Private Key

A part of the key pair that is generated to interact with the Box website.

Passphrase

The passphrase that is required to decrypt the private key if the private key is an encrypted file.

5. Click **Next**.
6. Name the collection.
7. If the language of the documents in Box is not English, select the appropriate language.
For a list of supported languages, see [Language support](#).
8. **Optional:** Change the synchronization schedule.
For more information, see [Crawl schedule options](#).
9. Choose the folders that you want to crawl.
10. If you want to limit the types of files to add to the collection, you can list the file extensions for file types to either include or exclude.



Important: When you choose to list extensions for file types to exclude, you must add at least one file extension.

For a list of supported file types, see [Supported file types](#).

11. If you want the web crawl to extract text from images on the site, expand **More processing settings**, and set **Apply optical character recognition (OCR)** to **On**.



Note: When OCR is enabled and your documents contain images, processing takes longer. For more information, see [Optical character recognition](#).

12. Click **Finish**.

The collection is created quickly. It takes more time for the data to be processed as it is added to the collection.

If you want to check the progress, go to the Activity page. From the navigation pane, click **Manage collections**, and then click to open the collection.

Currently, not all documents are refreshed during scheduled recrawls. For more information, see the [release note](#).

Database

Crawl documents that are stored in a database that supports the Java Database Connectivity (JDBC) API.

IBM Cloud Pak for Data [IBM Cloud Pak for Data only](#)



Note: This information applies only to installed deployments.

What documents are crawled

- Each row in the database is crawled and added to the collection as one document. The columns are indexed as metadata.
- The crawler attempts to crawl and index content, such as BLOB/BINARY, that is stored in the database. File types that are supported by Discovery are indexed. For more information, see [Supported file types](#).
- When a source is recrawled, new documents are added, updated documents are modified to the current version, and deleted documents are deleted from the collection's index.
- All Discovery data source connectors are read-only. Regardless of the permissions that are granted to the crawl account, Discovery never writes, updates, or deletes any content in the original data source.

Data source requirements

In addition to the [data source requirements](#) for all installed deployments, your database data source must meet the following requirements:

- Discovery supports the following data source versions:
 - Data Virtualization on IBM Cloud Pak for Data 1.8.0, 1.8.3 which use Db2 11.5
 - IBM Db2: 10.5, 11.1, 11.5
 - Microsoft SQL Server: 2012, 2014, 2016, 2017
 - Oracle Database: 12c, 18c, 19c
 - PostgreSQL: 9.6, 10, 11



Note: Support for Data Virtualization was added with IBM Cloud Pak for Data 4.5.x releases

- You must obtain any required service licenses for the data source that you want to connect to. For more information about licenses, contact the system administrator of the data source.

Prerequisite step

- Decide which database tables you want to crawl. You can crawl multiple tables in a collection, and you can specify tables that have different schemas, or sets of columns. You must know the following information:
 - Schema names
 - Table names

For Data Virtualization on IBM Cloud Pak for Data, you can get these details from the IBM Cloud Pak for Data web client. Click the main menu icon, expand Data, and then select **Data virtualization**. At the start of the page, choose to show **Virtualized data**.

Virtualized data ▾

Table	Schema name
<input type="checkbox"/> NHTSA	ADMIN

Figure 1. Virtualized data view in Cloud Pak for Data

- Be careful if you plan to crawl multiple tables that have columns with the same name but different data types. In Content Mining projects, columns with the same name but different data types are assigned to fields that have a data type suffix in the name, such as **DATA_string**. In all other project types, the data in one of the tables is excluded from the index. For example, if you have two tables that have columns that are called **DATA** and the **DATA** column in one table is populated with dates and the column in the other table is populated with strings, the data in one of the tables is excluded from the index.
- Get the user credentials for a user who has permission to access the tables that you want to crawl.
- Before you can connect to a database, you must get the JDBC driver library for the database. When you set up the database data source, you are asked to specify the JDBC driver class path.
- Before you can connect to the Data Virtualization service by using JDBC, you must install IBM Data Server driver packages. For more information, see [Connecting applications to the Data Virtualization service](#).
- If you want to connect to an instance of Data Virtualization that is hosted in a different cluster from your Discovery service, you must forward traffic that is routed for Data Virtualization from an external infrastructure node to the master nodes of your cluster. For more information, see [Updating HAProxy configuration file](#).

1. Download the JAR files for the JDBC driver library from the database server or vendor's website.

The following files are associated with each database:

- Db2 and Data Virtualization: **db2jcc4.jar**
- Oracle: **ojdbc8.jar**
- SQL Server: **mssql-jdbc-7.2.2.jre8.jar**
- PostgreSQL: **postgresql-42.2.6.jar**

2. Compress the JAR files into a single compressed file.

If you have a JDBC driver that has only one JAR file, skip this step.

3. Make a note of where the driver is stored. You must specify the directory where you store this JAR or compressed file in the next procedure so that Discovery can upload it.

Connecting to a database data source

⚠ Important: Before you begin, if you plan to apply enrichments to your data, create the collection in a Content Mining project type. If you are using a different project type and plan to apply enrichments, stop here. For more information, see [Applying enrichments to content from a database](#).

From your Discovery project, complete the following steps:

1. From the navigation pane, choose **Manage collections**.
2. Click **New collection**.
3. Click **Database**, and then click **Next**.
4. Name the collection.
5. If the language of the documents in the database is not English, select the appropriate language.

For a list of supported languages, see [Language support](#).

6. **Optional:** Change the synchronization schedule.

For more information, see [Crawl schedule options](#).

7. Complete the following fields in the **Enter your credentials** section:

Database URL

The URL of the database server.

The following table shows example database URLs:

Database	Syntax	Example
Data virtualization (same cluster)	<code>jdbc:db2://{{fully-qualified-hostname-of-dv-service}}:{jdbc-nonssl-internal-port}/bigsql</code>	<code>jdbc:db2://c-db2u-dv-db2u-engn-svc.myproject.svc.cluster.local:50000/bigsql</code>
Data virtualization (separate cluster)	<code>jdbc:db2://{{cluster-address}}:{jdbc-nonssl-external-port}/bigsql</code>	<code>jdbc:db2://api.conn.cp.example.com:30269/bigsql</code>
Db2	<code>jdbc:db2://{{server}}:{port}/{database_name}</code>	<code>jdbc:db2://localhost:50000/sample</code>
Oracle	<code>jdbc:oracle:thin:@//{{host}}:{TCPport}/{service_name}</code>	<code>jdbc:oracle:thin:@localhost:1521/sample</code>
SQL Server	<code>jdbc:sqlserver://{{serverName}}[{{instanceName}}]:{{port}}[;property=value]</code>	<code>jdbc:sqlserver://localhost:1433;DatabaseName=sample</code>
Postgresql	<code>jdbc:postgresql://{{host}}:{port}/{database}</code>	<code>jdbc:postgresql://localhost/sample</code>

Example database URLs

User

The username that you obtain from the database you selected. You use this username to crawl the source. Your username is different from database to database.

Password

The password that is associated with your username. Your password is different from database to database.

8. Complete the following fields in the **Connection settings** section:

JDBC driver type

Choose the database.

Db2 is selected by default. If you want to crawl from a database type that is not listed, select **OTHER**. To crawl data that is managed by Data Virtualization on IBM Cloud Pak for Data, keep **Db2** selected.

JDBC driver classname

The JDBC driver class name that is associated with the database you selected. This field is autofilled, unless you select **OTHER**.

JDBC driver classpath

Upload a JDBC driver file, which can have a .jar or .zip file extension. Alternatively, you can reuse a .jar or .zip file that you uploaded previously.

9. Complete the following fields in the **Specify what you want to crawl** section, and then click **Add**:

Schema Name

The schema that you want to crawl.

Table Name

The table within a schema that you want to crawl.

Click the edit icon to specify more table crawl settings, including:

Primary key

The primary key of the target database table. If the primary key is not configured in the target database table, you must specify the key in this field. The JDBC database crawler appends this primary key value to the URL of each crawled row to keep its uniqueness. When the primary key is a composite key, concatenate the key names by using a comma, for example **key1, key2**. If unspecified, the project defaults to the primary key fields of the table. If the primary key is configured in the target database table, this key is automatically detected.

Row filter

Optional. Specify the **SQL WHERE** clause to designate which table rows to crawl. You must specify a Boolean expression that can be the condition of a **WHERE** clause in a **SELECT** statement. If there is an error in syntax or column names, the table is excluded from the crawl, and no documents are indexed.

Column with data to extract

Name of the column with data that you want to crawl. If you don't specify the column, a column with text or with a single large object is chosen to be crawled.

MIME type of data

Optional. The MIME type is detected if not specified.

The values that you specify in the table crawl settings dialog are not displayed with the schema and tables names, but the values are applied to the database connection.



Note: The **Column with data to extract** and **MIME type of data** fields were added with the 4.6.5 release.

10. If you want the crawler to extract text from images in documents, expand **More processing settings**, and set **Apply optical character recognition (OCR)** to **On**.



Note: When OCR is enabled and your documents contain images, processing takes longer. For more information, see [Optical character recognition](#).

11. Click **Finish**.

The collection is created quickly. It takes more time for the data to be processed as it is added to the collection.

If you want to check the progress, go to the Activity page. From the navigation pane, click **Manage collections**, and then click to open the collection.

Using Windows Authentication on Linux

The JDBC driver from Microsoft does not support Windows Authentication on Linux. If you want to use Microsoft Windows authentication to access your SQL Server on Linux, you can use a third-party JDBC driver called jTDS from [Sourceforge](#). Specify the following values during the configuration:

- Database URL: **jdbc:jtds:sqlserver://<host>:<port>;databaseName=<database>;domain=<domain>;useNTLMv2=true;**
- JDBC driver type: **OTHER**
- JDBC driver class name: **net.sourceforge.jtds.jdbc.Driver**

Applying enrichments to content from a database

If you use a database as your data source and want to apply enrichments to the nested fields that are indexed from the database, you must use a Content Mining project type.

If your goal is to create a search application by using a Document Retrieval project type, create a Content Mining project type first. From the Content Mining project, you can connect to the database and enrich the data. Then, you can reuse the enriched collection from a Document Retrieval project.

To enrich database content for use in a Document Retrieval project, complete the following steps:

1. Create a Content Mining project.

For more information, see [Creating a project](#).

2. Connect to a database data source.

For more information, see [Configuring a data source: Database](#).

3. Apply enrichments.

For more information, see the following topics:

- [Adding domain-specific resources](#)
- [Applying prebuilt enrichments](#)

4. Create a Document Retrieval project.

For more information, see [Creating a project](#).



Note: When you are prompted to choose a collection, choose **Reuse data from an existing collection**. If necessary, scroll to see this option.

5. Select the collection that you created and enriched by using the Content Mining project, and then click **Finish**.

FileNet P8

Crawl documents that are stored in FileNet P8.

IBM Cloud Pak for Data [IBM Cloud Pak for Data only](#)



Note: This information applies only to installed deployments.

What documents are crawled

- Only file types that are supported by Discovery are crawled; all others are ignored. For more information, see [Supported file types](#).
- Document-level security is supported. When this option is enabled, your users can crawl and query the same content that they can access when they are logged in to FileNet. Discovery does not support role-based security when you crawl FileNet P8.
For more information about document-level security, see [Supporting document-level security](#).
- Only files with file extensions that match the file extension filter rules that you specify are crawled. **Added with the 4.7.0 release**.
- When a source is recrawled, new documents are added, updated documents are modified to the current version, and deleted documents are deleted from the collection's index.
- All Discovery data source connectors are read-only. Regardless of the permissions that are granted to the crawl account, Discovery never writes, updates, or deletes any content in the original data source.

Data source requirements

In addition to the [data source requirements](#) for all installed deployments, your FileNet P8 data source must meet the following requirements:

- The data source can crawl FileNet P8 5.5.0 and the Content Engine Web Services (CEWS) of a FileNet server that is installed on IBM Cloud Pak for Automation.
- FileNet P8 5.5.0 and FileNet on Cloud Pak for Automation support the HTTP and HTTPS protocols.

Prerequisite steps

If you want to enable document-level security, you must take some steps to set it up. For more information, see [About document-level](#)

[security](#).

Connecting to a FileNet P8 data source

From your Discovery project, complete the following steps:

1. From the navigation pane, choose **Manage collections**.
2. Click **New collection**.
3. Click **FileNet P8**, and then click **Next**.
4. Name the collection.
5. If the language of the documents in FileNet is not English, select the appropriate language.

For a list of supported languages, see [Language support](#).

6. **Optional:** Change the synchronization schedule.

For more information, see [Crawl schedule options](#).

7. Complete the following fields in the **Enter your credentials** section:

Content Engine Web Service URL

The Content Engine web service URL of the IBM FileNet P8 server.

When you enter the URL, use the format: `<protocol>://<server>:<port>/wsi/FNCEWS40MTOM`. You can use the HTTP or HTTPS protocol. The `<server>` is the hostname of the server where the Content Platform Engine is deployed and the `<port>` is the HTTP port that the application server uses, or where the Content Platform Engine is deployed.

User

The username to use to crawl the FileNet P8 server. You can obtain your username from your FileNet administrator.

Password

The password that is associated with the user.

8. In the **Specify what you want to crawl** section, enter the display name of the object store that you want to use to create, search, retrieve, and store documents in the **ObjectStore Name** field.
9. In **Crawler Space Type**, select either **Folder** or **Class**.
10. Complete the following field:

Folder subpath or Subclass name

The subfolder path that you can specify under RootFolder that crawls all documents that belong to the specified folder or the custom subclass of the **Document** class that crawls all documents that belong to the specified class. Before you specify anything in this field, keep in mind the following items:

- You can specify multiple crawler spaces by using both the **Class** and **Folder** types and crawl the documents belonging to the folder name and class name.
- You cannot specify a class outside the object store that you defined.
- No support is available for specifying a class that is a subclass of a **Custom Object** and **Folder**.

11. After you enter one or more paths, click **Add**.
12. **Optional:** In the **Security** section, if you want to enable document-level security, set the **Enable Document Level Security** switch to **On**.

When set to **On**, your users can crawl the same content that they have access to in FileNet.

13. If you want to limit the types of files to add to the collection, you can list the file extensions for file types to either include or exclude.

For a list of supported file types, see [Supported file types](#).



Note: Support for this option was added with the 4.7.0 release.

14. If you want the crawler to extract text from images in documents, expand **More processing settings**, and set **Apply optical character recognition (OCR)** to **On**.



Note: When OCR is enabled and your documents contain images, processing takes longer. For more information, see [Optical character recognition](#).

15. Click **Finish**.

The collection is created quickly. It takes more time for the data to be processed as it is added to the collection.

If you want to check the progress, go to the Activity page. From the navigation pane, click **Manage collections**, and then click to open the collection.

LDAP directory

Crawl records in an external directory that supports the Lightweight Directory Access Protocol (LDAP).

IBM Cloud Pak for Data [IBM Cloud Pak for Data only](#)



Note: This information applies only to installed deployments.

As the directory data is added to your collection, Discovery interprets and stores key attributes of each record according to the configuration that you specify. Later, you can find relevant records by filtering on the attributes that are of interest to you. For example, you can capture department and location information, and then filter records by location later.

For more information about the Lightweight Directory Access Protocol, see [RFC 4511](#).

What documents are crawled

- Each LDAP record is crawled and added to the collection as one document.
- All Discovery data source connectors are read-only. Regardless of the permissions that are granted to the crawl account, Discovery never writes, updates, or deletes any content in the original data source.

Data source requirements

In addition to the [data source requirements](#) for all installed deployments, your Salesforce data source must meet the following requirements:

- The LDAP directory data source supports connections to the following types of directories:
 - IBM Security Directory Server
 - Microsoft Active Directory (On premises only)
 - Oracle Directory Server
- The LDAP directory data source collection does **not** support the following capabilities:
 - Document-level security
 - Mutual authentication. Verifying the server certificate is supported, but also verifying the client certificate is not.
 - Proxy server access to the data source

Prerequisite step

When you set up the collection, you must provide details such as the LDAP host name and port, for your directory server type. For more information about how to discover these values, see the documentation from the vendor:

- [IBM Security Directory Server](#)
- [Microsoft Active Directory](#)
- [Oracle Directory Server](#)

Connecting to an LDAP directory data source

From your Discovery project, complete the following steps:

1. From the navigation pane, choose **Manage collections**.
2. Click **New collection**.
3. Click **LDAP directory**, and then click **Next**.

4. Name the collection.
5. If the language of the documents in Salesforce is not English, select the appropriate language.

For a list of supported languages, see [Language support](#).

6. **Optional:** Change the synchronization schedule.

The crawler schedule options work as follows for LDAP directories:

Full crawling

Crawls all entries.

Crawling updates

Crawls all entries, then filters out any entries that were inserted, updated, or deleted since the last crawl.

Crawling new and modified content

Runs an LDAP query against the data source server to pick up any entries that were inserted or updated only.

For more information, see [Crawl schedule options](#).

7. Configure a secure connection to the directory.

Server type

Choose your server type from the following options:

- IBM Security Directory Server
- Microsoft Active Directory
- Oracle Directory Server

LDAP protocol

If you want to encrypt data and verify the server certificate over Transport Layer Security (TLS), choose **ldaps**.

LDAP host name

Specify the hostname of the directory server. For example: <ldap-hostname>.mydomain.com.

LDAP host port

By default, the LDAP port is **389** and the LDAP-S port is **636**.

LDAP binding username

If the directory server requires credentials, the username that is used to bind to the directory service.

In most cases, this username is a distinguished name (DN). The username is case-sensitive.

LDAP binding user password

The password that is associated with the username.

8. Specify the information that you want to index from the directory.

LDAP Base DN

The object where you want to start the crawl.

LDAP directories have a hierarchical tree structure of objects. The base search distinguished name specifies the subtree in which you want the crawl to be constrained.

DN is a **distinguished name** that is defined by a series of **relative distinguished names** separated by commas. Each relative distinguished name consists of an **attribute** name-and-value pair that represents an object in a directory.

For example, in Active Directory, attributes can include a common name (CN) such as **Jane Doe** and an organizational unit

(OU) such as **Research**. Most distinguished names include one or more domain component (DC) attributes, which define the namespace where the LDAP directory is hosted.

Here's an example of a distinguished name for Jane:

CN=Jane Doe,OU=Research,DC=IBM,DC=COM

LDAP user filter

A filter to apply to the search to use to find LDAP entries that you want to crawl.

If unspecified, a default value is applied that is considered the best filter for the server type that you selected. You can edit the predefined filter value.

- Expand the **Advanced configuration** section to list specific attributes to include or exclude from the search.

For example, you might need to know the country in which an employee works, so you want to include a **c** attribute that stores the ISO country code. Or maybe you never want to return an employee's serial number, so you exclude the **serialnumber** attribute.

- Specify the search scope. You can choose to crawl records that are one level from the search base DN or to crawl the entire subtree that is associated with the search base DN.
- If the LDAP directory data source has binary attributes, you can enable the **Allow binary attributes** option.

When enabled, the crawler creates a separate document for each binary attribute that is specified. The document also contains any other non-binary LDAP attribute values.

For more information about the binary option, see [RTF 4522](#).

In the **Binary attributes** field, specify the names of the binary attributes that you want to index.

9. If you want the crawler to extract text from images on the site, expand **More processing settings**, and set **Apply optical character recognition (OCR)** to **On**.



Note: When OCR is enabled and your documents contain images, processing takes longer. For more information, see [Optical character recognition](#).

10. Click **Finish**.

The collection is created quickly. It takes more time for the data to be processed as it is added to the collection.

If you want to check the progress, go to the Activity page. From the navigation pane, click **Manage collections**, and then click to open the collection.

Local File System

Crawl documents that are stored in a local file system.

IBM Cloud Pak for Data [IBM Cloud Pak for Data only](#)



Note: This information applies only to installed deployments.

What documents are crawled

- Only file types that are supported by Discovery in your file path are crawled; all others are ignored. For more information, see [Supported file types](#).
- Only files in the **/mnt** directory or one of its subdirectories can be accessed by the crawler.
- Only files with file extensions that match the file extension filter rules that you specify are crawled. **Added with the 4.7.0 release**.
- When a source is recrawled, new documents are added, updated documents are modified to the current version, and deleted documents are deleted from the collection's index.
- All Discovery data source connectors are read-only. Regardless of the permissions that are granted to the crawl account, Discovery never writes, updates, or deletes any content in the original data source.

Prerequisite steps

Before you connect to the Local File System data source, complete the following step:

- [Create a persistent volume claim on the crawler pod](#)

The service uses Portworx storage by default. However, if you are using Network File System (NFS) storage, see [Prerequisite steps for NFS storage](#) instead.

Creating and mounting a persistent volume claim on the crawler pod

Before you can crawl a local file system, you must create a persistent volume claim and mount it on the **crawler** pod. You also need to copy the files that you want to crawl to the Discovery cluster that you are working on. If you have multiple Discovery clusters, you must copy the files along with the **crawler-pvc-portworx.yaml** file that you will create in this task to each cluster.

Complete the following steps:

1. Enter the following command to check the **storageclass** name of the Portworx provisioner:

```
$ oc get storageclass | grep portworx-gp3-sc
```

You might see output similar to the following:

NAME	PROVISIONER	RECLAIMPOLICY	VOLUMEBINDINGMODE	ALLOWVOLUMEEXPANSION	AGE
portworx-gp3-sc	kubernetes.io/portworx-volume	Retain	Immediate	true	51d

2. Create a file named **crawler-pvc-portworx.yaml** to define the persistent volume claim (PVC) with the following content:

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: <name-of-portworx-pvc>
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
  storageClassName: portworx-gp3-sc
```

Replace **<name-of-portworx-pvc>** with the name of your dynamic Portworx persistent volume claim. For example, **jdoe-pvc-portworx**

3. Enter the following command to create the persistent volume claim:

```
$ oc create -f crawler-pvc-portworx.yaml
```

A message is displayed:

```
persistentvolumeclaim/jdoe-pvc-portworx created
```

4. Enter the following command to mount the persistent volume claim to the **crawler** pod:

```
$ oc patch wd wd --type=merge \
--patch='{"spec": {"ingestion": {"crawler": {"mount": {"enabled": true, "persistentVolumeClaimName": "<name-of-portworx-pvc>"}}}}}'
```

Replace **<name-of-portworx-pvc>** with the name of your dynamic Portworx persistent volume claim. For example, **jdoe-pvc-portworx**.

5. Enter the following command to copy the files that you want to crawl to your dynamic Portworx persistent volume claim.

You only need to run this command one time against one of the existing **crawler** pods. The persistent volume claim is shared among all **crawler** and **ingestion-api** pods. Replace the variables in the command with the appropriate information.

```
$ oc rsync <path-to-local-file-system-folder> <crawler-pod>:/mnt
```

You mounted the persistent volume claim (PVC) and copied the files that you want to crawl to the PVC.

Connecting to a local file system data source

From your Discovery project, complete the following steps:

1. From the navigation pane, choose **Manage collections**.
2. Click **New collection**.

3. Click **Local File System**, and then click **Next**.
4. Name the collection.
5. If the language of the documents that you want to crawl is not English, select the appropriate language.
For a list of supported languages, see [Language support](#).
6. **Optional:** Change the synchronization schedule.
For more information, see [Crawl schedule options](#).
7. In the **Specify what you want to crawl** section, enter the file path that you want to crawl in the **Path** field, and then click **Add**.
The file path is case-sensitive. Remember, only files in the `/mnt` directory or one of its subdirectories can be accessed by the crawler.
8. Optionally, add more file paths.
9. If you want to limit the types of files to add to the collection, you can list the file extensions for file types to either include or exclude.
For a list of supported file types, see [Supported file types](#).



Note: Support for this option was added with the 4.7.0 release.

10. If you want the crawler to extract text from images in documents, expand **More processing settings**, and set **Apply optical character recognition (OCR)** to **On**.



Note: When OCR is enabled and your documents contain images, processing takes longer. For more information, see [Optical character recognition](#).

11. Click **Finish**.

The collection is created quickly. It takes more time for the data to be processed as it is added to the collection.

If you want to check the progress, go to the Activity page. From the navigation pane, click **Manage collections**, and then click to open the collection.

Prerequisite steps for NFS storage

Choose one of the following methods to enable the `crawler` pod to access the file system:

- [Configure an external NFS server](#)
- [Configure dynamic provisioning with an NFS storage class](#)

Configuring an external NFS server

If the local file system files or folders that you want to crawl are stored in an external Network File System (NFS), you can use the external NFS server to create the persistent volume claim.

1. Create a file named `crawler-pv-nfs.yaml` with the following content:

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: <persistent-volume-name>
  labels:
    pv-name: <persistent-volume-name>
spec:
  capacity:
    storage: 10Gi
  accessModes:
    - ReadWriteMany
  persistentVolumeReclaimPolicy: Retain
  nfs:
    server: <NFS server hostname or IP address>
    path: <Path of NFS exported folder>
```

Replace references to `<persistent-volume-name>` with the name of your persistent volume. For example, `jdoe-nfs-pv` and add the missing external NFS details.

2. Enter the following command to create the persistent volume claim:

```
$ oc create -f crawler-pv-nfs.yaml
```

The following message is displayed:

```
persistentvolume/jdoe-nfs-pv created
```

3. Create a file called `crawler-pvc-nfs.yaml` with the following content:

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: <persistent-volume-claim-name>
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
  selector:
    matchLabels:
      pv-name: <persistent-volume-name>
```

Replace the following variables:

- `<persistent-volume-claim-name>`: Specify the name of your persistent volume claim. For example, `jdoe-nfs-pvc`.
- `<persistent-volume-name>`: Specify the name of your persistent volume. For example, `jdoe-nfs-pv`.

4. Enter the following command to create the persistent volume claim:

```
$ oc create -f crawler-pvc-nfs.yaml
```

The following message is displayed:

```
persistentvolumeclaim/jdoe-nfs-pvc created
```

5. Enter the following command to mount the persistent volume claim to the `crawler` pod.

This command also mounts the persistent volume claim to all `ingestion-api` pods. Replace `<persistent-volume-claim-name>` with the name of your persistent volume claim. For example, `jdoe-nfs-pvc`.

```
$ oc patch wd wd --type=merge \
--patch='{"spec": {"ingestion": {"crawler": {"mount": {"enabled": true, "persistentVolumeClaimName": "<persistent-volume-claim-name>" } } } }'
```

Configuring dynamic provisioning with an NFS storage class

If you want to crawl your local file system files or folders but you do not want to prepare an extra NFS server to store those files or folders, you can configure dynamic storage by using an NFS storage class.

For more information about storage providers that Discovery supports and for storage comparisons, see [Storage considerations](#).

Before you complete this task, copy the files that you want to crawl to the Discovery cluster that you are working on. If you have multiple Discovery clusters, you must copy the files along with the `crawler-pvc-dynamic.yaml` file that you create in this task to each cluster.

Complete the following steps:

1. Enter the following command to check the `storageclass` name of the NFS provisioner:

```
$ oc get storageclass
```

A message is displayed.

NAME	PROVISIONER	RECLAIMPOLICY	VOLUMEBINDINGMODE
allowvolumeexpansion	age		

2. Create a file that is named `crawler-pvc-dynamic.yaml` and add the following content to it:

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: <name-of-dynamic-pvc>
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
  storageClassName: nfs-client
```

Replace `<name-of-dynamic-pvc>` with the name of your dynamic NFS persistent volume claim. For example, `jdoe-dynamic-pvc`.

3. Enter the following command to create the persistent volume claim:

```
$ oc create -f crawler-pvc-dynamic.yaml
```

A message is displayed.

```
persistentvolumeclaim/jdoe-dynamic-pvc created
```

4. Enter the following command to mount the persistent volume claim to the `crawler` pod.

This command also mounts the persistent volume claim to all `ingestion-api` pods.

```
$ oc patch wd wd --type=merge \
--patch='{"spec": {"ingestion": {"crawler": {"mount": {"enabled": true, "persistentVolumeClaimName": "<name-of-dynamic-pvc>"}}}}}'
```

Replace `<name-of-dynamic-pvc>` with the name of your dynamic NFS persistent volume claim in the previous step. For example, `jdoe-dynamic-pvc`.

5. Enter the following command to copy the files that you want to crawl to your dynamic NFS persistent volume claim.

You must run this command only one time against one of the existing `crawler` pods. The persistent volume claim is shared among all `crawler` and `ingestion-api` pods. Replace the variables in the command with the appropriate information.

```
$ oc rsync <path-to-local-file-system-folder> <crawler-pod>:/mnt
```

You mounted the persistent volume claim (PVC) and copied all of the files that you want to crawl to the PVC.

HCL Notes

Crawl an HCL Notes (formerly Lotus Notes) database.

IBM Cloud Pak for Data



Note: This information applies only to installed deployments.

What documents are crawled

- Each document in the HCL Notes database is crawled and added to the collection as a document.
- If an HCL Notes document has a file attachment, and you choose to process file attachments, only documents that are supported by Discovery are crawled; all others are ignored. For more information, see [Supported file types](#).
- If you choose to process attachments, the crawler attempts to crawl and index files that are attached to HCL Notes documents. File types that are supported by Discovery are indexed. For more information, see [Supported file types](#).
- Document-level security is supported. When this option is enabled, your users can crawl and query the same content that they can access when they are logged in to HCL Notes. For more information, see [Supporting document-level security](#).
- When a source is recrawled, new documents are added, updated documents are modified to the current version, and deleted documents are deleted from the collection's index.
- All Discovery data source connectors are read-only. Regardless of the permissions that are granted to the crawl account, Discovery never writes, updates, or deletes any content in the original data source.

Data source requirements

In addition to the [data source requirements](#) for all installed deployments, your HCL Notes data source must meet the following requirements:

- The data source can crawl HCL Notes 9.0.1 databases.
- The HCL Notes data source supports the Domino Internet Inter-ORB Protocol (DIIOP) protocol only.
- To crawl documents, including ACLs, you must have at least **Reader** level access to server, database, and document access on the Domino server.
- For group extractions from the internal Domino LDAP directory, you must have **Reader** access to the **names.nsf** directory database.
- For group extractions from the external LDAP directory, you must have the credential for the external LDAP server.

Prerequisite steps

- If you want to enable document-level security, you must take some steps to set it up. For more information, see [Supporting document-level security](#).

You can use the LDAP server that is used by HCL Notes (either the internal Domino LDAP or an external LDAP directory) as a remote LDAP directory to manage document-level security. Users who search the collection can be listed in an external LDAP directory. However, the user credentials that you use to set up the crawl must belong to a user who is listed in the internal Domino LDAP directory.

To configure document-level security, you need to collect the following information:

LDAP server URL

The LDAP server URL to connect to. For example, `ldap://<ldap_server>:<port>`.

LDAP binding username

The username to use to bind to the directory service. This user must have administrative access and be listed in the internal Domino LDAP directory.

LDAP binding user password

The password that is associated with the user.

LDAP base DN

The starting point for searching user entries in LDAP. For example, `CN=Users,DC=example,DC=com`.

LDAP user filter

The filter to apply to searches for user entries in LDAP. If unspecified, the default value is `(userPrincipalName=\{0\})`.

LDAP group filter

The filter to apply to searches for group entries in LDAP.

- Before you can crawl servers by using the Domino Internet Inter-ORB Protocol (DIIOP) protocol, you must configure the HCL Notes server to use the protocol. The server that you want to crawl must be running the DIIOP and HTTP tasks.

To configure the HCL Notes server to use DIIOP, complete the following steps:

1. Configure the HCL Notes server document.

- In HCL Notes, open the **server** document on the HCL Notes server that you want to crawl. This document is stored in the Domino directory.
- On the Configuration page, expand the **server** section.
- On the Security page in the Programmability Restrictions section, specify the appropriate security restrictions for your environment in the following three fields:
 - **Run restricted Lotus Script/Java agents**
 - **Run restricted Java/Javascript/COM**
 - **Run unrestricted Java/Javascript/COM**

For example, you might specify an asterisk (`*`) to allow unrestricted access by LotusScript/Java agents and specify usernames that are registered in the Domino directory for the Java/JavaScript/COM restrictions.

Important: To crawl a server that uses the DIIOP protocol, your configured crawler must be able to access the usernames that you specify in these fields.

- Open the Internet Protocol page, and then open the HTTP page. Set the **Allow HTTP clients to browse database** option to **Yes**.

2. Configure the user document.

- Open the **user** document for the user whose credentials that you want to use for LDAP binding. This document is stored in the Domino directory.

- On the Basics page in the **Internet password** field, specify a password.

You specify this user and password information when you set up the data source.

3. Restart the DIIOP task on the HCL Notes server.

For more information, see [Running server tasks](#) in the HCL Notes documentation.

Connecting to an HCL Notes data source

From your Discovery project, complete the following steps:

1. From the navigation pane, choose **Manage collections**.
2. Click **New collection**.
3. Click **Notes**, and then click **Next**.
4. Name the collection.
5. If the language of the documents in HCL Notes is not English, select the appropriate language.

For a list of supported languages, see [Language support](#).

6. **Optional:** Change the synchronization schedule.

For more information, see [Crawl schedule options](#).

7. In the **Enter your credentials** section, add values to the following fields:

Host name

The hostname of the HCL Notes server.

User name

The username to use to crawl the HCL Notes server.

Password

The password that is associated with the user.

8. In the **Crawl type**, choose what you want to crawl from the following options:

- If you want to crawl a specific HCL Notes database, choose **Database**, and then add the file name of the database to the **Database file name** field.
- If you want to crawl multiple databases, choose **Directory**. Specify the directory in which the databases that you want to crawl are stored in the **Directory name** field.

9. **Optional:** In the **Security** section, specify whether you want to enable document-level security.

- If you want to enable document-level security, set the **Enable Document Level Security** switch to **On**.

When set to **On**, your users can crawl the same content that they have access to in a HCL Notes database or directory.

- To use the Domino LDAP directory, set the **Use remote LDAP directory** switch to **On**. Provide details about the Domino LDAP directory. You collected this information when you performed the prerequisite step.

LDAP server URL

The LDAP server URL to connect to. For example, `ldap://<ldap_server>:<port>`.

LDAP binding username

The username to use to bind to the directory service.

LDAP binding user password

The password that is associated with the user.

LDAP base DN

The starting point for searching user entries in LDAP. For example, `CN=Users,DC=example,DC=com`.

LDAP user filter

The filter to apply to searches for user entries in LDAP. If unspecified, the default value is `(userPrincipalName=\{0\})`.

LDAP group filter

The filter to apply to searches for group entries in LDAP.

10. **Optional:** In the **Advanced options** section, make choices about the following configuration settings:

Crawl attachments

If you want to crawl files that are attached to HCL Notes documents, set the switcher to **On**.

Automatic code page detection

If you want the encoding converter to detect the code of pages to crawl, keep the switch set to **On**. If you set the switcher to **Off**, specify values for the following fields:

Code page to use

Specify the character encoding of the pages that you want to crawl. If unspecified, the default value of **UTF-8** is used.

Notes formula

Specify a HCL Notes formula to use to filter the data that you want to crawl. For example, `SELECT @IsAvailable(Year) & Year > 2003`.

For more information, see [Formula language](#) in the HCL Notes documentation.

11. Specify the date that you want to use when you filter the documents. The date is stored in a field that is named `__Date__` in HCL Notes documents. By default, the field stores the last modified date of the document. You can choose a different date to store in the field instead.

Document modification date

Uses the date that the document was last modified. This option is selected by default.

Document crawl date

Uses the last crawled date.

Document creation date

Uses the creation date of the document.

12. If you want the crawler to extract text from images in documents, expand **More processing settings**, and set **Apply optical character recognition (OCR)** to **On**.



Note: When OCR is enabled and your documents contain images, processing takes longer. For more information, see [Optical character recognition](#).

13. Click **Finish**.

The collection is created quickly. It takes more time for the data to be processed as it is added to the collection.

If you want to check the progress, go to the Activity page. From the navigation pane, click **Manage collections**, and then click to open the collection.

Salesforce

Crawl documents that are stored in a Salesforce data source.

 **Note:** This information applies only to managed deployments. For more information about connecting to Salesforce from an installed deployment, see [Salesforce](#).

What documents are crawled

During the initial crawl of the content, documents from all of the objects that can be accessed from the URL that you specify are crawled and added to your collection. Knowledge Articles are crawled only if their **version** is **published** and their languages is **en-us**.

During subsequent scheduled recrawls, only new and modified documents are crawled and any changes are reflected in your collection. Documents that are deleted from the external data source are not deleted from the collection.

All Discovery data source connectors are read-only. Regardless of the permissions that are granted to the crawl account, Discovery never writes, updates, or deletes any content in the original data source.

Discovery can crawl the following objects:

- Any default and custom objects that you have access to
- Accounts
- Contacts
- Cases
- Contracts
- Knowledge articles
- Attachments

Data source requirements

In addition to the [data source requirements](#) for all managed deployments, your Salesforce data source must meet the following requirements:

- The instance that you plan to connect to must be part of an Enterprise plan or higher.
- You must obtain any required service licenses for the data source that you want to connect to. For more information about licenses, contact the system administrator of the data source.

What you need before you begin

You must have the following information ready. If you don't know it, ask your Salesforce administrator to provide the information or consult the [Salesforce developer documentation](#).

Username

The **username** of an account that has access to the Salesforce site. For example, `jdoe@example.com`

Password

The password associated with the username. For example, `myP@ssw0rd`.

Service token

A valid Salesforce security token. For example, `mna08jsRET5CiJww9JnURLNN`.

URL

The URL of the Salesforce site that you want to crawl. For example, `https://my.salesforce.com`

Connecting to the data source

To configure the Salesforce data source, complete the following steps in Discovery:

1. From the navigation pane, choose **Manage collections**.
2. Click **New collection**.
3. Click **Salesforce**, and then click **Next**.
4. Add values to the following fields:
 - Username
 - Password plus service token

To form the password, concatenate the Password and Service token values that you noted earlier. For example, `myP@ssw0rdmna08jsRET5CiJww9JnURLNN`. The password and token values are never returned and are used only when credentials are created or modified.

- URL

Click **Next**.

5. Name the collection.
6. If the language of the documents on the site is not English, select the appropriate language.

For a list of supported languages, see [Language support](#).

7. **Optional:** Change the synchronization schedule.

For more information, see [Crawl schedule options](#).

8. Select the objects that you want to crawl.

The more objects that you select, the longer the processing of the documents takes.

9. If you want to limit the types of files to add to the collection, you can list the file extensions for file types to either include or exclude.



Important: When you choose to list extensions for file types to exclude, you must add at least one file extension.

For a list of supported file types, see [Supported file types](#).

10. If you want the crawler to extract text from images on the site, expand **More processing settings**, and set **Apply optical character recognition (OCR)** to **On**.



Note: When OCR is enabled and your documents contain images, processing takes longer. For more information, see [Optical character recognition](#).

11. Click **Finish**.

The collection is created quickly. It takes more time for the data to be processed as it is added to the collection.

If you want to check the progress, go to the Activity page. From the navigation pane, click **Manage collections**, and then click to open the collection.

Microsoft SharePoint Online

Crawl documents that are stored in a Microsoft SharePoint Online data source.

IBM Cloud **IBM Cloud only**



Note: This information applies only to managed deployments. For more information about connecting to SharePoint Online from an installed deployment, see [SharePoint Online](#).

What documents are crawled

During the initial crawl of the content, documents from all of the objects that can be accessed from the site collection path that you specify are crawled and added to your collection. You cannot limit the crawl to one library within a site collection, for example. All objects in the specified Site collection path are crawled. Custom metadata that is associated with the SharePoint content is crawled also. You can crawl one site collection path per collection. You cannot crawl **Personal SiteCollections**.

During subsequent scheduled recrawls, only new and modified documents are crawled and any changes are reflected in your collection. Documents that are deleted from the external data source are not deleted from the collection.

All Discovery data source connectors are read-only. Regardless of the permissions that are granted to the crawl account, Discovery never writes, updates, or deletes any content in the original data source.

Discovery can crawl the following objects:

- SiteCollections
- Sites
- SubSites
- Lists
- List Items

- Document Libraries
- List Item Attachments

Data source requirements

In addition to the [data source requirements](#) for all managed deployments, your SharePoint Online data source must meet the following requirements:

- The Site Collection that you connect to must be one that was created with an Enterprise plan. It cannot be a collection that was created with a frontline worker plan.
- You must have an Azure Active Directory user ID with permission to read all of the objects that you want to crawl. For example, `<admin_user>@.onmicrosoft.com`. The user ID does not need **SiteCollection Administrator** permission.

You can choose how to authenticate with the external Microsoft SharePoint account from the following options:

Open Authentication (OAuth v2)

Authenticates with the external data source by using a token so that your user credentials do not need to be shared. With this authentication method, you can log in to your Microsoft account directly to generate a token that is used by Discovery to connect to your data.

The **Sign in with Microsoft** option that uses Open Authentication v2 to authenticate with the external data source is a beta feature.

Before anyone can create connectors that use this authentication method, a user with the **Global Administrator** role must complete a one-time [prerequisite steps](#) to authorize the connection for all projects in the Discovery service instance.

Security Assertion Markup Language (SAML)

An older mechanism for authentication and authorization that requires user credentials to be shared with the Discovery service.

If you choose to use this authentication method, your Microsoft SharePoint account must meet the following requirements:

- Unless you created your SharePoint Online account before January 2020, two-factor authentication is enabled for the account by default. You must disable two-factor authentication.
To view and change your multifactor authentication status, see [View the status for a user](#) or [Change the status for a user](#).
- The crawl user account must have legacy authentication and **Contribute** level permissions enabled.
To enable legacy authentication, go to the [Azure portal](#) or contact your SharePoint administrator.
- The connector supports the **Password hash synchronization (PHS)** method for enabling hybrid identity only. Use any other type (such as Pass-through authentication or Federation) at your own risk.
- You must know the following information:

Username

The username of the user account to use to connect to the SharePoint Online SiteCollection that you want to crawl.

For example, `<janedoe>@exampledomain.onmicrosoft.com`.

Password

The password to connect to the SharePoint Online SiteCollection that you want to crawl.

This value is never returned and is only used when credentials are created or modified.

What you need before you begin

You must have the following information ready. If you don't know it, ask your SharePoint administrator to provide the information or consult the [Microsoft SharePoint developer documentation](#):

Organization URL

The root URL of the source that you want to crawl. Specify the domain name of the URL, for example `https://<company>`.

<domain>.com

Site collection path

The **site_collection_path** to the section of the site where you want to start the crawl.

For example, if the content that you want to crawl is available from <https://<company>.<domain>.com/sites/test>, then you can specify <https://<company>.<domain>.com> as the Organization URL and `/sites/test` as the Site collection path.

- You cannot specify folder paths as input.
- You cannot specify a path to an Active Server Page Extended (ASPx) file, such as URLs to document libraries, lists, and subsites.
- If you don't specify a path, the default value of `/` is used, and the root site collection is crawled.

- **Application ID:** ID of the data source that you want to crawl. This information is required only if you want to store ACL information that is associated with the source documents.

One-time prerequisite step for OAuth

Before anyone can configure the connector to use OAuth v2 authentication method, a user with the **Global Administrator** role in Microsoft Azure Directory where the data source is located must complete steps to register the Discovery enterprise application in Microsoft Azure. This step must be completed once per Discovery service instance.

The administrator does not need to create the application in Azure. When they choose SharePoint Online as the data source, the Discovery service generates the app automatically. As described in the procedure to follow, during the set up of the connector, the administrator must log in to Microsoft with credentials for a user with the **Global Administrator** role in Microsoft Azure Directory and allow the enterprise application to be registered.

The following steps must be completed by a global administrator one time only per service instance:

1. Review the default user access settings that will be applied to the enterprise application in Microsoft Azure.

Enterprise applications can handle user access in many ways. Check the default settings to ensure that they are appropriate for your deployment by completing the following steps:

1. Log in to [Microsoft Azure](#).
2. From the *Enterprise applications* page in *Azure Active Directory*, click *Consent and permissions*.

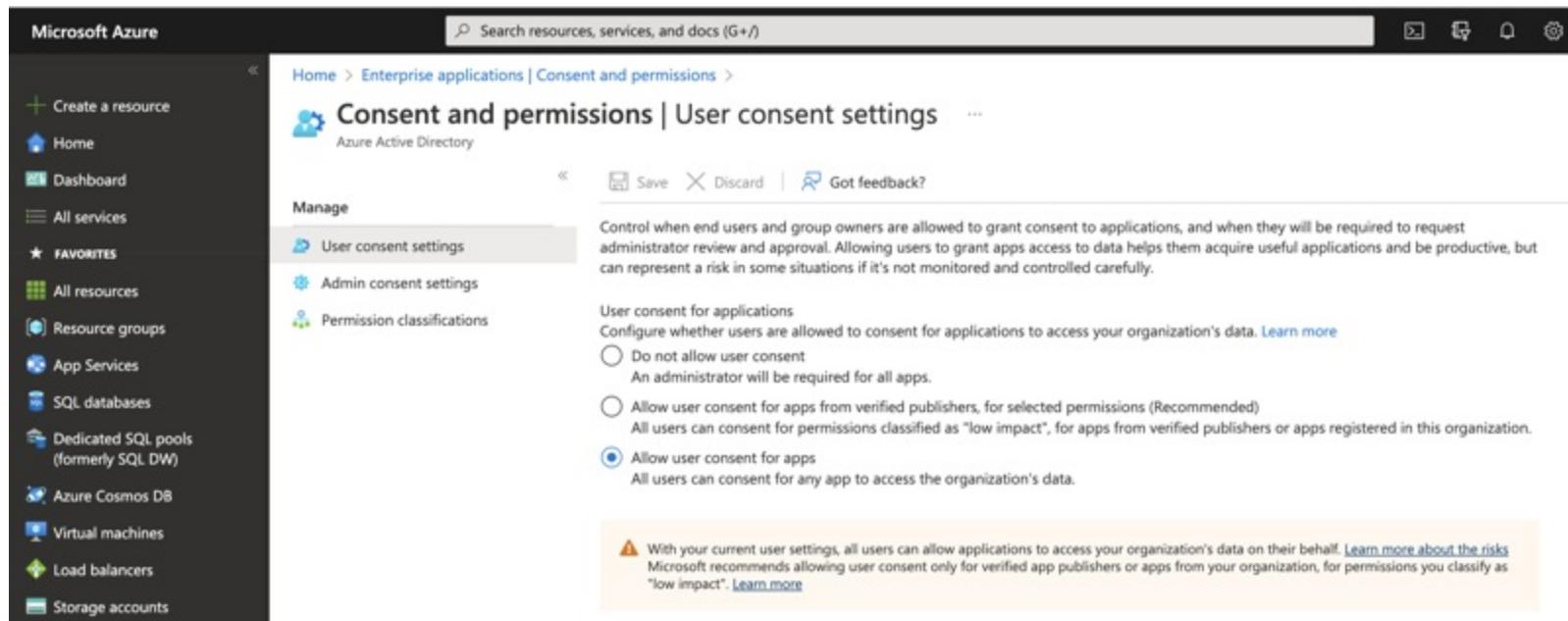


Figure 1. Microsoft Azure Enterprise application permissions user interface

1. Do one of the following things:

- If *Allow user consent for apps* is selected, no more action is needed.
- If *Allow user consent for apps from verified publishers, for selected permissions* is selected, then complete the following steps:

Click *Permissions classifications* link, and then ensure that the following permissions are configured at a minimum:

- Office 365 SharePoint Online: MyFiles.Read
- Office 365 SharePoint Online: AllSites.Read
- Microsoft Graph: offline_access
- Microsoft Graph: profile

The *Do not allow user consent option* is not supported.

The settings that you specify will be applied to the enterprise application that is created by Discovery in subsequent steps.

2. From the navigation pane of Discovery, choose **Manage collections**.
3. Click **New collection**.
4. Click **SharePoint Online**, and then click **Next**.
5. Add a URL to the **Organization URL** field.
6. Click **Sign in with Microsoft**.

Pop-ups must be enabled for this site in your web browser.

The **Sign in with Microsoft** option that uses Open Authentication to authenticate with the external data source is a beta feature.

Log in to your Microsoft SharePoint account with your user name and password, and then complete two-factor authentication, if necessary.

Important: Remember, the credentials you use must have the **Global Administrator** role in Microsoft Azure Directory. If you are not prompted for a user name and password, take note. You might be logged in to a Microsoft Sharepoint account already. If you are logged in to an account that you don't want to use for this connector, stop here. (Any account where you are logged in will be used automatically. And you cannot change the account configuration later.) Open a web browser in incognito mode and start this procedure over from step 1.

Discovery generates an enterprise application that it will register with the SharePoint organization that you specify. The enterprise application name has the format **IBM App Connect_{unique name}**.

7. Review the permissions that are associated with the enterprise application that Discovery will register, and then select **Consent on behalf of your organization**.

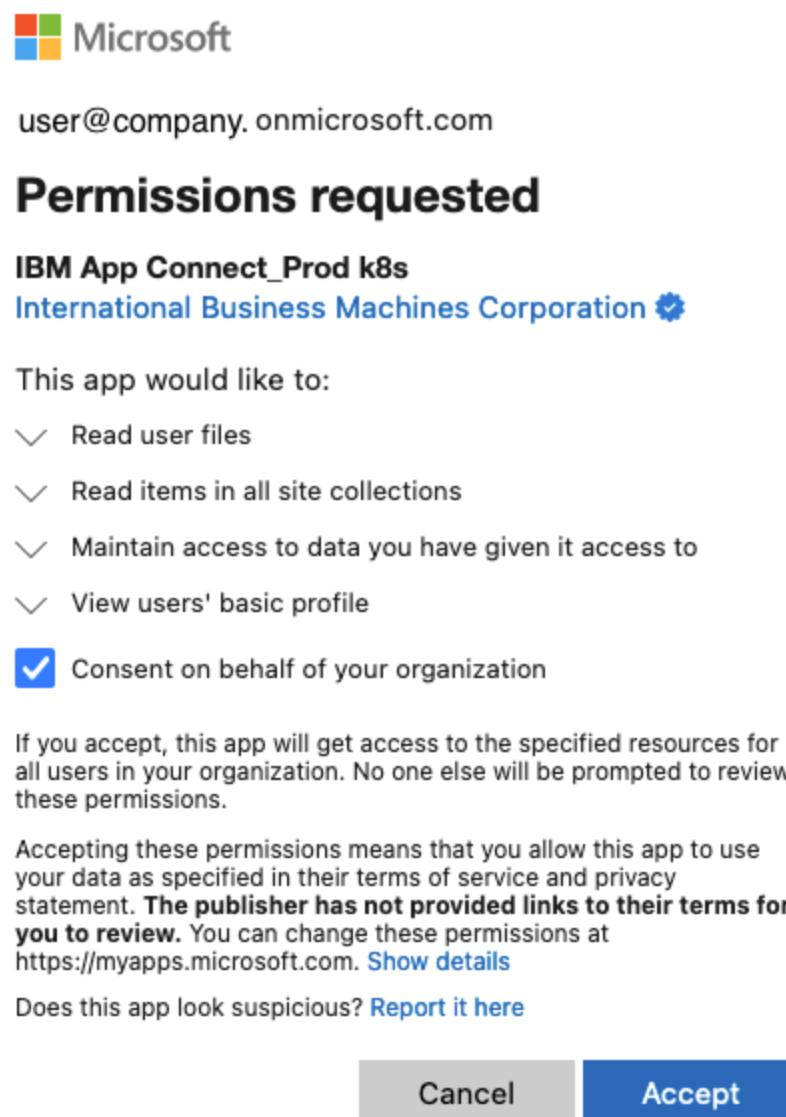


Figure 2. Discovery permission request dialog

8. Click **Accept**.
9. If you want to create a collection, you can name the collection, and then click **Finish**.

Otherwise, you can click **Back** to exit the collection creation process.

Now, anyone from your organization who works in a project that is hosted by the same Discovery service instance can create a collection by using the SharePoint Online connector.

OAuth support revisions

Support for the OAuth method of authentication was added with a software update in February 2022. If you want to update an existing

connector to use OAuth instead of SAML, you must re-create the connector. You cannot change the authentication mechanism for an existing connector.

The OAuth method of authentication was updated in January 2023. The enterprise application that is registered with Microsoft Azure now requires **Read** access only. Previously, the enterprise application required **Write** access. If you want to take advantage of this change, delete your current enterprise application and recreate the connector. For more information about how to delete an enterprise application, see [the Microsoft documentation](#).

Connecting to the data source

To configure the Microsoft SharePoint Online data source, complete the following steps in Discovery:

1. From the navigation pane, choose **Manage collections**.
2. Click **New collection**.
3. Click **SharePoint Online**, and then click **Next**.
4. Add a URL to the **Organization URL** field.
5. To enable access to your external data source, choose the method that you want to use to authenticate with the data source from the following options:

Open Authentication (OAuth v2)

Click **Sign in with Microsoft**.

Pop-ups must be enabled for this site in your web browser.

The **Sign in with Microsoft** option that uses Open Authentication to authenticate with the external data source is a beta feature.

Log in to your Microsoft SharePoint account with your user name and password, and then complete two-factor authentication, if necessary.

Security Assertion Markup Language (SAML)

Specify a username and password for a user that is authorized to access the site you want to crawl, and then click **Next**.

6. Specify the path you want to crawl in the **Site collection path** field.
7. Name the collection.
8. If the language of the documents on the site is not English, select the appropriate language.

For a list of supported languages, see [Language support](#).

9. **Optional:** Change the synchronization schedule.

For more information, see [Crawl schedule options](#).

10. **Optional:** If you want to store any access control information that exists in the SharePoint documents that you crawl, in the **Security** section, set the **Include Access Control List** switch to **On**.

When you enable this option, information about SharePoint access rules that is stored in SharePoint source documents is retained and stored as metadata in the documents that are added to your collection.

This feature is not the same as enabling document-level security for the collection. The access rules in the document metadata are not used by Discovery search. Enabling this feature merely stores the information so that you can leverage the access rules when you build a custom search solution.



Important: Use of this feature increases the size of the documents that are generated in the collection and increases the crawl time. Only enable the feature if your use case requires that you store the SharePoint document ACL information.

If you enable this feature, someone with the administrator role in Microsoft SharePoint must take extra steps to ensure that users who crawl the site have the right permissions to access ACL metadata.

An administrator must complete the following steps:

1. Log in to Microsoft SharePoint.
2. Open the page for your SharePoint site.

3. From the settings menu, choose ***Site permissions***.
4. Click ***Advanced permission settings***.
5. Make sure that people who want to collect access control information during a crawl have or are members of a group that has the ***Full Control*** permission for the site.

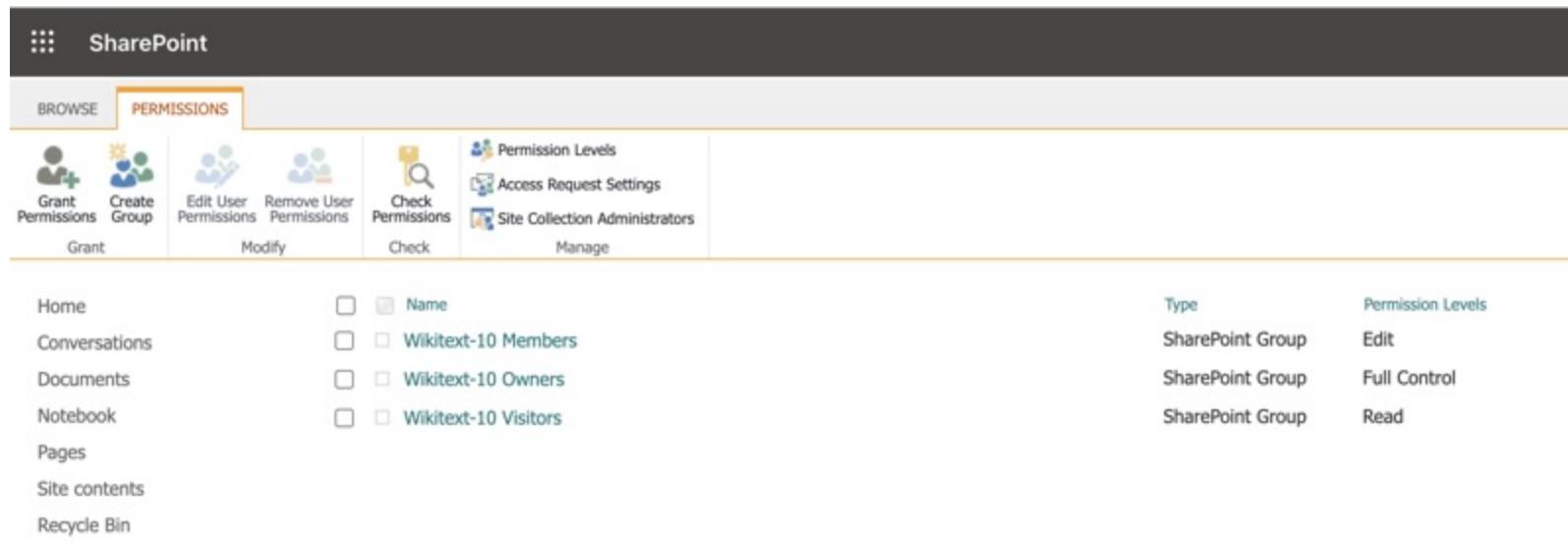


Figure 3. Microsoft SharePoint permissions user interface

Note: When access control list information is not extracted, ***Read*** permission is sufficient for all users who crawl the content.

11. If you want to limit the types of files to add to the collection, you can list the file extensions for file types to either include or exclude.

Important: When you choose to list extensions for file types to exclude, you must add at least one file extension.

For a list of supported file types, see [Supported file types](#).

12. If you want the crawler to extract text from images on the site, expand ***More processing settings***, and set ***Apply optical character recognition (OCR)*** to ***On***.

Note: When OCR is enabled and your documents contain images, processing takes longer. For more information, see [Optical character recognition](#).

13. Click ***Finish***.

The collection is created quickly. It takes more time for the data to be processed as it is added to the collection.

If you want to check the progress, go to the Activity page. From the navigation pane, click ***Manage collections***, and then click to open the collection.

Note: You cannot currently change the user account that is associated with the OAuth setup later, nor any of the details of the existing user account that the connector is configured to use. For example, you cannot update the password that was used to set up the connection after a password change in SharePoint.

Sample access control list information

The following screen capture illustrates the type of ACL information that is stored in the document when you include the access control list.

```

"document_id": "sharepoint_filecollection_c088dd58-5a12-476a-847d-38030f1211eb",
  "result_metadata": {
    "collection_id": "0e36fdd2-7fb0-812b-0000-017edabfa1ab"
  },
  "enriched_text": [
    {...}
  ],
  "metadata": {
    "parent_document_id": "sharepoint_filecollection_c088dd58-5a12-476a-847d-38030f1211eb",
    "source": {
      "LinkingUrl": "",
      "Modified": "2020-07-07T03:18:14Z",
      "TimeLastModified": "2020-07-07T03:18:13Z",
      "ContentTypeId": "0x010100036B86C6B029AA42831269188B39583E",
      "acl": [
        "c:0o.c|federateddirectoryclaimprovider|",
        "...",
        "i:0#.f|membership|",
        "SHAREPOINT\\system",
        "c:0t.c|tenant|",
        "c:0o.c|federateddirectoryclaimprovider|",
        "...",
        "i:0#.f|membership|",
        "i:0#.f|membership|"
      ],
      ...
    }
  }
}

```

Figure 4. Representation of ACL information in document metadata

Microsoft SharePoint On Prem

Crawl documents that are stored in a Microsoft SharePoint data source that is hosted on premises.

IBM Cloud **IBM Cloud only**



Note: This information applies only to managed deployments. For more information about connecting to an on-premises SharePoint data source from an installed deployment, see [SharePoint On Prem](#).

What documents are crawled

During the initial crawl of the content, documents from all of the objects that can be accessed from the site collection path that you specify are crawled and added to your collection. Custom metadata that is associated with the SharePoint content is crawled also. You can crawl one site collection path per collection. You cannot crawl **Personal SiteCollections**.

During subsequent scheduled recrawls, only new and modified documents are crawled and any changes are reflected in your collection. Documents that are deleted from the external data source are not deleted from the collection.

All Discovery data source connectors are read-only. Regardless of the permissions that are granted to the crawl account, Discovery never writes, updates, or deletes any content in the original data source.

The following table illustrates the objects that Discovery can crawl.

Data source	Objects that are crawled
Microsoft SharePoint On Prem	SiteCollections, Sites, SubSites, Lists, List Items, Document Libraries, List Item Attachments

Table 1. Data sources crawling support

Data source requirements

In addition to the [data source requirements](#) for all managed deployments, your SharePoint On Prem data source must meet the following requirements:

- You can connect to a SharePoint 2013, 2016, or 2019 on-premises data source.
- The user ID must have **SiteCollection Administrator** permission and be able to access all of the sites and lists that they want to crawl.
- The crawler supports Windows New Technology LAN Manager (NTLM) v1 authentication only. It does not support NTLM v2 or Security Assertion Markup Language (SAML) authentication.

What you need before you begin

You must have the following information ready. If you don't know it, ask your SharePoint administrator to provide the information or consult the [Microsoft SharePoint developer documentation](#):

Username

The username to use to connect to the SharePoint On Prem web application that you want to crawl. For example, `siteadmin01`.

Password

The password to connect to the SharePoint On Prem web application that you want to crawl. This value is never returned and is only used when credentials are created or modified.

Web Application URL

The SharePoint web application URL. For example, `https://sharepointwebapp.com:8443`. If you do not enter a port number, the default value of `80` is used for an HTTP URL and `443` for HTTPS.

Domain

The domain name of the SharePoint On Prem account. For example, `sharepoint.mycointernal`.

Prerequisite step

Before you can connect to a SharePoint On Prem data source, you must install and configure IBM® Secure Gateway for IBM Cloud®.

For more information, see [Installing IBM Secure Gateway for on-premises data](#).

Connecting to the data source

To configure the Microsoft SharePoint On Prem data source, complete the following steps in Discovery:

1. From the navigation pane, choose **Manage collections**.
2. Click **New collection**.
3. Click **SharePoint On Prem**, and then click **Next**.
4. Add values to the following fields:
 - Username
 - Password
 - Web Application URL
 - Domain

Click **Next**.

5. Name the collection.
6. If the language of the documents on the site is not English, select the appropriate language.

For a list of supported languages, see [Language support](#).

7. **Optional:** Change the synchronization schedule.

For more information, see [Crawl schedule options](#).

8. If you want to limit the types of files to add to the collection, you can list the file extensions for file types to either include or exclude.



Important: When you choose to list extensions for file types to exclude, you must add at least one file extension.

For a list of supported file types, see [Supported file types](#).

9. If you want the crawler to extract text from images on the site, expand **More processing settings**, and set **Apply optical character recognition (OCR)** to **On**.



Note: When OCR is enabled and your documents contain images, processing takes longer. For more information, see [Optical character recognition](#).

10. Click **Finish**.

The collection is created quickly. It takes more time for the data to be processed as it is added to the collection.

If you want to check the progress, go to the Activity page. From the navigation pane, click **Manage collections**, and then click to open the collection.

Web crawl

Add a web crawl collection to crawl a website, analyze its page content, and store meaningful information. Specify one or more base web page URLs and configure how many linked pages for the web crawl to follow. You can configure how often to synchronize with the website, so you control how up to date the data in your collection is.

IBM Cloud **IBM Cloud only**



Note: This information applies only to managed deployments. For more information about connecting to a website from an installed deployment, see [Web crawl](#).

What documents are crawled

You can connect to the following types of web content:

- Public websites
- Private company websites or other sites that require authentication
- Websites that are behind a corporate firewall

During the initial crawl of the content, all website pages that match your search settings are crawled and added to the document index of your collection. The crawl starts on the web page that you specify in the **Starting URLs** field. If your collection is configured to follow links, the crawl follows links on the starting page that share the same subtree as the starting page. For example, if you specify `https://www.example.com/banking/faqs.html`, links with URLs that begin with `https://www.example.com/banking/` are crawled. If you specify `https://www.example.com/banking`, links with URLs that begin with `https://www.example.com/` are crawled.

The crawl cannot access secure subdirectories. For example, if a subdirectory that you expect the crawl to access, such as `https://www.example.com/banking/pdfs`, isn't being crawled, check whether you can access the subdirectory URL from a web browser directly. If you can't access it, the crawl can't access it.

During subsequent scheduled recrawls, a full recrawl is performed and any changes are reflected in your collection. Documents that were added to your collection from website pages that are later deleted from the external website are not deleted from the collection. However, starting with collections that were created after April 2022, when you remove a starting URL from the web crawl configuration, any associated documents are deleted. Deleted documents include indexed documents that were added to the collection based on the content of the web page at the starting URL and documents that were derived from web pages that the starting URL linked to. You cannot limit the number of indexed documents by changing other settings, such as changing the existing URL to include a path with a more limited scope than before or reducing the maximum number of links to follow to 0. Only by deleting the URL can you remove the indexed documents that are associated with it.

The web crawler can crawl web pages that use JavaScript to render content, but the crawler works best on individual pages, not entire websites. It cannot crawl sites that use dynamic URLs; if you can't see any content when you view the source code of a web page in your browser, then the service cannot crawl it.

If you want to crawl a group of URLs that includes some websites that require authentication and some that don't, consider creating a different collection for each authentication type. The connector does not support cookie-based crawling.

All Discovery data source connectors are read-only. Regardless of the permissions that are granted to the crawl account, Discovery never writes, updates, or deletes any content in the original data source.

The following table illustrates the objects that Discovery can crawl.

Objects that are crawled

Websites, website subdirectories

Table 1. Data sources crawling support

Prerequisite step

If you want to connect to a website that is hosted behind a firewall, set up an IBM® Secure Gateway for IBM Cloud® connection first.

Valuable content is often stored on your company's internal website. Typically, such intranet websites are accessible only from a computer that is connected to your office network or through a VPN connection. You can establish a persistent and more secure connection between the web crawler and this type of internal site by using Secure Gateway.

For more information about how to set up the connection, see [Installing IBM Secure Gateway for on-premises data](#).

Connecting to the data source

To configure the web crawl collection, complete the following steps:

1. From the navigation pane, choose **Manage collections**.

2. Click **New collection**.
3. Click **Web crawl**, and then click **Next**.
4. Name the collection.
5. If the language of the content on the website is not English, select the appropriate language.
For a list of supported languages, see [Language support](#).
6. **Optional:** You can change the synchronization schedule.
For more information, see [Crawl schedule options](#).

7. Specify the URL of the website that you want to crawl.

- If the site you want to crawl requires a login, set **Basic authentication** to **On**, add the URL of the page to the **Starting URL** field, and then click **Add**.

Add a username and password with access to the site, and then click **Save credentials**. You can specify only one set of credentials per collection.

For example, you can specify `https://cloud.ibm.com` as the starting URL and add your IBMid as the credentials.

If you want to start the crawl from a specific section of the site, specify it in the **Starting URLs** field. The domain name of the subsection must match the domain in the URL you specified earlier.

For example, you might change the starting URL to `https://cloud.ibm.com/unifiedsupport/supportcenter`.

- For any public web pages that you want to crawl, add the URL for the root page of the website to the **Starting URLs** field, and then click **Add**. You can add more than one starting page.

The final forward slash (`/`) in the URL determines the subtree to crawl. If you specify

`https://www.example.com/banking/faqs.html`, all URLs that begin with `https://www.example.com/banking/` are crawled, for example. If you specify `https://www.example.com/banking` all URLs that begin with `https://www.example.com/` are crawled.

By default, the number of consecutive links that the crawl follows from the starting URL is **2**. To change the number of hops or to list website sections to exclude from the crawl, click the edit icon.

- The maximum number of hops allowed is **20**.
- To specify URL paths to exclude, add the site path. For example, if the starting URL is `https://example.com`, you can exclude `https://example.com/pricing` by entering `/pricing/`.

Any section of the web address that contains the site path you specify is excluded. For example, if you specify `/licenses/`, the page `https://example.com/products/licenses/europe` is excluded, among others.

- If you want to restrict the crawl to a single page, add the URL to the **Starting URLs** field. For example, `https://www.example.com/banking/faqs.html`. Click the edit icon to set the **Maximum number of links to follow** to **0**.

- If the website that you want to crawl uses JavaScript to customize the page content before it is displayed, you must take an extra step.

After you enter the starting URL and click **Add**, edit the URL by clicking the edit icon  Set the **Execute JavaScript during crawl** switcher to **On**, and then click **Save**.

 **Note:** When JavaScript processing is enabled, it takes 3 to 4 times longer to crawl a page. Use it only on individual web pages where you know it is necessary because the page renders its content dynamically. If you see timeout messages or the crawl ends without adding content to the collection, decrease the number of web pages that are included in the crawl. For example, you can specify the exact page to crawl in the **Starting URLs** field, and set **Maximum number of links to follow** to 0.

- To connect to a website that is hosted behind a firewall, [set up an IBM® Secure Gateway for IBM Cloud® connection first](#).

Expand **More connection settings**, and then set **Connect to on-premises network** to **On**. Provide details about your Secure Gateway connection.

8. Optional: Add another web address to the **Starting URLs** field.



Important: The number of starting URLs for a single collection must be less than 100. If you have a requirement to crawl a large number of websites, see [I need to crawl lots of sites. What's my limit?](#).

The number of web pages that are crawled is limited to 250,000, so the web crawler might not crawl all the specified websites.

The number of child URLs per URL that are crawled is limited to 10,000. If the number of child URLs within any crawled URL exceeds 10,000, the crawler cannot process any of the content in the child URLs.

9. If you want to limit the types of files to add to the collection, you can list the file extensions for file types to either include or exclude.

 **Important:** If the URLs for your website pages do not end in `.html`, use the exclude filter instead of the include filter. You must add at least one file extension to exclude.

For a list of supported file types, see [Supported file types](#).

10. If you want the web crawl to extract text from images on the site, expand **More processing settings**, and set **Apply optical character recognition (OCR)** to **On**.

 **Note:** When OCR is enabled and your documents contain images, processing takes longer. For more information, see [Optical character recognition](#).

11. Click **Finish**.

The collection is created quickly. It takes more time for the data to be processed as it is added to the collection.

If you want to check the progress, go to the Activity page. From the navigation pane, click **Manage collections**, and then click to open the collection.

I need to crawl lots of sites. What's my limit?

The service can support a total of 500 crawler connections per Discovery service instance. All of the data sources except Web crawl use one crawler connection each. For Web crawl, one connection is required for every 5 starting URLs. If you add 10 starting URLs, for example, Discovery generates the extra crawler connection that is needed to support the extra 5 URLs. Therefore, the maximum number of starting URLs that you can use depends on the other data collections that are configured in your service instance. You can calculate the limit yourself.

To calculate the starting URL limit, complete the following steps:

1. Calculate the number of other data source collections in the service instance, meaning this project and any other projects in the same Discovery instance.

For example, you might have 2 IBM Cloud Object Store collections in one project and 2 Salesforce collections and 1 SharePoint Online collection in another project. In this example, the total number of other data source collections is 5.

2. Subtract the number of other data source collections from the maximum allowed number of crawler connections, which is 500.

For example, $500 - 5 = 495$.

3. Multiply the remainder by 5 to determine the total number of starting URLs that you can use.

For example, $495 \times 5 = 2,475$.

 **Note:** To use the maximum-allowed number of starting URLs in the example, you would need 25 web crawl collections because each collection allows a maximum of 100 starting URLs to be configured. However, don't configure your instance to use the absolute maximum number allowed. If one or more additional data sources are added subsequently to a project in this service instance, it will impact the number of starting URLs that the instance can crawl successfully.

Troubleshooting crawler issues

A 403 Forbidden error is returned

The website that you want to crawl might block requests from all but a specific set of named entities. If possible, add the crawler to the allowlist for the site. The identifying header for the crawler is **User-Agent: IBM-AppConnect/V1**.

Windows File System

Crawl documents that are stored in a Microsoft Windows file system.

IBM Cloud Pak for Data **IBM Cloud Pak for Data only**



Note: This information applies only to installed deployments.

What documents are crawled

- Only documents that are supported by Discovery in your file path are crawled; all others are ignored. For more information, see [Supported file types](#).
- Document-level security is supported. When this option is enabled, your users can crawl and query the same content that they can access when they access the file system directly.
- When a source is recrawled, new documents are added, updated documents are modified to the current version, and deleted documents are deleted from the collection's index.
- All Discovery data source connectors are read-only. Regardless of the permissions that are granted to the crawl account, Discovery never writes, updates, or deletes any content in the original data source.

Data source requirements

In addition to the [data source requirements](#) for all installed deployments, your Windows File System data source must meet the following requirements:

- The connector supports Microsoft Windows Server 2012 R2, 2016, 2019, and 2022.
- The remote agent server and the file servers to be crawled must belong to the same Windows domain. The crawler can gather access control list (ACL) data from a single Windows domain only.



Note: Support for Microsoft Windows Server 2022 was added with the 4.6 release. Starting with the 4.7 release, you can secure traffic that is sent between the Windows Agent service and its crawler by enabling support for the transport layer security (TLS) protocol.

Prerequisite steps

- If you want to enable document-level security, you must take some steps to set it up. For more information, see [Supporting document-level security](#).

To configure document-level security, you need to collect the following information:

LDAP server URL

The LDAP server URL to connect to. For example, `ldap://<ldap_server>:<port>`.

LDAP binding username

The username to use to bind to the directory service.

In most cases, this username is a distinguished name (DN). An Active Directory username might work, but, unlike the general Windows logon, it is case sensitive.

LDAP binding user password

The password that is associated with the binding username.

LDAP base DN

The starting point for searching user entries in LDAP. For example, `CN=Users,DC=example,DC=com`.

LDAP user filter

The user filter to search user entries in LDAP. If empty, the default value is `(userPrincipalName={0})`.

- Before you configure a Windows File System collection, you must install the IBM Watson Discovery Agent for Windows File Systems on a remote Windows file server or on a remote Windows server. The agent is a Windows service that retrieves data from data source servers and sends it to Discovery. The agent can crawl remote Windows file systems, drives that are local to the agent, and shared network folders.

If you install the agent on a remote Windows server, the remote Windows server must be able to mount one or more file servers so that the agent can crawl the remote Windows file systems.

To install and configure the agent, complete the following tasks:

- [Install the agent](#).

- [Configure shared directories on the agent server](#).
- [Start and monitor the status of the agent server](#).

Install the agent

With the 4.6 release, the IBM Watson Discovery Agent for Windows File Systems was updated to run with 64-bit versions of Windows. If you installed the agent with a release prior to 4.6, you must uninstall the previous version, delete it, and then reinstall the agent.

Do one of the following tasks:

- You have a previous installation that is earlier than 4.6: [Replace the pre-4.6 agent](#)
- You are using the connector for the first time: [Install the agent](#)

Replace the pre-4.6 agent

Required for deployments where a version of the IBM Watson Discovery Agent for Windows File Systems that is earlier than 4.6.0.0 is installed.

To replace an earlier version of the agent, complete the following steps:

1. Copy the configuration file that defines the shared network directories that the Windows File System agent can access to a directory that is outside the agent's file path, which is `C:\Program Files (x86)\IBM\es`.
For example, copy the `C:\Program Files (x86)\IBM\es\distributed\esadmin\config\esfsexport.txt` file to a directory such as `C:\temp` directory.
2. From the Microsoft Windows *Apps & features* utility, find the earlier version of *IBM Watson Discovery Agent for Windows File Systems*, and then click *Uninstall*.
3. Choose *Completely delete IBM Watson Discovery Agent for Windows File Systems*, and then click *Uninstall*.
4. Restart your system.
5. Complete the steps in [Installing the agent](#) to install the latest version of the agent.
6. Replace the new version of the `C:\Program Files\IBM\es\distributed\esadmin\config\esfsexport.txt` file with the file that you copied in Step 1.

This step adds the configuration of the shared directories that you set up for the previous version of the agent to the new installation. When you reuse the file share, you can skip the step of configuring the shared directories.

7. Run the following command to verify that the directory is shared with the agent service:

```
C:\Users\Administrator> esagent --lsshare
```

Installing the agent

To install the IBM Watson Discovery Agent for Windows File Systems for the first time, complete the following steps:

1. From the navigation pane, choose **Manage collections**.
2. Click **New collection**.
3. Click **Windows File System**, and then click **Next**.
4. Scroll to the **Download & install Windows Agent** section, and then click **Download Windows Agent Installer**.

A ZIP file is downloaded.

5. Decompress the `WindowsAgentServer.zip` file.
6. You can choose one of the following methods to run the installation program:
 - Double-click the `install.exe` file to launch the installation wizard.
 - To run the installation program in text mode from a console, complete the following steps:
 - Change to the agent directory.
 - Enter the following command:

```
$ install.exe -i console
```

The screens are rendered in text and prompt you for the same information as the graphical installation.

 **Note:** After you enter the command, a process runs in the background for several seconds before the console installation program is displayed.

- To install the agent server silently, complete the following steps:

- Change to the **Agent/responseFiles** directory.
- Edit the **DistributedFileSystemCrawler.properties** template response file to provide information about your environment. To run the installation program, change to the agent directory, and then specify the name of the file that you edited.

See the following example:

```
$ install.exe -i silent -f responseFiles/DistributedFileSystemCrawler.properties
```

If you copy a template file to another location to edit, specify the fully qualified path for the file when you run the installation program. If the response file path includes a space, enclose the path in double quotation marks (""). See the following example:

```
$ install.exe -i silent -f "c:\My Documents\DistributionFileSystemCrawler.properties"
```

7. You must provide the following information during the installation process:

- **hostname**: Enter or verify the fully-qualified hostname of the computer you are installing the agent server on.

 **Important:** You cannot specify an IPv6 address as the hostname of the server.

- **username**: Enter the username of an account that can be used to authorize access to the agent server.

If the username does not exist, select the checkbox to create the account.

 **Important:** To crawl a domain in a secure collection, the username must be an existing domain user with administration privileges for the Windows system to be crawled. To specify a domain user, use the format <username>@<domain name>.

- **password**: Provide the password that is associated with the username.

8. **Optional:** If you want to change the default path and port settings, click **Advanced Options**.

- You can change the paths for the installation directory and data directory.
- The agent server uses three TCP/IP ports for authenticating connections to the server, transferring data between the file systems and Discovery, and monitoring the agent server. The default port numbers are **8397** and **8398**. If those values conflict with other port assignments in your system, change the port numbers.

9. On the summary page, review the options that you selected, and click **Install** to start installing the software.

10. **Optional:** If you want to secure traffic between the Windows Agent service and the crawler, enable TLS support.

Copy the file named **tls.p12** from the decompressed directory to the root directory where the agent is installed. For example, the root directory might be **C:\Program Files\IBM\es\distributed\esadmin**.

 **Note:** TLS support is available starting with the 4.7 release.

11. Restart your computer.

Configuring shared directories on the agent server

After the software is installed, you must set up shared network directories that the Windows File System agent can access. To define a new file system share, export a local or remote network directory.

 **Important:** If you are replacing an agent that you installed with a release that is earlier than 4.6.0.0, skip this procedure. The replacement instructions explain how to reuse the file share that was defined previously.

1. Export a local directory from the server where the agent is installed:

```
$ esagent --addshare <d:><\example>
```

Where **d:** represents the drive letter you want to use and where **\example** represents the path to the local directory.

2. Export a remote network directory that is accessible from the server where the agent is installed:

```
$ esagent --addshare <\\files.example.com\data>
```

Where **\\files.example.com\data** represents the hostname or IP address of the remote server or the path to the remote directory.

3. List shares that are defined on the server where the agent is installed:

```
$ esagent --lsshare
```

4. If you want to delete a share that is defined on the server where the agent is installed, you can use the following command:

```
$ esagent --rmshare \\files.example.com\data
```

Server status commands

After you install the agent server, you can enter commands to start, stop, and check the status of the server.

Stopping the agent server also stops the crawler. For example, if the crawler stops unexpectedly, you can close connections and release resources for that crawler.

- To start the server, enter the following command:

```
$ esagent start
```

- To stop the server, enter the following command:

```
$ esagent stop
```

- To get the status of the agent server, enter the following command:

```
$ esagent getStatus
```

The output of the **getStatus** command is an XML file with the following output:

```
<AgentStatus>
  <SpaceStatus>
    <SpaceId>012</SpaceId>
    <RootFolder>E:\\Projects\\Analytics\\data\\test1</RootFolder>
    <ConnectionNumber>9</ConnectionNumber>
    <StartTime>1244709336093</StartTime>
    <LastTime>1244709385843</LastTime>
    <IdlePeriod>219</IdlePeriod>
  </SpaceStatus>
  <SpaceStatus>
    <SpaceId>013</SpaceId>
    <RootFolder>E:\\Projects\\Analytics\\data\\test2</RootFolder>
    <ConnectionNumber>10</ConnectionNumber>
    <StartTime>1244709336093</StartTime>
    <LastTime>1244709385843</LastTime>
    <IdlePeriod>219</IdlePeriod>
  </SpaceStatus>
```

Connecting to a Windows File System data source

From your Discovery project, complete the following steps.

 **Note:** If you completed the prerequisite steps, return to the Windows File System data source collection that you started to create, and then skip to Step 4.

1. From the navigation pane, choose **Manage collections**.
2. Click **New collection**.
3. Click **Windows File System**, and then click **Next**.
4. Name the collection.

5. If the language of the documents that you want to crawl is not English, select the appropriate language.

For a list of supported languages, see [Language support](#).

6. **Optional:** Change the synchronization schedule.

For more information, see [Crawl schedule options](#).

7. In the **Enter your credentials** section, add values to the following fields. You provided these fields during the installation of the agent server, which was described in the [Prerequisite steps](#) section.

Host

The hostname of the remote Microsoft Windows server, for example `<hostname>.mydomain.com`.

Username

The username to connect the agent server. You use the username to connect Discovery to the shared network folders and crawl content.

Password

The password that is associated with the username.

Agent Authentication Port

The port to use for authentication. The default port value is **8397**.

Port

The port to use for transferring data. The default port value is **8398**.

8. In the **Specify what you want to crawl** section, enter the file path that you want to crawl in the **Path** field, and then click **Add**.

The file path is case sensitive.

Optionally, add more file paths.

9. **Optional:** Customize the types of files that are crawled.

The crawler is configured automatically to exclude a list of file extensions for file types that can be unsafe to crawl. You can add more file extensions to the excluded filter list, or list only the file extensions for file types that you want to include in the crawl. Listing the types of files to include is even more secure.

To change the file types that are crawled, in the **Extension filter** section, choose whether to use an Excluded or Included filter list. And then list the file extensions for the types of files you want to exclude or include.



Note: This configuration option was introduced with the 4.0.3 release.

10. **Optional:** Specify the character set of the data to crawl.

The converter that is used by the crawler is configured automatically to detect the character set of the files before it converts them. However, you can choose to specify a different character encoding to use for the data conversion. To specify a character encoding, complete the following steps:

- Set the **Automatic code page detection** switch to **Off**.
- In the **Code page to use** field, specify the character encoding as a [Java Charset](#) value. For example, **UTF-8** or **UTF-16**. If you don't specify a character set, ISO-8859-1 is used.



Note: This configuration option was introduced with the 4.0.3 release.

11. **Optional:** If you want to enable document-level security, in the **Security** section, set the **Enable Document Level Security** switch to **On**.

When you enable this option, your users can crawl and query content that they have access to. You must provide the details about the LDAP directory you want to use.

LDAP server URL

The LDAP server URL to connect to. For example, `ldap://<ldap_server>:<port>`.

LDAP binding username

The username to use to bind to the directory service.

LDAP binding user password

The password that is associated with the binding username.

LDAP base DN

The starting point for searching user entries in LDAP. For example, `CN=Users,DC=example,DC=com`.

LDAP user filter

The user filter to search user entries in LDAP. If empty, the default value is `(userPrincipalName={0})`.

12. If you want the crawler to extract text from images in documents, expand **More processing settings**, and set **Apply optical character recognition (OCR)** to **On**.



Note: When OCR is enabled and your documents contain images, processing takes longer. For more information, see [Optical character recognition](#).

13. Click **Finish**.

The collection is created quickly. It takes more time for the data to be processed as it is added to the collection.

If you want to check the progress, go to the Activity page. From the navigation pane, click **Manage collections**, and then click to open the collection.

Enabling TLS for an existing collection

To ensure that all traffic that is sent between the Windows Agent service and the crawler is sent over the transport layer security (TLS) protocol, enable TLS support.

This capability is available starting with version 4.7. Do not complete this task until after you upgrade your service software to 4.7.



Important: After you enable TLS for the Windows Agent service, any existing collections in deployments with earlier versions of Discovery will not be able to connect to this Windows Agent service.

To add TLS support to an existing collection, complete the following steps:

1. Open the **Processing settings** page for the existing Window File System collection.
2. Install the latest version of the agent.

Complete the steps in the [Installing the agent](#) procedure, starting with Step 4 and including the optional step to enable TLS support.



Important: Do not complete the last step that asks you to restart your computer.

3. Find and open the `as.cfg` file in a text editor, and then add the following lines to the file:

```
agent_key_store=%ES_AGENT_NODE_ROOT%\tls.p12  
agent_key_store_password=changeit
```

where `%ES_AGENT_NODE_ROOT%` is the root directory for the Windows Agent server. For example:

```
agent_key_store="C:\Program Files\IBM\es\distributed\esadmin\tls.p12"  
agent_key_store_password=changeit
```

4. Restart the Windows Agent service by using the following commands:

```
esagent stop  
esagent start
```

Troubleshooting ingestion

Learn about solutions and workarounds to warnings or errors that you might encounter when you add data to a collection.



Note: This information applies both to managed and installed instances of Discovery. For more troubleshooting tips for installed deployments only, see [Troubleshooting IBM Watson® Discovery Cartridge for IBM Cloud Pak® for Data deployments](#).

Unable to process one or more documents

This notification is displayed in the page header when a processing delay of any kind occurs in any project across the entire service instance. If the message is displayed while you are adding data to a collection, you can ignore it. If any problems occur that are related to the creation of your collection, a message is displayed in the **Activity** page for the collection. Check there for any pertinent messages.

This document exceeds the 1MB limit for non-HTML fields

The `html` field in the document index stores structural information about the document. If you add a single document with complex tables, images, or other objects that need to be represented in HTML, you might hit the size limit for this field. To work around this issue, consider breaking the source file up into 2 or more smaller files, and then add the files to the same collection separately so that you can apply enrichments and search them together.

Microsoft document troubleshooting tips

Failed to prepare document for SDU processing

Some DOC, PPT, and XLS files that use older features which are no longer supported by Microsoft Office can cause ingestion issues. If you encounter this issue, open the file in a more recent version of Microsoft Office and convert the file to the DOCX, PPTX, or XLSX format respectively, and then upload the DOCX, PPTX, or XLSX file.

Line breaks are inserted randomly

When some files in Microsoft Office format are added to a collection, line breaks are inserted seemingly at random to the text that is stored in the `html` field in the collection's index. The unexpected line breaks can impact the efficiency of enrichments, such as custom rule recognition.

Cause: As part of their ingestion into Discovery, such files are converted from Office format to PDF format. When the conversion happens, textual content is sometimes lost due to the nature of a PDF file. While the new lines appear to be added at random, they typically get inserted in areas where text wraps in the original document, such as in narrow text boxes or to accommodate other inline elements, such as images or diagrams.

Solution: To avoid new line insertions, increase the width of text boxes in the original document. If the original document has a section where text wraps to accommodate an inline element, such as an image, move the image so that it is situated in its own section and the nearby text doesn't need to wrap around it. To test whether your fixes address the issue, you can convert the original file to a PDF file to check for unexpected carriage returns in the text.

After applying a pretrained Smart Document Understanding model to a PPT file, table boundaries are not recognized properly

During the conversion process, text that is extracted from the table is confused with text that is outside the table in some PPT pages. This issue is more likely to occur in tables with a lot of text and that have footnotes displayed just outside the table border. If you encounter this issue, export the PPT file as a PDF file, and then upload the PDF file instead. Apply a user-trained Smart Document Understanding (SDU) model to the document, and then use the SDU tool to identify the tables in the document. The resulting model handles table boundaries properly and can extract text from the tables cleanly.

PDF file troubleshooting tips

Failed to parse document due to invalid encoding

Enable OCR for the file.

Enrichment troubleshooting tips

Table Understanding: n input tables excluded by enrichment

If tables in a document have inconsistent column and row spans or are too large for the system to process completely, the table understanding enrichment is not applied to them. Information from such tables cannot be returned in search results. If you want the table understanding enrichment to be applied to a table that was skipped, consider editing the table. Change a table with inconsistent column and row spans to have a simpler table format. Split a large table into many smaller tables.

To find the table where the enrichment was not applied, check the warning message. It lists the character offsets where the table begins and ends in the HTML representation of the document. To see the full warning message and get the document ID, click **View all**, and then make a note of the document ID. From the **Improve and customize** page, submit an empty search query to return all of the indexed documents. Look for the document ID. (You can change the search result settings to show the document ID as the result title.) Click the **View passage in document** link for your document, and then click **Open advanced view**. Choose to view the document as JSON and then look for the `html` field. Copy and paste the HTML representation of the document into a text editor. Look for the character offsets that were listed in the original warning message to find the table.

Managing data collections

After the processing of a new data collection is finished, you can see a summary of the settings that are applied to your collection from the **Manage collections** page.

For more information about how to create a collection, see [Creating collections](#).

Managing data

 **Note:** The **Manage data** page is available in installed deployments starting with the 4.6.5 release of IBM Cloud Pak for Data.

After you create a collection and the documents in the collection are indexed, you can see a list of the documents from the **Manage data** page.

1. Open the **Manage collections** page.
 2. Click to open the collection that you want to change.
 3. Click the **Manage data** tab.
- A list of the documents in the collection is displayed.
4. **Optional:** You can change the information that is displayed.

To change the fields that are shown in the view, click the **Settings** icon at the start of the view. Choose a different field for the first and second columns, and then click **Apply**.

For example, you can change the fields in the view to accomplish the following goals:

- Get the document ID for a document that you want to work with by using the API.
- Find the parent document for a document. Some file types, such as CSV or JSON files, generate subdocuments when they are added to a collection, for example. And splitting a document turns one document into multiple document segments.
- Retrieve the original file name for a document.
- Find out how many pages are in a document.

 **Note:** The custom settings that you apply are not retained. The default field settings are shown the next time that you access the page.

5. **Optional:** You can delete a document from the collection from this page. For more information, see [Excluding content from query results](#).

Changing how a data source is processed

You can change settings that were applied to a collection when it was created. You might want to change the schedule at which an external data source is crawled, for example.

To change how a data source is processed, complete the following steps:

1. Open the **Manage collections** page.
2. Click to open the collection that you want to change.
3. Click the **Processing settings** tab.
4. Make any changes that you want to make to the processing settings.

For example, you might want to enable or disable optical character recognition (OCR), which is a feature that extracts text from images. For more information, see [Optical character recognition](#).

For more information about changing data synchronization schedules, see [Crawl schedule options](#).

Other setting options differ by data source type.

5. Click **Apply changes and reprocess**.

Finding where a collection is used

To find out whether a collection is being shared, open the **My Projects** page, and then complete the appropriate step for your deployment:

- IBM Cloud Pak for Data [Click Collection usage and sharing](#).
- IBM Cloud Click [Data usage and GDPR](#), and then review the [Collection usage](#) page.

Collections can be associated with a single project, shared by two or more projects, or not associated with any project.

Reusing data from a collection

When you share collections across multiple projects, the following resources are shared:

- The processed data
- Configured connector

If you make any of the following changes to a shared collection, the changes are applied to the collection in every project where it is shared:

- Changing the Optical Character Recognition (OCR) setting
- Annotating fields or adding fields by using Smart Document Understanding
- Enabling or disabling fields
- Changing the setting for document splitting
- Changing any of the connector settings



Important: Enrichments and improvement tool settings are not included when a collection is shared because they are set at the project level.

For more information about the other tabs, see the following topics:

- **GDPR data label** IBM Cloud: For more information about GDPR and labeling data, see [European Union General Data Protection Regulation \(GDPR\)](#).
- **API usage** IBM Cloud Pak for Data For more information about monitoring Analyze API usage, see [Monitoring usage](#).

Deleting collections

Find out whether a collection is being used anywhere before you delete it from the **Collection usage** page. Unshared collections can be deleted directly from this page.

- To delete a single collection from a project, open the **Manage collections** page from the navigation panel, find the collection tile, and then click the delete icon.

Decide whether to keep the underlying data and configuration settings. If you choose to keep the data, you can find the collection in the unshared list on the **Collection usage** page. You might need to wait a few minutes before the collection is displayed.

Click **Delete from project**.

- IBM Cloud Pak for Data To delete all of the collections in your environment, select the Environment details icon , and then choose **Delete environment**.



Tip: *Environment* refers to the Discovery instance that you provisioned in IBM Cloud Pak for Data.

You cannot delete the **Sample Project** collection.

How your data is processed

When you connect to a data source, Discovery **processes** the information from the data source to create a **collection**.

The goal of processing a data source is to identify meaningful information and tag it as it is added to the collection so it is easier to find and retrieve the information later.

The processing that is applied to all data sources includes the following steps:

- Identify individual documents in the data source
- Find fields in the documents
- Index the fields

You can see a list of the fields that were indexed from the **Manage fields** page.

1. Go to the **Manage collections** page, and then choose the collection.

Make sure that the processing of the collection is finished first. The Activity page shows the processing status.

2. Click the **Manage fields** tab.

The fields that are shown can differ based on your data. However, one subset of fields is always listed. These fields, with names such as **footer** and **header**, are derived from the Smart Document Understanding (SDU) tool, and are listed even when you don't explicitly apply an SDU model to the collection. (For the full list of SDU-generated fields, see [Available fields](#).) Only the fields with a data type specified are stored in the collection's index.

One of the SDU-generated fields that is stored in the index is the **text** field. The **text** field typically contains the main body of text from the original document. Most of the content that is returned in search results that you submit from the **Improve and customize** page originates from this one field. How to parse and return only relevant chunks of information from this field is determined by the query result configuration that is used by the project. For more information, see [Previewing the default query results](#).

More processing adds more fields. And more processing is applied automatically depending on the project type. When processes run on documents in a collection, extra fields are added to store information that is associated with the process. For example, when the built-in Entities enrichment is applied to a collection, it starts a process that adds fields with names that begin with **enriched_{field_name}.entities** to the documents in the collection.

- For more information about the enrichments that are applied by default, see [Default project settings](#).

How fields are handled

For most unstructured file types, the bulk of the content from the file is added to a field named **text**. For file types that have an inherent data structure, such JSON files, names from the source file are used to name the fields in which the content is stored. When you upload files of this type, be aware of some naming limitations that exist for fields.

The following field names have special meaning. If possible, do not use these names in your structured source files.

- **document_id**
- **highlight**
- **html**
- **metadata**
- **parent_document_id**
- **result_metadata**
- **score**
- **spans**

Avoid field names that meet the following conditions. Field names with these restricted characters are not queried.

- Start with the characters **_**, **+**, and **-**. For example, **+extracted-content**.
- Contain the characters **.**, **,**, **#**, **?**, **(**, **)**, or **:** or spaces. For example, **extracted content** or **new:extracted-content**.
- End with numbers, for example, **extracted-content2**.

HTML fields

The **html** field in the document index stores structural information about the document.

- If you use the Smart Document Understanding tool to annotate a collection, the document representation is indexed in the **html** field.
- If you use the Smart Document Understanding tool to apply a pretrained model to a collection, the document representation is indexed both in the **html** field and **text** field.
- The **html** field has a size limit. For more information, see [Field limits](#).

Note about enhancing data:

- If you want to apply an enrichment that can understand the tables in a document, the document must contain an `html` field.

How dates are handled

Dates are captured in different ways by different file types.

Unstructured files

The best way to capture date information from the body of a document with unstructured data is to use a natural language processing model enrichment. For example, the prebuilt Entities enrichment recognizes dates and annotates them in the `text` field (or other body fields with the `String` data type). In a document where the enrichment is applied, you can find dates by looking for fields that are labeled as `enriched_{fieldname}.entities.type = Date`.

Dates from metadata date fields, such as `extracted_metadata.publicationdate`, are stored in the index as dates as long as the date format matches one of the supported date data type formats. You can't see nested fields from the **Manage fields** page. And when you view a search result as JSON, date field values are displayed as string values because the JSON editor shows the date as a string. However, values from date fields behave like dates. You can use greater than (`>`) or less than (`<`) operators with such fields in Discovery Query Language queries, for example.

Structured files

Structure files that you import, such as CSV or JSON files, might contain date fields that you want to store as date data types. Discovery can recognize many date formats. However, you might need to add a format to the list. For more information, see **Date format settings**.

Date format settings

If your documents have a root-level field with date information in it, you can set the field to be a `Date` data type field in the index.

Discovery recognizes the following date formats automatically:

```
yyyy-MM-dd'T'HH:mm:ssZ
yyyy-MM-dd'T'HH:mm:ssXXX
yyyy-MM-dd'T'HH:mm:ss.SSSZ
yyyy-MM-dd'T'HH:mm:ss.SSSX
yyyy-MM-dd
M/d/yy
yyyyMMdd
yyyy/MM/dd
```

If you store dates in other formats, you can add the format to the list of supported formats.

To add more date formats, complete the following steps:

1. From the **Manage fields** page for the collection, add a format as a new line in the **Date formats** field.

Specify a date format that is supported by the Java [SimpleDateFormat](#) class.

For example, if your records store only year values for dates, add `yyyy` to the supported date formats list. You can then set the data type for the field that contains a year value to `Date`, and reprocess your collection. As a result, an occurrence of `2019` in the date field is stored as `2019-01-01T00:00:00Z` in the index.

When you add a date format, you must specify an associated time zone for the date.

2. Specify a time zone.
3. Optionally, select a date locale.

The locale that you choose is used to parse a string value that represents the date for the date-type data set fields. For example, by using the `EEE, MM dd, yyyy` format, the **English (United States)** locale can parse the string value of `"Wednesday, 07 01, 2020"`, and the **Japanese (Japan)** locale can parse the same string value of `"木曜, 07 01, 2020"`.

4. If you already imported documents with dates in formats that were not recognized, reprocess the documents.

Discovery cannot store a date that is mentioned within a text field as a `Date` field in the index. However, you can use an enrichment such as the **Entities** enrichment to identify dates that are mentioned in text.

How file types are handled

When you upload a document, data in the file is indexed. Different file types are handled differently by Discovery.

- [CSV files](#)
- [HTML files](#)
- [JSON files](#)

CSV files

Notes about adding data:

- Each line that is defined in the CSV file is added to the index as a separate document, each with the same `parent_document_id`. The child documents typically have a document ID with the syntax `{parent-ID}_n` where `{parent-ID}` is the document ID of the original file that was added and `n` is a sequential number. For example, if you upload a CSV file with 5 rows, then five documents are added to the collection with document IDs such as `f5214225c1e03e25190ffcdfad8e84ff_0` through `f5214225c1e03e25190ffcdfad8e84ff_4`.
- You cannot enable the Optical Character Recognition (OCR) feature for CSV files.
- If the CSV file has headers, the header names are used to name the fields in which the content from the corresponding column is stored. Do not use names that have special meaning in Discovery. Be sure that the field names conform to the naming rules, such as having no spaces and no appended numbers. For example, you can rename the `start date` header to `start_date` and `label1` to `label-one` before you add the file. For more information, see [How fields are handled](#).
- When a CSV file header name contains restricted characters, the document converter automatically removes the restricted characters from the field name when it adds the resulting field to the index.

Note about enhancing data:

- You cannot apply prebuilt or user-trained Smart Document Understanding models to CSV files.

HTML files

If you upload an HTML file or crawl a data source with HTML files, such as a website, an `html` field is generated along with the `text` field. For more information, see [HTML fields](#).

JSON files

Notes about adding data:

- Object names from the source JSON file are used to name the fields in which the content is stored. Do not use names that have special meaning in Discovery. Be sure that the names conform to the naming rules, such as having no spaces and no appended numbers. For example, you can rename the `updated on` object to `updated_on` and `answer2` to `answer-two` before you add the file. For more information, see [How fields are handled](#).
- If a root-level field is an array but contains no items, the field is omitted from the index.
- If a root-level field is an array and contains only one item, the array is indexed as the data type of the one item. For example, a string array with one string is indexed as a string.
- If a nested field contains an array, even if the array has only one value, it is indexed as an array.
- If a root-level field is an array and contains more than one item, the data is indexed as an array.
- If you copy JSON that is generated by Discovery and then upload it as a JSON file, remove these system-generated fields from the file first: `document_id`, `parent_document_id`, `filename`, and `title`.
- You cannot enable the Optical Character Recognition (OCR) feature for JSON files.
- If your source document has a field with the name `document_id`, the field is skipped and not added to the index in the collection.



Note: How the `document_id` field in a JSON file is handled changed with the `2023-03-31` version update of the API. Before the update, when you uploaded a JSON file from the product user interface or used the API to add it with the `Add document` method, the value in the `document_id` field from the file was shown as the `document_id` value in query results. However, a different document ID was assigned to it and stored in the `parent_document_id` field. The assigned document ID is what was returned when you called the `List documents` method and is what had to be used as the `document_id` in the endpoint URL for a `Delete document` method request. When you used the `Update document` method to assign a new `document_id`, the original ID continued to be returned in query results. However, the assigned ID had to be used to delete the document. If you have an application that relies on the previous behavior, you can specify a version number earlier than 2023-03-31, such as `2020-08-30`, in your API calls.

Notes about enhancing data:

- You cannot apply prebuilt or user-trained Smart Document Understanding models to JSON files.
- When you apply an enrichment to a field from the JSON file, the field data type is converted to an array. The field is converted to an array even if it contains a single value. For example, "field1": "Discovery" becomes "field1": ["Discovery"].
- Only the first 50,000 characters of a custom field from a JSON file are enriched.
- In project types where the **Part of Speech** (POS) enrichment is applied automatically, the enrichment is applied to the field that contains the bulk of the file content in the first JSON file that is added to the collection. This field is determined by the following rules:
 - If a field is named **text**, the POS enrichment is applied to it.
 - The field with the longest string value and highest number of distinct values is chosen.
 - If more than one field meets the previous condition, one of the fields is chosen at random.
- If you want to apply an enrichment to a nested field, you must create a Content Mining project, and then apply the enrichment to the field. If you want to use a project type other than Content Mining, you can reuse the collection that you created with the Content Mining project type elsewhere. For more information, see [Applying enrichments](#).



Note: You can specify the **normalizations** and **conversions** objects in the [Update a collection](#) method of the API to move or merge JSON fields.

How passages are derived

Discovery uses sophisticated algorithms to determine the best passages of text from all of the documents that are returned by a query. Passages are returned per document by default. They are displayed as a section within each document query result and are ordered by passage relevance.

Discovery uses sentence boundary detection to pick a passage that includes a full sentence. It searches for passages that have an approximate length of 200 characters, then looks at chunks of content that are twice that length to find passages that contain full sentences. Sentence boundary detection works for all supported languages and uses language-specific logic.

For all project types except **Conversational Search**, you can change how the passages are displayed in the search results from the **Customize display > Search results** page. For example, you can configure the number of passages that are shown per document and the maximum character size per passage.

Querying your data from the UI

Previewing your query results

See the types of query results that are returned automatically and learn about how they are derived.

When a document is ingested, the text is extracted and indexed in the **text** field. To return only the subset of information that is relevant to the query, Discovery returns **passages** from the **text** field. For more information about passages, see [How passages are derived](#).

When you enter a query from the product user interface, the UI submits the text as a natural language query. For more information about how to query your data programmatically, see [Query API](#).

To query your data from the product user interface, complete the following steps:

1. From the navigation panel, open the **Improve and customize** page.
2. Take the appropriate next steps for your project type.
 - [Document Retrieval](#)
 - [Conversational Search](#)
 - [Document Retrieval for Contracts](#)
 - [Content Mining](#)

Document Retrieval

1. Do one of the following things:
 - Click **Run search** for one of the keywords that Discovery calculated to have special meaning in your collection.
 - Submit your own phrase or keyword from the search bar.

You can see that the query results that are returned consist of passages.

Entities that are recognized in your documents (based on the Entities enrichment that is applied to the project by default) are displayed as facets by which you can filter the query results.

2. To explore a query result in more detail, click **View passage in document**.
3. Click **Open advanced view** to explore the entity mentions that are recognized by Discovery.

Excerpt unavailable

Passages displayed in search results are extracted from the content that is indexed in the **title** and **text** fields of the documents. If the content is indexed in other fields, the search displays the **Excerpt unavailable** message.

Content is indexed in other fields in the following scenarios:

- Your collection contains structured files, such as JSON or CSV files. When you ingest structured files, content is stored in custom fields with names taken from the original object names (JSON) or column headers (CSV).
- You applied a Smart Document Understanding model that moves content from the **text** field into new fields, such as **section** or **results**, based on the document's structure.

To improve the search results, first choose how you want to extract content from the documents. If your documents have targeted fields with succinct content in them, such as **answer** fields for an FAQ use case, configure the search to return those specific fields. If your documents have custom fields with lots of content in them, such as **chapter** fields, configure the search to find the best passages from the custom fields.

To customize the results, complete the following steps:

1. From the **Improvement tools** panel, expand **Customize display**.
2. Click **Search results**.
3. In the **Select source of result content** option, do one of the following things:
 - Select **Passages**, and then specify one or more fields from which to extract passages.
 - Select **Field**, and then choose one or more fields to return.

customize

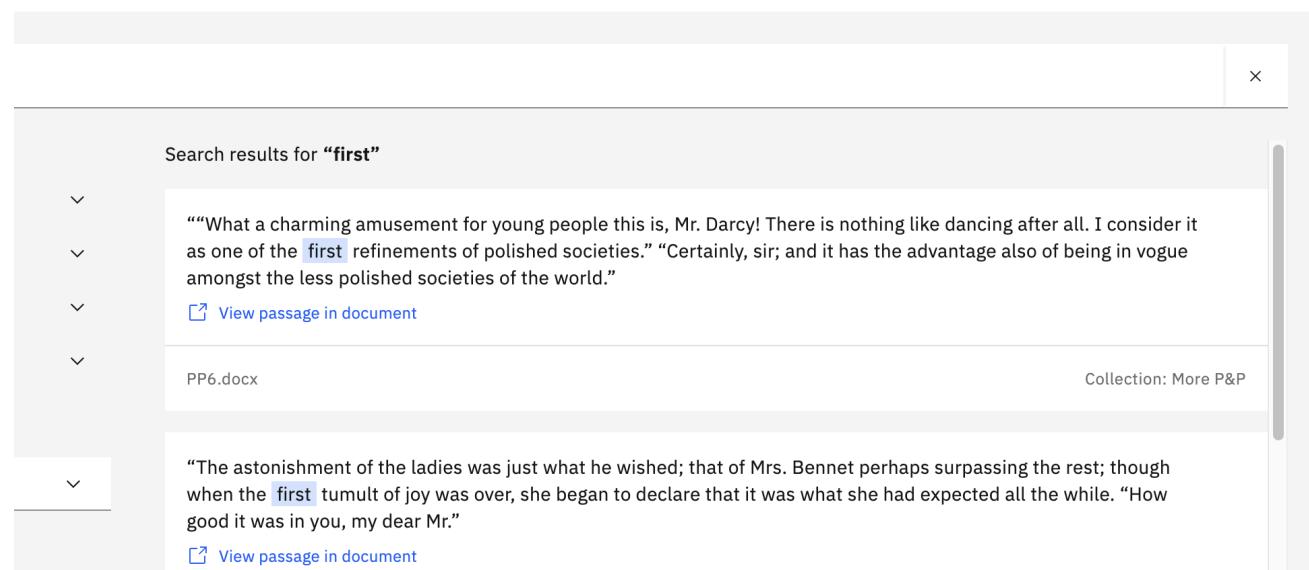


Figure 1. Search results dialog

4. Click **Apply**.

Conversational Search

A single search field is displayed that mimics the user interface of a virtual assistant.

1. Submit a phrase or keyword.

The query results are returned as passages by default. You can [configure the search to return a specific field](#) instead.

If you want to investigate the results a bit more, you might want to use a different project type. For more information, see [Improving results for a chatbot](#).

Document Retrieval for Contracts

Contract-related elements that are recognized in your collection are displayed.

1. Do one of the following things:

- Filter the documents by one of the highlighted elements or by entities that are recognized in your documents (based on the Entities enrichment that is applied to the project by default).
- To view the contract elements in more detail, click a document result to open it. Open the **Contract Data** tab.

For more information about the elements, see [Understanding contracts](#).

Content Mining

Facets based on the **Part of Speech** enrichment are shown.

1. To analyze your data, open the Content Mining application. Click **Launch application**.

For more information, see [Analyzing your data](#).

What to do next

- For more information about how to enrich your documents so that you can find key information, see [Choosing enrichments](#).
- To explore ways to improve the query results, see [Improving your query results](#).

Improving your query results

Learn about actions you can take to improve the quality of your query results.

You can use the tools that are built in to Discovery to make improvements.

Results include more than exact matches

Unlike some other search applications, adding quotation marks to a phrase that you submit does **not** return only exact matches. Queries that are submitted from the product user interface are natural language queries. When quoted text is submitted in a natural language query, the phrase is used to boost result scores. However, results are not limited to documents that contain the entire phrase.

If you want more control over how queries are handled, you must use the query API. For more information about the **phrase** operator of the query API, see [Query operators](#).

A short query returns irrelevant results

It might be that your query contains too many stop words and not enough distinct terms to trigger a meaningful search. When you submit a query, the query text is analyzed and optimized before it is submitted to the project. One of the changes that occurs is the removal of any stop words from the text. A **stop word** is a word that is considered to be not useful in distinguishing the semantic meaning of the content. Examples of stop words include terms such as `and`, `the`, and `about`. Discovery defines a list of stop words that it ignores automatically both when the data is indexed and when it is searched. When you submit a query that contains mostly or only stop words, such as `About us`, it is equivalent to submitting an empty query.



Note: Although `us` is not included in the stop words list, it is lemmatized to `we`, which is listed as a stop word.

You can edit the stop words that are used by your collection. However, you can only augment the stop words list; you cannot remove stop words. And the stop words that you define are used only at query time. They do not affect the stop word list that is used by Discovery when data is added to a collection and the index is created.

For more information, see [Identifying words to ignore](#).

Results have too much text

If the source document is large, consider splitting the document into smaller chunks.

To do so, you can create a Smart Document Understanding user-trained model. Find content in the document that can be used to consistently break your document into subsections. For example, maybe your document has chapters or subtitles. You can label the chapters with a custom label named `chapter`. After you teach the model to recognize the `chapter` content type, apply the model to your entire collection. For more information, see [Using Smart Document Understanding](#).

You can then split the document by the `chapter` field to create many subdocuments segmented by chapter. For more information, see [Split documents to make query results more succinct](#).

Information from tables is not found

The table understanding enrichment must be applied to your collection for information from tables to be searchable. The table understanding enrichment is applied to collections automatically in some situations. If it isn't and your collection has an HTML field in its index, you can apply the **table understanding** enrichment yourself.

For more information, see [Understanding tables](#).

Information from diagrams is not represented

Text from diagrams and other images is not captured unless you enable the optical character recognition (OCR) setting for the collection. You can apply the setting to a collection after its initial creation. For more information, see [Managing data collections](#).

Search does not recognize significant terms

If the results suggest that keywords, common nouns, or domain-specific terms in the query are not being recognized as significant, enrich your collection.

Use Watson Natural Language Understanding to find and tag terms that are generally understood to have special meaning, such as locations or company names. For more information, see [Applying prebuilt enrichments](#).

Teach Discovery about terms and patterns that have special meaning to your use case. For more information, see [Adding domain-specific resources](#).

Default facets aren't useful

You can add facets that categorize documents based on data from enrichments that you apply to a collection. For example, you might want to show facets based on keywords or dictionary categories. For more information, see [Facets](#).

Explore other search features

When you test your project from the Discovery user interface, you submit a natural language query. Search features are available that you can enable to influence how the natural language query search is done. And Discovery Query Language search is another type of search that you can leverage by using the API. If the initial results don't meet your needs, experiment with another search method.

- Discovery Query Language (DQL) search: A search mechanism that accepts more complex queries. You must use the query API to submit DQL queries.

For example, you can search for specific values in fields that are generated by enrichments that are applied to a collection.

- Natural language query is the type of search that is triggered from the [Improve and customize](#) page.

For more information about the Query API, see [Query API overview](#).

Adding facets

Add more facets that you can use to filter your data.

When you apply custom enrichments to your collection, annotations are added to its documents. The annotations feed into new facets that you can use to sort your data.

The following table describes the types of facets that you can create from annotations.

Information to recognize	Annotator type
Commonly understood terms, such as organization or people names.	Built-in Natural Language Processing models
Phrases that express an opinion and evaluate whether the opinion is positive or negative.	Phrase sentiment
Alternative words that share a meaning with terms in a finite list.	Dictionary
Terms that match a syntactical pattern	Regular expression
Custom terms by the context in which they are used.	Machine learning model
Documents that fit into categories that you define.	Document classifier
Custom facet types	

Grouping facets

To organize your facets, you can group them in folders.

Grouping facets does not combine the data from the facets. It merely makes the facets easier to find because they are organized in named folders.

To associate facets such that you can combine data from multiple facets, the facets must have a facet and subfacet relationship. Such hierarchical relationships must be defined at the time that the facet enrichment or annotation is created and applied to the collection.

To group facets, complete the following steps:

1. From the initial search page, submit a search.
2. From the **Facet analysis** pane, click the **Edit** icon.
3. Name the group, and then select the facets that you want to group together.

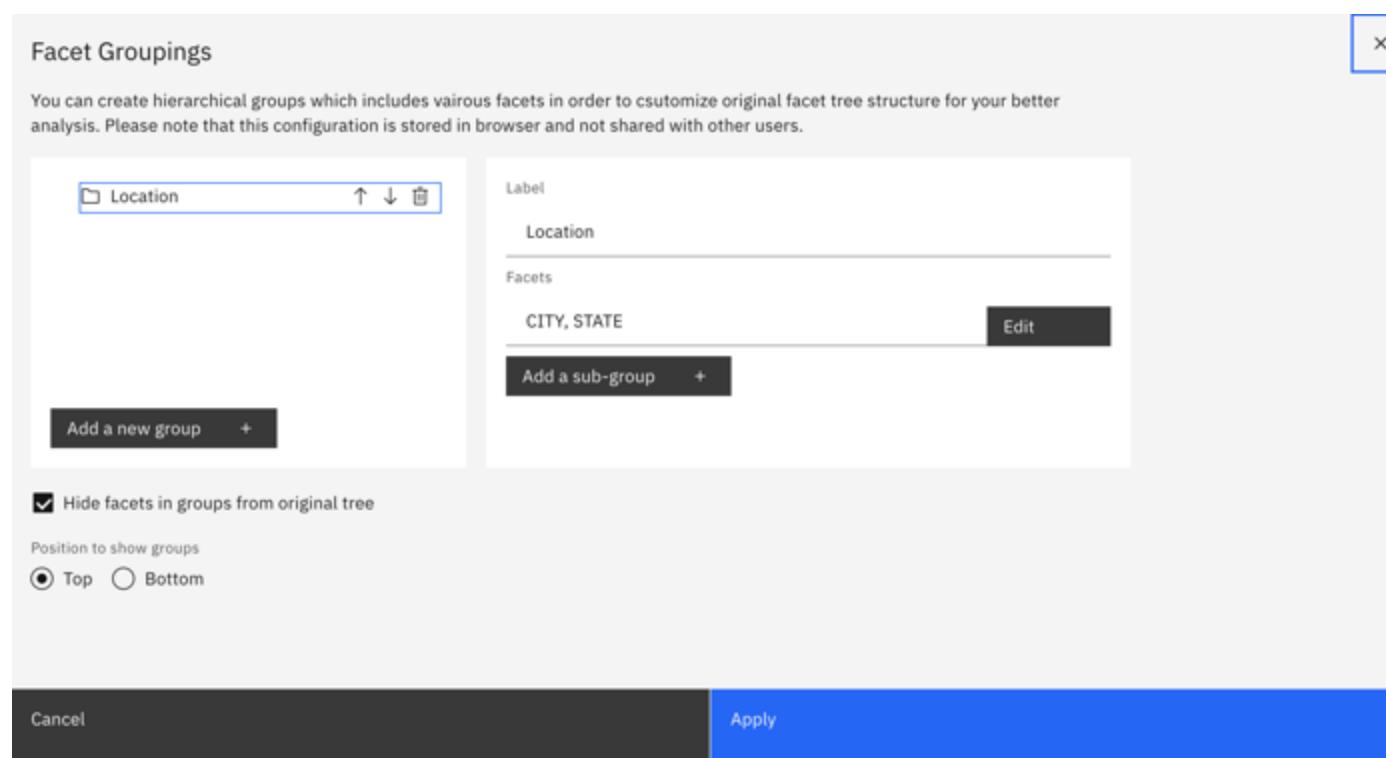


Figure 1. Facet grouping dialog

4. Click **Apply**.

5. The facets that you grouped are now available from a folder with the group name.

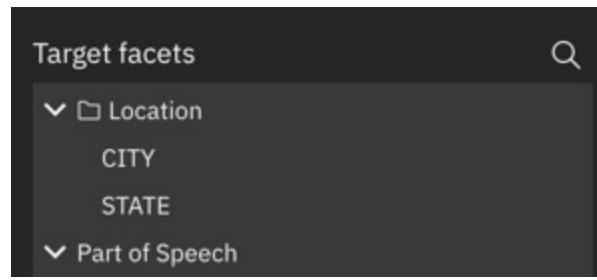


Figure 2. Facet folder from the facet list

Customizing the search bar

Control how customers interact with the search bar.

Decide whether the search bar can interact with customer query submissions in the following ways:

- Propose alternative search terms when a misspelling is detected
- Propose better query wording with type-ahead



Note: These search bar customizations are available for all project types except *Conversational Search*. Similar features, such as autocorrection, can be configured for the chat widget where you deploy the project.

To customize search bar behavior, complete the following steps:

1. From the navigation pane, open the **Improve and customize** page.
2. Expand **Customize display** from the *Improvement tools* pane, and then click **Search bar**.
3. Turn the following features on or off by setting the associated switcher:

Autocompletion

As the customer types a word as part of a query into the search bar, completed words are displayed as suggestions. The user can click a suggestion to add it to the query. The suggested words are based on terms from the project documents. Suggestions are not based on terms from the user's search history and the project does not learn from the user's choices. This setting is enabled by default.

Spelling suggestions

Recognizes words that are misspelled in the customer query. After the query is submitted, a **Did you mean:** link is displayed that shows a corrected version of the original query. The customer can click the corrected query to submit it. This setting is disabled by default.

Expanding the meaning of queries

You can improve the quality of search results by expanding the meaning of the queries that are submitted by customers.

To expand the scope of a query beyond exact matches, add a synonyms list to your collection. When synonyms are defined, the customer does not need to submit an exact phrase or keyword that your project is trained to understand. Even variations of the term are recognized and used to find the best results. For example, you can expand a query for **ibm** to include **international business machines** and **big blue**. Query expansion terms are typically synonyms, antonyms, or common misspellings for terms.



Note: Synonyms that you add to improve the search results function differently from synonyms that you add to a dictionary. Dictionary synonyms are recognized and tagged at the time that a document is ingested. The synonyms that you define are recognized and tagged as occurrences of the associated dictionary term, so that they can be retrieved later by search. For more information about adding synonyms that are recognized when documents are processed, see [Dictionaries](#).

You can define two types of expansions:

Bidirectional

Each entry in the **expanded_terms** list expands to include all expanded terms. For example, a query for **ibm** expands to **ibm OR international business machines OR big blue**.

Bidirectional example:

```
{  
  "expansions": [  
    {  
      "expanded_terms": [  
        "ibm",  
        "international business machines",  
        "big blue"  
      ]  
    ]  
  ]  
}
```

Unidirectional

The **input_terms** in the query is replaced by the **expanded_terms**. For example, a query for **banana** is converted to **plantain OR fruit** and does not contain the original term, **banana**. If you want an input term to be included in the query, then repeat the input term in the expanded terms list.

Unidirectional example:

```
{  
  "expansions": [  
    {  
      "input_terms": [  
        "banana"  
      ],  
      "expanded_terms": [  
        "plantain",  
        "fruit"  
      ]  
    },  
    {  
      "input_terms": [  
        "car"  
      ],  
      "expanded_terms": [  
        "car",  
        "automobile",  
        "vehicle"  
      ]  
    }  
  ]  
}
```

To enable query expansion, complete the following steps:

1. Create a synonyms list file. The file must be a JSON file with the **json** file extension.

Follow these guidelines:

- Specify the **input_terms** and **expanded_terms** in lowercase. Lowercase terms expand to uppercase.
- The synonyms files cannot contain terms that are specified as stop words. For example, if **on** is included in your stop words file, and you specify in your synonyms file that **rotfl** expands to **rolling on the floor laughing**, the expansion won't return the expected results. Check the words in the stop words file that is used by your collection by default to make sure that you don't use any of the same words. For more information, see [Identifying words to ignore](#).

You can use the [expansions.json](#) file as a starting point when you build a query expansion list.

2. From the navigation pane, open the **Improve and customize** page.
3. Expand **Improve relevance** from the Improvement tools pane.
4. Click **Synonyms**, and then click **Upload synonyms** for the collection.

Do not upload a synonyms file while documents are being added to your collection. The ingestion processing that occurs when documents are added can cause the index to be unavailable.

Only one synonyms list can be uploaded per collection. If a second expansion list is uploaded, the second list replaces the first.

5. Run a test query to verify that the query expansion is working as expected.

Query expansions are applied at query time, not during indexing, so you can add synonyms without reprocessing your collection.

To disable query expansion, delete the synonyms file. However, do not delete a synonyms file while new documents are being

processed.

Identifying words to ignore

To ignore meaningless terms during searches, add a list of custom stop words. Stop words are words that are not useful in distinguishing the semantic meaning of the content.

In English, `the`, `is` and `and` are examples of stop words.

The stop words that you define are filtered out of queries and improve the relevance of natural language query results.

For example, a company has three tiers of service. The documents in one of the collections pertain to only one tier, the Silver tier. You might want to add `"silver"` to the stop words list because the term doesn't help to distinguish the significance of one document over another, given that all of the documents relate to the Silver service tier. When a customer mentions the Silver tier in a query string, it is ignored. Other terms in the query that are more significant are used to search the data instead. Or maybe the document collection consists of car accident reports only. You might want to add `"car"` to the stop words list to prevent mentions of `car` in queries from adding noise to the search.

Discovery applies a list of default stop words for many of the supported languages automatically. These stop words are applied both at indexing time and at query time. The predefined stop words are ignored when content is indexed and they are filtered out of queries. However, stop words that you define are used at query time only. Your list doesn't replace the default list; it augments the default list. You can add stop words, but you cannot remove stop words.

Example custom stop word list:

```
{  
  "stopwords": [  
    "a", "an", "the", "ibm", "what", "how", "when", "can", "should", ...  
  ]  
}
```

Default stop word lists

You can access the default stop words list for English from the [Watson Developer Cloud GitHub repository](#).

For the following languages, Discovery uses the default stop words list that is defined by Apache Lucene. For more information about what words are included in the list, see the Lucene reference documentation:

- Arabic: [stopwords_ar.txt](#)
- Czech: [stopwords_cs.txt](#)
- Danish: [stopwords_da.txt](#)
- Dutch: [stopwords_nl.txt](#)
- Finnish: [stopwords_fi.txt](#)
- French: [stopwords_fr.txt](#)
- German: [stopwords_de.txt](#)
- Hindi: [stopwords_hi.txt](#)
- Italian: [stopwords_it.txt](#)
- Norwegian (both supported dialects): [stopwords_no.txt](#)
- Portuguese: [stopwords_pt.txt](#)
- Romanian: [stopwords_ro.txt](#)
- Russian: [stopwords_ru.txt](#)
- Spanish: [stopwords_es.txt](#)
- Swedish: [stopwords_sv.txt](#)
- Turkish: [stopwords_tr.txt](#)



Note: These default stop words are documented in TXT format, but if you want to augment the list and submit it for use by Discovery, you must submit a JSON file. To see an example of the syntax of stop words list file, see the custom English stop words list file.

For the remaining supported languages, no default stop words are used. You can specify a stop words list to use at query time for these languages. The list that you submit is not used when data is ingested.

Examples of stop word lists that you might want to apply at query time include:

- Japanese: [custom_stopwords_ja.json](#)
- Polish: [custom_stopwords_pl.json](#)

See [supported languages](#) for the list of the languages that are supported by Discovery.

Defining query-time stop words

To define stop words, complete the following steps:

1. Create a stop words file. The file must be a JSON file with the **json** file extension.

Follow these guidelines:

- Specify stop words in lowercase.
- In general, keep your list of stop words under **200** total words. The size limit is one million characters. However, if you specify too many terms, you might negatively affect search accuracy.

You can use the default English stop words list file, [custom_stopwords_en.json](#), as a starting point when you build a custom stop word list in English.

2. From the navigation pane, open the **Improve and customize** page.
3. Expand **Improve relevance** from the Improvement tools pane.
4. Click **Stopwords**, and then click **Upload stopwords** for the collection.

Only one stop words list can be uploaded per collection. The stop words list that you upload augments the default stop words list for your collection; it does not replace the default list.

5. Click **Done**.

To disable a custom stop words file and revert to using the default stop words, delete the custom stop words file.

Split documents to make query results more succinct

Split your documents so that the search function can find more concise information to return in query results.

For more information about the benefits of splitting documents, read the [Using IBM Watson Discovery's New Document Segmentation Feature](#) blog post on Medium.com.



Note: You can split only documents to which a user-trained Smart Document Understanding model is applied.

When you split a document, the original document is broken into segments. Each segment contains a more uniform set of information. By splitting the content in your documents into segmented groups, you can enrich and index your data at a more granular level.

To control how your documents are split, you specify a field, such as **subtitle** or **question**, to use as the page break marker. The page break options are populated with fields that are created when you apply a user-trained Smart Document Understanding (SDU) model to the documents. For more information, see [Using Smart Document Understanding](#). You cannot split documents with fields that are generated by a pretrained Smart Document Understanding model.

As a document is reprocessed, it is evaluated from start to end. Whenever the page break marker field occurs, the original document is split and a new segment is created. The splitting continues at each marker field until the original document is broken into multiple segments.

Before you begin, decide which field to use as the page break marker.

- You can use any of the fields that are indexed by default. To see your choices, check the **Fields to index** list. Fields that have a **Type** value are stored in the index.
- The number of segments per document is limited to **1,000**. After segment number **999** is created, any remaining document content is stored within segment **1,000**.
- Metadata from PDF and Microsoft Word documents and any custom metadata is extracted and included in the index with each segment.

Be careful with documents that contain repeating sections, such as a catalog that has a description and specifications section for each product entry. If you split the document at too granular a level, the subsections, such as a section with specification details, can be disassociated from the product to which it belongs.

To split the documents in a collection, complete the following steps:

1. Click **Manage collections** from the navigation panel, and then click to open a collection.
 2. Open the **Manage fields** page.
- A list of the identified fields is displayed.
3. From the **Improve query results by splitting your documents** section, click **Split document**.
 4. Choose the field that you want to use as your page break marker from the **Select field** dropdown.

The list that you can choose from includes a subset of all the identified fields.

5. Click **Apply changes and reprocess**.

You can check the status of the splitting process from the **Activity** page.

The metadata field includes the parent document ID. Each resulting segment of the original document can contain different information. For example, if you split the document based on the subtitle field, the first segment might contain only a title field. The next segment might contain a subtitle and a text field. The third might contain a subtitle field, a text field, and a footer field.

Updating documents that were split

If a document that was split changes and you want to upload the document again, work with a developer to replace the document by using the API. A developer can use the **Update a document** method to replace the original parent document. For more information, see the [API reference](#). To provide the `{document_id}` path variable that must be sent with the request, copy the contents of the `parent_document_id` field of one of the document's segments.

When you replace the original document, all of the segments are overwritten, unless the updated version of the document has fewer total segments than the original. Those older segments remain in the index.

Deleting document segments from the index

You can delete documents in a collection from the **Manage data** page. To find all of the document segments that were generated from a single document, check for documents with the same `metadata.parent_document_id` field value. For more information, see [Excluding content from query results](#).

IBM Cloud Pak for Data [IBM Cloud Pak for Data before the 4.6.5 release](#)

The **Manage data** page is available in installed deployments starting with the 4.6.5 release. In earlier releases, a developer can delete document segments by using the API. For more information, see the [delete document API](#).

Excluding content from query results

Prevent content that you don't want customers to see from being included in query results.

You can prevent content from being included in query results in the following ways:

- Delete an entire collection.
For more information, see [Deleting collections](#).
- Remove a field from the index that contains data that you don't want to share with customers.

You can control which fields are indexed. If you want to prevent a field from being indexed, you can set it to be excluded. For example, if your PDF files contain a running header or footer that does not contain useful information, you can exclude the `header` and `footer` fields from the index.

To manage the fields to index, complete the following steps:

1. From the navigation pane, open the **Manage collections** page, and then click a collection to open it.
 2. Click the **Manage fields** tab.
A list of the identified fields is displayed. You can see which fields are included in the index and which are not.
 3. To remove a field from the index, set the **Include** switch to off.
- Delete a single document.



Note: If you use the Smart Document Understanding tool to annotate a document, and then decide that you want to delete the document and its associated SDU annotations, you must remove the annotations before you delete the document. To remove the annotations, annotate the document again. This time, label all of the content as `text`.

To delete a document, complete the following steps:

1. From the navigation pane, open the **Manage collections** page, and then click a collection to open it.
2. Click the **Manage data** tab.

A list of information from each document in the collection is displayed. If the information that is displayed doesn't help you identify the document that you want to delete, you can change what is displayed.

- Click the **Settings** icon in the table header.
- Choose fields from which you want to fetch data to display in the first and second columns. You can choose fields such as `extracted_metadata.filename` to show the document file name, or `document_id`, for example.



Tip: You can page through the documents in the collection by using the controls in the table footer.

3. After you identify the document that you want to delete, select the checkbox that is associated with the document, and then click **Delete**. Confirm the deletion.

Documents that are added to a collection from an external data source will be added back to the collection with the next scheduled crawl of the data source. The delete function removes the document from the index of the collection, not from the external data source.



Note: Some file types, such as CSV or JSON files, generate subdocuments when they are added to a collection. Splitting a document turns one document into multiple document segments. If you delete one of these generated documents, and then repeat the action that created it, the deleted document is added back in to your collection.

IBM Cloud Pak for Data [IBM Cloud Pak for Data releases before 4.6.5](#)

The **Manage data** page is not available in installed deployments before the 4.6.5 release. You must use the [Discovery API](#) to delete a document. And you must know the document ID of the document that you want to delete. To get the document ID, use the [List documents](#) API method.

If the document is a subdocument of another document and you want to remove it, its parent, and any other subdocuments that are associated with the parent, delete the parent document. To get the document ID of the parent document, look for the **metadata.parent_document_id** field for the document. It is specified in the JSON representation of the document when it is returned as a response in the [Improve and customize](#) page of the product user interface.

Improving result relevance with training

The relevance of natural language query results can be improved in IBM Watson® Discovery with training.

A relevancy model determines the most relevant documents to return in search results. Without relevancy training, a standard mechanism is used to determine relevance based on common factors. When you train a relevancy model, you help Discovery to use features that are unique to your documents as it determines relevance.

The relevancy training model that is associated with a project is used at run time only when natural language queries are submitted. The model is not applied to Discovery Query Language (DQL) queries.



Important: You cannot apply relevancy training to **Content Mining** project types.

To train a relevancy model, you provide sample natural language queries, submit them to get results from your documents, and then rate those results. As you add more examples, the information you provide about result relevance for each query is used to learn about your project. The system uses your assessments to assign importance to different types of structural information within the documents. For example, it learns the importance of when a keyword from the search query appears in the title versus the header, body, or in the metadata of the document. It also learns from the importance of the distance between one matching keyword and another. After a successful relevancy training session, a ranker model is created. The model is used automatically by Discovery with the next natural language query. Discovery reorders the document results so that the most relevant results according to the relevancy training model are displayed first.

Training applies to an entire project. It cannot be skipped for one collection and applied to other collections in the same project. You do not enable use of the training model by specifying a query parameter. If present, the model is used for every natural language query that is submitted for the project. The model is used whether you limit the search to one collection or all of the collections. For this reason, it is important that your training data represents queries that are likely to be answered by all of the collections in your project. To stop a project from using the relevancy training model, you can delete the model by using the API.

Relevancy training does not run continuously. Training occurs only when you initiate it. At most one trained relevancy model is used at a time per project. If you retrain a model, the existing model is used until the new model is successfully trained, at which time the new model replaces the old model.

The set of documents that constitute the training data are used only during the training process. If a subsequent change is made to a document that was used to train the model, it does not change the trained model and does not trigger a new training session. Keep in mind that if many of the documents in your project change, it might be time to retrain the model to use the features from the updated documents.

Stop words and query expansions that you add to a collection do not affect the relevancy training model directly. However, they can change which documents are returned from a search, which affects the documents that are ranked by the relevancy model. The model ranks the top 100 documents that are returned for a query. Changes that you make to stop words or query expansions do not initiate a relevancy training update. If you add artifacts that drastically change the documents that are returned by search, consider retraining the model.

If documents that were used previously to train the model are removed from a collection, you must remove any references to them

from the training data before you start to retrain the model. The model expects both the documents and queries from training data pairs to continue to exist. To remove these references, delete the training queries that returned the deleted documents. If the queries continue to be relevant, you can add them back to the training data and pair them with other documents.

For more information about the relevancy training API, see the [API reference documentation](#).

When to use relevancy training

Relevancy training is optional. Test the quality of your search results. If the results of your queries meet your needs, no further training is necessary.

The training improves the relevancy of the documents that are returned in query responses. It does not improve the passages or answers that are returned per document. If you're using passage retrieval and your test results are returning good documents, but the wrong passages from the documents, relevancy training will not help.

For more information about when to use relevancy training, read the [Relevancy training for time-sensitive users](#) blog post on Medium.

How fields are handled

When you train a project from the product user interface, the results are always taken from the `text` field of the documents. If your documents don't have a `text` field, use the API to train your project instead. Your documents might not have a `text` field if you uploaded a CSV file that doesn't have a column named `text`, or uploaded a JSON file that doesn't have an object named `text`, or if you used the Smart Document Understanding tool to define fields with other names in which the bulk of the content of your documents now are stored.

When you train a project from the API, results are taken from all of the root-level fields and they are all considered to have equal significance. Unlike Discovery Query Language queries, with natural language queries you cannot specify which fields from the document you care about or how much significance to give to each one. When you teach Discovery with examples, the service figures out for you how much weight to give to each field.

Discovery builds a model that assigns different weights to term, bigram, and skip-gram matches for each of the root-level fields and balances them against matches from all of the other document fields. With enough examples, Discovery can return better answers because it knows where the best answers are typically stored.



Note: Relevancy training cannot be used to give more weight to nested fields. Nested fields are grouped and assigned one overall score. No matter how much you train, Discovery never gives a nested field more weight than it gives to a root-level field. For more information about nested fields, see the [FAQ](#).

Training a project

The training data that is used to train the relevancy model includes these parts:

- A natural language query that is representative of a query that your users might submit
- Results of the query which are returned by the service
- The rating that you apply to the result that indicates whether the result is `relevant` or `not relevant`

To apply relevancy training to a project, complete the following steps:

1. Go to the **Improve and customize** page. On the **Improvement tools** panel, select **Improve relevance**, then **Relevancy training**.
2. Enter a natural language query in the **Enter a question to train** field.

Do not include a question mark in your query. Use the same wording as your users. For example, `IBM Watson in healthcare`. Write queries that include some of the terms that are mentioned in the target answer. Term overlap improves the initial results when the natural language query is evaluated.

3. Click **Add+**.
4. Click **Rate results**.
5. After the results are displayed, assess each result, and then select **Relevant** or **Not relevant**, whichever option applies given the quality of the result.

When you select **Relevant**, you apply a score of `10` to the result. **Not relevant** applies a score of `0`. You can use a different scoring scale if you use the API to rate results, but you can't mix scoring scales within the same project.

If the result shows the message, "No content preview available for this document", it means that the document that was returned does not contain a `text` field or that its `text` field is empty. If none of the documents in your collection have a `text` field, use the API to train the project instead of training it from the product user interface.

6. When you are finished, click **Back to queries**.

7. Continue adding queries and rating them.

As you rate results, your progress is shown. Check your progress to see when enough rating information is available to meet the training threshold needs. Your progress is broken into the following tasks:

- Add more queries
- Rate more results
- Add more variety to your ratings

You must evaluate at least 50 unique queries, maybe more, depending on the complexity of your data. You cannot add more than 10,000 training queries.

8. You can continue adding queries and rating results after you reach the threshold. Enter all of the queries that you think your users will ask.

To delete a training query, click the **Delete** icon. To delete all of the training queries in your collection at one time, use the API. For more information, see [Delete training queries](#).

 **Note:** If two or more users attempt to train identical queries at the same time, the ratings that are submitted by one of the users overwrites the others.

Testing and iterating on the relevancy of results

When you are done rating results, and training is completed, test to see whether your query results are better. To do so, run test natural language queries that are related (but not identical) to your training queries. Review the results.

If you want to continue to improve the results after testing, you can:

- Add more documents to your collection.
- Add more training queries.
- Rate more results, making sure to use both the **Relevant** and **Not relevant** ratings.

Confidence scores

Discovery returns a **confidence** score for natural language queries of trained collections. This **confidence** score is not interchangeable with **confidence** scores that are returned by untrained collections.

The **confidence** score can range from **0.0** to **1.0**. The higher the number, the more relevant the result.

The **confidence** score can be found in the query results, under the **result_metadata** for each document. This number is calculated based on how relevant the result is estimated to be, compared to the trained model.

```
{  
  "matching_results": 4,  
  "retrieval_details": {  
    "document_retrieval_strategy": "trained"  
  },  
  "results": [  
    {  
      "id": "eea16dfd5fe6139a25324e7481a32f89_13",  
      "result_metadata": {  
        "confidence": 0.08793  
      }  
    }  
  ]  
}
```

The **document_retrieval_strategy** can be found under the **retrieval_details**. If you query a trained collection by using the Discovery Query Language, or the trained model is temporarily disabled, the **document_retrieval_strategy** is **untrained**.

For more information on querying a project, see the [Query overview](#).

Relevancy training limits

The following limits apply to relevancy training models:

- One model per project
- 10,000 queries per model
- 40 models per service instance for Enterprise and Premium plans; 20 models for Plus plan instances

Other ways to improve relevancy

If you prefer to use the Discovery API to train Discovery, see the [API reference](#).

You also can use the API to add curations. Curations is a beta feature that you can use to teach Discovery to return a specific document every time a certain query is submitted. For more information, see [Curations](#).

Adding a custom stopwords list can also improve the relevance of results for natural language queries. For more information, see [Identifying words to ignore](#).

Understanding relevancy training

Answers to common questions about training a project.

How do I know whether my system is trained?

Run a natural language query and check the `document_retrieval_strategy`. See [confidence scores](#).

If you are using the API, see [List training queries](#).

How long does it take to train a model?

It can take between 45 minutes to an hour for the training to finish. The duration of the training differs depending on the amount and variety of the data that is used to train the relevancy model. Also, the training occurs asynchronously. It can be delayed if other data that it needs is unavailable because it is being searched or processed in some other way.

How do I stop relevancy training from being applied to my project?

Use the API to delete the relevancy model that is associated with your project. To delete the model, you delete that training data that is associated with the ranker model. For more information, see [Deleting training queries](#).

Does relevancy training impact passage search?

No. Relevancy training is used for document search only. It has no impact on passage search.

Does relevancy training impact answer finding?

Not directly. Relevancy training indirectly impacts answer finding because it changes the order of the documents from which answers are retrieved. It reranks the returned documents from most to least relevant.

How do I check errors and warnings?

Open the [Manage collections](#) page. Choose your collection, then open the [Activity](#) tab.

How do I interpret the confidence score that appears in natural language query results after training?

See [confidence scores](#).

Interpreting relevancy training errors and warnings

The following list has explanations for some common error and warning messages.

Warning: `Invalid training data found: The document was not returned in the top 100 search results for the given query, and will not be used for training`

This warning occurs when the `document_ids` in your training data do not match the `document_ids` in a search that is performed against the collection. Check your queries to make sure that the `document_id` of the document you are rating is returned in the top 100 results for that query. If it is not, then you might want to check two things:

- If the document is not returned in the top 100, it might not be an example of a high-quality result. Reevaluate whether to use the document.
- If the document is not returned at all, then review why it is not returned and see whether any text in the document matches portions of the query.



Note: This warning indicates that you might have one or more failed queries. It doesn't mean that the training cannot be completed.

Error: `Invalid training data found: Syntax error when parsing query`

A syntax error means that the query is invalid. Syntax errors can occur when you increase the complexity of the query by adding a filter to the natural language query, for example. Run the query against the collection outside of relevancy training by using the API. After

you confirm that the query is valid and returns results, you can add it as a relevancy training query.

Error: Training data quality standards not met: You will need additional training queries with labeled examples. (To be considered for training, each example must appear in the top 100 search results for its query.)

You need to add more training data to train successfully. You need at least 49 unique training queries at a minimum, and each one needs at least one rated document. Minimum does not mean optimal; the size of the collection and other factors can increase the number of training examples that are needed to meet the minimum.

Error: Training data quality standards not met: Insufficient number of unique training queries. Expected at least n, but found m.

To meet the minimum training requirements, you need at least 50 unique training queries, and each query must have at least one rated document. If you have more queries than the minimum and are still receiving this error message, check your notices for other errors.

Error: Training data quality standards not met: No documents found with non-zero relevance labels.

Training data needs enough labeled data that specifies what documents are high value. Therefore, you need to rate some documents with nonzero values. You need to rate some documents as **Relevant** and some as **Not relevant**. At least one document must be rated **Relevant**.

Error: Training data quality standards not met: Training examples have no relevance label variety for X queries.

One of the requirements for training is to have sufficient label diversity. At least 25% of the training queries must include both **Relevant** and **Not relevant** labels. If you use the API, at least 25% of the queries must include two different numeric labels.

Default query settings

Learn about how the search query is configured for each project type by default.

When you submit a search from the product user interface, your text is passed as a natural language query value to the Query API. Other parameters that you can define when you use the API are assigned default values for queries that are made from the user interface. The following tables explain which values are specified by default for each project type. For more information about the Query API, see [Query reference](#).

You can override some of the default values by using improvement tools in the user interface. For example, you can use the **Search results** tool to change parameters such as **passages.enabled**. For more information, see [Changing the result content](#).

The enrichments that are applied to your data automatically differ by project type. For more information, see [Default project settings](#).

Default query settings

Query default	Document Retrieval	Document Retrieval for Contracts
aggregation	[term(enriched_text.entities.text,name:entities)]	[term(enriched_html.contract.elements.categories.label,count:25]
count	10	10
highlight	false	false
passages.characters	200	200
passages.count	10	10
passages.enabled	true	true
passages.fields	["text", "title"]	["text", "title"]
passages.find_answers	false	false
passages.max_answers_per_passage	1	1
passages.max_per_document	1	1

passages.per_document	true	true
return	[]	[]
spellingSuggestions	false	true
sort	""	""
table_results.count	10	10
table_results.enabled	false	true
table_results.per_document	0	0

Default query settings

Default query settings continued

Query default	Conversational Search	Content Mining	Custom
aggregation	[]	[]	[]
count	10	10	10
highlight	false	false	false
passages.characters	200	200	200
passages.count	10	10	10
passages.enabled	true	false	true
passages.fields	["text", "title"]	["text", "title"]	["text", "title"]
passages.find_answers	false	false	false
passages.max_answers_per_passage	1	1	1
passages.max_per_document	1	1	1
passages.per_document	true	true	true
return	[]	[]	[]
spellingSuggestions	false	true	true
sort	""	""	""
table_results.count	10	10	10
table_results.enabled	false	false	false
table_results.per_document	0	0	0

Default query settings continued

Project component settings

Default	Document Retrieval	Document Retrieval for Contracts
Aggregations	See table	See table

autocomplete	true	true
fields_shown.body.field	""	""
fields_shown.body.use_passage	true	true
fields_shown.title.field	"title"	"title"
results_per_page	5	5
structured_search	false	false

Project component settings

Project component settings continued



Note: The Custom project type has no project component default settings.

Default	Conversational search	Content Mining
Aggregations	[]	[]
autocomplete	false	true
fields_shown.body.field	""	text
fields_shown.body.use_passage	true	false
fields_shown.title.field	"title"	"document_id"
results_per_page	0	0
structured_search	false	false

Project component settings continued

Document Retrieval project aggregations

aggregations.name	aggregations.label	aggregations.multiple_selections_allowed
"name": "entities"	"label": "Top Entities"	"multiple_selections_allowed": false
"name": "_system_collections"	"label": "Collections"	"multiple_selections_allowed": true

Document Retrieval project aggregations

Document Retrieval for Contracts project aggregations

aggregations.name	aggregations.label	aggregations.multiple_selections_allowed
"name": "categories"	"label": "Category"	"multiple_selections_allowed": true
"name": "natures"	"label": "Nature"	"multiple_selections_allowed": false
"name": "contract_terms"	"label": "Contract Term"	"multiple_selections_allowed": false
"name": "contract_payment_terms"	"label": "Contract Payment Term"	"multiple_selections_allowed": false

"name": "contract_types"	"label": "Contract Type"	"multiple_selections_allowed": false
"name": "contract_currencies"	"label": "Contract Currency"	"multiple_selections_allowed": false
"name": "invoice_buyers"	"label": "Invoice Buyer"	"multiple_selections_allowed": false
"name": "invoice_suppliers"	"label": "Invoice Supplier"	"multiple_selections_allowed": false
"name": "invoice_currencies"	"label": "Invoice Currency"	"multiple_selections_allowed": false
"name": "po_payment_terms"	"label": "Purchase Order Payment Term"	"multiple_selections_allowed": false
"name": "po_buyers"	"label": "Purchase Order Buyer"	"multiple_selections_allowed": false
"name": "po_suppliers"	"label": "Purchase Order Supplier"	"multiple_selections_allowed": false
"name": "po_currencies"	"label": "Purchase Order Currency"	"multiple_selections_allowed": false

Document Retrieval for Contracts project aggregations

Enriching your data

Choose enrichments

Add resources that can teach Discovery about terms or patterns that have special meaning to your application.

The following table describes the best resources to add to address different needs.

Goal	Resource	Notes
Define categories by which text in your documents can be classified.	Classifier	N/A
Recognize terms and synonyms for terms that are significant to you, such as the names of products that you sell.	Dictionary	Term suggestions are displayed if the <i>Part of Speech</i> enrichment is applied to the collection.
Define regular expressions that capture patterns of significance, such as that AB10045 is the syntax that is used for your order numbers.	Regular expressions	N/A
Recognize and tag entities and relationships that are defined in a custom machine learning model.	Machine learning models	Requires a model that is built and exported from another IBM tool.
Apply rules to fields that are based on rules you defined by creating an advanced rules model in IBM Watson® Knowledge Studio.	Advanced rules models	Requires an advanced rules model that is built and exported from IBM Watson® Knowledge Studio or that uses an exported Patterns resource.
IBM Cloud Recognize terms that are mentioned in sentences that match a syntactic pattern that you teach Discovery to recognize.	Patterns (beta)	Available as a beta feature for English-language collections in managed deployments only. The enrichment that is derived by defining patterns cannot be applied to Content Mining projects. You can export the resource and use it as an advanced rules model.
Recognizes entities that you identify as being significant by training an entity extractor machine learning model.	Entity extractor	Supports starting from an imported Knowledge Studio corpus.

Domain tools overview

Alternatively, you can apply built-in Watson NLP enrichments that find the following information in your collection:

- [Entities and keywords](#)
- [Sentiment](#)

You can extract meaning from documents based on the document structure by defining a Smart Document Understanding (SDU) model. Use the Smart Document Understanding tool to identify new fields by which to target enrichments or to split large documents into more manageable chunks. For more information, see [Structural meaning with SDU](#).

Dictionaries and classifiers that you add to one project can be used by other projects.

For more information about how to get the most from enrichments, read the [Enriching your documents can make search more effective](#) blog post.

Choosing the right enrichment type

The following diagram helps you to choose the right enrichment for your use case.

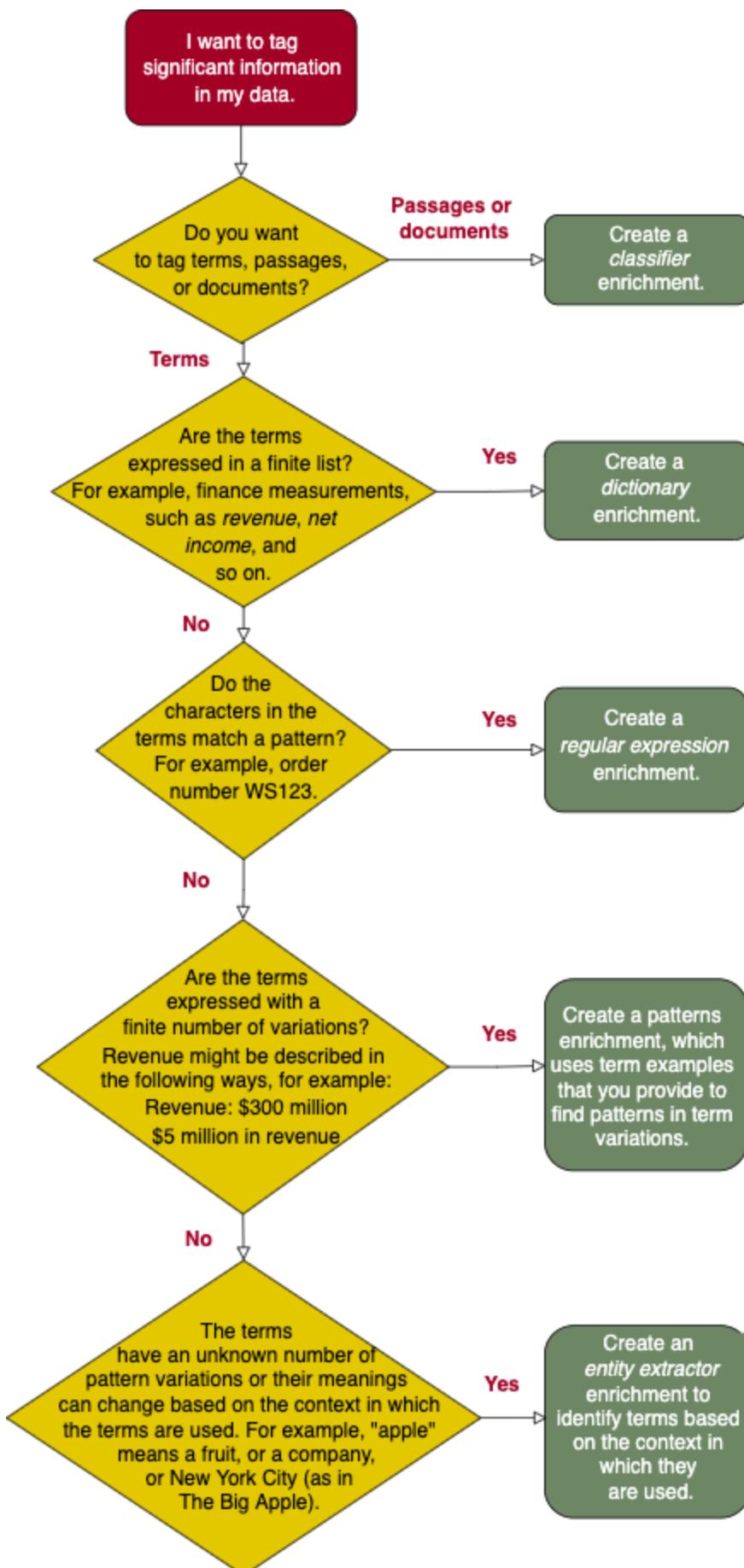


Figure 1. Flow diagram for choosing the right enrichment

Using enrichments together

You can use many enrichments together to tackle various challenges that you might encounter as you develop a search application.

Many teams start by creating a **dictionary** enrichment. Dictionaries are a great tool for identifying important terms and tagging them so they can be retrieved later. Let's say you're building a search application that needs to extract ingredients from recipes. A dictionary enrichment can recognize mentions of most ingredients. However, the dictionary enrichment might partially match against two-word terms. For terms such as **olive oil** or **mustard greens**, it might incorrectly recognize only **olive** and **mustard**. To improve the accuracy of the search, you can augment the dictionary enrichment with a **pattern** enrichment that can recognize two-word ingredient mentions. Maybe a few recipes mention food coloring codes in European format (**E104**). You can add a **regular expression** enrichment to recognize occurrences of codes with the syntax **E1nn**. Finally, to catch terms that no other enrichment can recognize, you can use a **machine learning** enrichment. The enrichment can be one that you build in an external tool and import to Discovery or one that you build in Discovery by creating an **entity extractor** enrichment.

The entity extractor enrichment is more sophisticated than the other enrichments. For example, a dictionary enrichment recognizes only exact matches of dictionary terms and synonyms that occur in your documents. A regular expression enrichment recognizes only specific patterns. In contrast, occurrences of an entity are recognized based on the context in which an entity example is mentioned in a sentence.

For example, maybe you want to recognize locations and the document you want to process contains the following types of sentences:

- I live in **Massachusetts**.
- We're traveling from **New York City** to **Paris** next week.

To use a dictionary enrichment to recognize location names successfully, the dictionary must list every possible location. However, if

When you use an entity extractor enrichment, you can identify when a location is mentioned based on how the location is referenced in a sentence. With phrases such as, "I live in **x**" or "I'm from **x**" or "I'm traveling to **x**" in its training data, the entity extractor can learn that **x** is a reference to a location.

When you need to choose between using a dictionary or an entity extractor enrichment, follow these guidelines:

- If the list of possible examples is short, use a dictionary.

It is more efficient to define a dictionary term **planet** with synonyms such as **Earth** and **Saturn** than to create a **planet** entity because only 8 planets exist in our solar system. However, defining a list of every possible location on Earth is not feasible. An entity extractor can recognize more location mentions.

- If the list of possible examples is static, use a dictionary.

Controversy over Pluto aside, the **planet** category is a good example here too because the list of planets in our solar system is static. Or maybe you want to monitor general customer sentiment about your products. You need to be able to recognize product name mentions, but might not need specifics. If you have a large variety of product names, you can create a **product name** entity. As new products are added to your portfolio, or product names change over time, you do not need to maintain an overall product list. The entity extractor can continue to recognize general feedback about your products based on the context of the sentences in which products are mentioned.

Add a resource

When you add a custom enrichment to a project it is available to any collection in the project.

To add a resource, complete the following steps:

1. Open your project and go to the **Improve and customize** page.
 2. On the **Improvement tools** panel, expand **Teach domain concepts**, and then choose the resource that you want to add.
- After you create the resource, it becomes a new type of enrichment that you can apply to your data.
3. Specify the collection and field in which to apply the enrichment.

 **Tip:** You can apply enrichments to the **text** and **html** fields, and to custom fields that were added from uploaded JSON or CSV files or from the Smart Document Understanding (SDU) tool. Only the first 50,000 characters of a custom field from a JSON file are enriched.

For example, if you add a dictionary and choose to apply it to the **text** field of a collection, the documents in the collection are reprocessed. If the term **vehicle** is specified as a synonym of the **car** dictionary entry and occurs in the document text, **vehicle** is tagged as a mention of the **car** dictionary entry type. If a customer later searches for **car**, the passage that contains the **vehicle** mention is included in the search results.

 **Note:** If the field that you choose comes from a JSON file, after you apply the enrichment, the field data type is converted to an array. The field is converted to an array even if it contains a single value. For example, `"field1": "Discovery"` becomes `"field1": ["Discovery"]`.

You can choose to apply resource-derived enrichments to your data later. Enrichments that you add to a project are available for use from any collection in the project. Go to the **Manage collections** page, choose the collection where you want to apply the enrichment, and then open the **Enrichments** tab. Make sure the status of the enrichment shows that it is **Ready**, and then apply the enrichment to a field in the collection. Enrichments that you enable are applied to the documents in random order. For more information, see [Managing enrichments](#).

From the deployed Content Mining application, you can create a classifier or a custom annotator from a dictionary, regular expression, machine learning, or PEAR file and use it as an enrichment in collections that are stored in other project types. For more information, see [Adding facets](#).

Use built-in Watson NLP to find common terms

Take advantage of award-winning Watson Natural Language Processing (NLP) capabilities by adding prebuilt enrichments to your documents.

With Watson NLP, you can identify and tag meaningful information in your collections so you can understand what it all means and make more informed decisions.

The following Watson NLP enrichments are available:

- [Entities](#): Recognizes proper nouns such as people, cities, and organizations that are mentioned in the content.
- [Keywords](#): Recognizes significant terms in your content.

- [Part of Speech](#): Identifies the parts of speech (nouns and verbs, for example) in the content.
- [Sentiment](#): Understands the overall sentiment of the content.

The following other pretrained enrichments are available with Discovery:

- [Contracts](#)
- [Document structure](#)
- [Table understanding](#)

Watson NLP enrichments

For example, the following screen capture shows a transcript of the US Declaration of Independence that was added to a Discovery collection where the Entities and Keywords enrichments are enabled. The mentions that are recognized by the enrichments are highlighted in the document text.

The screenshot shows the IBM Watson Discovery Premium interface. At the top, there's a navigation bar with 'IBM Watson Discovery Premium', 'My projects', 'Share feedback', 'Guided tours', and user icons. Below the navigation, the page title is 'Web crawl / Improve and customize / Declaration of Independence: A Transcription | Nation...'. On the left, a sidebar titled 'Identified elements' lists 'Top entities' and 'Keywords'. The main content area displays the text of the Declaration of Independence. Several words are highlighted in blue, including 'Laws of Nature and of', 'Nature's God', 'unalienable Rights', 'Life, Liberty and the pursuit of Happiness', 'Governments are', 'instituted among Men', 'deriving their just powers from the consent of the governed', 'That whenever any Form of Government becomes destructive of these ends', 'the Right of the People to alter or to abolish it', 'to institute new Government', 'such principles and organizing its powers in such form', 'Safety and Happiness', 'Prudence, indeed, will dictate that Governments long established', 'should not be changed for light and transient causes', 'all experience hath shewn', 'mankind are more disposed to suffer', 'while evils are sufferable, than to right themselves by abolishing the forms to which they are accustomed', 'But when a long train of abuses and usurpations, pursuing invariably the same Object evinces a design to reduce them under absolute Despotism', 'it is their right, it is their duty, to throw off such Government, and to provide new Guards'. To the right of the text, a 'Matches found' section lists various entities with their counts: Organization (~ - of 28), Independence (~ - of 2), Congress (~ - of 1), Framers of the Constitution (~ - of 2), Constitution (~ - of 2), Laws of Nature (~ - of 1), Rights (~ - of 3), Declaration (~ - of 6), and America (~ 2 of 2). The 'Top entities' dropdown in the sidebar is currently set to 'Top entities'.

Figure 1. Excerpt of the US Declaration of Independence with highlighted terms

Some of the NLP enrichments are applied to projects automatically. You don't need to apply them yourself if you are using one of these project types.

Default enrichments per project type

Some prebuilt enrichments are applied automatically to collections in a project based on the project type. The following table shows the default enrichments that are applied to each project type.

Enrichment	Document Retrieval	Document Retrieval for Contracts	Conversational Search	Content Mining
Contracts	✓			
Entities	✓	✓		
Keywords				
Part of Speech	✓	✓	✓	✓
Sentiment of Document				
Table Understanding	✓			

Default enrichments per project type

For more information about the following prebuilt enrichments, see the following topics:

- [Contracts](#)
- [Table Understanding](#)

For more information about how to create custom enrichments, see [Adding domain-specific resources](#).

For more information about how to get the most from enrichments, read the [Enriching your documents can make search more effective](#) blog post.

For more information about how to apply enrichments by using the API, see [Applying enrichments by using the API](#).

Add enrichments

To add an NLP enrichment, complete the following steps:

1. Open your project and go to the **Manage collections** page.
2. Click to open the collection that you want to enrich.
3. Open the **Enrichments** tab.
4. Scroll to find the NLP enrichment that you want to apply to your documents.



Note: Both built-in enrichments and custom enrichments are listed. Built-in enrichments have a type value of **System**.

5. Choose one or more fields to apply the enrichment to.

You can apply enrichments to the **text** and **html** fields, and to custom fields that were added from uploaded JSON or CSV files or from the Smart Document Understanding (SDU) tool.

6. Click **Apply changes and reprocess**.

Enrichments that you enable are applied to the documents in random order. For information about how to remove an enrichment, see [Managing enrichments](#).

Entities

Identifies entities. **Entities** are terms that typically represent proper nouns such as people, cities, and organizations that are mentioned in the data collection. Discovery can recognize entities that are part of an entity type system that is defined by the Watson Natural Language Processing (NLP) service.

If you want to be able to identify uncommon terms that are significant to your business, you can train your own model to recognize custom entities. For more information, see [Entity extractor](#).

The Watson NLP entity extractor service that is used by Discovery is called the **NLU type system**. The name originates from the fact that the type system is used by the Watson Natural Language Understanding (NLU) service in addition to the Watson Discovery service. However, it is the Watson NLP implementation of the type system that is used directly by Discovery, not the Watson NLU implementation. As a result, the two implementations can produce different results. To get a general idea of the types of entities that are recognized by the service, see [Entities](#).

The following screen capture shows that the Entities enrichment recognizes the terms **Systems of Government** and **King of Great Britain** (among others) and tags them as entity mentions.

The screenshot shows the IBM Watson Discovery Premium interface. On the left, there's a sidebar with a tree view and a search bar labeled "Web crawl / Improve and customize /". The main content area has a title "[← Declaration of Independence: A Transcription | Nation...](#)". On the left, a panel titled "Identified elements" lists categories like "Organization (28)", "Location (17)", etc. The main text area contains several paragraphs of the Declaration of Independence. In the top right, a panel titled "Matches found" shows a snippet of text with highlighted entities: "Systems of Government", "King of Great Britain", and "Great Britain". Below this, it says "Select arrows to find related matches based on your filtered elements." and "Organization 15 of 28".

Figure 2. The recognized entities, Governments and King of Great Britain, are highlighted

From the JSON view of the document, you can see the underlying JSON structure of the entity mentions.

```

    ↳ 30 : { ... } 4 items
    ↳ 31 : { ... } 4 items
    ↳ 32 : { 4 items
      "model_name" : "natural_language_understanding"
      ↳ "mentions" : [ ... ] 1 item
      "text" : "Systems of Government"
      "type" : "Organization"
    }
    ↳ 33 : { 4 items
      "model_name" : "natural_language_understanding"
      ↳ "mentions" : [ ... ] 1 item
      "text" : "King of Great Britain"
      "type" : "Organization"
    }
    ↳ 34 : { ... } 4 items
  
```

Figure 3. JSON representation of recognized entity mentions

If you want to search for the Organization entity type, for example, you can copy all of the JSON content into a text editor and search for **Organization**. Click the **Copy** icon from the root of the JSON tree view.

Example

Input

```
"IBM is an American multinational technology company headquartered in Armonk."
```

Response

In the JSON output:

- **text** = string. The entity text
- **type** = string. The entity type, such as **Organization**, **Location**, **Person**, **Number**.
- **mentions** = array. The entity mentions and locations
- **model_name** = string. For custom models, this field contains the user-provided model name. Otherwise, this field contains the default name of the model, such as **watson_knowledge_studio**, **dictionary**, **character_pattern**, or **natural_language_understanding**

```
{
  "entities": [
    {
      "model_name": "natural_language_understanding",
      "mentions": [
        {
          "confidence": 0.8317045,
          "location": {
            "end": 3,
            "begin": 0
          },
          "text": "IBM"
        }
      ]
    }
  ]
}
```

```

        },
        "text": "IBM",
        "type": "Organization"
    },
    {
        "model_name": "natural_language_understanding",
        "mentions": [
            {
                "confidence": 0.6114863,
                "location": {
                    "end": 75,
                    "begin": 69
                },
                "text": "Armonk"
            }
        ],
        "text": "Armonk",
        "type": "Location"
    }
]
}

```

Entity limits

The Entities enrichment can identify up to 50 entities, each with one or many mentions, per document.

Keywords

Returns important keywords in the content.

For example, the following screen capture shows highlighted terms from the US Declaration of Independence that are recognized by the Keywords enrichment.

The screenshot shows the IBM Watson Discovery Premium interface. The top navigation bar includes 'IBM Watson Discovery Premium', 'My projects', 'Share feedback', 'Guided tours', and help icons. The main content area displays the 'Declaration of Independence: A Transcription | Nation...' document. On the left, a sidebar titled 'Identified elements' lists terms like 'America', 'people', 'Constitution', 'Congress', and 'Independence'. The central text area shows the historical document with several words highlighted in blue, such as 'In', 'Congress', 'July', '4', '1776', 'the', 'unanimous', 'Declaration', 'of', 'the', 'thirteen', 'United', 'States', 'of', 'America', 'When', 'in', 'the', 'Course', 'of', 'human', 'events', 'it', 'becomes', 'necessary', 'for', 'one', 'people', 'to', 'dissolve', 'the', 'political', 'bands', 'which', 'have', 'connected', 'them', 'with', 'another', 'and', 'to', 'assume', 'among', 'the', 'powers', 'of', 'the', 'earth', 'the', 'separate', 'and', 'equal', 'station', 'to', 'which', 'the', 'Laws', 'of', 'Nature', 'and', 'of', 'Nature's', 'God', 'entitle', 'them', 'a', 'decent', 'respect', 'to', 'the', 'opinions', 'of', 'mankind', 'requires', 'that', 'they', 'should', 'declare', 'the', 'causes', 'which', 'impel', 'them', 'to', 'the', 'separation.', 'We', 'hold', 'these', 'truths', 'to', 'be', 'self-evident', 'that', 'all', 'men', 'are', 'created', 'equal', 'that', 'they', 'are', 'endowed', 'by', 'their', 'Creator', 'with', 'certain', 'unalienable', 'Rights', 'that', 'among', 'these', 'are', 'Life', 'Liberty', 'and', 'the', 'pursuit', 'of', 'Happiness.--That', 'to', 'secure', 'these', 'rights', 'Governments', 'are', 'instituted', 'among', 'Men', 'deriving', 'their', 'just', 'powers', 'from', 'the', 'consent', 'of', 'the', 'governed', '--That', 'whenever', 'any', 'Form', 'of', 'Government', 'becomes', 'destructive', 'of', 'these', 'ends', 'it', 'is', 'the', 'Right', 'of', 'the', 'People', 'to', 'alter', 'or', 'to', 'abolish', 'it', 'and', 'to', 'institute', 'new', 'Government', 'laying', 'its', 'foundation', 'on'. On the right, a 'Matches found' section lists terms with counts: Laws of Nature (1), Rights (3), Declaration (6), America (2), Constitution (2), Congress (1), Independence (2), truths (1), people (4), July (1).

Figure 4. Terms recognized by the Keywords enrichment

From the JSON view of the document, you can see the underlying JSON structure of the `Declaration` keyword mention.

```

    "enriched_text": [ 1 item
      0 : { 1 item
        "keywords": [ 50 items
          0 : {...} 3 items
          1 : {...} 3 items
          2 : {...} 3 items
          3 : {...} 3 items
          4 : {...} 3 items
          5 : {...} 3 items
          6 : {...} 3 items
        ]
        7 : { 3 items
          "mentions": [...] 6 items
          "text": "Declaration"
          "relevance": 0.511093
        }
        8 : {...} 3 items
        9 : {...} 3 items
      ]
    ]
  ]
}

```

Figure 5. JSON representation of Keywords enrichment mentions

Example

Input

```
"Watson Discovery is an award-winning AI search technology."
```

Response

In the JSON output:

- **text** = The keyword text
- **mentions** = The entity mentions and locations

```
{
  "keywords": [
    {
      "mentions": [
        {
          "location": {
            "end": 157,
            "begin": 141
          },
          "text": "Watson Discovery"
        }
      ],
      "text": "Watson Discovery",
      "relevance": 0.503613
    },
    {
      "mentions": [
        {
          "location": {
            "end": 177,
            "begin": 164
          },
          "text": "award-winning"
        }
      ],
      "text": "award-winning",
      "relevance": 0.728722
    },
    {
      "mentions": [
        {
          "location": {
            "end": 198,
            "begin": 181
          },
          "text": "search technology"
        }
      ],
      "text": "search technology",
      "relevance": 0.779356
    }
  ]
}
```

Keywords limits

The Keywords enrichment can identify up to 50 keywords, each with one or many mentions, per document.

Part of speech

Recognizes and tags parts of speech, including nouns, verbs, adjectives, adverbs, conjunctions, interjections, and numerals.

Identify custom terms

Define a finite set of terms with a dictionary

Recognize terms and synonyms for terms that are significant to you, such as the names of products that you sell.

Help Discovery find terms that have meaning to your use case by adding a dictionary. You can define multiple synonyms for a term or a set of words in the same category.

You can create a dictionary by adding the terms one by one or by uploading a CSV file that lists the terms.

To add dictionary terms one by one, complete the following steps:

1. From the **Teach domain concepts** section of the **Improvement tools** panel, choose **Dictionaries**.
2. Click **New**.
3. Name your dictionary.

For example, **Transportation**.

4. Choose the language. A dictionary can contain terms in only one language.
5. **Optional:** Expand **Advanced options**, and edit the facet name for the dictionary.

Facets are used to categorize documents. A user can choose a facet type to narrow their search results. The dictionary name in lowercase is used as the facet name by default. You might want to change the facet to be uppercase.

6. Enter a term, and then select the **+** button to add it.

For example **vehicle** and **engine**.

In English dictionaries, specify the dictionary terms in lowercase. Only use uppercase if you want Discovery to ignore lowercase mentions of the term when they occur in text. When terms are analyzed to determine whether they are occurrences of the dictionary enrichment, the surface form of the term with uppercase match is used. For example, a **vehicle** entry in the dictionary results in annotations for **vehicle**, **Vehicle**, or **VEHICLE** mentions when they occur in text. For a **Sat** entry in the dictionary, annotations are added for **Sat** or **SAT**, but not for **sat**.

Dictionary matching is case-sensitive for Arabic, Chinese, Korean, Japanese, and Hebrew.

7. To add synonyms for the term, click the **Edit** icon, and then enter synonyms in the **Other terms** field. Separate multiple synonyms with a comma. Click **Save term**.

The dictionary can contain terms and their synonyms or a category and terms that belong to the category.

For the term **vehicle**, you can specify synonyms such as **car**, **automobile**, **sedan**, **convertible**, **station wagon**, and so on. For **engine**, you can specify **gasket**, **carburetor**, **piston**, and **valves**.



Tip: Be careful not to add too many synonyms. Test the impact of any synonyms that you add. When you test, use data that is different from the data you use to derive the synonyms.

8. Continue adding terms.

Similar terms from all of the collections in the current project are suggested as new entries.



Note: Suggested terms are taken from the field to which the **Part of Speech** enrichment is applied. Suggestions are not displayed if the **Part of Speech** enrichment is not enabled.

9. Click **Save dictionary**.

10. Choose the collections and fields where you want to apply the dictionary, and then click **Apply**.

Example

A transportation dictionary is added to a project.

The screenshot shows the 'Transportation' dictionary configuration. At the top, there's a 'Dictionary name' field set to 'Transportation' and a 'Language' dropdown set to 'English'. Below this is an 'Advanced options' button, which is highlighted with a blue border. A 'Facet path' section follows, with 'Transportation' selected. A note below says: 'Used as a field value in the index. To indicate hierarchy, place a dot (.) between parent and child values.' The main area is titled 'Included terms' and contains a table. The table has columns: 'Base term', 'Other terms', and 'Actions'. It lists two entries: 'vehicle' with 'car, automobile, sedan, convertible, station wagon' and 'engine' with 'gasket, carburetor, piston, valves'. There are edit and delete icons in the 'Actions' column for each entry. At the bottom of the table are pagination controls: 'Terms per page: 10', '1–2 of 2 terms', '1 of 1 pages', and navigation arrows.

Figure 1. Transportation dictionary

The resulting facet that is created for the dictionary is displayed in the search page.

The screenshot shows a search results page with a facet sidebar on the left. The sidebar is titled 'Top Entities' and has a dropdown menu with 'Transportation' selected, which is highlighted with a blue border. Below the dropdown are two checkbox options: 'vehicle' and 'engine'.

Figure 2. Transportation facet

The document where the enrichment is applied contains the following sentence:

Some car fluids can be acidic, such as battery fluid.

The following JSON snippet illustrates how a Transportation dictionary enrichment mention is stored when the term `car`, which is a synonym for the `vehicle` dictionary entry, is found in the document. In this collection, the dictionary enrichment is applied to the `text` field, so the mention is listed in the `entities` array that is in the `enriched_text` array.

```
{
  "enriched_text": [
    {
      "entities": [
        {
          "model_name": "Dictionary:Transportation",
          "mentions": [
            {
              "confidence": 1,
              "location": {
                "end": 91122,
                "begin": 91119
              },
              "text": "car"
            }
          ],
          "text": "vehicle",
          "type": "Transportation"
        }
      ]
    }
  ]
}
```

Uploading dictionary terms

To add dictionary from a CSV file, complete the following steps:

1. Create a CSV file that contains the dictionary terms that you want to add.

Use UTF-8 encoding. Specify one entry per line.

- To define a set of synonymous terms, use the following syntax:

```
<term>,<synonym>,<synonym>,<synonym>,...
```

For example:

```
vehicle,car,automobile,sedan,convertible,station wagon
```

The entry in this example creates a `vehicle` dictionary entry. When the dictionary enrichment is applied to a document, any mentions of `vehicle`, `car`, `automobile`, `sedan`, `convertible`, or `station wagon` are tagged as instances of the `vehicle` dictionary entry.

- To define a set of terms in the same category, use the following syntax:

```
<category>,<related-term>,<related-term>,...
```

For example:

```
engine,gasket,carburetor,piston,valves
```

The entry in this example creates an `engine` dictionary entry. When the dictionary enrichment is applied to a document, any mentions of `engine`, `gasket`, `carburetor`, `piston`, or `valves` are tagged as instances of the `engine` dictionary entry.

2. From the **Teach domain concepts** section of the **Improvement tools** panel, choose **Dictionaries**.
3. Click **Upload**.
4. Name your dictionary and choose the language that was used in the CSV file.
5. **Optional:** Expand **Advanced options**, and specify edit the facet name for the dictionary. Facets are used to categorize documents. A user can choose a facet type to narrow their search. The dictionary name in lowercase is used as the facet name by default. You might want to change the facet to be uppercase.
6. Click **Upload** to browse for the CSV file that you created earlier.
7. Click **Create**.
8. Choose the collections and fields where you want to apply the dictionary, and then click **Apply**.



Note: If you add a dictionary by using the Enrichment API, after you apply the API-generated dictionary enrichment to a field, the dictionary is displayed in the Dictionaries page. However, you cannot edit the API-generated dictionary from the dictionary tool in the product user interface.

To delete a dictionary, you must use the [Delete an enrichment](#) method of the Discovery v2 API.



Note: There is a limitation in how words with Hankaku (half-width) characters in Japanese are handled by the dictionary enrichment. When you create a dictionary enrichment in the Japanese language, you can use the Katakana or alphanumeric characters in the dictionary entry. However, when a Katakana word is used in the dictionary entry, the synonyms are handled with Zenkaku characters, except for the same Katakana word, which is represented by Hankaku characters. The Hankaku word is treated as a separate term from the Zentaku words. It is displayed as a separate facet, for example. Similarly, when an alphanumeric word is used in the dictionary entry, the synonyms are handled with Hankaku characters, except for the same alphanumeric word, which is represented by Zenkaku characters. The Zenkaku word is treated as a separate term from the Hankaku words.

Dictionary enrichments that you add to one project can be applied to collections in other projects in the same service instance. In fact, you can apply them to collections in a Content Mining project from the deployed Content Mining application.

Dictionary limits

The number of dictionaries and term entries you can create per service instance depends on your Discovery plan type.

Plan	Number of dictionaries per service instance	Number of term entries per dictionary	Number of terms for which suggestions can be generated
Cloud Pak for Data	Unlimited	Unlimited	1,000

Premium	100	10,000	1,000
Enterprise	100	10,000	1,000
Plus (includes Trial)	20	1,000	50

Dictionary plan limits

Define custom entities

Teach Discovery about terms that are significant to your business by creating an entity extractor.

An **entity extractor** is a machine learning model that recognizes and tags terms that you indicate are significant to your business need or use case. When you create an entity extractor, you get to decide the content and scope of information to find and extract. Your extractor can extract any of the following things:

- Terms that represent objects, such as vegetable names from cooking recipes or the make and model of cars from accident reports
- Attributes of objects, such as color and quantity
- Short phrases, such as **107 deaths in France, revenue of \$343M**

An **entity type** is a type of thing. To create an entity extractor, you define a set of **entity types** that you care about. You then annotate a collection of your own documents by finding terms or phrases that represent the type of information you want to extract and labeling them as entity examples.

After you define entity types and label entity examples, you can generate a machine learning model. The model learns about the information that you care about based on how the terms or phrases that you label as examples are referenced in sentences. The model learns from the context and language with which the entity examples are referenced in the training data.

After the machine learning model is trained well enough to recognize your entity types, you can publish the model as an enrichment and apply the enrichment to new documents. The custom entity extractor enrichment recognizes and tags new mentions of the same and similar terms as occurrences of the entity types that you care about.

For more information about how to use the entity extractor to add domain customization to your AI applications, see the [Entity Extractor Feature in Watson Discovery v2](#) blog post.

Discovery also has a built-in **Entities** enrichment that can be applied directly to your collection. It doesn't require any training to recognize commonly-known proper nouns. For more information about the Watson NLP Entities enrichment, see [Entities](#).

You already built an entity type system in Knowledge Studio? You can use the corpus that is associated with your machine learning model as a starting point for your entity extractor training data. For more information, see [Importing a corpus](#).

For information about the languages with which the entity extractor can be used, see [Language support](#).

Entity extractor overview video

This video provides an overview of how to define custom entity types and then use them to extract terms of interest from your data.



[View video: Define custom entity types with Watson Discovery](#)

To read a transcript of the video, [open the video on YouTube.com](#), click the **More actions** icon, and then choose **Open transcript**.

Example

If you are familiar with the built-in Entities enrichment, you know that the enrichment can recognize terms that match generalized categories, such as **Person** and **Location**. With the entity extractor, you control what constitutes terms or phrases that are meaningful.

The following image shows the terms that an enrichment that recognizes **family members** entity type mentions might extract from text. The example illustrates how family member mentions and other entity mentions (that are recognized by the built-in Entities enrichment) both might be predicted.



Figure 1. Labeled entity examples

This excerpt comes from Chapter 3 of *Pride and Prejudice* by Jane Austen.

Before you begin

Find or create a collection with documents that have various examples of the entity types that you want Discovery to learn about. To teach the extractor, you must label examples of entity types. You can only label examples if your collection contains valid examples. Try to find documents that have many and varying terms that function as examples of every entity type that you want to define.

Adding an entity extractor

To add an entity extractor, complete the following steps:

1. Open the project where you want to create the entity extractor.

The project must have at least one collection with documents that are representative of your domain data.

2. From the *Improvement tools* panel of the *Improve and customize* page, expand **Teach domain concepts**, and then click **Extract entities**.

3. Click **New**.

If you want to create an entity extractor that is based on the entity type system from a IBM Watson® Knowledge Studio corpus, click the arrow, and then choose **Import a Knowledge Studio corpus**. For next steps, see [Importing a Knowledge Studio corpus](#).

4. Add an extractor name and optionally a description.

This name is used as the model name and as the name of the enrichment that is created when you publish the model. The name is displayed as the enrichment name in the Enrichments page where you and others can apply it to collections. It also is displayed as the model name in the JSON representation of documents where custom entities are found. The name is stored with the capitalization and spacing that you specify.

5. Choose a collection with documents that are representative of your domain data.
6. Choose fields from the document to show in the document view where you will label documents from the collection.

- **Document title** is shown in the page header as the document name. Choose a field that has a unique value per document, such as the file name, which is stored in the `extracted_metadata.filename` field.
- **Document body** is where you label entity examples. Choose a field that contains the bulk of the document content, such as the `text` field.

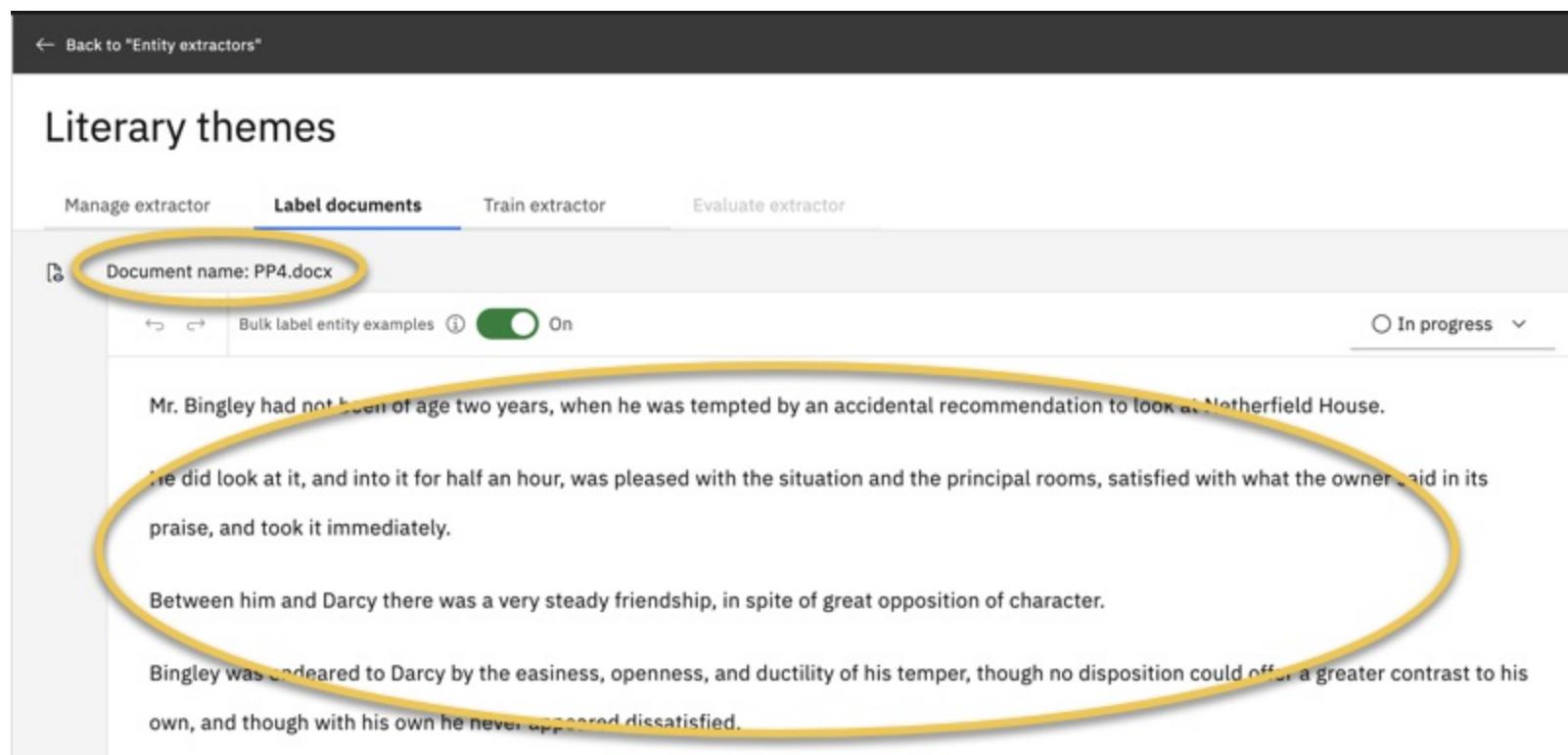


Figure 2. Label documents page

7. Click **Create**.

A document from the collection that you selected is displayed in the **Label documents** view. You will label occurrences of the entity types that you want Discovery to recognize from this and other documents in the collection.

Tip: If no text is displayed in the body of the page, start over now by creating a new entity extractor. This time, when you select a value for the **Document body** field, be sure to choose a field from your processed documents that contains text.

Defining entity types

Define entity types by completing the following steps:

1. Click **Add an entity type**.

2. Add the entity type name and an optional description.

Use a naming convention that works for your data. The built-in Entities enrichment uses initial capitals and no spaces, for example **EmailAddress**. To distinguish your entities from entities that are extracted by other enrichments, you might want to use a different convention.

3. Optional: Pick the color to use for highlighting text in the document that you want to label as an example of this entity type.

You can click a color from the **Label color** palette, click the **Renew color** icon to tab from one color to the next. To use a custom color, specify its hexadecimal color code (#fff0f7).

4. Click **Create**.

5. Repeat this process to add all of the entity types that you want the extractor to recognize.

If you aren't sure what to add for entity types, it might help to review the documents in the collection first. By reviewing the content, you can get a feel for which terms have significant meaning and look for logical ways to group such terms.

Label significant terms

From the **Label documents** view, find terms of significance in the documents from your collection and label them to indicate their entity types.

Before you begin labeling documents, decide whether you want to keep bulk labeling enabled. The bulk label feature is a great way to speed up the process of labeling your documents. When enabled, every term that you label is labeled automatically everywhere it occurs in the document. Otherwise, you must label each occurrence of the term one at a time.

If you decide that you don't want to bulk label examples, set the **Bulk label entity examples** switch to **Off**. For more information, see [Labeling examples in bulk](#).

Labeling tips

Review these tips before you begin:

- The document collection that you label must contain a representative set of documents. The documents must have many and varied examples of the entity types that you want the entity extractor to recognize. If the collection you selected when you started to create the entity extractor does not meet the requirement, stop now and start over with a different document collection.
- Define entity types that are clearly distinct from one another.

- Aim to label at least 40 examples of each entity type.
- Label every valid example of an entity type. Do not skip any occurrences. To speed up the process, use the bulk label feature.

Labeling entity examples

Label terms in the document that represent examples of the entity types you defined. When you are done with one document, switch the document status from **In progress** to **Complete**, and then move on to the next document.

To label entity examples, complete the following steps:

1. Review the text of the document. Look for entity examples to label.

The following table shows some examples.

Entity type	Examples to label in the document
color	white, green, purple
car	convertible, SUV, sedan
auto_model	Explorer, Civic, Sorrento
auto_manufacturer	Ford, Honda, Kia
clothing	shirt, blouse, skirt
instruments	bonds, stocks, ETFs, munis
Entity types and examples	

If an entity type that you want to identify is not created yet, add the entity type. From the **Entity types** panel, click **Create new**. For more information about adding entity types, see [Defining entity types](#).

2. First, click the entity type from the **Entity types** panel.
3. In the document body, select the word or phrase that represents the entity example.

The term is selected and a color label is applied to the term. The first two characters of the entity type name are shown in uppercase superscript within the label boundary. Both the 2-character ID and the label color help you to associate the example with the entity type it represents.

It is a truth universally acknowledged, that a single man in possession of a good fortune,
 must be in want of a wife^{FA}.

Figure 3. A label is applied to an entity example

The example text is also added to the **Entity types** panel. If you click the chevron to view details, you can see that the example is listed. The example text is saved in lowercase, regardless of the capitalization that is used in the original text.

4. If bulk labeling is enabled, a notification is displayed to show the number of occurrences of the term that were found and labeled in the current document.
5. If you want to label occurrences of the term in all of the documents in the collection, click **Apply to all documents**.

When you enable this option, occurrences of the term are labeled in all of the documents in the collection, including documents that you already reviewed and marked complete.

You are asked to confirm the action because it cannot be undone. If you don't want to have to confirm the action every time you choose to apply bulk labeling to all documents, select **Do not ask for confirmation again**. Click **Run**.

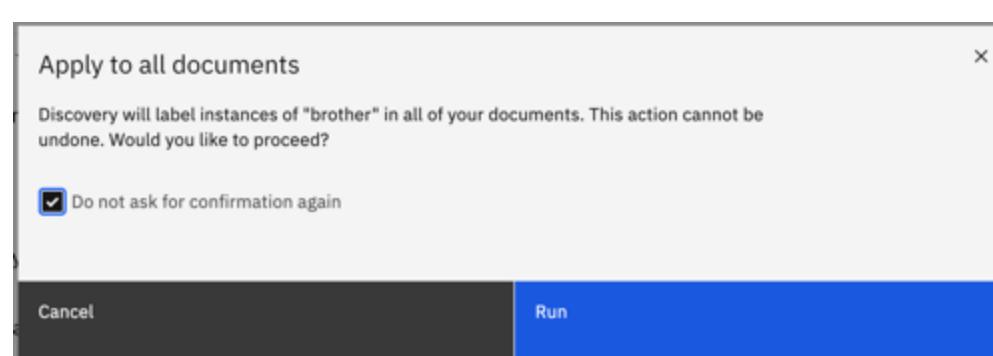


Figure 4. Bulk labeling configuration confirmation

For more information, see [Labeling examples in bulk](#).

6. Scroll through the document to label every valid example of every entity type that you want your extractor to recognize.

⚠️ Important: The machine learning model learns as much from the terms that you don't label as the terms that you do.

If you miss labeling a valid example, the model learns that when the term is used in that context, it is not a valid mention of the entity type. In some cases, an omission is appropriate. For example, some terms have different meanings in different contexts. You don't want to label the term when it is used in the wrong context. However, if the term is used in the right context and you don't label it, you are teaching the model to ignore it. You decrease the model's effectiveness when your training data is inconsistent.

After you label many examples, entity example suggestions are displayed. You can accept or reject entity example suggestions.

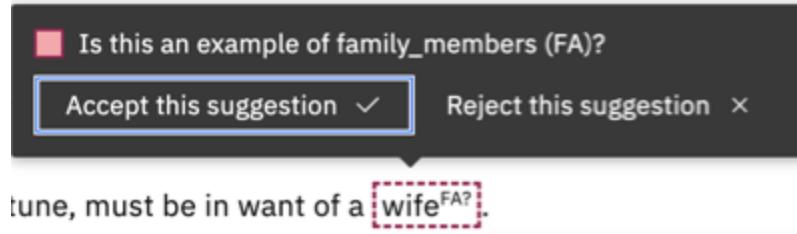


Figure 5. Decide whether to accept a suggestion

Accepting example suggestions is another way to speed up the labeling process. For more information, see [Entity example suggestions](#). After you accept a suggestion, you can bulk label the term.

7. If you make a mistake and label the wrong word or a word was labeled incorrectly by the bulk labeling process, you can delete the label.

Hover over the labeled word until the **Delete this example** option is displayed, and then click it. You can choose to delete only this mention or all of the mentions in the document. Make a choice, and then click **Delete**.

8. After you label all of the entity examples in the current document, change the document status from **In progress** to **Complete**.

Another document from the collection is displayed.

9. Label examples of your entity types in each document in the collection.

At any time during the labeling process, you can click **Save entity extractor** to save your work.

10. If you don't have enough examples in the current set of documents, you can add more documents.

From the **Document list** panel, click **Add documents**. The option is available only when more documents are available in the collection. You can add up to 20 documents. If bulk labeling for all documents is enabled, labels are applied to the newly-added documents automatically.

11. After you label examples in as many documents in the collection as you want, click **Save entity extractor**, and then open the **Train extractor** page.

Labeling examples in bulk

For most entity examples, enabling the bulk label feature is helpful. You might want to skip it if a term has more than one meaning in different contexts. In that case, evaluate each occurrence individually. Remember, if you enable the bulk label feature, you can check the accuracy of the labels that were added automatically and make corrections when necessary as you review the document.

After you enable the bulk label feature, a notification is displayed that indicates how many occurrences of an entity example were found in the current document. From the current page, the labeling tool cannot access other documents to report how many occurrences exist in other documents from the collection. However, the mention count is shown in the **Entity types** panel. When you first open other documents, you can check the mention counts to see how many mentions were labeled automatically.

Did the bulk label feature miss an occurrence?

Occurrences of the term are not labeled if they occur in the same phrase in which the term is already labeled. For example, the first occurrence of the term **husband** is not labeled when the bulk label feature is switched on for the second occurrence of the term in the following sentence.

"Your plan is a good one," replied Elizabeth, "where nothing is in question but the desire
of being well married; and if I were determined to get a rich **husband**, or any **husband^{FA}**,
I dare say I should adopt it."

Figure 6. Treatment of overlapping labels

Entity example suggestions

After you label enough examples, suggested entity type examples are displayed. The system learns from the types of examples you label, and applies what it learns to identify potential new examples. For example, after you label **red**, **orange**, **yellow**, **green**, and **blue** as examples of the **color** entity type, the **Example suggestions** panel might show **indigo** and **violet** as suggested examples for you to label. Suggestions are not displayed until after you label many examples of an entity type.

The following example shows suggestions that are made for family member mentions.

Suggestions for family members	
Entity examples	<input checked="" type="checkbox"/> <input type="checkbox"/>
✓ wives	<input checked="" type="checkbox"/> <input type="checkbox"/>
✓ daughters	<input checked="" type="checkbox"/> <input type="checkbox"/>
✓ father	<input checked="" type="checkbox"/> <input type="checkbox"/>

Figure 7. Entity example suggestions

You might notice that a term that you chose to bulk label is not labeled, but is displayed as a suggestion instead. A term is skipped in the following situations:

- The term might occur in different noun phrases in different sections of the document. For example, the term **father** might occur in the noun phrases **the kindest *father*** and **to her *father***. When a word is included in a noun phrase with adjectives, the meaning can change. Therefore, such terms sometimes are suggested rather than labeled automatically.
- A word might be a valid example on its own and as part of a multiple-word mention. For example, a mention of **IBM** might refer to the company **International Business Machines, Corp.** or might be used as part of a product name, such as **IBM Cloud Pak for Data**. However, a word or phrase can be part of only one example. Example labels cannot overlap one another. Therefore, you must choose which example suggestion is the most accurate. In this example, where the term **IBM** is used as part of a product name, it is more accurate to label the full phrase as an example of the **Product** entity type.
- The service might recognize that a term is a possible example of more than one entity type. For example, the word **top** might mean **the best** or might mean **shirt**.

To investigate a suggestion further, click it to see the word in context within the document. Seeing the term in context helps you to decide whether the occurrence is a valid entity example for you to label.

Importing a Knowledge Studio corpus



Note: For installed deployments, the import capability was added with the 4.6.2 release.

You can import a corpus of documents that were annotated in IBM Watson® Knowledge Studio to use as the training data for an entity extractor in Discovery.

Entity types that were defined in Knowledge Studio are shown as new entity types in Discovery. You can continue to annotate the imported documents when you customize the entity extractor model.

Entity subtypes and relations from the Knowledge Studio machine learning model are not represented, neither are any custom dictionaries that are associated with the model.

Before you can import a corpus, you must export the document set from Knowledge Studio as a .zip file. Follow the appropriate steps for exporting based on your Knowledge Studio deployment type:

- [IBM Cloud](#)
- [Cloud Pak for Data](#)

Although you must download both the document set and type system to include annotations in documents that you upload to another Knowledge Studio workspace, the same is not true in this use case. You import only the document set to Discovery. Any annotations in the documents are recreated in Discovery. The Knowledge Studio type system is not needed.

To import a Knowledge Studio corpus, complete the following steps:

1. Open the project where you want to import the corpus.
2. From the **Improvement tools** panel of the **Improve and customize** page, expand **Teach domain concepts**, and then click **Extract entities**.
3. Click the arrow that is associated with the **New** button, and then click **Import a Knowledge Studio corpus**.
4. Add an extractor name and optionally a description.

This name is used as the model name and as the name of the enrichment that is created when you publish the model. The name is displayed as the enrichment name in the Enrichments page where you and others can apply it to collections. It also is displayed as the model name in the JSON representation of documents where custom entities are found. The name is stored with the

capitalization and spacing that you specify.

5. Click **Upload**, and then browse to find and select the .zip file that you exported from Knowledge Studio. Click **Create**.

The annotated documents that you upload are stored with the entity extractor workspace, not as a new collection in the project. You can continue to annotate the documents.

Give Discovery some time to import and process the machine learning model corpus. After the entity extractor is created, the extractor is opened to the **Label documents** page.

Training the extractor

After you label documents, review the training data that will be used to train the entity extractor model.

To train the extractor, complete the following step:

1. Decide whether you want to apply an advanced option. Most models do not require changes to these options.

The following customizations are available from the **Review and finish** page:

- Include documents that were not reviewed by a person in the training set.

Typically, only documents that a person labeled, reviewed, and explicitly marked complete can be candidates for inclusion in the training set. However, if you want to allow documents that were not marked complete to be included in the training set, you can do so.

- Change the ratio of documents that are included in the document sets that comprise your training data.

The documents from your collection are split at random into the following sets:

- Training set: The documents that you label and that are used to train the entity extractor machine learning model. The goal of the training set is to teach the machine learning model about correct labels.
- Test set: The documents that are used to test the trained model. After you run a test, you can review the results, closely analyze areas where the model got something wrong, and find ways to improve the model's performance.
- Blind set: Documents that are set aside and used to test the model periodically after several iterations of testing and improvement are completed. The documents in the blind set are intentionally roped off. As you test the model with documents from the test set and analyze the results, you become familiar with the underlying test documents. Because the test documents are used iteratively to improve the model, they can start to influence the model training indirectly. That's why the blind set of documents is so important. The blind set gives you a way to generate an unbiased evaluation of the model periodically.

The default split applies a ratio (70%-23%-7%) that is commonly used for machine learning training.

2. Click **Train extractor**.

When you train the extractor, Discovery uses documents from the training set to build a machine learning model. After the model is generated, it runs a test against the documents from the test set automatically. The results of the test are displayed for you to review.

Troubleshoot training issues

Learn about possible error messages and how to address them.

The training data is too large

Your training data contains large text document or many entity types and resources that are needed to process the data is greater than the resources that are available to your service instance. This error can occur even when your workspace doesn't exceed the documented entity extractor limits. To resolve the issue, you can try one of the following approaches:

- Remove one or more entity types to decrease the size of your training data.
- Remove extra large documents from the training data. For example, if one of the labeled documents is extra large, change its status from **Completed** to **In progress** to omit it from the training data.
- Reduce the number of documents that are included in the training set. The default split ratio (70%-23%-7%) for the training data uses 70% of the documents in the training set. You can change the percentage of documents that are used in the training set to a smaller number. For example, you might change the split ratio to 60%-33%-7%.
- IBM Cloud Pak for Data Increase the capacity of your deployed service instance by scaling up service pods.

Evaluating the extractor

To review metrics from the test run of the entity extractor model that you created, click the **Evaluate extractor** tab.

The following table describes the available evaluation metrics.

Metric	Description
Confusion matrix	A table that provides a detailed numeric breakdown of annotated document sets. Use it to compare entity type mentions that are labeled by the machine learning model to entity type mentions that are labeled in the training data.
F1 Score	Measures whether the optimal balance between precision and recall is reached. The F1 score can be interpreted as a weighted average of the precision and recall values. An F1 score reaches its best value at 1 and worst value at 0. Overall scores are lower if the model doesn't have enough training data to learn from.
Precision	Measures how many of the overall extracted mentions are classified as the correct entity type. A false positive is when an entity label is incorrect, but was predicted to be correct (Predicted = Positive, Actual = Negative). False positives typically mean low precision.
Recall	Measures how often entity type mentions that should be extracted are extracted. A false negative is when an entity type label is correct, but was predicted to be incorrect (Predicted = Negative, Actual = Positive). False negatives typically mean low recall.

Metrics details

1. Review the metrics that are provided about the extractor model test run to determine whether more training is needed.
2. Explore the test results in more detail by clicking [Review training results in test set](#).

Documents from the test set are displayed with the predicted labels shown in one panel and the ground truth shown in the other.

- Predicted labels are the examples that the entity extractor identified and labeled as entity types.
- The **ground truth** has examples that a person labeled or that were bulk labeled and reviewed by a person. Labels in the ground truth are considered the correct labels.

The performance of the model is rated based on how closely the predicted labels match the ground truth.

Improving the extractor

The following table shows suggested fixes for common problems.

Problem	Action to remedy the problem
Low overall scores	You might not have enough documents with labeled examples in your training set. Label more examples in more of your documents.
Low recall	Label more documents with new examples of the entity types that the extractor missed.
Low precision	Look for entity types that are commonly confused. Find and label more examples of each entity type to help the entity extractor distinguish between the entity types.

Improvement actions

Adding documents to the training data

To add more documents, complete the following steps:

1. Open the **Label documents** tab.
2. From the **Document list** panel, choose **Add documents**.

This button is disabled if no other documents are available to add to the entity extractor from the current collection. To add more documents to the collection, go to the **Activity** page for the collection, and then click the **Upload data** tile to browse for and add more files.

 **Tip:** You cannot choose the documents from the collection to show in the **Document list** for labeling purposes. If there are specific types of documents that you want to label, consider adding representative documents to a collection that you can use to create the entity extractor.

There are limits to the number of documents that can be included in the training data. If your training data includes documents with a combination of sections that are labeled and others that are not, the system might sample some examples from unlabeled sentences. Subsampling helps to balance the number of positive and negative examples that are used for training. Balancing the examples in the training set improves the training performance.

Publishing the entity extractor as an enrichment

When you think the entity extractor is ready, publish the entity extractor. How do you know when it's ready? If the score doesn't change after several test runs in which you make improvements, the model is ready. You can return to update and retrain the model after you publish it.

1. From the *Evaluate extractor* page, click **Publish extractor**.
2. Click **Apply to data**.
3. Choose a collection, and then select the document field where you want the entity extractor enrichment to be applied.
4. Click **Apply**.

Exporting the entity extractor



Note: For installed deployments, the export capability was added with the 4.6.2 release.

An entity extractor model that you create and deploy in one project is available as an enrichment that can be applied to a collection from any project in the same service instance.

If you want to use the entity extractor model in a project from another service instance, you can export the entity extractor. To use it elsewhere, follow the steps to create a machine learning model from [Use imported ML models to find custom terms](#). You cannot continue to edit an entity extractor that you import into another project.

The entity extractor that you want to export must be fully trained.

To export an entity extractor, complete the following steps:

1. Open the project with the entity extractor that you want to export.
2. From the *Improvement tools* panel of the *Improve and customize* page, expand *Teach domain concepts*, and then click **Extract entities**.
3. From the *Entity extractors* list, find the entity extractor that you want to export.
4. Click the **Actions** icon for your extractor, and then choose **Download model** to save the model to your system.



Note: The **Download model** option is not available unless the model is trained.

The entity extractor model is saved as a .ent file. You can import it into a project in another service instance as a machine learning model, and then apply it to your collections. For more information about importing the model, see [Use imported ML models to find custom terms](#).

Applying an entity extractor enrichment

When you publish the extractor, you specify the field where you want the extractor to be applied. If you decide to apply the enrichment to different or more fields later, you can follow these steps to do so.

1. From the navigation panel, click **Manage collections**.
2. Click to open the collection where you want to apply the enrichment.
3. Click **Enrichments**.
4. Find the entity extractor name in the list, and then choose a field to apply the enrichment to.
5. Click **Apply changes and reprocess**.

For more information about how to remove an entity extractor enrichment from a collection, see [Managing enrichments](#).

Entity extractor output

When the enrichment recognizes one of your custom entities in a document, an entry is added to the `enriched_text.entities` section of the JSON representation of the document. The section contains occurrences of entities that are recognized by your custom model along with entities that are recognized by the built-in Entities enrichment. The built-in enrichment uses the Watson NLP service to identify entities that are part of what it calls the *Natural Language Understanding* type system. For more information about the built-in Entities enrichment, see [Entities](#).

The following JSON output is produced by a custom model that is named *literature* that recognizes family member mentions.

```

{
  "model_name": "natural_language_understanding",
  "mentions": [
    {
      "confidence": 0.4220163,
      "location": {
        "end": 12418,
        "begin": 12411
      },
      "text": "evening"
    },
    {
      "text": "evening",
      "type": "Time"
    },
    {
      "model_name": "literature",
      "mentions": [
        {
          "confidence": 1,
          "location": {
            "end": 443,
            "begin": 437
          },
          "text": "sister"
        }
      ]
    }
  ]
}

```

Figure 8. JSON representation of a custom entity mention

Monitoring performance over time

You can retrain your entity extractor model at any time. Each time that you train the model, review the performance metric scores to determine whether your most recent changes increase or decrease the model's scores.

1. To compare one test run against another, click [View score history](#).

The history view shows the last 5 training runs.

 **Tip:** To retain the score information for more than the most recent 5 training runs, you can export the metrics in comma-separated value format, and track the scores in a separate application. Click the tabular representation icon , and then click [Download as CSV](#).

If a subsequent training run results in lower scores, don't publish that version of the model.

Deleting an entity extractor

You can delete an entity extractor if it is not in use, meaning the enrichment that is published from the entity extractor is not applied to a collection.

You might want to delete an entity extractor if you hit the limit for the maximum number of extractors that are allowed for your plan, for example.

Remember, limits are defined per service instance, not per project. If you cannot create new entity extractors, but do not have the maximum number of extractors in the current project, check other projects in the same service instance. There might be entity extractors that aren't being used in other projects that can be deleted.

1. Remove the entity extractor enrichment that was published from the entity extractor that you want to delete from any collections where it is being used.

For more information, see [Deleting enrichments](#).

2. From the *Improvement tools* panel of the *Improve and customize* page, expand **Teach domain concepts**, and then click **Extract entities**.
3. Find the entity extractor that you want to delete, click the **Actions** icon, and then select **Delete**.

Entity extractor limits

The number of entity extractors that you can create per service instance depends on your Discovery plan type.

Plan	Entity extractors per service instance	Maximum entity types per extractor	Maximum documents in training data
	[1]		

Cloud Pak for Data	Unlimited	18	1,000
Premium	10	18	1,000
Enterprise	10	18	1,000
Plus (including Trial)	3	12	200

Entity extractor plan limits

1. This number reflects the number of published entity extractor enrichments for the service instance (including from imported entity extractor models) whether they are applied to a collection or not. [↳](#)

Use imported ML models to find custom terms

Use custom Machine Learning models that use rules or context to recognize and tag entities.

Add Machine Learning models that you created with IBM tools that you can use to define your own type system.

The type of models you can add depend on your deployment:

- IBM Cloud Pak for Data You can add models that were created with Watson Explorer Content Analytics Studio models, or with an instance of IBM Watson® Knowledge Studio that is hosted on IBM Cloud Pak® for Data or IBM Cloud. Starting with the 4.6.2 release, you can also add custom entity extractor models that were created in and exported from another instance of Discovery.
- IBM Cloud You can add models that were created with a IBM Watson® Knowledge Studio instance that is hosted in IBM Cloud only.



Tip: To use a Knowledge Studio model that was built with Knowledge Studio on IBM Cloud Pak for Data, migrate the ground truth to a IBM Cloud instance of Knowledge Studio. and then retrain the model.

The following types of models are supported:

- Rule-based models created in Knowledge Studio that find entities in documents based on rules that you define. (File format: .pear)
- Machine learning models created in Knowledge Studio that understand the linguistic nuances, meaning, and relationships specific to your industry (file format: .zip)
- Custom entity extractors that are created in and exported from Discovery. (File format: .ent)
- IBM Cloud Pak for Data Custom UIMA text analysis models created in Watson Explorer Content Analytics Studio. (File format: .pear)

From installed deployments, support for importing entity extractor models was added with the 4.6.2 release.



Important: Discovery cannot identify entity subtypes that are defined by a Knowledge Studio model.

To add a Machine Learning model, complete the following steps:

1. Create the model and export it from the tool you use to create it.

For more information, see the following documentation:

- Knowledge Studio for IBM Cloud Pak® for Data
 - [Creating a rule-based model](#)
 - [Creating a machine learning model](#)
- Knowledge Studio for IBM Cloud
 - [Creating a rule-based model](#)
 - [Creating a machine learning model](#)
- [Watson Explorer Content Analytics Studio](#)

You must export the model from Watson Explorer Content Analytics Studio as a UIMA PEAR file. For more information, see: [Creating Custom PEAR Files for use with Lexical Analysis Streams](#).

- [Discovery entity extractor](#)

2. From the **Teach domain concepts** section of the **Improvement tools** panel, and then click **Import machine learning models**.

3. Specify a name for the model, and then choose the language that was used to define the model.
4. Click **Upload** to browse for the file that you exported earlier.
5. Click **Create**.
6. Choose the collection and field where you want to apply the enrichments from the model, and then click **Apply**.



Note: If the model is too large to upload from the product user interface, you can use the [Create an enrichment](#) method of the API to import the file.

Rule-based model example

For example, when a machine learning model is applied as an enrichment to a field, it extracts all entity types in that field that were specified in a Knowledge Studio rule-based model. If the model recognizes entity types such as **person**, **surname**, and **job title** they are recognized in your documents and tagged.

In the output, the information that is extracted by the Machine Learning enrichment in the **enriched_{field_name}** array, within the **entities** array. In this example, the field that is selected for enrichment is **text**.

```
{
  "enriched_text": [
    {
      "entities": [
        {
          "path": ".wksrule.entities.PERSON",
          "text": "George Washington",
          "type": "PERSON"
        },
        {
          "path": ".wksrule.entities.GIVENNAME",
          "text": "George",
          "type": "GIVENNAME"
        },
        {
          "path": ".wksrule.entities.SURNAME",
          "text": "Washington",
          "type": "SURNAME"
        },
        {
          "path": ".wksrule.entities.POSITION",
          "text": "politician",
          "type": "POSITION"
        },
        {
          "path": ".wksrule.entities.POSITION",
          "text": "soldier",
          "type": "POSITION"
        },
        {
          "path": ".wksrule.entities.JOBTITLE",
          "text": "President of the United States",
          "type": "JOBTITLE"
        }
      ],
      "text": [
        "George Washington (February 22, 1732, December 14, 1799) was an American politician and soldier who served as the first President of the United States from 1789 to 1797 and was one of the Founding Fathers of the United States."
      ]
    }
  ]
}
```

As a result, if someone [uses the API](#) to submit a Discovery Query Language query to look for occurrences of the **enriched_{field_name}.entities.type:jobtitle** enrichment, any passages that discuss a person's job title are returned.

Machine learning model example

In this example, a Machine learning model extracts entity types such as **person**, **organization**, and **date**, and information about relationships between the entities. When this ML model is applied as an enrichment to a field, it uses machine learning to understand the linguistic nuances, meaning, and relationships that are mentioned in the document.

In the output, the information that is extracted by the Machine Learning enrichment in the **enriched_{field_name}** array, within the **entities** and the **relations** arrays. In this example, the field that is selected for enrichment is **text**.

```
{
```

```

"enriched_text": [
  {
    "entities": [
      {
        "count": 1,
        "text": "Democratic Party",
        "type": "ORGANIZATION"
      },
      {
        "count": 1,
        "text": "March 15, 1767",
        "type": "DATE"
      },
      {
        "count": 1,
        "text": "President",
        "type": "POSITION"
      },
      {
        "count": 1,
        "text": "Andrew Jackson",
        "type": "PERSON"
      }
    ],
    "relations": [
      {
        "sentence": "Andrew Jackson (March 15, 1767, June 8, 1845) was an American soldier and statesman who served as the seventh President of the United States from 1829 to 1837 and was the founder of the Democratic Party."
      }
    ]
  }
]

```

Machine learning model limits

The number of Machine Learning (ML) models you can create per service instance depends on your Discovery plan type.

Plan	ML models per service instance
Cloud Pak for Data	Unlimited
Premium	10
Enterprise	10
Plus (includes Trial)	3

ML model plan limits

For each Knowledge Studio machine learning model, the maximum number of entities that can be detected is 50.

Advanced rules models

Add an advanced rules model to apply a text extraction model that was created and exported from the Advanced Rule editor of IBM Watson® Knowledge Studio to your collection.

Your model must be created with the appropriate Knowledge Studio deployment:

- IBM Cloud Pak for Data You can add models that were created and exported from the following places:
 - IBM Watson® Knowledge Studio that was built with a IBM Cloud Pak® for Data deployment earlier than the 4.5 release.
 - IBM Watson® Knowledge Studio that is hosted on IBM Cloud
 - NLP Editor that is built by contributors to the Center for Open-source Data & AI Technologies
- IBM Cloud You can add models that were created with a IBM Watson® Knowledge Studio instance that is hosted on IBM Cloud only.

Removal from Knowledge Studio

Support for building models with the beta Advanced Rules Editor in Knowledge Studio ended. Any rules models that were exported from Knowledge Studio prior to the end of support date can continue to be used in Discovery.

End of support dates differ based on the deployment type:

- IBM Cloud 30 June 2022
- IBM Cloud Pak for Data IBM Cloud Pak for Data release 4.5.1 on 3 August 2022.

IBM Cloud As an alternative to using a model that is generated by the Knowledge Studio Advanced Rules Editor, you can define a rule by [adding a Patterns enrichment](#).

Adding an existing model

To add an advanced rule model, complete the following steps:

1. Create the model and export the ZIP file that contains the model resources.

For more information about how to export the model, see the instructions for your model source:

- [Knowledge Studio for IBM Cloud Pak® for Data](#)
- [Knowledge Studio for IBM Cloud](#)
- [Open source NLP Editor](#)

2. From the **Teach domain concepts** section of the **Improvement tools** panel, choose **Advanced rules model**.

3. Click **Upload**.

4. Specify a name for the model, and then choose the language that was used to define the model.

5. Specify a name for the result field, which is the field in the index where the output of this enrichment will be stored.

6. Click **Upload** to browse for the ZIP file that you exported earlier.

7. Click **Create**.

8. Choose the collection and field where you want to apply the enrichments from the model, and then click **Apply**.

Output format for advanced rules

Knowledge Studio uses the Annotation Query Language (AQL) to define the rules in an advanced rules model. Each model is defined by one or more views. Each view is a relational data structure that contains multiple data records. Each record is composed of values in columns that are defined by the view's schema. To facilitate representing these models, which are custom and therefore have various schemas, a uniform JSON output schema is used.

- Each JSON object represents an Annotation Query Language (AQL) view.
- The name-and-value pairs in the JSON objects represent the names and values of the attributes in the view.
- The tuples in an AQL view are represented as an array of JSON objects, with one object for each tuple in the view.

The following table describes how AQL data types are represented in JSON syntax.

AQL data	JSON syntax	JSON example
type		
Integer	number	5
Float	number	4.13
Boolean	boolean	true
Text	string	"some string"
Span	object with the form {"text": String, "location": {"begin": Integer, "end": Integer}}	{ "text": "Jane", "location": {"begin": 5, "end": 9} }
Special case: null value	null	null
List of Integer	array of number values	[1, 2, 3, 4, 5]
List of Float	array of number values	[4.13, 4.5]

List of Boolean	array of boolean values	[true, true, false]
List of Text	array of string values	["some string", "another string"]
List of Span	array of objects with the form {"text":String, "location": {"begin": Integer, "end": Integer}}	[{ "text":"Jane", "location": {"begin": 5, "end": 9} }, { "text":"...", "location": {"begin": 15, "end": 40} }]
Special case: empty List	array with 0 elements	[]

Advanced rules model JSON output schema

Advanced rules model limits

The number of advanced rules models that you can define per service instance depends on your Discovery plan type.

Plan	Advanced rules models per service instance
Cloud Pak for Data	Unlimited
Premium	3
Enterprise	3
Plus (includes Trial)	1

Advanced rules model plan limits

Identify terms by pattern

Use patterns to find terms

Recognize terms that are mentioned in sentences that match a syntactic pattern that you teach Discovery to recognize.

IBM Cloud Patterns is a beta feature that is available in managed deployments only. The feature is available for English-language documents only.

Add a Patterns resource to teach Discovery to recognize patterns in your data. The Patterns feature uses pattern induction, which generates extraction patterns from examples that you provide as training data. After you specify a few examples, Discovery suggests more rules that you can review and accept to complete the pattern.

Patterns produces a model by using a human-in-the-loop process. You aren't asked to build a large set of training data up front. Instead, you provide a few examples, and then participate in an interactive process to define the training data. You passively accept or reject smart suggestions that are proposed by the system.

Pattern recognition works best on text with consistent structure in casing, length, text, or numeric values. Examples of patterns you can teach Discovery to identify in your documents:

- Standards numbers, such as **ISO 45001**, **ISO 22000**.
- Currency references, such as **\$50.5 million**, **\$29 million**.
- Date references, such as **8 September 2019**, **12 June 2020**.

If you need to identify specific terms or text, such as product names, add a [dictionary](#).

For more information, read the following blog posts:

- [Extracting Text Patterns with User Highlights with Pattern Induction](#)
- [Pattern Induction: Best Practices for Extracting Text Patterns](#)

To define a pattern, complete the following steps:

1. From the **Teach domain concepts** section of the **Improvement tools** panel, choose **Patterns**.

2. Click **New**.
3. Pick how you want to choose documents.
 - Allow Discovery to choose 10 random documents for you.
 - Choose the documents yourself (up to 20 can be selected).

Each document must be under 5,000 characters in length. Documents that exceed the limit are truncated to 5,000 characters.
4. Click **Next**.
5. Start selecting example words or phrases that fit the pattern you want to define.

For example, if you have a collection of articles that discuss **ISO** standards, you might start highlighting the numbers of the standards in each document.

If you annotate something, and then change your mind, hover over the selection, and then click **x** to delete it.
6. Continue selecting examples.

After you identify enough examples, Discovery shows a list of suggested examples for you to review and determine to be valid or not valid examples. Suggested examples are taken from the field that is configured to be used in search results. If the source of result content is configured to be passages, the **text** field is used. For more information, see [Changing the result content](#).
7. Choose **Yes** or **No** for each suggestion.

Click the **Preview document** icon if you want to see the example in context before you make a choice.
8. Continue highlighting examples and validating suggestions until a message is displayed to inform you that you identified enough examples.
9. Click the **Review examples** tab to review the lists of examples that were identified by you and Discovery.
10. If the examples are correct, click **Save pattern**.



Note: If Discovery cannot discern a consistent and valid pattern based on the information you provided, the **Save pattern** button is never enabled. A pattern might not be created if you provide contradictory examples, for example. To start over, click the **Reset** button. The documents are returned to their original state and any examples that you identified previously are removed.

11. To apply the pattern immediately, choose the collection and field where you want to apply the enrichments from the model, and then click **Apply**.

When Discovery finds text in a document that matches a pattern that you defined, it is annotated in the **enriched_{fieldname}.entities** field. You can find it by checking the **enriched_{fieldname}.entities.model_name** field for your pattern name.

Downloading a pattern

To download a pattern, complete the following step:

1. In the **Patterns view**, click the download icon.

A pattern model is downloaded as a ZIP file.

You can import the downloaded ZIP file as the source for an advanced rules model resource. For more information, see [Advanced rules models](#).

Pattern limits

The number of patterns that you can define per service instance depends on your Discovery plan type.

Plan	Patterns per service instance
Premium	100
Enterprise	100
Plus (includes Trial)	20

Pattern plan limits

Use regular expressions to find terms

Define regular expressions that capture patterns of significance, such as that **AB10045** is the syntax that is used for your order numbers.

Define regular expressions that can identify and extract information from fields in your collection.

For example, this regular expression finds occurrences of credit card numbers of a specific format and length.

```
4[0-9]{15}
```

The following regular expression finds occurrences of a US social security number.

```
(?!666|000|9\d{2})\d{3}-(?!00)\d{2}-(!0{4})\d{4}
```

To add a regular expression, complete the following steps:

1. From the **Teach domain concepts** section of the **Improvement tools** panel, choose **Regular expression**.
2. Click **Upload**.
3. Optional: Specify a facet path to categorize any text that matches the regular expression. The text can be filtered by this facet later.

If you use a hierarchy of categories, add a period between category names in the facet path to represent the hierarchy. For example, if you are adding a regex that can recognize phone numbers, you might have a facet path such as **international.europe**.

4. Add the regular expression.
 - o Use a Java™ regular expression.
For more information, see the [Java documentation](#). Another useful resource is [Regex 101](#).
 - o Keep the regular expression as short and understandable as possible.
 - o The best regular expressions resolve to a match or non-match quickly.
 - o Use common patterns. For example, use **a(b|c|d)** instead of **(ab|ac|ad)**.
 - o The regular expression engine might fail if it backtracks because it can't make a negative match toward the end of the string and then attempts too many permutations. To prevent backtracks, consider using a possessive quantifier, such as **(a+b*)++c**.
5. Click **Create**.
6. Choose the collection and field to search for occurrences of text that match this regular expression pattern.

In the output, the information that is extracted by the regular expression enrichment can be found under **enriched_{field_name}**, within the **entities** array.

In this example, the **Facet Path** is **regex.cccardnumber**, and the field that is selected for enrichment is **text**.

```
{
  "enriched_text": [
    {
      "entities": [
        {
          "path": ".regex.cccardnumber",
          "type": "cccardnumber",
          "text": "4000000000000000"
        }
      ]
    },
    "text": [
      "He has 2 phones, 090-1234-5678 and 080-1234-5678. His credit card number is 4000000000000000."
    ]
  }
}
```

When you submit test queries from the **Improve and customize** page, you can add a facet that is based on the **enriched_text.entities.model_name** field. As a result, the **cccardnumber** regular expression enrichment that you created is displayed as a facet value by which documents can be filtered. For more information about creating facets, see [Adding facets](#).

Regular expression limits

The number of regular expressions that you can define per service instance depends on your Discovery plan type.

Plan	Regular expressions per service instance
Cloud Pak for Data	Unlimited
Premium	100
Enterprise	100
Plus (includes Trial)	20

[Regular expression plan details](#)

Classify text

Define categories by which text in your documents can be classified.

This topic describes how to classify text. If you want to classify documents, use the Content Mining application. For more information, see [Classifier types](#).

Add a text classifier to assign text from documents in your collection into categories. Discovery uses the labels and text examples that you provide to predict the categories of text in your collection.

To create a text classifier, complete the following steps:

1. Create a CSV file that contains example text followed by its category label per line.

The CSV file must be in UTF-8 encoding format and must meet the following requirements:

- The format must be `text,label`. The `text` is the example text, and the `label` is the category name.

Add complete sentences as text entries. Do not include any blank lines in the CSV file.

You can add more `label` columns if you need to apply more than one label to the sentence in the `text` column. For example, `text,label,label`.

- The file must have at least two columns with no headers.
- Add 10 or more entries for each category that you want to define. The minimum number of entries that are required per category is 3. The more examples that you provide for each category, the better the classifier can predict the categories of other content in your collection.

The following example is a CSV file that defines two categories, named `facility_temperature` and `catering`. The example text consists of feedback from conference attendees.

```
The rooms were too cold.,facility_temperature
Breakfast did not include gluten-free options.,catering
The rooms were too warm.,facility_temperature
I was very comfortable in the session rooms.,facility_temperature
The awards dinner was delicious.,catering
Coffee ran out during one of the breaks.,catering
The temperature was not comfortable.,facility_temperature
I was very happy with the selection at lunch.,catering
It was nice that you provided tea and coffee. Tea drinkers are often ignored.,catering
Can you turn up the air conditioning? I was very warm.,facility_temperature
My teeth were chattering because I was so cold.,facility_temperature
The speaker left the room to find someone to adjust the temperature.,facility_temperature
Would you consider an all-vegan menu next year?,catering
I would like lemonade and iced tea to be served during the breaks.,catering
The lunch staff was excellent.,catering
Appreciated the fresh blueberry muffins at breakfast.,catering
The hotel staff adjusted the temperature in my session room as soon as I asked. Excellent service!,facility_temperature
Every meal was delicious and there was something for everyone.,catering
The seats under the skylights were not comfortable. Too hot.,facility_temperature
I was comfortable everywhere in the conference center. I never needed my emergency sweater.,facility_temperature
```

2. From the **Teach domain concepts** section of the **Improvement tools** panel, and then click **Text classifiers**.

3. Click **Upload**.
 4. Specify a name for the classifier, and then choose the language that was used in the CSV file.
 5. Click **Upload** to browse for the CSV file that you created earlier.
 6. Click **Create**.
- A classifier enrichment is created based on the training data that you provided.
7. Choose the collection and field where you want to apply the text classifier enrichment, and then click **Apply**.

The following example shows how an enrichment that is created with the sample CSV file as its training data might classify text in a document. In the output, the classifier enrichment applies the `facility_temperature` label to the document text. The `label` is stored in the `enriched_{field_name}` array, within the `classes` array.

```
{
  "enriched_text": [
    {
      "classes": [
        {
          "confidence": 0.999692440032959,
          "label": "facility_temperature"
        }
      ]
    },
    "text": [
      "I think more attendees would stay awake in the sessions if the rooms were colder."
    ]
  }
}
```

Classifier types

The classifier that you add from the Discovery user interface is a **text classifier**. A text classifier can classify documents based on words and phrases that are extracted from the body text with their part of speech information taken into account.

You can create another classifier type, a **document classifier**, only from the deployed Content Mining application. A document classifier can classify documents based on words and phrases that are extracted from the body text fields with information from their part of speech and the other enrichments that are applied to the body text taken into account. The information from the other non-body fields are also used.

You can apply a document classifier to a collection in a project type other than a Content Mining project. To do so, you must create the classifier in the deployed Content Mining application and export it. You can then import the classifier and apply it to your collection as an enrichment. For more information, see [Creating and applying a document classifier](#).

The text classifier uses Part of Speech information regardless of whether the Part of Speech enrichment is applied to the project.

Text classifiers that you add to one project can be used by other projects, including Content Mining projects.

 **Tip:** A text classifier does not classify the target text field with confidence scores that are lower than 0.5. You cannot change the confidence threshold that is used by the text classifier. If you expected certain types of passages to be classified that weren't, you can add passages with similar characteristics to your training data and train another classifier.

Text classifier limits

The number of text classifiers and labels that you can create per service instance depends on your Discovery plan type.

Limit	Plus	Enterprise	Premium	Cloud Pak for Data
Number of text classifiers per service instance	5	20	20	Unlimited
Number of labeled data rows	2,000	20,000	20,000	20,000
Maximum size in MB of training data after enrichment	16	1,024	1,024	1,024
Number of labels	100	1,000	1,000	1,000

Text classifier plan limits

Detect sentiment

Use the built-in Watson Natural Language Processing (NLP) sentiment enrichment to analyze the sentiment that is expressed in text and indicate whether the text is **positive**, **neutral**, or **negative**.

To understand the sentiment of an entire document, apply this enrichment to a field that contains as much of the text from the document as possible, such as the **text** field.

To analyze sentiment in text from multiple fields at one time and capture the overall sentiment of the document, use the Content Mining application. For more information, see [Detecting phrases that express sentiment](#).

Adding the enrichment

To add the sentiment enrichment, complete the following steps:

1. Open your project and go to the **Manage collections** page.
2. Click to open the collection that you want to enrich.
3. Open the **Enrichments** tab.
4. Scroll to find and select the Sentiment enrichment.
5. Choose one or more fields to apply the enrichment to.

You can apply enrichments to the **text** and **html** fields, and to custom fields that were added from uploaded JSON or CSV files or from the Smart Document Understanding (SDU) tool.

6. Click **Apply changes and reprocess**.

Enrichments that you enable are applied to the documents in random order. For information about how to remove an enrichment, see [Managing enrichments](#).

Example

Input

```
"It is powerful and easy to use and integrate with third party applications."
```

Response

In the JSON output:

- **score** = Sentiment score from **-1** (negative) to **1** (positive)
- **label** = **positive**, **negative**, or **neutral**
- **mixed** = Indicates that the document expresses a combination of different sentiments

```
{  
  "sentiment": {  
    "score": 0.9255063900060722,  
    "mixed": false,  
    "label": "positive"  
  }  
}
```

Read contracts

The **Contracts** enrichment identifies contract-related elements in a document.

To use the Contracts enrichment, create a **Document Retrieval** project type and select the **Apply contracts enrichment** option. When you make this selection, a **Document Retrieval for Contracts** project is created.



Note: Only users of installed deployments (IBM Cloud Pak for Data) or Premium or Enterprise plan managed deployments can create a **Document Retrieval for Contract** project type.

To see the elements that are identified by the **Contracts** enrichment, complete the following steps:

1. From the **Improve and Customize** page, submit a search query.

You can submit any keyword you want or pick one of the suggested keyword search terms.

2. Click **View passage in document** for one of the search results that are displayed for the document that you want to review.

3. Do one of the following things:

IBM Cloud

1. Click **Open advanced view** to see the **Contract Data** page.

The screenshot shows the 'Contract Data' view for the 'IBM Cloud Service Agreement'. The left sidebar contains filters for 'Category' (e.g., Assignments, Business Continuity, Communication, Dispute Resolution, Intellectual Property, Liability, Order of Precedence, Payment Terms & Billing, Pricing & Taxes, Privacy, Safety and Security, Term & Termination, Warranties) and 'Nature' (Disclaimer, Exclusion). The main content area displays the 'Cloud Services Agreement' with a detailed description of its components: Complete Agreement, Transaction Documents, Attachments, and specific sections like 1. Cloud Services and a. IBM Cloud Services.

Figure 1. Contract Data view

IBM Cloud Pak for Data

1. Click **Contract Data**.

A list of the elements that the **Contracts** enrichment identified in the document is displayed.

Contract schema information

Contract enrichments are applied to the **html** field of documents that are added to the project.

After a document is processed by the Contracts enrichment, the service generates JSON output in the following schema:

```
{  
  "elements": [  
    {  
      "location": {  
        "begin": int,  
        "end": int  
      },  
      "text": string,  
      "types": [  
        {  
          "label": { "nature": string, "party": string },  
          "provenance_ids": [string, string, ...]  
          ...  
        }  
      ]  
    }  
    ...  
  ],  
  "categories": [  
    {  
      "label": string,  
      "provenance_ids": [string, string, ...]  
    }  
    ...  
  ],  
  "attributes": [  
    {  
      "type": string,  
      "text": string,  
    }  
  ]  
}
```

```
        "location": { "begin": int, "end": int }
    }
]
...
],
"effective_dates": [
{
    "confidence_level": string,
    "text": string,
    "text_normalized": string,
    "provenance_ids": [ string, string, ... ],
    "location": { "begin": int, "end": int }
},
...
],
"contract_amounts": [
{
    "confidence_level": string,
    "text": string,
    "text_normalized": string,
    "interpretation": {
        "value": string,
        "numeric_value": number,
        "unit": string,
    },
    "provenance_ids": [ string, string, ... ],
    "location": { "begin": int, "end": int }
},
...
],
"termination_dates": [
{
    "confidence_level": string,
    "text": string,
    "text_normalized": string,
    "provenance_ids": [ string, string, ... ],
    "location": { "begin": int, "end": int }
},
...
],
"contract_types": [
{
    "confidence_level": string,
    "text": string,
    "provenance_ids": [ string, string, ... ],
    "location": { "begin": int, "end": int }
},
...
],
"contract_terms": [
{
    "confidence_level": string,
    "text": string,
    "text_normalized": string,
    "interpretation": {
        "value": string,
        "numeric_value": number,
        "unit": string,
    },
    "provenance_ids": [ string, string, ... ],
    "location": { "begin": int, "end": int }
},
...
],
"payment_terms": [
{
    "confidence_level": string,
    "text": string,
    "text_normalized": string,
    "interpretation": {
        "value": string,
        "numeric_value": number,
        "unit": string,
    },
    "provenance_ids": [ string, string, ... ],
    "location": { "begin": int, "end": int }
},
...
],
"contract_currencies": [
{
    "confidence_level" : string,
```

```

    "text" : string,
    "text_normalized" : string,
    "provenance_ids": [string, string ...],
    "location": { "begin": int, "end": int }
},
...
],
"tables": [],
"document_structure": {
    "section_titles": [
        {
            "text": string,
            "location": {
                "begin": int,
                "end": int
            },
            "level": int,
            "element_locations": [
                {
                    "begin": int,
                    "end": int
                },
                ...
            ]
        },
        ...
    ],
    "leading_sentences": [
        {
            "text": string,
            "location": {
                "begin": int,
                "end": int
            },
            "element_locations": [
                {
                    "begin": int,
                    "end": int
                },
                ...
            ],
            "paragraphs": [
                {
                    "location": {
                        "begin": int,
                        "end": int
                    }
                },
                ...
            ]
        },
        ...
    ],
    "parties": [
        {
            "party": string,
            "role": string,
            "importance": string,
            "addresses": [
                {
                    "text": string,
                    "location": {
                        "begin": int,
                        "end": int
                    }
                },
                ...
            ],
            "contacts": [
                {
                    "name": string,
                    "role": string
                },
                ...
            ],
            "mentions": [
                {
                    "text": string,
                    "location": {
                        "begin": int,
                        "end": int
                    }
                }
            ]
        }
    ]
}

```

```
    },
    ...
],
},
...
}
```

Schema arrangement

The `contracts` schema is arranged as follows.

- `elements`: An array of the document elements detected by the service.
 - `location`: An object that identifies the location of the element. The object contains two index numbers, `begin` and `end`. The index numbers indicate the beginning and ending positions of the characters that constitute the element in HTML.
 - `text`: The text of the element.
 - `types`: An array that describes what the element is and whom it affects.
 - `label`: An object that defines the type by using a pair of the following elements:
 - `nature`: The type of action the sentence requires. Current values are `Definition`, `Disclaimer`, `Exclusion`, `Obligation`, and `Right`.
 - `party`: A string that identifies the party to whom the sentence applies.
 - `provenance_ids`: An array of one or more hashed values that you can send to IBM to provide feedback or receive support.
 - `categories`: An array that lists the functional categories into which the element falls; in other words, the subject matter of the element.
 - `label`: A string that lists the identified category. For a list of categories, see [Categories](#).
 - `provenance_ids`: An array of one or more hashed values that you can send to IBM to provide feedback or receive support.
 - `attributes`: An array that identifies document attributes. Each object in the array consists of three elements:
 - `type`: The type of attribute. Possible values are `Currency`, `DateTime`, `Duration`, `Location`, `Number`, `Organization`, `Percentage`, and `Person` as described at [Attributes](#).
 - `text`: The text that is associated with the attribute.
 - `location`: The location of the attribute as defined by its `begin` and `end` indexes.
- `effective_dates`: An array that identifies the date or dates on which the document becomes effective.
 - `confidence_level`: The confidence level of the identification of the effective date. Possible values include `High`, `Medium`, and `Low`.
 - `text`: An effective date, which is listed as a string.
 - `text_normalized`: The normalized form of the effective date, which is listed as a string. This element is optional; that is, the service output lists it only if normalized text exists.
 - `location`: The location of the date as defined by its `begin` and `end` indexes.
 - `provenance_ids`: An array that contains zero or more keys. Each key is a hashed value that you can send to IBM to provide feedback or receive support.
- `contract_amounts`: An array that monetary amounts that identify the total amount of the contract that needs to be paid from one party to another.
 - `confidence_level`: The confidence level of the identification of the contract amount. Possible values include `High`, `Medium`, and `Low`.
 - `text`: A contract amount, which is listed as a string.
 - `location`: The location of the amount or amounts as defined by its `begin` and `end` indexes.
 - `provenance_ids`: An array that contains zero or more keys. Each key is a hashed value that you can send to IBM to provide feedback or receive support.
- `termination_dates`: An array that identifies the date or dates on which the document is to be terminated.
 - `confidence_level`: The confidence level of the identification of the termination date. Possible values include `High`, `Medium`, and `Low`.
 - `text`: A termination date, which is listed as a string.
 - `text_normalized`: The normalized form of the termination date, which is listed as a string. This element is optional; that is, the service output lists it only if normalized text exists.
 - `location`: The location of the date as defined by its `begin` and `end` indexes.
 - `provenance_ids`: An array that contains zero or more keys. Each key is a hashed value that you can send to IBM to provide feedback or receive support.
- `contract_types`: An array that identifies the document's contract type or types.

- **confidence_level**: The confidence level of the identification of the contract type. Possible values include **High**, **Medium**, and **Low**.
- **text**: A contract type, which is listed as a string.
- **provenance_ids**: An array that contains zero or more keys. Each key is a hashed value that you can send to IBM to provide feedback or receive support.
- **location**: The location of the contract type as defined by its **begin** and **end** indexes.
- **contract_terms**: An array that identifies the duration or durations of the contract.
 - **confidence_level**: The confidence level of the identification of the contract terms. Possible values include **High**, **Medium**, and **Low**.
 - **text**: A contract term, which is listed as a string.
 - **provenance_ids**: An array that contains zero or more keys. Each key is a hashed value that you can send to IBM to provide feedback or receive support.
 - **location**: The location of the contract term as defined by its **begin** and **end** indexes.
- **payment_terms**: An array that identifies the document's payment duration or durations.
 - **confidence_level**: The confidence level of the identification of the payment term. Possible values include **High**, **Medium**, and **Low**.
 - **text**: A payment term, which is listed as a string.
 - **text_normalized**: The normalized text, if applicable.
 - **interpretation**: The details of the normalized text, if applicable.
 - **value**: A string that lists the value that was found in the normalized text.
 - **numeric_value**: An integer or double that expresses the numeric value of the **value** key.
 - **unit**: A string that lists the unit of the value that was found in the normalized text.

The value of **unit** is the [ISO-4217 currency code](#) that is identified for the currency amount (for example, **USD** or **EUR**). If the service cannot disambiguate a currency symbol (for example, **\$** or **£**), the ambiguous symbol itself is stored as the **unit** value.
- **provenance_ids**: An array that contains zero or more keys. Each key is a hashed value that you can send to IBM to provide feedback or receive support.
- **location**: The location of the contract term as defined by its **begin** and **end** indexes.
- **contract_currencies**: An array that identifies the document's contract currency or currencies.
 - **confidence_level**: The confidence level of the identification of the contract currency. Possible values include **High**, **Medium**, and **Low**.
 - **text**: A contract currency, which is listed as a string.
 - **text_normalized**: The normalized text, if applicable. It is listed as a string in [ISO-4217](#) format
 - **provenance_ids**: An array that contains zero or more keys. Each key is a hashed value that you can send to IBM to provide feedback or receive support.
 - **location**: The location of the contract currency as defined by its **begin** and **end** indexes.
- **document_structure**: An object that describes the structure of the input document.
 - **section_titles**: An array that contains one object per section or subsection that is detected in the input document. Sections and subsections are not nested. Instead, they are flattened out and can be placed back in order by using the **begin** and **end** values of the element and the **level** value of the section.
 - **text**: A string that lists the section title, if detected.
 - **location**: The location of the title in the input document as defined by its **begin** and **end** indexes.
 - **level**: An integer that indicates the level at which the section is located in the input document. For example, represents a root-level section, represents a subsection within the level section.
 - **element_locations**: An array that specifies the **begin** and **end** values of the sentences in the section.
 - **leading_sentences**: An array that contains one object per leading sentence of a list or subsection, in parallel with the **section_titles** and **paragraph** arrays. The object details the leading sentences in the matching section or subsection. As in the **section_titles** array, the objects are not nested. Instead, they are flattened out and can be placed back in order by using the **begin** and **end** values of the element or any level markers in the input document.
 - **text**: A string that lists the leading sentence, if detected.
 - **location**: The location of the leading sentence in the input document as defined by its **begin** and **end** indexes.
 - **element_locations**: An array that specifies the **begin** and **end** values of the leading sentences in the section.
 - **paragraphs**: An array that contains one object per paragraph, in parallel with the **section_titles** and **leading_sentences** arrays. Each object lists the span (beginning and end location) of the corresponding paragraph.

- **location**: The location of the paragraph in the input document as defined by its **begin** and **end** indexes.
- **parties**: An array that defines the parties that are identified by the service.
 - **party**: A string that provides the normalized form of the party's name.
 - **role**: A string that identifies the role of the party.
 - **importance**: A string that identifies the importance of the party. Possible values include **Primary** for a primary party and **Unknown** for a non-primary party.
 - **addresses**: An array of objects that identify addresses.
 - **text**: A string that contains the address.
 - **location**: The location of the address as defined by its **begin** and **end** indexes.
 - **contacts**: An array that defines the name and role of contacts that are identified in the input document.
 - **name**: A string that lists the name of an identified contact.
 - **role**: A string that lists the role of the identified contact.
 - **mentions**: An array of objects that identify mentions of the party.
 - **text**: A string that lists the name of the party.
 - **location**: The location of the mention as defined by its **begin** and **end** indexes.

location object

The **location** object is included with most of element definitions. The object identifies the location of the text string or number that represents the element. The object contains two index numbers, **begin** and **end**. The index numbers indicate the beginning and ending positions of the characters in the mention.

For example, a **text** string with the value **Amount due** might have a corresponding **location** object that looks as follows:

```
{
  ...
  "location": {
    "begin": 2510,
    "end": 2519
  }
  ...
}
```

The **begin** index indicates that the string begins at character position **2510** in the transformed HTML, which is the location of the letter **A** in **Amount**. The **end** index indicates that the string ends at character position **2519**, which is the location of the letter **e** in **due**.

Document structure

The output of the **Contracts** enrichment includes a **document_structure** object that details the structural composition of the input document. The document structure information is represented in the following JSON sample. The object is located immediately after the root-level **tables** array.

```
"document_structure": {
  "section_titles": [
    {
      "text": string,
      "location": {
        "begin": int,
        "end": int
      },
      "level": int
      "element_locations": [
        {
          "begin": int,
          "end": int
        },
        ...
      ]
    },
    ...
  ],
  "leading_sentences": [
    {
      "text": string,
      "location": {
        "begin": int,
        ...
      }
    }
  ]
}
```

```

    "end": int
},
"element_locations": [
{
  "begin": int,
  "end": int
},
...
]
},
...
],
"paragraphs": [
{
  "location": {
    "begin": int,
    "end": int
  }
},
...
]
}

```

Document structure elements

The elements of the `document_structure` object contain the following information:

- `document_structure`: An object that describes the structure of the input document.
 - `section_titles`: An array that contains one object per section or subsection that is detected in the input document. Sections and subsections are not nested. Instead, they are flattened out and can be placed back in order by using the `begin` and `end` values of the element and the `level` value of the section.
 - `text`: A string that lists the section title, if detected.
 - `location`: The location of the title in the input document as defined by its `begin` and `end` indexes.
 - `level`: An integer that indicates the level at which the section is located in the input document. For example, `1` represents a root-level section, `2` represents a subsection within the level `1` section, and so on.
 - `element_locations`: An array that contains objects that specify the `begin` and `end` values of the sentences in the section.
 - `leading_sentences`: An array that contains one object per leading sentence of a list or subsection, in parallel with the `section_titles` and `paragraph` arrays. The object details the leading sentences in the matching section or subsection. As in the `section_titles` array, the objects are not nested. Instead, they are flattened out and can be placed back in order by using the `begin` and `end` values of the element or any level markers in the input document.
 - `text`: A string that lists the leading sentence, if detected.
 - `location`: The location of the leading sentence in the input document as defined by its `begin` and `end` indexes.
 - `element_locations`: An array that contains objects that specify the `begin` and `end` values of the leading sentences in the section.
 - `paragraphs`: An array that contains one object per paragraph, in parallel with the `section_titles` and `leading_sentences` arrays. Each object lists the span (beginning and end location) of the corresponding paragraph.
 - `location`: The location of the paragraph in the input document as defined by its `begin` and `end` indexes.

Elements

The Contract enrichment generates an analysis of each identified element in the contract. The following sections describe each type of element that is generated.

Types

The `types` array includes a number of objects, each of which contains `nature` and `party` keys whose values identify a couplet for the element.

The following tables list the possible values of the `nature` and `party` keys.

<code>nature</code>	Description
<code>Definition</code>	This element adds clarity for a term, relationship, or similar. No action is required to fulfill the element, nor is any party affected.

Disclaimer The **party** in the element is not obligated to fulfill the terms that are specified by the element but is not prohibited from doing so.

Exclusion The **party** in the element will not fulfill the terms that are specified by the element.

Obligation The **party** in the element is required to fulfill the terms specified by the element.

Right The **party** in the element is guaranteed to receive the terms specified by the element.

Supported keys

Each **nature** key is paired with a **party** key, which contains either the name or the role of the party or parties that apply to the nature (examples include, but are not limited to, **Buyer**, **IBM**, or **All Parties**). For the **Definition** nature, the party is always **None**.

Parties

The **parties** array specifies the participants that are listed in the contract. Each **party** object is associated with other objects that provide details about the party, including:

- **role**: The party's role. Values are listed in the table that follows this list.
- **importance**: The importance of the party. Possible values are **Primary** for a primary party and **Unknown** for a non-primary party.
- **addresses**: An array that identifies addresses.
 - **text**: An address.
 - **location**: The location of the address as defined by its **begin** and **end** indexes.
- **contacts**: An array that defines the names and roles of contacts that are identified in the input document.
 - **name**: The name of a contact.
 - **role**: The role of the contact.
- **mentions**: An array of objects that identify mentions of the party.
 - **text**: A string that lists the name of the party.
 - **location**: The location of the mention as defined by its **begin** and **end** indexes.

The values of **role** that can be returned for contracts include, but are not limited to:

role	Description
Buyer	The party responsible for paying for the goods or services that are listed in the contract.
End User	The party who interacts with the provided goods or services, explicitly distinguished from the Buyer .
None	No party was identified for the element.
Supplier	The party responsible for providing the goods or services that are listed in the contract.

Supported role values

Categories

The **categories** array defines the subject matter of the sentence. Currently, supported categories include:

categories	Description
Amendments	Elements that specify changes to the contract after it was signed, or alterations to a standard contract. Includes discussions of the conditions for changing the terms of a contract.

Asset Use	Elements that refer to how one party may or may not use the assets of another party. This category specifically applies to one party having access to or using assets such as licenses, equipment, tools, or personnel of the other party while conducting their duties under the agreement, including permissions and restrictions thereon. This category does not extend to specifications of a party's obligations or rights regarding any purchased goods, services, licenses, and so on, as those goods are the party's own assets, rather than assets of another party.
Assignments	Elements that describe the transfer of rights, obligations, or both to a third party.
Audits	Elements referring to either the right of a party to examine or review compliance, or requirements that a party be available for inspection or compliance review. This category includes references to record keeping (primarily as it relates to the right of inspection) and the maintenance and retention of activity records that may be examined.
Business Continuity	Elements referring to the consequences if the entire business of one of the parties is sold.
Communication	Elements referring to requirements to communicate, respond, notify, or provide notice; contact information; or information regarding changes to the contract. Also includes references to details about communication methods, the act or process of exchanging information, and acceptable means of exchanging information between parties (and others who are not necessarily direct parties to the contract).
Confidentiality	Elements describing how parties can or cannot use information that is learned in the course of completing the contract and going forward. Also includes discussion of information that must be kept confidential, such as maintaining trade secrets or nondisclosure of business information.
Deliverables	Elements specifying the items, such as goods or services, that one party provides to another under the terms of the contract, usually in exchange for payment. Includes discussion of preparation of deliverables.
Delivery	Elements that specify the means or modes of transferring deliverables (things, as opposed to personal services) from one party to another. Includes discussions of characteristics of delivery, such as scheduling or location.
Dispute Resolution	Elements discussing provisions for settling any dispute (for example, regarding labor, invoices, or billing) arising between contracting parties. Provision examples may include settlement by a defined procedure such as an arbitration panel, a process for obtaining an injunction, waiving a right to trial, or prohibiting the pursuit of a class action. Also includes references to the contract's governing law or choice of law, such as a particular country or jurisdiction.
Force Majeure	Elements that refer to unexpected or disruptive events outside a party's control that would relieve the party from performing their contractual obligation.
Indemnification	Elements that specify the remediation of certain liabilities, when one party of the contract becomes responsible for compensating another party as a result of incurred loss or damages during the term of, or arising from the circumstances of the contract. Also includes references to any legal exemptions from loss or damages.
Insurance	Elements referring to insurance coverage or terms of coverage that is provided by one party to another party (including to third parties such as subcontractors or others). Includes various types of insurance including, but not limited to, medical insurance.
Intellectual Property	Elements that discuss the assignment of rights (such as copyrights, patents, and trade secrets) to parties to the contract. Includes references to patents, rights to apply for patents, trademarks, trade names, service marks, domain names, copyrights, and all applications and registration of such schematics, industrial models, inventions, authorship, know-how, trade secrets, computer software programs, and other intangible proprietary information. Also includes discussion of the consequences of violation of intellectual property rights.
Liability	Elements that describe the method for determining when and how fault attaches to any party. Examples may include, but are not limited to, statements regarding limitations of liability, third-party claims, and repairs, replacements, or reimbursements as required of the party at fault.

Payment Terms & Billing	Elements that detail how and when a party is to pay or get paid, and the items or fees the parties will be paying or billed for. Includes references to modes of payment or payment mechanisms.
Pricing & Taxes	Elements that refer to specific amounts or figures that are associated with individual deliverables that are exchanged (for example, how much something costs) as part of satisfying the terms of the contract. Includes references to specific figures or methods for calculating prices or tax amounts.
Privacy	Elements that are particularly concerned with the treatment of sensitive personal information, usually regarding its protection (for example, to satisfy regulations such as GDPR).
Responsibilities	Elements that discuss tasks ancillary to the contract that are in only one party's control and are focused on discussion of employee oversight.
Safety and Security	Elements referring to physical safety or cybersecurity protection for people, data, or systems. Examples include discussions of background checks, safety precautions, workplace security, secure access protocols, and product defects that might pose a danger.
Scope of Work	Elements that define what is in the contract versus is not in the contract; consequently, what is promised to be done. Examples include statements that define an order, or describe the goals or aims outlined in the contract.
Subcontracts	Elements referring to the hiring of third parties to perform certain duties under the contract, and the permissions, rights, restrictions, and consequences thereto and arising therefrom.
Term & Termination	Elements referring to duration of the contract, the schedule and terms of contract termination, and any consequences of termination, including any obligations that apply at or after termination.
Warranties	Elements that refer to ongoing promises and obligations that are made in the contract that are currently true and will continue to be true in the future. Also, elements that discuss the consequences of such promises or obligations that are broken, and the rights to remedy the situation (for example, but not limited to, seeking damages). This category does not apply to elements that are concerned with representation statements (statements of fact about the past or the present), or to elements that lay out assumptions about things that occurred in the past.

Supported categories

Attributes

The `attributes` array specifies any attributes that are identified in the sentence. Each object in the array includes three keys: `type` (the type of attribute from the following table), `text` (the applicable text), and `location` (the `begin` and `end` indexes of the attribute in the input document). Currently, supported attributes include:

attributes	Description
<code>Currency</code>	Monetary value and units.
<code>DateTime</code>	A date, time, date range, or time range.
<code>DefinedTerm</code>	A term that is defined in the input document.
<code>Duration</code>	A time duration.
<code>Location</code>	A geographical location or region.
<code>Number</code>	A digital or textual number that describes a quantity of countable things and is not classified as one of the other numerical <code>attribute</code> types.
<code>Organization</code>	An organization.
<code>Percentage</code>	A percentage.

Person A person.

Supported attributes

Effective dates

The `effective_dates` array identifies the date or dates on which the document becomes effective.

effective_dates	Description
<code>confidence_level</code>	The confidence level of the identification of the effective date. Possible values include <code>High</code> , <code>Medium</code> , and <code>Low</code> .
<code>text</code>	An effective date, listed as a string.
<code>text_normalized</code>	The normalized text of the <code>text</code> if available, listed as a string.
<code>location</code>	The location of the date as defined by its <code>begin</code> and <code>end</code> indexes.
<code>provenance_ids</code>	An array of hashed values that you can send to IBM to provide feedback or receive support.

Effective date values

Contract amounts

The `contract_amounts` array identifies the monetary amounts specified in the document.

contract_amounts	Description
<code>confidence_level</code>	The confidence level of the identification of the contract amount. Possible values include <code>High</code> , <code>Medium</code> , and <code>Low</code> .
<code>text</code>	A contract amount, listed as a string.
<code>normalized_text</code>	The normalized text, if applicable.
<code>interpretation</code>	The details of the normalized text, if applicable.
<code>value</code>	A string that lists the value that was found in the normalized text.
<code>numeric_value</code>	An integer or float expressing the numeric value of the <code>value</code> key.
<code>unit</code>	A string that lists the unit of the value that was found in the normalized text.
<code>location</code>	The location of the contract amount as defined by its <code>begin</code> and <code>end</code> indexes.
<code>provenance_ids</code>	An array of hashed values that you can send to IBM to provide feedback or receive support.

Amount values

Termination dates

The `termination_dates` array identifies the document's termination dates.

termination_dates	Description
<code>confidence_level</code>	The confidence level of the identification of the termination date. Possible values include <code>High</code> , <code>Medium</code> , and <code>Low</code> .
<code>text</code>	The termination date, listed as a string.

text_normalized	The normalized text of the text if available, listed as a string.
location	The location of the termination date as defined by its begin and end indexes.
provenance_ids	An array of hashed values that you can send to IBM to provide feedback or receive support.

Termination datae values

Contract types

The **contract_types** array identifies the document's contract type or types as declared in the document.

contract_types	Description
confidence_level	The confidence level of the identification of the contract type. Possible values include High , Medium , and Low .
text	The contract type, listed as a string.
location	The location of the contract type as defined by its begin and end indexes.
provenance_ids	An array of hashed values that you can send to IBM to provide feedback or receive support.

Contract type values

Contract terms

The **contract_terms** array identifies the duration or durations of the contract as declared in the document.

contract_terms	Description
confidence_level	The confidence level of the identification of the contract term. Possible values include High , Medium , and Low .
text	The contract term, listed as a string.
normalized_text	The normalized text, if applicable.
interpretation	The details of the normalized text, if applicable.
value	A string that lists the value that was found in the normalized text.
numeric_value	An integer or float expressing the numeric value of the value key.
unit	A string that lists the unit of the value that was found in the normalized text.
location	The location of the contract term as defined by its begin and end indexes.
provenance_ids	An array of hashed values that you can send to IBM to provide feedback or receive support.

Contract term values

Payment terms

The **payment_terms** array identifies the payment duration or durations as declared in the document.

payment_terms	Description
confidence_level	The confidence level of the identification of the payment term. Possible values include High , Medium , and Low .

text	The payment term, listed as a string.
normalized_text	The normalized text, if applicable.
interpretation	The details of the normalized text, if applicable.
value	A string that lists the value that was found in the normalized text.
numeric_value	An integer or float expressing the numeric value of the value key.
unit	A string that lists the unit of the value that was found in the normalized text.
location	The location of the payment term as defined by its begin and end indexes.
provenance_ids	An array of hashed values that you can send to IBM to provide feedback or receive support.

Payment term values

Contract currencies

The **contract_currencies** array identifies the contract currency or currencies as declared in the document.

contract_currencies	Description
confidence_level	The confidence level of the identification of the contract currency. Possible values include High , Medium , and Low .
text	The contract currency, listed as a string.
text_normalized	The normalized text of the text if applicable, listed as a string in ISO-4217 format.
location	The location of the contract currency as defined by its begin and end indexes.
provenance_ids	An array of hashed values that you can send to IBM to provide feedback or receive support.

Contract currency values

Provenance

Each object in the **types** and **categories** arrays includes a **provenance_ids** array. The **provenance_ids** array has one or more keys. Each key is a hashed value that you can send to IBM to provide feedback or receive support.

Understand tables

Apply the **Table Understanding** enrichment to get detailed information about tables and table-related data within documents.

The following tasks generate an HTML field with table information and apply the Table Understanding enrichment to it for your collection automatically:

- If you use the Smart Document Understanding tool to define a user-trained or pretrained SDU model, the **Table Understanding** enrichment is applied to the **html** field that is generated for the collection.
- If you create a **Document Retrieval for Contracts** project type, a pretrained SDU model is applied to your collection automatically. As a result, the **Table Understanding** enrichment is applied to the **html** field that is generated for the collection.

For more information, see [Smart Document Understanding](#).

Before you begin

The documents in your collection must contain a field with HTML representations of your tables. This information often is stored in the **html** field. If your collection consists of CSV or JSON files, it might have a field other than the **html** field that contains table information in HTML format.

Applying the table understanding enrichment

You can apply the enrichment only to a field that contains an HTML representation of the table.

To apply the enrichment, complete the following steps:

1. From the navigation pane, open the **Manage collections** page, and then click a collection to open it.
2. Click the **Enrichments** tab.
3. Find the **Table Understanding** enrichment.
4. Select the **html** field from the field list.

Choose the field that contains HTML representations of the tables.

After the enrichment is applied, you can get valid results when you submit queries that require Discovery to find information that is stored in tables.

A developer can query tables by using the API. For more information, see [Query parameters](#).

For more information about how to apply the table understanding enrichment by using the API, see [Applying enrichments by using the API](#).

Working with tabular data in Python

Use [Text Extensions for Pandas](#), an open-source library from IBM, to read the tables that were parsed from documents in Discovery into pandas DataFrame objects. A pandas DataFrame is an object that represents two-dimensional tabular data in a form that can be transformed and manipulated for downstream analysis in Python.

For example, you can extract content from tables in many annual report documents and reconstruct it into a single table that includes multiyear data points of interest. For more information, read the [Structured Information Extraction from Tables in PDF Documents with Pandas and IBM Watson](#) blog post on Medium.com.

Output schema

The output schema from the **Table Understanding** enrichment is as follows.

```
{  
  "tables": [  
    {  
      "location": {  
        "begin": int,  
        "end": int  
      },  
      "text": string,  
      "section_title": {  
        "text": string,  
        "location": {  
          "begin": int,  
          "end": int  
        }  
      },  
      "title": {  
        "location": {  
          "begin": int,  
          "end": int,  
        },  
        "text": string  
      },  
      "table_headers": [  
        {  
          "cell_id": string,  
          "location": {  
            "begin": int,  
            "end": int  
          },  
          "text": string,  
          "row_index_begin": int,  
          "row_index_end": int,  
          "column_index_begin": int,  
          "column_index_end": int  
        },  
        ...  
      ],  
      "column_headers": [  
        {  
          "cell_id": string,  
        }  
      ]  
    }  
  ]  
}
```

```

"location" : {
    "begin" : int,
    "end" : int
},
"text" : string,
"text_normalized" : string,
"row_index_begin" : int,
"row_index_end" : int,
"column_index_begin" : int,
"column_index_end" : int
},
...
],
"row_headers" : [
{
    "cell_id" : string,
    "location" : {
        "begin" : int,
        "end" : int
    },
    "text" : string,
    "text_normalized" : string,
    "row_index_begin" : int,
    "row_index_end" : int,
    "column_index_begin" : int,
    "column_index_end" : int
},
...
],
"body_cells" : [
{
    "cell_id" : string,
    "location" : {
        "begin" : int,
        "end" : int
    },
    "text" : string,
    "row_index_begin" : int,
    "row_index_end" : int,
    "column_index_begin" : int,
    "column_index_end" : int,
    "row_header_ids": [ string ],
    "row_header_texts": [ string ],
    "row_header_texts_normalized": [ string ],
    "column_header_ids": [ string ],
    "column_header_texts": [ string ],
    "column_header_texts_normalized": [ string ],
    "attributes" : [
        {
            "type" : string,
            "text" : string,
            "location" : {
                "begin" : int,
                "end" : int
            }
        },
        ...
    ]
},
...
],
"key_value_pairs": [
{
    "key": {
        "cell_id": string,
        "location": {
            "begin": int,
            "end": int
        },
        "text": string
    },
    "value": [
        {
            "cell_id": string,
            "location": {
                "begin": int,
                "end": int
            },
            "text": string
        },
        ...
    ]
}
]
...
]

```

```

],
"contexts": [
  {
    "text": string,
    "location": {
      "begin": int,
      "end": int
    }
  },
  ...
]
}

```

Schema arrangement

The schema is arranged as follows.

- **tables**: An array that defines the tables that are identified in the input document.
 - **location**: The location of the current table as defined by its **begin** and **end** indexes in the input document.
 - **text**: The textual contents of the current table from the input document without associated markup content.
 - **section_title**: If identified, the location of a section title contained in the current table. Empty if no section title is identified.
 - **text**: The text of the identified section title.
 - **location**: The location of the section title in the input document as defined by its **begin** and **end** indexes.
 - **title**: If identified, the title or caption of the current table of the form **Table x.: ...**. Empty when no title is identified. When present, the **title** is excluded from the **contexts** array of the same table.
 - **location**: The location of the title in the input document as defined by its **begin** and **end** indexes.
 - **text**: The text of the identified table title or caption.
 - **table_headers**: An array of table-level cells applicable as headers to all the other cells of the current table. Each table header is defined as a collection of the following elements:
 - **cell_id**: The unique ID of the cell in the current table.
 - **location**: The location of the cell in the input document as defined by its **begin** and **end** indexes.
 - **text**: The textual contents of the cell from the input document without associated markup content.
 - **row_index_begin**: The **begin** index of the cell's **row** location in the current table.
 - **row_index_end**: The **end** index of the cell's **row** location in the current table.
 - **column_index_begin**: The **begin** index of the cell's **column** location in the current table.
 - **column_index_end**: The **end** index of the cell's **column** location in the current table.
 - **column_headers**: An array of column-level cells, each applicable as a header to other cells in the same column as itself, of the current table. Each column header is defined as a collection of the following items:
 - **cell_id**: The unique ID of the cell in the current table.
 - **location**: The location of the cell in the input document as defined by its **begin** and **end** indexes.
 - **text**: The textual contents of the cell from the input document without associated markup content.
 - **text_normalized**: Normalized column header text.
 - **row_index_begin**: The **begin** index of the cell's **row** location in the current table.
 - **row_index_end**: The **end** index of the cell's **row** location in the current table.
 - **column_index_begin**: The **begin** index of the cell's **column** location in the current table.
 - **column_index_end**: The **end** index of the cell's **column** location in the current table.
 - **row_headers**: An array of row-level cells, each applicable as a header to other cells in the same row as itself, of the current table. Each row header is defined as a collection of the following items:
 - **cell_id**: The unique ID of the cell in the current table.
 - **location**: The location of the cell in the input document as defined by its **begin** and **end** indexes.
 - **text**: The textual contents of the cell from the input document without associated markup content.
 - **text_normalized**: Normalized row header text.
 - **row_index_begin**: The **begin** index of the cell's **row** location in the current table.
 - **row_index_end**: The **end** index of the cell's **row** location in the current table.
 - **column_index_begin**: The **begin** index of the cell's **column** location in the current table.
 - **column_index_end**: The **end** index of the cell's **column** location in the current table.
 - **body_cells**: An array of cells that are not table header or column header or row header cells, of the current table with corresponding row and column header associations. Each body cell is defined as a collection of the following items:

- **cell_id**: The unique ID of the cell in the current table.
- **location**: The location of the cell in the input document as defined by its **begin** and **end** indexes.
- **text**: The textual contents of the cell from the input document without associated markup content.
- **row_index_begin**: The **begin** index of this cell's **row** location in the current table.
- **row_index_end**: The **end** index of this cell's **row** location in the current table.
- **column_index_begin**: The **begin** index of this cell's **column** location in the current table.
- **column_index_end**: The **end** index of this cell's **column** location in the current table.
- **row_header_ids**: An array of values, where each value is the cell ID value of a row header that is associated with this body cell.
- **row_header_texts**: An array of values, where each value is the text from a row header for this body cell.
- **row_header_texts_normalized**: An array of values, where each value is the normalized text from a row header for this body cell.
- **column_header_ids**: An array of values, where each value is the cell ID value of a column header that is associated with this body cell.
- **column_header_texts**: An array of values, where each value is the text from a column header for this body cell.
- **column_header_texts_normalized**: An array of values, where each value is the normalized text from a column header for this body cell.
- **attributes**: An array that identifies document attributes. Each object in the array consists of three elements:
 - **type**: The type of attribute. Possible values are **Address**, **Currency**, **DateTime**, **Duration**, **Location**, **Number**, **Organization**, **Percentage**, and **Person**.
 - **text**: The text that is associated with the attribute.
 - **location**: The location of the attribute as defined by its **begin** and **end** indexes.
- **key_value_pairs**: An array that specifies any key-value pairs in tables in the input document. For more information, see [Understanding key-value pairs](#).
 - **key**: An object that specifies a key for a key-value pair.
 - **cell_id**: The unique ID of the key in the table.
 - **location**: The location of the key cell in the input document as defined by its **begin** and **end** indexes.
 - **text**: The text content of the table cell without HTML markup.
 - **value**: An array that specifies the value or values for a key-value pair.
 - **cell_id**: The unique ID of the value in the table.
 - **location**: The location of the value cell in the input document as defined by its **begin** and **end** indexes.
 - **text**: The text content of the table cell without HTML markup.
- **contexts**: A list of related material that precedes and follows the table, excluding its section title, which is provided in the **section_title** field. Related material includes related sentences; footnotes; and sentences from other parts of the document that refer to the table. The list is represented as an array. Each object in the array consists of the following elements:
 - **text**: The text contents of a related material from the input document, without HTML markup.
 - **location**: The location of the related material in the input document as defined by its **begin** and **end** indexes.

Notes on the table output schema

- Row and column index values per cell are zero-based and so begin with **0**.
- Multiple values in arrays of **row_header_ids** and **row_header_texts** elements indicate a possible hierarchy of row headers.
- Multiple values in arrays of **column_header_ids** and **column_header_texts** elements indicate a possible hierarchy of column headers.

Examples

The following table is an example table from an input document.

	Three months ended September 30,		Nine months ended September 30,	
	2005	2004	2005	2004
Statutory tax rate	35.0%	35.0%	35.0%	35.0%
IRS audit settlement	(97.9)%	(36.0)%	(58.4)%	(15.2)%
Dividends received deduction	(13.2)%	(3.3)%	(15.4)%	(4.7)%
Total effective tax rate	(76.1)%	(4.3)%	(38.8)%	15.1%

Figure 1. Table example

The table is composed as follows:

	Column header		Column header	
	Column header	Column header	Column header	Column header
<i>Row header</i>	Body cell	Body cell	Body cell	Body cell
<i>Row header</i>	Body cell	Body cell	Body cell	Body cell
<i>Row header</i>	Body cell	Body cell	Body cell	Body cell
<i>Row header</i>	Body cell	Body cell	Body cell	Body cell

Figure 2. Anatomy of the example table

The following syntax is used in the table:

- **Bold text** indicates a column header
- *Italic text* indicates a row header
- Unstyled text indicates a body cell

The output from service represents the example's first body cell (that is, the first cell in row 3 with a value of **35.0%**) as follows:

```
{
  "tables": [ {
    "location": {
      "begin": 872,
      "end": 5879
    },
    "text": "...",
    "section_title": {
      "text": "",
      "location": {
        "begin": 0,
        "end": 0
      }
    },
    "table_headers": [ ],
    "column_headers": [ {
      "cell_id": "colHeader-1050-1082",
      "location": {
        "begin": 1050,
        "end": 1083
      },
      "text": "Three months ended September 30",
      "text_normalized": "Three months ended September 30",
      "row_index_begin": 0,
      "row_index_end": 0,
      "column_index_begin": 1,
      "column_index_end": 2
    }, {
      "cell_id": "colHeader-1270-1301",
      "location": {
        "begin": 1270,
        "end": 1302
      },
      "text": "Nine months ended September 30",
      "text_normalized": "Nine months ended September 30",
      "row_index_begin": 0,
      "row_index_end": 0,
      "column_index_begin": 3,
      "column_index_end": 4
    }, {
      "cell_id": "colHeader-1544-1548",
      "location": {
        "begin": 1544,
        "end": 1548
      }
    } ]
  } ]
}
```

```
        "end" : 1549
    },
    "text" : "2005",
    "text_normalized" : "Year 1",
    "row_index_begin" : 1,
    "row_index_end" : 1,
    "column_index_begin" : 1,
    "column_index_end" : 1
}, {
    "cell_id" : "colHeader-1712-1716",
    "location" : {
        "begin" : 1712,
        "end" : 1717
    },
    "text" : "2004",
    "text_normalized" : "Year 2",
    "row_index_begin" : 1,
    "row_index_end" : 1,
    "column_index_begin" : 2,
    "column_index_end" : 2
}, {
    "cell_id" : "colHeader-1889-1893",
    "location" : {
        "begin" : 1889,
        "end" : 1894
    },
    "text" : "2005",
    "text_normalized" : "Year 1",
    "row_index_begin" : 1,
    "row_index_end" : 1,
    "column_index_begin" : 3,
    "column_index_end" : 3
}, {
    "cell_id" : "colHeader-2057-2061",
    "location" : {
        "begin" : 2057,
        "end" : 2062
    },
    "text" : "2004",
    "text_normalized" : "Year 2",
    "row_index_begin" : 1,
    "row_index_end" : 1,
    "column_index_begin" : 4,
    "column_index_end" : 4
} ],
"row_headers" : [ {
    "cell_id" : "rowHeader-2244-2262",
    "location" : {
        "begin" : 2244,
        "end" : 2263
    },
    "text" : "Statutory tax rate",
    "text_normalized" : "Statutory tax rate",
    "row_index_begin" : 2,
    "row_index_end" : 2,
    "column_index_begin" : 0,
    "column_index_end" : 0
}, {
    "cell_id" : "rowHeader-3197-3217",
    "location" : {
        "begin" : 3197,
        "end" : 3218
    },
    "text" : "IRS audit settlement",
    "text_normalized" : "IRS audit settlement",
    "row_index_begin" : 3,
    "row_index_end" : 3,
    "column_index_begin" : 0,
    "column_index_end" : 0
}, {
    "cell_id" : "rowHeader-4148-4176",
    "location" : {
        "begin" : 4148,
        "end" : 4177
    },
    "text" : "Dividends received deduction",
    "text_normalized" : "Dividends received deduction",
    "row_index_begin" : 4,
    "row_index_end" : 4,
    "column_index_begin" : 0,
    "column_index_end" : 0
}, {
    "cell_id" : "rowHeader-5106-5130",
```

```

"location" : {
    "begin" : 5106,
    "end" : 5131
},
"text" : "Total effective tax rate",
"text_normalized" : "Total effective tax rate",
"row_index_begin" : 5,
"row_index_end" : 5,
"column_index_begin" : 0,
"column_index_end" : 0
} ],
"key_value_pairs" : [ ],
"body_cells" : [ {
    "cell_id" : "bodyCell-2450-2455",
    "location" : {
        "begin" : 2450,
        "end" : 2456
    },
    "text" : "35.0%",
    "row_index_begin" : 2,
    "row_index_end" : 2,
    "column_index_begin" : 1,
    "column_index_end" : 1,
    "row_header_ids" : [ "rowHeader-2244-2262" ],
    "row_header_texts" : [ "Statutory tax rate" ],
    "row_header_texts_normalized" : [ "Statutory tax rate" ],
    "column_header_ids" : [ "colHeader-1050-1082", "colHeader-1544-1548" ],
    "column_header_texts" : [ "Three months ended September 30,", "2005" ],
    "column_header_texts_normalized" : [ "Three months ended September 30,", "Year 1" ],
    "attributes": [ ]
}, {
    "cell_id" : "bodyCell-2633-2638",
    "location" : {
        "begin" : 2633,
        "end" : 2639
    },
    "text" : "35.0%",
    "row_index_begin" : 2,
    "row_index_end" : 2,
    "column_index_begin" : 2,
    "column_index_end" : 2,
    "row_header_ids" : [ "rowHeader-2244-2262" ],
    "row_header_texts" : [ "Statutory tax rate" ],
    "row_header_texts_normalized" : [ "Statutory tax rate" ],
    "column_header_ids" : [ "colHeader-1050-1082", "colHeader-1712-1716" ],
    "column_header_texts" : [ "Three months ended September 30,", "2004" ],
    "column_header_texts_normalized" : [ "Three months ended September 30,", "Year 2" ],
    "attributes": [ ]
}, {
    "cell_id" : "bodyCell-2825-2830",
    "location" : {
        "begin" : 2825,
        "end" : 2831
    },
    "text" : "35.0%",
    "row_index_begin" : 2,
    "row_index_end" : 2,
    "column_index_begin" : 3,
    "column_index_end" : 3,
    "row_header_ids" : [ "rowHeader-2244-2262" ],
    "row_header_texts" : [ "Statutory tax rate" ],
    "row_header_texts_normalized" : [ "Statutory tax rate" ],
    "column_header_ids" : [ "colHeader-1270-1301", "colHeader-1889-1893" ],
    "column_header_texts" : [ "Nine months ended September 30,", "2005" ],
    "column_header_texts_normalized" : [ "Nine months ended September 30,", "Year 1" ],
    "attributes": [ ]
}, {
    "cell_id" : "bodyCell-3008-3013",
    "location" : {
        "begin" : 3008,
        "end" : 3014
    },
    "text" : "35.0%",
    "row_index_begin" : 2,
    "row_index_end" : 2,
    "column_index_begin" : 4,
    "column_index_end" : 4,
    "row_header_ids" : [ "rowHeader-2244-2262" ],
    "row_header_texts" : [ "Statutory tax rate" ],
    "row_header_texts_normalized" : [ "Statutory tax rate" ],
    "column_header_ids" : [ "colHeader-1270-1301", "colHeader-2057-2061" ],
    "column_header_texts" : [ "Nine months ended September 30,", "2004" ],
    "column_header_texts_normalized" : [ "Nine months ended September 30,", "Year 2" ]
}
]

```

```
    "attributes": [ ]  
},  
...  
],  
"contexts": [ ]  
}
```

Understanding key-and-value pairs

Tables sometimes contain key-and-value pairs that span multiple table cells. **Table Understanding** can detect the following types of tabular pairs.

- Simple key-and-value pairs in adjacent cells, as in the following example table:

Key	Value
Item number	123456789
Date	1/1/2019
Amount	\$1,000

Basic table

- Key-and-value pairs in the same cell, as in the following example table:

Key-value pairs	Key-value pairs
Item number: 123456789	Amount: \$1000
Date: 1/1/2019	Address: 123 Anywhere Dr

Complex table

Use SDU to analyze document structure

Analyzing documents based on their structure

Create a model that understands the content of a document based on the document's format and structure.

First, decide whether you want to use a pretrained model or define your own.

Pretrained model

Applies a noncustomizable model that extracts text and identifies tables, lists, and sections.

Instead of training the model yourself, you can apply an existing model that is trained to identify tables, lists, and sections in various types of documents.

If capturing information from tables is critical to your use case, consider using a pretrained model.

For more information, see [Apply a pretrained SDU model](#).

User-trained model

Opens the Smart Document Understanding tool that you can use to pick certain types of text to store in fields other than the **text** field.

When you label a section of a document as a custom field, later you can apply enrichments to the field or split your documents on each occurrence of the field. You can search or filter by the field, or omit the field from the index.

For more information, see [Define a user-trained SDU model](#).

Text extraction only

Indexes any text that is recognized in the source documents in the **text** field. This option is used by default.

Define a user-trained SDU model

Create a Smart Document Understanding (SDU) model that learns about the content of a document based on the document's structure.

Use the Smart Document Understanding tool to add custom fields to a collection so you can do the following things:

- Target prebuilt or custom enrichments at specific sections of a document.
- Break large documents into smaller documents.

For help with deciding whether SDU can help your use case, read [When to use Smart Document Understanding](#).

If capturing information from tables is critical to your use case, consider using a pretrained model. For more information about creating a pretrained SDU model, see [Apply a pretrained SDU model](#).

When to use Smart Document Understanding

The Smart Document Understanding (SDU) tool works better with some project types.

- The tool is most beneficial when used with **Document Retrieval** projects. Use the tool to break your documents into smaller, more consumable chunks of information. When you help Discovery index the correct set of information in your documents, you improve the answers that your application can find and return.

For example, your documents might contain tips that are shown in sections with an H4 heading. If you want to extract the information from these tips separately, you can add a field that is named **tips**, and teach the model to recognize it. After you apply the model to your collection, you can apply an enrichment to the **tips** field only. Later, you can limit the search to return content from only the **tips** field.

Or maybe you have extra large documents that contain subsections. You can teach the SDU model to recognize these subsections, and then split the large document into multiple, smaller, and easier-to-manage documents that begin with one of these subsections.

- The best way to prepare a collection for use in **Conversational Search** projects is to identify discrete question-and-answer pairs. You can use the SDU tool to find and annotate them. If you configure the project to contain answers in an answer field, you must update the search configuration in Watson Assistant to get the body of the response from the custom answer field.
- A pretrained SDU model is applied to **Document Retrieval for Contracts** projects automatically. The pretrained SDU model knows how to recognize terms and concepts that are significant to contracts. As a result, you cannot apply a user-trained SDU model to this project type, but you also don't need to.
- The SDU tool is rarely used with **Content Mining** projects.

You can use the SDU tool to annotate the following file types only:

- Image files (PNG, TIFF, JPG)
- Microsoft PowerPoint
- Microsoft Word
- PDF

For a complete list of file types that Discovery supports, see [Supported file types](#).

The Smart Document Understanding tool uses optical character recognition (OCR) to extract text from images in the files that it analyzes. Images must meet the minimum quality requirements that are supported by OCR. For more information, see [Optical character recognition](#).

The tool cannot read documents with the following characteristics; remove them from your collection before you begin:

- Documents that appear to have text that overlays other text are considered **double overlaid** and cannot be annotated.
- Documents that contain multiple columns of text on a single page cannot be annotated.



Note: When you build a custom Smart Document Understanding model, the conversion time for your collection can increase due to the resources that are required to apply the AI model to your documents.

Start with representative documents

Documents come in all shapes and sizes. Your collection might have a mix of different document structures. Smart Document Understanding works best when the documents in a single collection have similar style characteristics. For example, the documents use consistent font sizes and colors for titles and headers, and tables in the document have similar layouts. To create the best model for your collection, take this prerequisite step:

1. Review your documents to look for style and layout patterns, and then separate the documents into groups based on their style.

For example, if your data contains documents that follow four different formatting styles, break the documents into four separate collections, one for each style. Add documents with a uniform layout and style to each collection. A good target size per collection

is 40 documents.

2. Use the SDU tool to annotate this representative set of documents and train Watson to recognize custom content in your data.
3. Apply the custom SDU model to the full collection. For more information, see [Reusing SDU models](#).

Creating the model

To apply a user-trained Smart Document Understanding model to your collection, complete the following steps:

1. Open the **Manage collections** page from the navigation panel.
2. If your project has more than one collection, select the collection with documents that you want to annotate.
3. Open the **Identify fields** page.
4. Choose **User-trained models**.

The **Text extraction only** option is used by default. With this model, any text that is recognized in the source documents is indexed in the **text** field.

5. Click **Submit**, and then click **Apply changes and reprocess**.

A subset of documents is available for you to annotate. A set of 20 - 50 documents is displayed in a list. The number of documents that are available differs based on several factors, including the overall number of documents in your collection and how many of them are supported file types.

Labeling video

The following video shows you how to select a label, and then apply it to a representation of the text in your document.

In the video, the user clicks the **title** field label, and then clicks the text block that represents the **Table of Contents** page title to label the text as a title. Next, the user clicks the **table_of_contents** field label and selects the table of contents text block to label it. Then, the user clicks the **footer** field label and clicks the text block that represents the page footer. After the text is labeled, the user clicks the **Submit page** button.

Your browser does not support the video tag.

Labeling the documents

Before you begin, get a feel for the structure of the document you plan to annotate. Are there subtitled sections that you want Discovery to return per answer? If so, identify all subtitles. Later you can split the document into discrete subdocuments, each starting with a subtitle. For more information, see [When to use Smart Document Understanding](#).

To label documents, complete the following steps:

1. Review the document preview.

A view of the original document is displayed along with a representation of the document, where the text is replaced by blocks.

The blocks are all the color of the **text** field label because all of the current text is considered to be standard text and will be indexed in the **text** field.

Label blocks that represent specific types of information, such as titles or page footers, with other field labels. For example, when you apply the title field label to a document title that would otherwise be indexed as text, you are defining a more precise representation of the document content.

The process of using labels to identify different parts of the document's structure is called **annotating** the document.

2. Review the field labels that you can use to annotate the document. They are displayed in the **Field labels** panel.

See the [Default field labels](#) table for a list of the fields and their descriptions.

3. To create a custom field label, click **Create new**.

- Specify a field label with no spaces. For example, **complex_task** is a valid field label.



Note: Avoid using a field label name or including characters, such as a number sign (#) or period (.), in the name that have special meaning for Discovery. For more information, see [How fields are handled](#).

- If you want to change the color that is used to represent the field, repeatedly click the color block until it is displayed in the color that you want to use.



Important: You cannot change the field label color later.

- Click **Create**.
4. First, click a field label to activate it.
 5. Next, click the block that represents the content that you want to label as the field type.

The block changes to the color of the field label. You successfully labeled the field!
 6. Repeat this process to annotate more fields in the document.
- Don't worry. You don't need to label every page. As you apply labels and submit pages, Watson learns from what you annotate and starts to predict annotations.
- Follow these guidelines:
- If there's nothing special about a section, leave it labeled as **text**, which is applied by default.
 - A label cannot span multiple pages.
 - Do not treat **bold**, **italic**, or underlined text differently. Label based on the context, not the style.
 - Use consistent labeling on all documents.
 - Work from the first page of a multipage document to the last.
 - To remove a single annotation, choose another label (such as **text**) and apply it to the item to overwrite the previous annotation.
 - To remove annotations that you added to an entire page, click the **Clear changes** icon in the toolbar.
 - To annotate a table, click the text at the start of the table and then drag to select the text in the entire table.
 - When you label one or more tables, the **Table Understanding** enrichment is enabled for the entire collection automatically. For more information, see [Understanding tables](#).
 - Images from the source documents are not rendered in the preview. If Optical Character Recognition (OCR) is enabled, any text from the image or diagram is extracted and rendered in the preview.
 - Do not label white space.
7. When everything that you want to label is labeled, submit the page. Click **Submit page**.



Note: Continue annotating documents until Watson can correctly and consistently map different types of content to the appropriate fields for you.

8. After you teach Watson to identify fields, click **Apply changes and reprocess**.

Custom fields that you define by using the SDU tool are indexed as root-level fields.

What to do next

When you build a user-trained model, you change where information is stored in your documents. Next, change how the search results are configured. By default, search results are retrieved from passages or the text field. You might have a better field to use as the source of the result body. For more information, see [Changing the result content](#).

If your project is being used by a virtual assistant, update the search skill configuration to pull the answer body from a different field. For more information, see [Configure the search](#).

You can apply enrichments, either custom or prebuilt enrichments, to the new root fields that are generated by the SDU model.

If you want to return shorter text snippet with a search result, you can split your documents based on one of the new fields that you defined, such as chapter or section.

Available fields

The following fields are available for you to apply to documents by using the Smart Document Understanding tool.

The fields are arbitrary. You can apply the **image** field to every title in the document if you want. Although, it might be difficult to know which field to search later for information that you need if the field names don't match the content. The default set are representative field types that are meant to help you get started. Only the **text** and **table** fields have special significance. Do not use them to identify anything other than text and tables.

Field	Definition
answer	In a question-and-answer pair (often in an FAQ), the answer to the question.
author	Name of author or authors.

footer	Use this tag to denote meta-information about the document (such as the page number or references), that appear at the end of the page.
header	Use this tag to denote meta-information about the document that appears at the start of the page.
question	In a question-and-answer pair (often in an FAQ), the question.
subtitle	The secondary title of the document.
table_of_contents	Use this tag on lists in the document table of contents.
text	By default, every block of text in the document is labeled as text. Apply different labels only to blocks of text with special meaning.
title	The main title of the document.
table	Use this tag to annotate tables in your document.
image	Images are not shown in the document preview. If you enable OCR, text from an image or diagram is displayed in the preview instead. If you want to prevent text from some images from being included in search results, tag the image text as an image. You can exclude the image field from the index later.

Default field labels

Reusing SDU models

After you define a model with the SDU tool, you can save it and reuse it in other collections by exporting it from one collection and importing it to another.

To reuse a model, complete the following steps:

1. Export the model that you want to reuse. From the SDU toolbar menu, select **Export model**.



Figure 1. Import and export menu

2. Create the collection where you want to reuse the model. Add only one document to the collection at first.
3. Import the model from the SDU toolbar. The exported model has a file extension of **.sdumodel**.
4. Add the rest of the documents to the collection. Open the **Activity** tab of the **Manage collections** page, and then click **Upload data** to add more files to the collection.

Use the imported model as-is. Do not make any more annotations. If you make annotations after you import the **.sdumodel** file, the imported model will be overwritten.

Smart Document Understanding limits

The number of custom fields that you can create per Smart Document Understanding model depends on your Discovery plan type.

Plan	Custom fields per SDU model
Cloud Pak for Data	Unlimited
Premium	100
Enterprise	100
Plus (includes Trial)	40

Custom field limits

The maximum number of documents that you can annotate to train an SDU model per collection depends on your Discovery plan type.

Plan	Documents per collection
Cloud Pak for Data	40
Premium	40
Enterprise	40
Plus (includes Trial)	40

Training set limits

Managing fields

The **Manage fields** tab contains several options:

Identify fields to index

For more information, see [Excluding content from query results](#).

Improve query results by splitting your documents

For more information, see [Split documents to make query results more succinct](#).

Date format settings

For more information, see [Date format settings](#).

To access the **Manage fields** page, click the **Manage collections** icon on the navigation panel and open a collection. Click the **Manage fields** tab. For more information about collections, see [Creating collections](#).

Apply a pretrained SDU model

Apply a prebuilt Smart Document Understanding (SDU) model that can extract text and is trained to identify tables, lists, and sections in documents.

Use the pretrained model if your documents contain tables with valuable information that you want to capture. The model is also able to preserve the meaning inherent in the nesting structure of tables, lists, and sections. Using the pretrained model speeds up the process of capturing information from the structure of a document.

If you want to customize how the document structure is used to infer meaning from a document or you want to split documents with a field that is generated by an SDU model, create a user-trained model instead. For more information, see [Define a user-trained SDU model](#).

A pretrained model is applied to **Document Retrieval for Contracts** projects automatically. Instead of you annotating contract-related content in your documents, the project applies a model that already knows how to recognize terms and concepts that are significant to contracts.

Preparing documents

You can apply a pretrained SDU model to the following file types only:

- Image files (PNG, TIFF, JPG)
- Microsoft PowerPoint
- Microsoft Word
- PDF

For a complete list of file types that Discovery supports, see [Supported file types](#).

The Smart Document Understanding tool uses optical character recognition (OCR) to extract text from images in the files that it analyzes. Images must meet the minimum quality requirements that are supported by OCR. For more information, see [Optical character recognition](#).

The tool cannot read documents with the following characteristics; remove them from your collection before you begin:

- Documents that appear to have text that overlays other text are considered **double overlaid** and cannot be annotated.
- Documents that contain multiple columns of text on a single page cannot be annotated.



Note: When you apply a Smart Document Understanding model, the conversion time for your collection can increase due to the resources that are required to apply the AI model to your documents.

Apply a pretrained model

To apply a pretrained Smart Document Understanding model to your collection, complete the following steps:

1. Open the **Manage collections** page from the navigation panel.
2. Select the collection to which you want to apply the model.
3. Open the **Identify fields** page.
4. Choose **Pre-trained models**

The **Text extraction only** option is used by default. With this model, any text that is recognized in the source documents is indexed in the **text** field.

5. Click **Submit**, and then click **Apply changes and reprocess**.

Understanding the output

If the SDU model finds and processes a structure, such as a table, in the document, it stores a representation of the structure in a field named **enriched_{field}**, where **{field}** is the field where the structure was stored.

The following excerpt shows the JSON representation of a table from the **enriched_html** field of a document that was processed by the pretrained SDU model.

The screenshot shows a user interface for 'Identified elements'. On the left, there's a sidebar with 'Entities v2' and a list of entity types: Organization (76), JobTitle (51), Number (51), Date (29), Person (18), and a 'Show all' link. On the right, the main area displays a JSON tree under the heading 'enriched_html'. The tree shows a complex nested structure of tables, rows, and cells. A specific section of the JSON is highlighted in yellow, showing the following code:

```
        "enriched_html" : [ 1 item
          0 : { 1 item
            "tables" : [ 7 items
              0 : { 10 items
                "section_title" : {...} 2 items
                "row_headers" : [...] 9 items
                "table_headers" : [] 0 items
                "location" : {...} 2 items
                "text" :
                  "Using this Guide 06 Introduction 08 Our Organizational
                  Strategy 12 Focus Areas 16 Build Diverse Teams 20 Sponsor
                  Underrepresented 26 Designers Design Inclusive Culture 32
                  Closing 40 Foundational Terms 44 References 50 "
                "body_cells" : [...] 9 items
                "contexts" : [...] 3 items
                "key_value_pairs" : [] 0 items
                "title" : {} 0 items
                "column_headers" : [...] 4 items
              }
              1 : {...} 10 items
              2 : {...} 10 items
              3 : {...} 10 items
            ]
          }
        ]
```

Figure 1. JSON table representation

If you want to extract text from the processed structure, you can use the **location** field to find the index values that identify where the text string starts and ends.

For more information about the structure of indexed tables, see [Understanding tables](#).

Troubleshooting issues

Follow these workarounds if you experience problems when working with the Smart Document Understanding tool.

Insufficient resources to process document

Error

When you apply a pretrained model to your collection, document processing does not complete successfully and an **Insufficient resources to process document** message is displayed.

Cause

The error is displayed because out of memory errors occur during the parse, structure identification, or assembly phases of the process that builds the machine learning model. Resources are insufficient when one or more of the documents in your collection are too large or have too many complex tables for the tool to handle.

Solution

Review your collection for large documents or documents with many tables and break them up into more smaller documents before you apply the pretrained model to the collection. Exact limits differ based on the complexity of your documents. Generally,

split documents that are over 400 pages long and avoid including more than 20 complex tables in a single document.

Manage enrichments

Apply enrichments to fields in your documents to make meaningful information easier to find and return in searches.

You typically apply enrichments to fields at the time that you create the enrichment. However, you can apply enrichments to fields later. You might want to apply an existing enrichment to a new custom field that you create by using the SDU tool, for example. You can remove enrichments that you applied to fields also.

For more information about available enrichments, see the following topics:

- [Adding domain-specific resources](#)
- [Applying prebuilt enrichments](#)

To manage enrichments, complete the following steps:

1. From the navigation pane, open the **Manage collections** page, and then click a collection to open it.
2. Click the **Enrichments** tab.

A list of available enrichments is displayed.

You can identify built-in enrichments because they are categorized as type **System**. The list also includes any custom enrichments that were created in any of the projects in your service instance.

3. Find the enrichment that you want to apply or remove.
4. Click the twistie in the associated field to expand a list of fields.
5. Do one of the following things:

- To apply an enrichment to documents, select the field or fields that you want to enrich. You can apply enrichments to the **text** and **html** fields, and to custom fields that were added from uploaded JSON or CSV files or from the Smart Document Understanding (SDU) tool.

 **Note:** If the field that you choose comes from a JSON file, after you apply the enrichment, the field data type is converted to an array. The field is converted to an array even if it contains a single value. For example, `"field1": "Discovery"` becomes `"field1": ["Discovery"]`. Only the first 50,000 characters of a custom field from a JSON file are enriched.

- To remove an enrichment, clear the checkboxes for fields that you want to remove the enrichment from.
6. Click **Apply changes and reprocess** to apply your changes to the collection.

Deleting an enrichment

You can delete a custom enrichment that you built to teach Discovery about your service. Custom enrichments include dictionaries, regular expressions, patterns, and so on. For more information, see [Adding domain-specific resources](#).

You cannot delete a prebuilt enrichment. Prebuilt enrichments include the Natural Language Understanding enrichments, such as the Entities enrichment, that are built into the product. To determine which enrichments are built in, check the **Type** column of the **Enrichments** page for a collection. Prebuilt enrichments have the **System** type.

To delete a custom enrichment, complete the following steps:

1. Open a project that uses the custom enrichment.
2. Click **Manage collections**, and then open a collection where the enrichment is in use.
3. From the **Enrichments** page of a collection, remove the enrichment from any fields where it is applied.
4. Click **Apply changes and reprocess**, and then wait for the system to process the documents in your collection without the enrichment.
5. Repeat the previous step for every collection in every project where the enrichment is used.

Remember, a custom enrichment can be used by any collection in any project within a single Discovery service instance.

6. From any of the collections that used the enrichment, open the **Enrichments** page, and then click the delete icon to delete the enrichment.

The custom enrichment is removed from the **Enrichments** list everywhere in this service instance.

Using the API to manage enrichments

To apply an enrichment to data by using the API, complete the following steps:

1. First, you must know the unique ID of the enrichment that you want to apply. For more information, see [Enrichment IDs](#).
2. Use the **Create collection** or **Update collection** methods to apply an enrichment to the documents in a collection. For more information, see [Apply enrichment by using the API](#)

Enrichment IDs

If you want to apply a custom enrichment that you created for one collection to another collection, you must know the unique ID that was generated for the enrichment when it was created. Use the API to [list enrichments](#) from the project where the custom enrichment is in use. The list that is returned includes enrichment ID information.

For prebuilt enrichments, the unique IDs do not change. The following table lists the IDs that are associated with each prebuilt enrichment type and identifies the collection languages for which the enrichment is supported. An enrichment cannot be applied to a collection unless the collection language is supported by the enrichment.

Name	Enrichment ID	Supported languages
Contracts	701db916-fc83-57ab-0000-000000000014	en
Entities	701db916-fc83-57ab-0000-00000000001e	ar, de, en, es, fr, it, ja, ko, nl, pt, zh-CN
Keywords	701db916-fc83-57ab-0000-000000000018	ar, de, en, es, fr, it, ja, ko, nl, pt, zh-CN
Part of Speech	701db916-fc83-57ab-0000-000000000002	All supported languages
Sentiment of Document	701db916-fc83-57ab-0000-000000000016	ar, de, en, es, fr, it, ja, ko, nl, pt, zh-CN
Table Understanding	701db916-fc83-57ab-0000-000000000012	All supported languages

Prebuilt enrichment IDs

For more information about all of the supported languages, see [Language support](#).

Apply enrichments by using the API

To apply an enrichment by using the API, complete the following steps:

1. Determine the base URL for the endpoint and the token or API key for your deployment.

For more information, see [Building custom applications with the API](#).

2. Get the project ID for your project.

From the product user interface, go to the **Integrate and Deploy > View API information** page, and then copy the project ID.

3. If you don't know the ID of the collection that you want to apply the enrichment to, get a list of your collections to find it.

For example:

```
GET $authentication $url/v2/projects/$project_id/collections?version=2019-11-22
```

The **collection_id** is returned.

4. Send a GET request to return the configuration of the collection that lists the applied enrichments.

For example:

```
GET $authentication $url/v2/projects/$project_id/collections/$collection_id?version=2019-11-22
```

For the enrichment IDs for the prebuilt enrichments, see [Enrichment IDs](#).

5. Add the enrichment that you want to apply.

For example, to add the **Keywords** enrichment, you can include the enrichment in the enrichments list. First, get its ID from the table.

The Keywords enrichment ID is **701db916-fc83-57ab-0000-000000000018**. To indicate that you want to apply the Keywords enrichment to the content in the **text** field of the documents in the collection, you can represent it in JSON format as follows:

```
{  
  "enrichment_id" : "701db916-fc83-57ab-0000-000000000018",  
  "fields" : [ "text" ]  
}
```

Note: Any enrichments that you specify replace the default enrichments. Therefore, if you want to retain a default enrichment, don't forget to include it in the list of enrichments that you apply to the collection. For a list of default enrichments per project type, see [Default enrichments per project type](#).

For example, to retain the **Entities** enrichment and add the **Keywords** enrichment, you might specify the following in the request body.

Note: The **Part of Speech** enrichment is included also because it is applied to all collection automatically.

```
"enrichments": [  
  {  
    "enrichment_id": "701db916-fc83-57ab-0000-000000000002",  
    "fields": [  
      "text"  
    ]  
  },  
  {  
    "enrichment_id": "701db916-fc83-57ab-0000-00000000001e",  
    "fields": [  
      "text"  
    ]  
  },  
  {  
    "enrichment_id": "701db916-fc83-57ab-0000-000000000018",  
    "fields": [  
      "text"  
    ]  
  }  
]
```

6. Submit the updated JSON request body with the [update collection](#) method to apply the enrichment to your collection.

For example:

```
POST $authentication -d '$requestBody' $url/v2/projects/$project_id/collections/$collection_id?version=2019-11-22
```

Default enrichments per project type

Some prebuilt enrichments are applied automatically to collections in a project based on the project type. The following table shows the default enrichments that are applied to each project type.

Enrichment	Document Retrieval	Document Retrieval for Contracts	Conversational Search	Content Mining
Contracts	✓			
Entities	✓	✓		
Keywords				
Part of Speech	✓	✓	✓	✓
Sentiment of Document				
Table Understanding	✓			

Default enrichments per project type

Testing and sharing your project

As you improve your project, periodically test how enrichments and search setting changes impact the query results.

For all project types except Conversational Search, you can see the fields that are associated with an indexed document by looking at the JSON view of a document that is returned by a query. Checking the JSON structure of a document can be useful if you want to check whether certain types of information are being captured.

After you enrich your collection, you can use the JSON view of a query result to check whether your enrichments are being applied and retrieved properly. For example, you can check the JSON to confirm that a synonym that you defined in a dictionary is being tagged as an occurrence of the corresponding dictionary term.

To test your project, complete the following steps:

1. From the navigation panel, open the **Improve and customize** page.
2. Retrieve query results by doing one of the following things:
 - **Content Mining** project: Choose or add a facet to apply to the documents, and then click **View filtered documents**. To analyze your data in more depth, open the Content Mining application in a new window or tab by clicking **Launch application**.
 - Other project types: Enter a test query to submit or leave the field empty and press Enter to submit an empty query.
3. From the query result list, click the link to view the document.

A representation of the original document is displayed.

4. IBM Cloud Click **Open advanced view** to see useful summary information, such as the number of occurrences of any enrichments that are detected in the document.

Optional: Select an enrichment to highlight every occurrence of the element within the document text.

For a **Document Retrieval for Contracts** project, the **Contract Data** page is displayed. For more information about Contract filter options, see [Understanding contracts](#).

5. You can learn more about information that is identified by the enrichments that are applied to your documents by reviewing the JSON representation of a document that is returned in a search result.

IBM Cloud

1. Click the **Display options** menu from the advanced view header, and then select **JSON**.

IBM Cloud Pak for Data

1. Click **JSON**.

For a **Document Retrieval for Contracts** project, click the **Contract Data** tab. For more information about Contract filter options, see [Understanding contracts](#).

Sharing your project

Try out your project and share it with others on your team for testing purposes. A test implementation of your project is created and hosted by IBM. Use this application preview to test your search results.

To preview and share your project, complete the following steps:

1. From the **Integrate and Deploy > Preview Link** page, follow the instructions to give your team members access to your project. (In Content Mining projects, the page is named **Share Link**.)

IBM Cloud For more information about access in IBM Cloud, see [Managing access to resources](#).

2. Click the copy icon for the **Copy Link** field to copy the URL of the preview application.
3. Paste the URL into a web browser to test it yourself or send the URL to team members.

Don't forget to send any login credentials that are needed to access the project when you send the link to your colleagues.

Querying your data programmatically

Building custom applications with the API

Use the Discovery API to build a custom application or component that searches your data.

Service API Versioning

API requests require a version parameter that takes a date in the format `version=YYYY-MM-DD`. Whenever a backwards-incompatible API change occurs, a new minor version of the API is released.

Send the version parameter with every API request. The service uses the API version for the date you specify, or the most recent version before that date. Don't default to the current date. Instead, specify a date that matches a version that is compatible with your app, and don't change it until your app is ready for a later version.

The current version is `2023-03-31`.

Getting your project ID IBM Cloud

 **Important:** This information applies to IBM Cloud only.

To use the API, you must construct the URL to use in your requests. Many of the API methods require the project ID.

1. From the [IBM Cloud Resource list](#), expand **AI/Machine Learning**, and then find the service page for your Discovery service instance.
2. From the **Credentials** section, copy the URL. You specify this value as the `{url}` in your API requests.
3. While you're on this page, copy the API key. You specify this value as the `{apikey}`.
4. Open your project in Discovery, and then go to the **Integrate and deploy > API Information** page.
5. Copy the project ID. You specify this value as the `{project_id}`.

If you're using a Content Mining project, stay on the **Share Link** page. From the web browser's **location** field, copy the URL starting with `/projects`. For example, `projects/a8ce5fed-7f33-4405-aa4b-88ffba322712/deploy/beta`. The ID that is specified after the `/projects/` segment of the URL is your project ID.

6. Construct a request URL by using the IDs you copied.

For example, the following request lists the collections in the project:

```
curl -X {request_method} -u "apikey:{apikey}" \
"{url}/v2/projects/{project_id}/collections?version=2019-11-29 -k"
```

To get the `{collection_id}`, you can use the [List collections](#) API method. Alternatively, open the collection in the product user interface, and then copy the collection ID, which is displayed after the `/collections/` segment of the page URL, from the web browser location field.

Using the API from Cloud Pak for Data IBM Cloud Pak for Data

 **Important:** This information applies to Discovery for Cloud Pak for Data only.

To use the API, you must construct the URL to use in your requests.

1. From the IBM Cloud Pak for Data web client main menu, expand **Services**, and then click **Instances**.
2. Find your instance, and then click it to open its summary page.
3. Scroll to the **Access information** section of the page, and then copy the URL. You will specify this value as the `{url}`.
4. Copy the bearer token also. You will need to pass the token when you make an API call.
5. From the launched application instance, go to the **Integrate and Deploy > API Information** page.
6. Copy the project ID. You will specify this value as the `{project_id}`.

If you're using a Content Mining project, stay on the **Share Link** page. From the web browser's **location** field, copy the URL starting with `/projects`. For example, `projects/a8ce5fed-7f33-4405-aa4b-88ffba322712/deploy/beta`. The ID that is specified after the `/projects/` segment of the URL is your project ID.

7. Construct a request URL by using the IDs you copied.

For example, the following request lists the collections in the project:

```
curl -H "Authorization: Bearer {token}" \
"{url}/v2/projects/{project_id}/collections?version=2019-11-29 -k"
```

 **Important:** The bearer token that is generated for an administrator can access any instance regardless of the access settings that are configured for the instance.

The bearer token expires after 12 hours. For more information about customizing the length of a session, see [Setting the idle session timeout](#).

Next steps

A developer can make the following enhancements:

- Use the API to [define more complex queries with the Discovery Query Language](#).
- Specify the exact document to return in response to a specific query with [curations](#).
- Process documents without storing them in a collection by [using the Analyze API](#).

Query API

Query overview

IBM Watson® Discovery offers powerful content search capabilities through search queries.

To retrieve data from Discovery after it is ingested, indexed, and enriched, submit a query.

As data is added to Discovery, a representation of each file is stored in the index as a JSON-formatted document. Enrichments that are applied to your collections identify meaningful information in the data and store it in new fields in these documents. To search your data, submit a query to return the most relevant documents and extract the information you're looking for.

Query types

Discovery accepts one of the following supported query types:

Query

Finds documents with values of interest in specific fields in your documents. Queries of this type use Discovery Query Language syntax to define the search criteria.

Parameter name: `query`

Natural Language Query (NLQ)

Finds answers to queries that are written in natural language. NLQ requests accept a text string value.

Parameter name: `natural_language_query`

Along with the query that you specify by using one of the supported query types, you can include one or both of the following parameters. The values for these parameters are also specified by using the Discovery Query Language (DQL) syntax:

- `filter`
- `aggregation`

For more information about the Discovery Query Language, see [DQL overview](#).

Queries that are submitted from the product user interface are natural language queries. A few other supported parameters are specified and given default values based on the project type in use. For more information, see [Default query settings](#).



Note: Discovery does not log query request data. You cannot opt in to request logging.

Choosing the right query type

The following table summarizes the capabilities that are supported for each query type. Use it to help you determine which type of

query to submit.

Goal	Natural Language Query (NLQ)	Discovery Query Language (DQL)
Return passages from documents	✓	✓
Highlight terms in responses (unless passages per document is enabled)	✓	✓
Define custom stop words or query expansions	✓	✓
Search specific document fields or enrichments		✓
Use operators, such as boolean clauses in the search		✓
Enable spelling correction	✓	
Add curations to return hardcoded answers to certain questions	✓	
Use relevancy training	✓	
Enable answer finding to return a succinct answer from a passage	✓	
Use table retrieval	✓	

Query types comparison

Query analysis

When you submit a query, the query text string is analyzed. During query analysis, the root (or lemma) of each key term in the query is identified. Any stop words that occur in the original query string are removed and synonym expansions that are defined for any terms that occur in the original query string are added. This enhanced version of the query is what gets submitted to Discovery.

The same analysis is performed on all queries, whether they are submitted as natural language queries or by using Discovery Query Language syntax.

Query flow

The following diagram shows a conceptual illustration of how a search request is handled by Discovery.

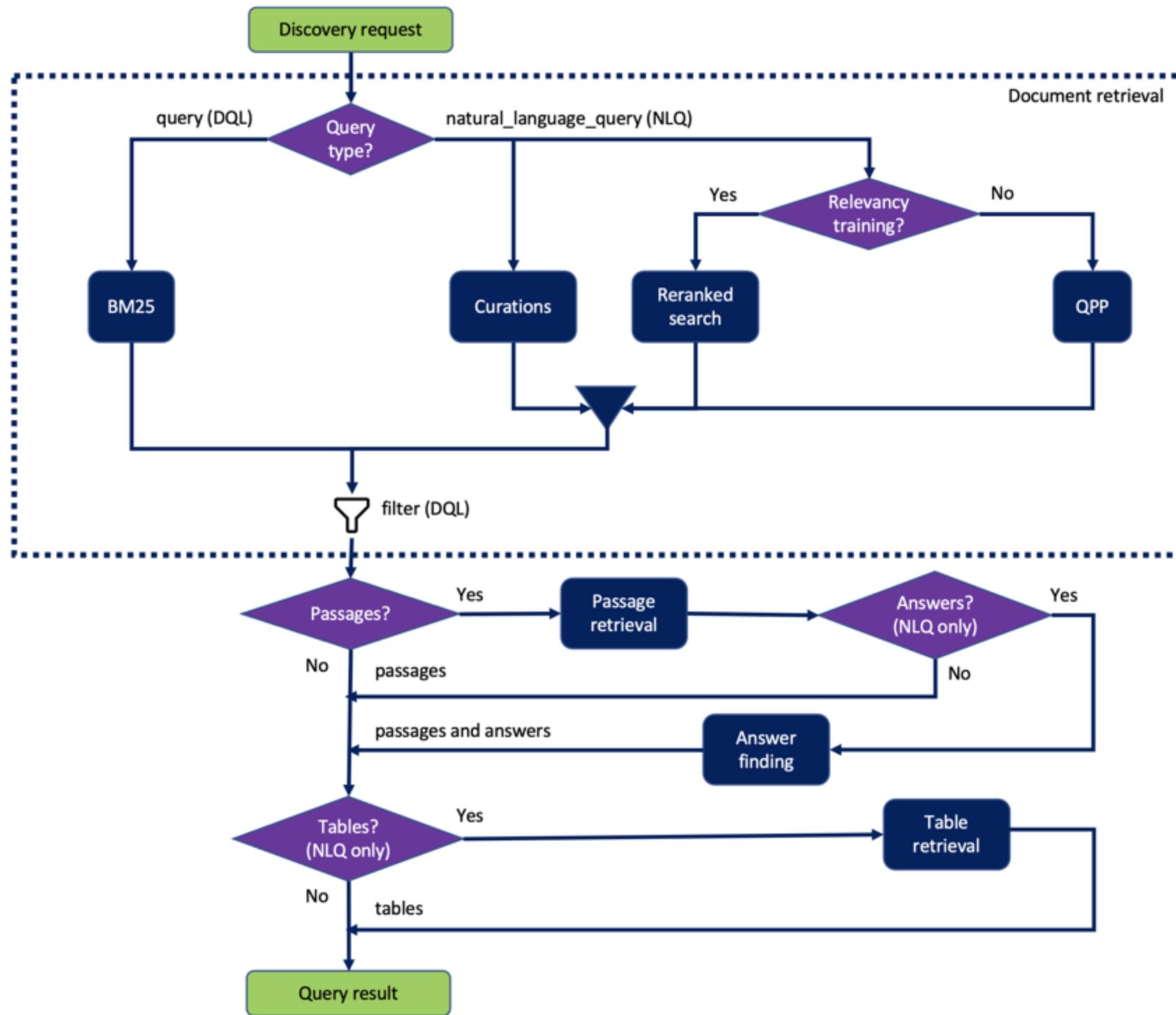


Figure 1. Flow chart that shows the processes that are used for Natural Language Queries versus Discovery Query Language queries

The following processes are shown in the flow diagram:

BM25

Uses Best Match 25 (a probabilistic information retrieval algorithm) to compute a relevance score for each document returned by search. The diagram shows that BM25 is applied to document results from the query requests, but it is not limited to query requests. It also is used along with other techniques as part of the relevancy training ranker process that is applied to natural language query results.

Curations

If the natural language query matches a predefined curation query, then certain documents and possibly a hardcoded snippet are returned. There is no query parameter to enable a curation. For curations to be used, you must define them programmatically ([Create curation method](#)). The output of any curations is merged with the output of the Relevancy training ranker or QPP results.

Relevancy training

A model that you can optionally define and apply to a project to score documents for relevance. There is no query parameter to enable relevancy training. For relevancy training to be used, you must successfully train the project either programmatically ([Create training query method](#)) or by using the product user interface.

QPP

A Query Performance Prediction algorithm that, given a query and a list of top results, produces a score that determines how relevant a document is. Used only if no Relevancy training ranker is available.

filter

The **filter** parameter can be passed along with **query** and **natural_language_query** requests to remove documents that don't meet certain criteria from the result set. The filter is shown as the last step within the document retrieval phase. However, it is used at different times in the flow. Its placement in the diagram is chosen to emphasize the fact that any documents that don't match the filter definition are excluded from the result set. The exclusion applies even to documents that might be specified in a curation.

Passage retrieval

Returns passages from documents when the `passages.enabled=true` parameter is included with a natural language query request.

Answer finding

When the `passages.find_answers=true` parameter is included with a natural language query request, returns succinct answers from passages along with the passages that are extracted from documents. If answer finding is enabled, then the final confidence score for each search result is a combination of the confidence scores from answer finding, passage retrieval, and QPP or Reranked search, whichever method is used.

Table retrieval

Returns information from tables in documents when the `table_results.enabled=true` parameter is included with a natural language query request.

Query limits

A query is any operation that submits a `POST` request to the `/query` endpoint of the API. Such operations include queries that are submitted by using the API. It does not include queries that are submitted from the search bar on the *Improve and customize* page of the product user interface.

A query is counted only if the request is successful, meaning it returns a response (with message code 200).

The number of search queries that you can submit per month per service instance depends on your Discovery plan type.

Plan	Queries per month per service instance
Cloud Pak for Data	Unlimited
Premium	Unlimited
Enterprise	Unlimited
Plus (includes Trial)	500,000

Number of queries per month



Note: For Enterprise plans only, your bill labels requests that are generated from both query searches and analyze API calls as "Queries". For more information about Analyze API calls, see [Analyze API limits](#).

The number of queries that can be processed per second per service instance depends on your Discovery plan type.

Plan	Concurrent queries per service instance
Cloud Pak for Data	Unlimited
Premium	50
Enterprise	5
Plus (includes Trial)	5

Number of concurrent queries

For information about pricing, see [Discovery pricing plans](#).

Estimating query usage

How to estimate the number of queries your application will use per month depends on your use case.

- For use cases that focus more on data enrichment and analysis or where the output from the document processing is not heavily searched, you can estimate query numbers based on the total number of documents.
- For use cases where many users interact with the application that uses Discovery, you can estimate by calculating the number of searches per user times the number of expected users. For example, 50% of the questions that are submitted by users to a virtual

assistant are likely to be answered by Discovery. With 100,000 users per month and an average of 3 questions per user, you can expect 15,000 queries per month. (10,000 users/mo * 3 queries/user * 50% to Discovery = 15,000)

Querying with document-level security enabled

IBM Cloud Pak for Data **IBM Cloud Pak for Data only**



Note: This information applies only to installed deployments.

If you enable document-level security for a collection, only documents that the current user has permission to access are returned in search results. For more information, see [Configuring document-level security](#).

To return search results that adhere to the security restrictions, the current user must meet these requirements:

- Have access to your Discovery instance.
- Have access to the data source.

If the current user does not meet these requirements, no search results are returned.

The username that is associated with your Discovery instance is used to generate an authorization token. The token is used to authenticate Discovery queries.

To generate each access token, run the following command:

```
$ curl -u "{username}:{password}" \
"https://{{hostname}}:{port}/v1/preauth/validateAuth"
```

Replace **{username}** and **{password}** with the user's Discovery credentials.

Use the bearer token that is associated with the user when you run the query.

```
$ curl -H "Authorization: Bearer {token}" \
"https://{{hostname}}/{{instance_name}}/v2/projects/{{project_id}}/collections/{{Collection_ID}}/query?version=2019-11-29"
```

DQL overview

The Discovery Query Language defines syntax you can use to filter, search, and analyze your data.

How to write a Discovery Query Language query

The Discovery Query Language leverages the structure of indexed documents. The following JSON snippet shows an indexed document from a collection where the **Entities** enrichment is applied. As a result of the enrichment, the JSON structure captures any mentions of known entities, such as city names, companies, or famous people.

In this example, the recognized entity is the company name **IBM**.

```
{
  "document": {
    "document_id": "f7f27ea30eb3e4c0ce21830618d9ee99",
    "enriched_text": [
      {
        "entities": [
          {
            "model_name": "natural_language_understanding",
            "mentions": [],
            "text": "IBM",
            "type": "Organization"
          }
        ]
      }
    ]
  }
}
```

To create a query that returns all of the documents in which the entity **IBM** is mentioned, use the following syntax:

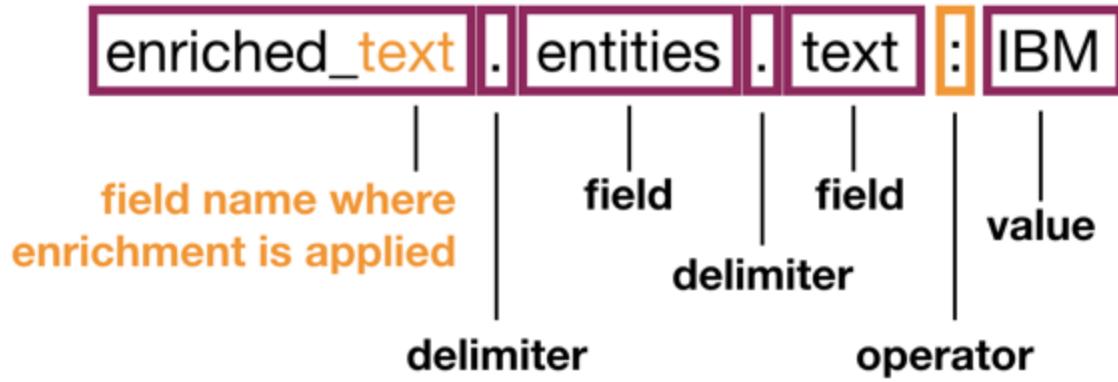


Figure 1. Example query structure

This basic query contains a nested path expression before the `:` operator. Each path element is the name of a field in the document separated by a period (`.`). The `:` operator indicates that the text that follows the operator must be included in the result.

The `::` operator indicates that the text must be matched exactly in the result. For more information, see [Query operators](#). You can see how the two operators are used in the following examples.

- To return matching documents in order of relevance, pass the following data object in the `POST` request:

```
{
  "query": "enriched_text.entities.text::IBM"
}
```

- To return matching documents in any order, pass the following data object in the `POST` request as the query body:

```
{
  "filter": "enriched_text.entities.text::IBM"
}
```

Using the filter and query parameters together

The `filter` parameter returns faster than the `query` parameter and its results are cached. If you submit queries that use the `filter` and `query` parameters separately on a small data set, each request returns similar (if not identical) results.

In large data sets, if you need results to be returned in order of relevance, combine the `filter` and `query` parameters. Using the parameters together improves performance because the `filter` parameter is applied first. It filters the documents and caches the results. The `query` parameter then ranks the cached results.

Filter example: Get a document by its ID

Query body:

```
{
  "filter": "document_id::b6d8c6e3-1097-421b-9e39-75717d2554aa"
}
```

If the document exists, the query returns 1 matching result. If it doesn't, the query returns no matching results.

Filter example: Find a document ID by its file name

If you don't know the `document_id` of a document, but you know the original `filename` of the document, you can use the `filter` and `return` parameters together to discover the `document_id`.

Query body:

```
{
  "filter": "extracted_metadata.filename::100674.txt",
  "return": [ "document_id", "extracted_metadata" ]
}
```

Response:

```
{
  "matching_results": 1,
  "results": [
    {
      "document_id": "b6d8c6e3-1097-421b-9e39-75717d2554aa",
      "extracted_metadata": {
        "sha1": "AD447F7592A17CDCBF0A589C4E6EC2087AF7H35F",
        "filename": "100674.txt",
      }
    }
  ]
}
```

```
        "file_type": "text"
    }
]
}
```

Filter example: Find documents that mention an entity value

The query looks for documents that mention the entity `Gilroy` and finds 4 matching documents.

Query body:

```
{
  "filter": "enriched_text.entities.text::Gilroy"
}
```

Response:

```
{
  "matching_results": 4
}
```

Filtering nested values

You can nest one filter inside another to ensure that the documents that are returned match more than one condition.

In the documents used for these examples, the entity `"Gilroy"` appears as both a `"Location"` (a town in California) and as a `"Person"` (a surname) entity type. To find documents where `"Gilroy"` appears as a location, write a query that filters on two nested fields at the same time: the entity text must be `"Gilroy"` and the entity type must be `"Location"`.

One way to write the query is as follows:

```
{
  "filter": "enriched_text.entities.text::Gilroy,enriched_text.entities.type::Location"
}
```

This query matches documents where some path `enriched_text.entities.text` is `Gilroy` and some path `enriched_text.entities.type::Location` is `Location`. However, there is no guarantee that those two paths will be under the same `entities` object. For example, the query matches documents that have `Gilroy` as a `Person` entity type and, at the same time, have some other `Location` entity type object.

To accurately capture the nested semantics of this query, nest the filter values by using the following syntax:

Query body:

```
{
  "filter": "enriched_text.entities:(text::Gilroy,type::Location)"
}
```

This stricter query matches only those documents in which there is an `entities` object with `text` equal to `Gilroy` and `type` equal to `Location`.

As another example, if you want to match documents that contain an `entities` object with `text` equal to `Gilroy` but `type` not equal to `Location`, you can use the `not equal` operator in the query, for example:

```
{
  "filter": "enriched_text.entities:(text::Gilroy,type::!Location)"
}
```

You can also use aggregations to do more sophisticated filtering of the results. For more information about the available aggregation types, see [Query aggregations](#).

For more information about the Discovery Query Language, see the following topics:

- [Query parameters](#)
- [Query operators](#)

Query parameters

You can use these parameters when you write queries with the Discovery Query Language. For more information, see the Discovery API reference. For an overview of query concepts, see the [Query overview](#).

Queries that are written in the Discovery Query Language can include both search and structure parameters.

The default values for query parameters can differ by project type. For more information about default values, see [Default query settings](#).

Search parameters

Use search parameters to search your collection, identify a result set, and analyze the result set.

The **results set** is the group of documents that are identified by the combined searches of the search parameters. The results set might be significantly larger than the returned results. If an empty query is submitted, the results set is equal to all the documents in the collection.



Important: Documents that you do not have permissions to access are not returned in query results.

Answer finding

IBM Cloud The `find_answers` parameter is supported in managed deployments only.

By default, Discovery provides answers by returning the entire [passage](#) that contains the answer to a natural language query. When the answer-finding feature is enabled, Discovery also provides a "short answer" within the passage, and a confidence score to show whether the "short answer" answers the question that is explicit or implicit in the user query. Applications that use the answer-finding feature can display the short answer alone or can display the short answer emphasized in the context of the full passage. For most applications, displaying the short answer emphasized within the full passage is preferable, because answers generally make more sense in context.

The answer finding feature behaves in the following ways:



Note: In the passage examples that follow, the short answers are shown in bold font.

- Finds answers. It doesn't create answers. The answer must be part of the text; it can't be inferred.

"What was IBM's revenue in 2022?" can get a correct answer if you have a document that states what IBM's revenue was in 2022. However, if you have a document that lists what IBM's revenue was in each quarter of 2022, it doesn't add them up and give you a total.

- Handles synonyms and lexical variations if the answer is available.
 - Example question: "When did IBM purchase Red Hat?"
 - Passage: "IBM closed its \$34 billion acquisition of Red Hat in **July of 2019**."
- Combines information across multiple sentences if they are close together (within approximately 2,000 characters).
 - Example question: "When did IBM purchase Red Hat?"
 - Passage: "IBM acquired Red Hat for \$34 billion. The deal closed in **July of 2019**."
- Handles implicit questions similar to the way it would handle the equivalent explicit question.

Example questions:

- `company that developed the AS/400`
- `What company developed the AS/400?`

- Works well with questions with longer phrase or clause answers.
 - Example question: How do I flip a pancake?
 - Passage: The key to getting a world-class pancake is flipping it properly. The best way to flip a pancake is to **stick a spatula under it, lift it at least 4 inches in the air, and quickly rotate the spatula 180 degrees**.
- Many how or why questions are only fully answered by much longer spans of text. The answer-finding feature does not return a whole document as the answer (and it doesn't summarize a document length answer).
- Handles yes or no questions that are factual and have a concise answer in the text
 - Example question: Is there a library in Timbuktu
 - Passage: Timbuktu's **main library, officially called the Ahmed Baba Institute of Higher Islamic Studies and Research**, is a treasure house that contains more than 20,000 manuscripts that cover centuries of Mali's history.
- Handles questions with very short answers, such as names and dates, especially when the type of answer that is required is explicit in the text.
- Handles opinion questions, but only by finding a statement of that opinion; it does not assess the validity of the opinion.
 - Example question: Should I try blue eyeshadow?

- Passage: We think blue eye shadow is trending this year.

How the answer-finding feature works

After a user submits a query, the query is analyzed by the Discovery service. Query analysis transforms the user's original query in ways that improve the chances of finding the best search results. For example, it lemmatizing words, removes stop words, and adds query expansions. The search is performed and the resulting documents and passages are returned.

Answer finding is applied to the returned passages. Up to 60 passages are sent to the answer-finding service. How these 60 passages are chosen differs based on the `passages.per_document` parameter value.

- If `passages.per_document` is `false`, the top 60 passages from all of the documents that are returned by search are chosen based on their passage scores only.
- If `passages.per_document` is `true`, the returned documents are ranked first, and then the top 60 passages from these top documents are chosen.

For example, if you set the query to return 100 documents (`count=100`) and ask for 2 passages from each document (`passages.max_per_document=2`), then 2 passages are chosen from each of the 30 top-ranked documents ($2 \times 30 = 60$ passages) only. No passages are chosen from the remaining 70 documents.

 **Tip:** If your goal is to get the best 10 short answers, a good approach is to give the answer-finding feature various passages from more documents than just the top 10. To do so, set `passages.per_document` to `true`, and then request 20 documents and up to 3 passages from each document with the answer-finding feature enabled. The answer-finding feature searches for answers in up to $20 \times 3 = 60$ passages.

Answer finding does not use the transformed query string that is generated by query analysis. Instead, it uses a copy of the user's original input that is stored at query time to find the best short answer. If the answer-finding module is confident that it found an answer in one of the passages, the answer confidence score is combined with the document and passage scores to produce a final ranking, which can promote a document or passage that might otherwise be missed.

Answer-finding API details

The answer-finding API adds the following parameters to the `passage` section of the query API:

- `find_answers` is optional and defaults to `false`. If it is set to `true` (and the `natural_language_query` parameter is set to a query string), the answer-finding feature is enabled.
- `max_answers_per_passage` is optional and defaults to `1`. In this case, the answer-finding feature finds the number of answers that are specified at most from any one passage.

A section is also added to the return value within each `passage` object. That section is called `answers`, and it is a list of answer objects. The list can be up to `max_answers_per_passage` in length. Each answer object contains the following fields:

- `answer_text` is the text of the concise answer to the query.
- `confidence` is a number between `0` and `1` that is an estimate of the probability that the answer is correct. Some answers have low confidence and are unlikely to be correct. Be selective about what you do with answers based on this value. The confidence and order of documents in the search results are adjusted based on this combination if the `per_document` parameter of passage retrieval is set to `true` (which is the default).
- `start_offset` is the start character offset (the index of the first character) of the answer within the field that the passage came from. It is greater than or equal to the start offset of the passage (since the answer must be within the passage).
- `end_offset` is the end character offset (the index of the last character, plus one) of the answer within the field that the passage came from. It is less than or equal to the end offset of the passage.

To find answers across the entire project:

- Set `passages.enabled` to `true`
- Set `passages.find_answers` to `true`

To find answers within a single known document (for example, a document review application with long, complex documents):

- Set `passages.enabled` to `true`
- Set `passages.find_answers` to `true`
- Set `filter` to select the `document_id` for the document

The following example shows a query that uses this API:

```
POST /v2/projects/{project_id}/query{
  "natural_language_query": "Why did Nixon resign?",
  "passages": {
    "enabled": true, "find_answers":true
  }
}
```

Example response:

```
{
  "matching_results": 74, "retrieval_details": { "document_retrieval_strategy": "untrained"}, "results": [
    {
      "document_id": "63919442-7d5b-4cae-ab7e-56f58b1390fe",
      "result_metadata": {"collection_id": "collection_id1234", "document_retrieval_source": "search", "confidence": 0.78214},
      "metadata": { "parent_document_id": "63919442-7d5b-4cae-ab7e-56f58b1390fg" },
      "title": "Watergate scandal",
      "document_passages": [
        {
          "passage_text": "With his complicity in the cover-up made public and his political support completely eroded, Nixon resigned from office on August 9, 1974. It is believed that, had he not done so, he would have been impeached by the House and removed from office by a trial in the Senate.",
          "field": "text",
          "start_offset": 281,
          "end_offset": 553,
          "answers": [
            {
              "answer_text": "his complicity in the cover-up made public and his political support completely eroded",
              "start_offset": 286, "end_offset": 373, "confidence": 0.78214
            }
          ]
        }
      ]
    }
  ]
}
```

natural_language_query

Use a natural language query to enter queries that are expressed in natural language, as might be received from a user in a conversational or free-text interface, such as IBM Watson Assistant. The parameter uses the entire input as the query text. It does not recognize operators.

The maximum query string length for a natural language query is **2048**.

Result confidence scores

When the query type is a natural language query, each result has a confidence score. The confidence score is a measure of the relevancy of the result. Each query result is evaluated and scored independently.

A variety of techniques are used to evaluate confidence. One important factor is the frequency of word matches between the query and the document.

Because a variety of techniques are used in different contexts to evaluate the result, the number range of result scores can vary widely from query to query. This variability means that comparing the confidence score to a static threshold value is an unsuitable method by which to delimit the results that are returned by your application. Results are ordered from highest to lowest confidence. You can find the best candidate answers by taking the top results, regardless of their confidence score values.

The **natural_language_query** parameter enables capabilities such as relevancy training. For more information, see [Improving result relevance with training](#).

query

A query search returns all documents in your data set with full enrichments and full text in order of relevance. A query also excludes any documents that don't mention the query content.

aggregation

Aggregation queries return a count of documents that match a set of data values. For the full list of aggregation options, see [Query aggregations](#).

filter

A cacheable query that excludes any documents that don't mention the query content. Filter search results are **not** returned in order of relevance.

When you write a query that includes both a **filter**, and an **aggregation**, **query**, or **natural_language_query** parameter, the **filter** parameter runs first, and then any **aggregation**, **query**, or **natural_language_query** parameters run in parallel.

With a simple query, especially on a small data set, the **filter** and **query** parameters often return the exact same (or similar) results. If the **filter** and **query** calls return similar results, and you don't need the responses to be returned in order of relevance, use the **filter** parameter. Filter calls are faster and are cached. Caching means that the next time you make the same call, you get a much quicker response, particularly in a big data set.

Structure parameters

Structure parameters define the content and organization of the documents in the returned JSON. Structure parameters don't affect

which documents are part of the entire results set.

return

A comma-separated list of the portion of the document hierarchy to return. Any of the document hierarchies are valid values. If this parameter is an empty list, then all fields are returned.

count

The number of documents that you want to return in the response. The default is **10**. The maximum for the **count** and **offset** values together in any one query is **10000**.

offset

Index value of the position of the search result where the set of results to return begins. For example, if the total number of results that are returned is 10, and the offset is 8, it returns the last two results. The default is **0**. The maximum allowed value for the **count** and **offset** together in any one query is **10000**.

spell correction

In natural language queries, checks the query that is submitted for misspelled terms. The query is processed as-is. However, likely corrections to the original query, if any exist, are returned in the **suggested_query** field of the response. The suggestions are not used automatically, but your application can make use of them.

sort

A comma-separated list of fields in the document to sort by. You can optionally specify a sort direction by prefixing the field with **-** for descending order or **+** for ascending order. Ascending order is the default sort direction.

highlight

A Boolean value that specifies whether to include a **highlight** object in the returned output. When included, the highlight returns keys that are field names and values that are arrays. The arrays contain segments of query-matching text that is highlighted by using the HTML emphasis (****) tag.

This parameter is ignored if **passages.enabled** and **passages.per_document** are **true**, in which case passages are returned for each document instead of highlights.



Note: Currently, if the query searches for an **exact match** of an enrichment mention, only lowercase matches are highlighted. When the **includes** operator is used, upper- and lowercase matches are highlighted.

The output lists the **highlight** object after the **enriched_text** object, as shown in the following example.

```
$ curl -H "Authorization: Bearer {token}" \
'https://{{hostname}}/{{instance_name}}/v2/projects/{{project_id}}/collections/{{collection_id}}/query?version=2019-11-29&natural_language_query=Hybrid%20cloud%20companies&highlight=true'
```

The JSON that is returned has the following format:

```
{
  "highlight": {
    "extracted_metadata.title": [
      "IBM to Acquire Sanovi Technologies to Expand Disaster Recovery Services for <em>Hybrid</em> <em>Cloud</em>"
    ],
    "enriched_text.concepts.text": [
      "Privately held <em>company</em>",
      "<em>Cloud</em> computing"
    ],
    "text": [
      " Sanovi Technologies, a privately held <em>company</em> that provides <em>hybrid</em> <em>cloud</em> recovery, <em>cloud</em> migration",
      "IBM to Acquire Sanovi Technologies to Expand Disaster Recovery Services for <em>Hybrid</em> <em>Cloud</em>\n\nPublished",
      " undergoing digital and <em>hybrid</em> <em>cloud</em> transformation.\n\nURL: http://www.ibm.com/press/us/en/pressrelease/50837.wss",
      " and business continuity software for enterprise data centers and <em>cloud</em> infrastructure. Adding"
    ],
    "enriched_text.categories.label": [
      "/business and industrial/<em>company</em>/bankruptcy"
    ],
    "enriched_text.entities.type": [
      "<em>Company</em>"
    ],
    "html": [
      " Technologies, a privately held <em>company</em> that provides <em>hybrid</em> <em>cloud</em>\n recovery,"
    ]
  }
}
```

```

<em>cloud</em> migration and business",
    " Disaster Recovery Services for <em>Hybrid</em> <em>Cloud</em></title></head>\n<body>\n\n\n<p>Published: Thu,
27 Oct 2016 07:01",
    " digital and <em>hybrid</em> <em>cloud</em> transformation.</p>\n<p>URL:
http://www.ibm.com/press/us/en/pressrelease/50837.wss</p>\n\n\n</body></html>",
    " continuity software for \nenterprise data centers and <em>cloud</em> infrastructure. Adding these \ncapabilities"
]
}
}

```

passages

A Boolean that specifies whether the service returns a set of the most relevant passages from the documents that are returned by a query that uses the `natural_language_query` parameter. The passages are generated by sophisticated Watson algorithms that determine the best passages of text from all of the documents returned by the query. The default value for the parameter differs based on your project type. For more information about default values, see [Default query settings](#).

Discovery attempts to return passages that start at the beginning of a sentence and stop at the end by using sentence boundary detection. To do so, it first searches for passages approximately the length specified in the `passages.characters` parameter (for most project types, the default is `200`). It then expands each passage to the limit of twice the specified length so as to return full sentences. If your `passages.characters` parameter is short or the sentences in your documents are long there might be no sentence boundaries close enough to return the full sentence without going over twice the requested length. In that case, Discovery stays within the limit of twice the `passages.characters` parameter, so the passages that are returned might not include the entire sentence and can omit the beginning, end, or both.

Since sentence boundary adjustments expand passage size, the average passage length can increase. If your application has limited screen space, you might want to set a smaller value for `passages.characters` or truncate the passages that are returned by Discovery. Sentence boundary detection works for all supported languages and uses language-specific logic.

Passages are grouped with each document result and are ordered by passage relevance. Including passage retrieval in queries increases the response time because it takes more time to score the passages.

You can adjust the fields in the documents for passage retrieval to search with the `passages.fields` parameter.

The `passages` parameter returns matching passages (`passage_text`), and the `score`, `document_id`, the name of the field that the passage was extracted from (`field`), and the starting and ending characters of the passage text within the field (`start_offset` and `end_offset`), as shown in the following example.

```
$ curl -H "Authorization: Bearer {token}"
'https://{{hostname}}/{{instance_name}}/v2/projects/{{project_id}}/collections/{{collection_id}}/query?version=2019-11-
29&natural_language_query=Hybrid%20cloud%20companies&passages=true&passages.per_document=false'
```

The JSON that is returned from the query has the following format:

```
{
  "matching_results":2,
  "passages":[
    {
      "document_id":"ab7be56bcc9476493516b511169739f0",
      "passage_score":15.230205287402338,
      "passage_text":"a privately held company that provides hybrid cloud recovery, cloud migration and business continuity
software for enterprise data centers and cloud infrastructure.",
      "start_offset":120,
      "end_offset":300,
      "field":"text"
    },
    {
      "passage_text":"Disaster Recovery Services for Hybrid Cloud</title></head>\n<body>\n\n\n<p>Published: Thu, 27 Oct
2016 07:01:21 GMT</p>\n",
      "passage_score":10.153470191601558,
      "document_id":"fb5dc4d8a6a29f572ebdeb6fbed20e",
      "start_offset":70,
      "end_offset":120,
      "field":"html"
    }
  ]
}
```

passages.fields

A comma-separated list of fields in the index that passages are drawn from. If this parameter is not specified, then passages from all root-level fields are included.

You can specify fields in both the `return` and `passages.fields` parameters. When you specify both parameters, each with different values, they are treated separately.

For example, the request might include the parameters `"return": ["docno"]` and `"passages": {"fields": ["body"]}`. The `body` field is specified in `passages.fields`, but not in `return`. In the result, passages from the document body are returned, but the contents of the body field itself is not returned.

`passages.count`

The maximum number of passages to return. The search returns fewer passages if the specified count is the total number found. The default value is `10`. The maximum value is `100`.

`passages.characters`

The approximate number of characters that any one passage can have. The default value is `200`. The minimum is `50`. The maximum is `2,000`. Passages that are returned can contain up to twice the requested length (if necessary) to get them to begin and end at sentence boundaries.

`passages.max_per_document`

One passage is returned per document by default. You can increase the maximum number of passages to return per document by specifying a higher number in the `passages.max_per_document` parameter.

`similar`

Finds documents that are similar to documents that you identify as being of interest to you. To find similar documents, Discovery identifies the 25 most relevant terms from the original document and then searches for documents with similar relevant terms.

If `similar.enabled` is `true`, you must specify the `similar.document_ids` field to include a comma-separated list of the documents of interest.



Note: In installed deployments, support for this parameter was added with the 4.6.0 release.

`table_retrieval`

If [Table understanding](#) is enabled in your collection, a `natural_language_query` finds tables with content or context that match a search query.

Example query:

```
$ curl -H "Authorization: Bearer {token}" \
'https://{{hostname}}/{{instance_name}}/v2/projects/{{project_id}}/collections/{{collection_id}}/query?version=2019-11-29&natural_language_query=interest%20appraised&table_results=true'
```

The JSON that is returned from the query has the following format:

```
{
  "matching_results": 1,
  "session_token": "1_FDjAVkn9SW6oH9y5_9Ek3KsNFG",
  "results": [
    {}
  ]
{
  "table_results": [
    {
      "table_id": "e883d3df1d45251121cd3d5aef86e4edc9658b21",
      "source_document_id": "c774c3df0c90255191cc0d4bb8b5e8edc6638d96",
      "collection_id": "collection_id",
      "table_html": "html snippet of the table info",
      "table_html_offset": 42500,
      "table": [
        {
          "location": {
            "begin": 42878,
            "end": 44757
          },
          "text": "Appraisal Premise Interest Appraised Date of Value Value Conclusion\\nMarket Value \\\"As Is\\\" Fee Simple Estate January 12, 2016 $1,100,000\\n",
          "section_title": {
            "location": {
              "begin": 42300,
              "end": 42323
            },
            "text": "MARKET VALUE CONCLUSION"
          },
          "title": {},
          "table_headers": []
        }
      ]
    }
  ]
}
```

```

"row_headers": [
  {
    "cell_id": "rowHeader-42878-42896",
    "location": {
      "begin": 42878,
      "end": 42896
    },
    "text": "Appraisal Premise",
    "text_normalized": "Appraisal Premise",
    "row_index_begin": 0,
    "row_index_end": 0,
    "column_index_begin": 0,
    "column_index_end": 0
  }
],
"column_headers": [],
"body_cells": [
  {
    "cell_id": "bodyCell-43410-43424",
    "location": {
      "begin": 43410,
      "end": 43424
    },
    "text": "Date of Value",
    "row_index_begin": 0,
    "row_index_end": 0,
    "column_index_begin": 2,
    "column_index_end": 2,
    "row_header_ids": [
      "rowHeader-42878-42896",
      "rowHeader-43145-43164"
    ],
    "row_header_texts": [
      "Appraisal Premise",
      "Interest Appraised"
    ],
    "row_header_texts_normalized": [
      "Appraisal Premise",
      "Interest Appraised"
    ],
    "column_header_ids": [],
    "column_header_texts": [],
    "column_header_texts_normalized": [],
    "attributes": []
  }
],
"contexts": [
  {
    "location": {
      "begin": 44980,
      "end": 44996
    },
    "text": "Compiled by CBRE"
  }
],
"key_value_pairs": []
}
]
}
}

```

`table_results.enabled`

When `true`, a `table_results` array is included in the response with a list of table objects that match the `natural_language_query` value in order of scored relevance. For all project types, except *Document Retrieval for Contracts*, the default value is `false`.

`table_results.count`

This parameter specifies the maximum number of tables that can be included in the `table_results` array. Only returned if `table_results.enabled = true`. The default value is `10`.

Query operators

You can use operators when you write queries to submit to Discovery by using the Query API.

The types of operators that are supported differ by query type:

- [Natural language queries](#)

- [Discovery Query Language \(DQL\) queries](#)

Natural Language Query (NLQ) operator

The `natural_language_query` parameter accepts a string value.

"" (Phrase query)

Use quotation marks to emphasize a single word or phrase in the query that is most important to match. For example, the following request boosts documents that contain the term "nomination" in them.

```
{
  "natural_language_query": "What is the process for \"nomination\" of bonds?"
}
```

Specifying a quoted phrase does not prevent documents without the phrase from being returned. It merely gives more weight to documents with the phrase than those without it. For example, the query results might also contain documents that mention "bonds" or "process" and do not contain the word "nomination".

The following request boosts the phrase "change in monetary policy" and also matches "change" or "monetary" or "policy".

```
{
  "natural_language_query": "\"change in monetary policy\""
}
```

Single quotation marks ('') are not supported. You cannot use wildcards (*) in phrase queries.

Discovery Query Language (DQL) operators

Operators are the separators between different parts of a query.

. (JSON delimiter)

This delimiter separates the levels of hierarchy in the JSON schema

For example, the following query argument identifies the section of the enriched_text object that contains entities and the text recognized as an entity.

```
enriched_text.entities.text
```

The JSON representation of this section looks as follows:



Figure 1. JSON representation of the enriched_text.entities.text field

: (Includes)

This operator specifies inclusion of the full query term.

For example, the following query searches for documents that contain the term `cloud computing` in the `text` field:

```
{
  "query": "enriched_text.entities.text:\"cloud computing\""
}
```

The **includes** operator does not return a partial match for the query term. If you want to find a partial match for a term, use a **wildcard** operator with the **includes** operator. For example, if you want to find any occurrences of `TP53` or `p53` in the `test_results` field, the following query will **not** find occurrences of both terms:

```
{
  "query": "test_results:P53"
}
```

Instead, include a wildcard in the request. For example, use the following query request. Because we are using the wildcard operator, we also changed the term to lowercase.

```
{  
  "query": "test_results:*p53"  
}
```

With this syntax, occurrences of `p53`, `tp53`, `P53`, or `TP53` are all returned.

`""` (Phrase query)

Phrase queries only match occurrences of the whole phrase. The order of the words in the phrase must match.

For example, the following query returns only documents that contain a field named `quotation` with the text, `There's no crying in baseball`.

```
{  
  "query": "quotation:There's no crying in baseball"  
}
```

A document with a `quotation` field that says `Jimmy Dugan said there's no crying in baseball` is also returned. However, documents that only mention `baseball` or `crying` without the entire phrase are not matched. Neither is a document with `In baseball, there's no crying`. Documents that contain the right text in the wrong field also are not matched. For example, a document with the text `There's no crying in baseball` in the `text` field is not returned.

Single quotation marks ('') are not supported. You cannot use wildcards (*) in phrase queries.

`::` (Exact match)

This operator specifies an exact match for the query term. Exact matches are case-sensitive.

For example, the following query searches for documents that contain entities of type `Organization`:

```
{  
  "query": "enriched_text.entities.type::Organization"  
}
```

The entire content of the field that you specify must match the phrase you specify. For example, the following query finds documents in which only entity mentions of `IBM Cloud` are detected, not `IBM Cloud Pak for Data` or `IBM cloud` or `Cloud`.

```
{  
  "query": "enriched_text.entities.text::\"IBM Cloud\""  
}
```

`::!` (Does not include)

This operator specifies that the results do not contain a match for the query term.

For example:

```
{  
  "query": "enriched_text.entities.text:!\"cloud computing\""  
}
```

`::::!` (Not an exact match)

This operator specifies that the results do not exactly match the query term.

For example:

```
{  
  "query": "enriched_text.entities.text::!\"Cloud computing\""  
}
```

Exact matches are case-sensitive.

`\` (Escape character)

Escape character that preserves the literal value of the character that follows it.

For example, you can place an escape character before a quotation mark in query text to include the quotation mark in the query string.

```
{  
  "query":"title::Dorothy said: \"There's no place like home\""  
}
```

([],) (Nested grouping)

Logical groupings can be formed to specify more specific information.

For example:

```
{  
  "query":"enriched_text.entities:(text:IBM,type:Company)"  
}
```

| (or)

Boolean operator for "or".

In the following example, documents in which **Google** or **IBM** are identified as entities are returned:

```
{  
  "query":"enriched_text.entities.text:Google|enriched_text.entities.text:IBM"  
}
```

 **Note:** The includes ([:], :!) and match (::, ::!) operators have precedence over the **OR** operator.

For example, the following syntax searches for documents in which **Google** is identified as an entity or the string **IBM** is present:

```
{  
  "query":"enriched_text.entities.text:Google|IBM"  
}
```

It is treated as follows:

```
(enriched_text.entities.text:Google) OR IBM
```

, (and)

Boolean operator for "and".

In the following example, documents in which **Google** and **IBM** both are identified as entities are returned:

```
{  
  "query":"enriched_text.entities.text:Google,enriched_text.entities.text:IBM"  
}
```

 **Note:** The includes ([:], :!) and match (::, ::!) operators have precedence over the **AND** operator.

For example, the following syntax searches for documents in which **Google** is identified as an entity and the string **IBM** is present:

```
{  
  "query":"enriched_text.entities.text:Google,IBM"  
}
```

It is treated as follows:

```
(enriched_text.entities.text:Google) AND IBM
```

<=, >=, >, < (Numerical comparisons)

Creates numerical comparisons of **less than** or **equal to**, **greater than** or **equal to**, **greater than**, and **less than**.

Only use numerical comparison operators when the value is a **number** or **date**.

 **Tip:** Any value that is surrounded by quotations is a String. Therefore, `score>=0.5` is a valid query and `score>="0.5"` is not.

For example:

```
{  
  "query": "invoice.total>100.50"  
}
```

`^x` (Score multiplier)

Increases the score value of a search term.

For example:

```
{  
  "query": "enriched_text.entities.text:IBM^3"  
}
```

`*` (Wildcard)

Matches unknown characters in a search expression. Do not use capital letters with wildcards.

For example:

```
{  
  "query": "enriched_text.entities.text:ib*"  
}
```

`~n` (String variation)

The number of character differences that are allowed when matching a string. The maximum variation number that can be used is 2.

For example, the following query returns documents that contain `car` in the title field, as well as `cap`, `cat`, `can`, `sat`, and so on:

```
{  
  "query": "title:cat~1"  
}
```

The normalized version of the word is used for matching. Therefore, if the input contains "cats", the search looks for "cat", which is the normalized form of the plural cats.

When a phrase is submitted, each term in the phrase is allowed the specified number of variations. For example, the following input matches `cat dog` and `far log` in addition to `car hog`.

For example:

```
{  
  "query": "title:\"car hog\"~1"  
}
```

`:*` (Exists)

Used to return all results where the specified field exists.

For example:

```
{  
  "query": "title:/*"  
}
```

`:!*` (Does not exist)

Used to return all results that do not include the specified field.

For example:

```
{  
  "query": "title:!/*"  
}
```

For more information, see the Discovery [API reference](#).

For an overview of query concepts, see the [Query overview](#).

Query aggregations

Use aggregations to group, analyze, or compare results that are returned by a query request.

An aggregation is defined by an **aggregation** parameter that you can specify in the Query API. The input to the aggregation parameter is the document set that is returned from the **query**, **filter**, or **natural_language_query** parameter that is specified as a separate parameter in the same query request. Otherwise, the aggregation is applied to all of the documents in the project.

You can use an aggregation to do calculations from values in the result document set. For example, to get information about the highest dollar amount in the **order.total** field of the documents that are returned as query results, use **max(order.total)** as the value of the **aggregation** parameter.

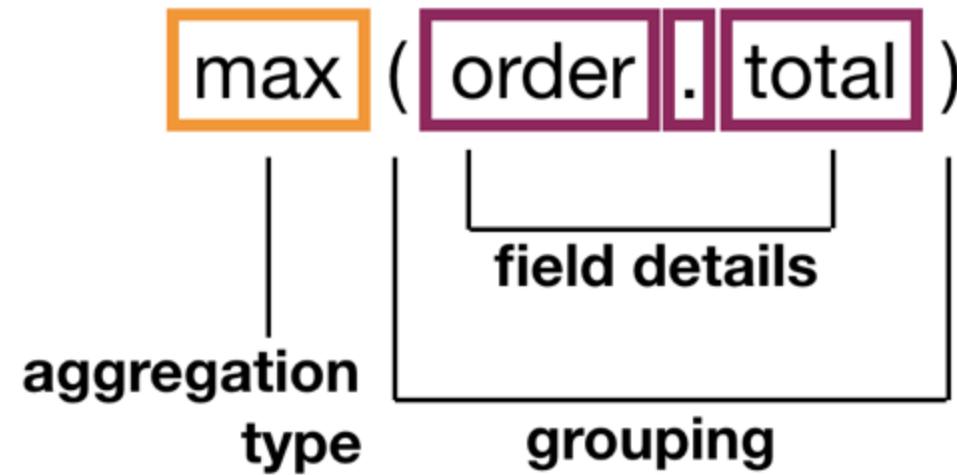


Figure 1. Aggregation query structure example

The aggregation parameter returns data about the field with the highest value.

```
"aggregations": [
  {
    "type": "max",
    "field": "order.total",
    "value": 100668.00
  }
]
```

Grouping documents

In addition to doing calculations, you can use an aggregation to group documents in the result set that match certain values, so you can count them or analyze them further. For example, you can use an aggregation to search a set of traffic incident reports for documents that mention the term **brake**. And from the returned documents, find reports from the US states with the most relevant mentions of the term.

In the following example request, the count parameter that returns only 3 aggregation results is included to make the example easier to follow.

```
{
  "query": "brake",
  "aggregation": "term(field:STATE,count:3,relevancy:true)"
}
```

The output of the aggregation parameter is returned in an **aggregations** object that is displayed before the **results** object, which contains the query results. A maximum of 50,000 values can be returned in the **aggregations** object for a single query.

The resulting **aggregations** object contains summary information about the query results. In this example, for instance, it shows that traffic incident reports from New York, California, and Florida have the most relevant mentions of the term **brake**.

```
{
  "matching_results": 9064,
  "retrieval_details": {
    "document_retrieval_strategy": "untrained"
  },
  "aggregations": [
    {
      "type": "term",
      "field": "STATE",
      "results": [
        {
          "key": "NY",
          "value": 1234
        },
        {
          "key": "CA",
          "value": 1234
        },
        {
          "key": "FL",
          "value": 1234
        }
      ]
    }
  ]
}
```

```

    "key": "NY",
    "matching_results": 693,
    "relevancy": 1.1649531567631084,
    "total_matching_documents": 2156,
    "estimated_matching_results": 542
  },
  {
    "key": "CA",
    "matching_results": 1210,
    "relevancy": 1.1170819184294765,
    "total_matching_documents": 4017,
    "estimated_matching_results": 1011
  },
  {
    "key": "FL",
    "matching_results": 511,
    "relevancy": 0.828014956418841,
    "total_matching_documents": 2199,
    "estimated_matching_results": 553
  }
],
"results": []

```

Combining aggregation types

There are different types of aggregations that you can use to analyze or group the query results. And you can combine more than one aggregation in a request to do more targeted analysis.

The following example shows a request that is composed of two term operators. The first term aggregation groups the input documents by US STATE values and selects 3 groups. The second term aggregation applies to each of those 3 groups and groups them further by the value of CITY. Only 2 of those CITY subgroups are returned per STATE group.

The relevancy parameter is being excluded to make the results easier to read.

```
{
  "query": "brake",
  "aggregation": "term(field:STATE,count:3).term(field:CITY,count:2)"
}
```

The response contains city information from each state.

```
{
  "matching_results": 9064,
  "retrieval_details": {
    "document_retrieval_strategy": "untrained"
  },
  "aggregations": [
    {
      "type": "term",
      "field": "STATE",
      "count": 3,
      "results": [
        {
          "key": "CA",
          "matching_results": 1210,
          "aggregations": [
            {
              "type": "term",
              "field": "CITY",
              "count": 2,
              "results": [
                {
                  "key": "LOS ANGELES",
                  "matching_results": 77
                },
                {
                  "key": "SAN DIEGO",
                  "matching_results": 66
                }
              ]
            }
          ]
        },
        {
          "key": "NY",
          "matching_results": 693,
          "aggregations": [

```

```
{
  "type": "term",
  "field": "CITY",
  "count": 2,
  "results": [
    {
      "key": "BROOKLYN",
      "matching_results": 35
    },
    {
      "key": "NEW YORK",
      "matching_results": 21
    }
  ]
},
{
  "key": "FL",
  "matching_results": 511,
  "aggregations": [
    {
      "type": "term",
      "field": "CITY",
      "count": 2,
      "results": [
        {
          "key": "JACKSONVILLE",
          "matching_results": 33
        },
        {
          "key": "TAMPA",
          "matching_results": 29
        }
      ]
    }
  ]
},
"results": []
}
```

The order in which you specify the aggregations matters. For example, if you reverse the order of the term aggregations from the previous example, you get different results.

```
{
  "query": "brake",
  "aggregation": "term(field:CITY,count:3).term(field:STATE,count:1)"
}
```

The new order produces results that surface Chicago, a city that wasn't included in the previous set of results. When the request starts by grouping by state, Illinois, which has only one city with a high number of traffic incident reports, is not included in the results. New York and Florida, which both have more than one city with many incident reports, produce a higher number of statewide matches and therefore, were returned. When you group by city first, the results change.

```
{
  "matching_results": 9064,
  "retrieval_details": {
    "document_retrieval_strategy": "untrained"
  },
  "aggregations": [
    {
      "type": "term",
      "field": "CITY",
      "count": 4,
      "results": [
        {
          "key": "LOS ANGELES",
          "matching_results": 77,
          "aggregations": [
            {
              "type": "term",
              "field": "STATE",
              "count": 1,
              "results": [
                {
                  "key": "CA",
                  "matching_results": 77
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

```

        }
    ]
},
{
  "key": "SAN DIEGO",
  "matching_results": 66,
  "aggregations": [
    {
      "type": "term",
      "field": "STATE",
      "count": 1,
      "results": [
        {
          "key": "CA",
          "matching_results": 66
        }
      ]
    }
  ],
  {
    "key": "CHICAGO",
    "matching_results": 59,
    "aggregations": [
      {
        "type": "term",
        "field": "STATE",
        "count": 1,
        "results": [
          {
            "key": "IL",
            "matching_results": 59
          }
        ]
      }
    ]
  }
],
"results": []

```

Using aggregations to explore enrichments

The `term()` aggregation is especially useful for analyzing results to find out how many enrichments are recognized in the documents. For example, to count how many times each entity type is recognized in the filtered documents, you can submit the following query parameters:

```
{
  "filter": "enriched_text.entities:(text::Gilroy,type::Location)",
  "aggregation": "term(enriched_text.entities.type)"
}
```

The query first selects the documents that have at least one entity of type `Location` and whose text is `Gilroy`. This action returns 3 documents. From the returned documents, the aggregation then counts the number of documents in which each entity type appears.

```
{
  "matching_results": 3,
  "retrieval_details": {
    "document_retrieval_strategy": "untrained"
  },
  "aggregations": [
    {
      "type": "term",
      "field": "enriched_text.entities.type",
      "results": [
        {
          "key": "Location",
          "matching_results": 3
        },
        {
          "key": "Person",
          "matching_results": 3
        },
        {
          "key": "Company",
          "matching_results": 2
        }
      ]
    }
  ]
}
```

```

},
{
  "key": "GeographicFeature",
  "matching_results": 2
},
{
  "key": "Organization",
  "matching_results": 2
},
{
  "key": "Quantity",
  "matching_results": 2
},
{
  "key": "Facility",
  "matching_results": 1
},
{
  "key": "PrintMedia",
  "matching_results": 1
}
]
}
}

```

The 3 matching documents all have a **Location** and a **Person** entity type (`"matching_results": 3`). However, only 2 of the matching documents have a **Company** entity type.

By default, the top 10 matches are returned, sorted by relevance. You can change the number of results by adding the **count** parameter to the aggregation.

```
{
  "filter": "enriched_text.entities:(text::Gilroy,type::Location)",
  "aggregation": "term(enriched_text.entities.type,count:20)"
}
```

Add a filter

Use the **filter()** in the aggregation clause to filter results. For example, you can specify the same filter that was submitted separately in the previous example directly in the **aggregation** clause.

```
{
  "aggregation": "filter(enriched_text.entities:(text::Gilroy,type::Location)).term(enriched_text.entities.type)"
}
```

In this case, the **filter().term()** aggregation finds the same result as the earlier example with the separate **filter** and **aggregation** clauses. However, results are ranked differently when the **filter** clause is used. You can leverage this difference by using the **filter()** clause within the **aggregation** clause to filter results from a sequence of expressions, as shown in the next example.

Start with nested objects

In the previous examples, the `"matching_counts"` value represents the number of documents that match the filter and aggregation. You might want to count how many **nested** objects are present in the query response. The **nested()** aggregation allows you to change the set of documents that is used as input to other aggregation terms.

For example, in the following query the **nested()** segment selects all **enriched_text.entities** nested objects as the input used by the **filter()** and **term()** segments.

```
{
  "aggregation":
  "nested(enriched_text.entities).filter(enriched_text.entities.type::Organization).term(enriched_text.entities.text,count:3)"
}
```

The query results in an **aggregations** object that looks as follows:

```
{
  "aggregations": [
    {
      "type": "nested",
      "path": "enriched_text.entities",
      "matching_results": 1993,
      "aggregations": [
        {
          "type": "term"
        }
      ]
    }
  ]
}
```

```

    "type": "filter",
    "match": "enriched_text.entities.type::Organization",
    "matching_results": 645,
    "aggregations": [
        {
            "type": "term",
            "field": "enriched_text.entities.text",
            "count": 3,
            "results": [
                {
                    "key": "IBM",
                    "matching_results": 36
                },
                {
                    "key": "Docker",
                    "matching_results": 12
                },
                {
                    "key": "OpenShift",
                    "matching_results": 12
                }
            ]
        }
    ]
}

```

The `nested()` segment of the query found 1993 `enriched_text.entities` nested objects. The filter was applied to those objects and found 645 `enriched_text.entities` of type `Organization`.

Terminal operations

For most aggregation types, when you construct a query with multiple aggregation operations, the first operation is applied to the documents. Then, the output of that operation is used as the input for the next operation. However, a subset of the aggregation types are ***terminal operations***. The output of a terminal operation is not used as input for the next aggregation. Instead, the output is returned in a discrete group.

For an example of a request that combines aggregation types and includes an aggregation that performs a terminal operation, see the second [example](#) for the `average` aggregation type.

Aggregation types

The following types of aggregations are supported:

- [average](#)
- [filter](#)
- [group_by](#)
- [histogram](#)
- [max](#)
- [min](#)
- [nested](#)
- [pair](#)
- [sum](#)
- [term](#)
- [timeslice](#)
- [top_hits](#)
- [trend](#)
- [topic](#)
- [unique_count](#)

For Document Retrieval project types, when you don't include an aggregation parameter in a query request, a default aggregation request is applied. For more information, see [Document Retrieval project aggregations](#).

For more information about how to submit a query, see the Discovery [API reference](#).

average

Returns the mean of values of the specified field across all matching documents.

Syntax

```
average(field)
```

Example

Product	Price
I Series	200
J Series	450
X Series	325

Table 1. Sample product prices

When the `average` aggregation type is applied to a set of documents in which the `price` field contains the values that are shown in Table 1, the result is `325`.

```
average(price)=325
```

This aggregation type performs a terminal operation. When combined with other aggregations, the output is not used as input for the next aggregation. The output is returned in a discrete group.

```
{
  "query": "brake",
  "aggregation": "term(field:STATE,count:3).average(field:VEH_SPEED).term(field:CITY,count:2)"
}
```

For each state returned by the first `term` aggregation operation, the response shows the average vehicle speed specified in the incident reports. Notice that the second `term` aggregation uses the output from the first `term` aggregation, not the `average` aggregation, as its input.

```
{
  "matching_results": 9064,
  "retrieval_details": {
    "document_retrieval_strategy": "untrained"
  },
  "aggregations": [
    {
      "type": "term",
      "field": "STATE",
      "count": 3,
      "results": [
        {
          "key": "CA",
          "matching_results": 1210,
          "aggregations": [
            {
              "type": "average",
              "field": "VEH_SPEED",
              "value": 26.239653512993264
            }
          ]
        },
        {
          "type": "term",
          "field": "CITY",
          "count": 2,
          "results": [
            {
              "key": "LOS ANGELES",
              "matching_results": 77
            },
            {
              "key": "SAN DIEGO",
              "matching_results": 66
            }
          ]
        }
      ]
    }
  ]
}
```

filter

A modifier that narrows the document set of the aggregation query that it precedes.

Syntax

```
filter(field)
```

Example

The following example filters the matching document set to include only documents that mention **IBM**.

```
filter(enriched_text.entities.text:IBM)
```

When combined with other aggregations, filters the matching documents set to include only those documents that meet the condition you specify.

```
{
  "query": "brake",
  "aggregation": "filter(VEH_SPEED>50).term(field:STATE,count:3).term(field:CITY,count:2)"
}
```

The query response shows cities where incidents happen that involve the brakes and the vehicle speed is over 50.

```
{
  "matching_results": 9064,
  "retrieval_details": {
    "document_retrieval_strategy": "untrained"
  },
  "aggregations": [
    {
      "type": "filter",
      "match": "VEH_SPEED>50",
      "matching_results": 1075,
      "aggregations": [
        {
          "type": "term",
          "field": "STATE",
          "count": 3,
          "results": [
            {
              "key": "CA",
              "matching_results": 176,
              "aggregations": [
                {
                  "type": "term",
                  "field": "CITY",
                  "count": 2,
                  "results": [
                    {
                      "key": "FONTANA",
                      "matching_results": 6
                    },
                    {
                      "key": "ALTA LOMA",
                      "matching_results": 5
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

group_by

Separates results into groups that you define.

Syntax

```
group_by(condition:[(condition 1),(condition 2)...])
```

Each condition must be specified as a valid Discovery Query Language expression surrounded by parentheses. For example, **(age<20)** or **(flavor:chocolate)**. The maximum number of conditions that you can define is 50.

You can optionally include the **relevancy** parameter and set it to **true** to return the relevancy value of the set of documents that meet the specified condition. When **true**, the results are sorted by relevance. When **false**, the results are sorted by the highest number of **matching_results**.

Example

The following request looks for documents that mention the term `engine`, and groups them by car manufacturing year. The documents are sorted into 3 groups, one group of traffic incident reports involving cars that were manufactured before 2000, one group for cars manufactured in 2000, and one group for cars manufactured after 2000.

```
{  
  "query": "engine",  
  "aggregation": "group_by(condition:[(YEARTXT<2000),(YEARTXT=2000),(YEARTXT>2000)],relevancy:true)"  
}
```

The results might look like this:

```
{  
  "type": "group_by",  
  "results": [  
    {  
      "key": "YEARTXT<2000",  
      "matching_results": 2034,  
      "relevancy": 1.0,  
      "total_matching_documents": 2034,  
      "estimated_matching_results": 2034  
    },  
    {  
      "key": "YEARTXT=2000",  
      "matching_results": 1738,  
      "relevancy": 1.0,  
      "total_matching_documents": 1738,  
      "estimated_matching_results": 1738  
    },  
    {  
      "key": "YEARTXT>2000",  
      "matching_results": 32708,  
      "relevancy": 1.0,  
      "total_matching_documents": 32708,  
      "estimated_matching_results": 32708  
    }  
  ]  
}
```

histogram

Creates numeric interval segments to categorize documents.

Syntax

```
histogram({field},{interval})
```

Uses field values from a single numeric field to describe the category. The field that is used to create the histogram must have a number data type, such as `integer`, `float`, `double`, or `date`.

Nonnumber types such as `string` are not supported. For example, `"price": 1.30` is a number value that works, and `"price": "1.30"` is a string, so it doesn't work.

Use the `interval` argument to define the size of the sections for the results to be split into. Interval values must be whole, nonnegative numbers. Choose a value that makes sense for segmenting the typical values from the field.

Histograms can process decimal values that are specified in a field, but the interval must be a whole number.

You can optionally include a custom name by including a `name` parameter.

Example

For example, if your data set includes the price of several items, like: `"price": 1.30`, `"price": 1.99`, and `"price": 2.99`, you might use intervals of `1`, so that you see everything that is grouped in the range `1 - 2`, and `2` and `3`. You do not want to use an interval of `100` because then all of the data ends up in the same segment.

```
histogram(product_price,interval:1)
```

max

Returns the highest value in the specified field across all matching documents.

Syntax

```
max(field)
```

Example

Product	Price
I Series	200
J Series	450
X Series	325

Table 2. Sample product prices

When the `max` aggregation type is applied to a set of documents in which the `price` field contains the values that are shown in Table 2, the result is `450`.

```
max(price)=450
```

This aggregation type performs a terminal operation. When combined with other aggregations, the output is not used as input for the next aggregation. The output is returned in a discrete group.

min

Returns the lowest value in the specified field across all matching documents.

Syntax

```
min(field)
```

Example

Product	Price
I Series	200
J Series	450
X Series	325

Table 3. Sample product prices

When the `min` aggregation type is applied to a set of documents in which the `price` field contains the values that are shown in Table 3, the result is `200`.

```
min(price)=200
```

This aggregation type performs a terminal operation. When combined with other aggregations, the output is not used as input for the next aggregation. The output is returned in a discrete group.

nested

Applying `nested` before an aggregation query restricts the aggregation to the area of the results that are specified.

For example, `nested(enriched_text.entities)` means that only the `enriched_text.entities` components of any result are used to aggregate against.

The following example checks how many mentions are returned per model type.

```
nested(enriched_text.entities).term(enriched_text.entities.model_name)
```

The result shows that there are a total of 50 recognized entities and all of them are of type NLU.

```
"aggregations": [
```

```
{
  "type": "nested",
  "path": "enriched_text.entities",
  "matching_results": 50,
  "aggregations": [
    {
      "type": "term",
      "field": "enriched_text.entities.model_name",
      "results": [
        {
          "key": "natural_language_understanding",
          "matching_results": 50
        }
      ]
    }
  ]
}
```

For another example, see [Starting with nested objects](#).

pair

Analyzes relationships between two fields.

Syntax

```
pair(first:{aggregation},second:{aggregation})
```

The first and second `{aggregation}` values must be one of the following aggregation types:

- `term`
- `group_by`
- `histogram`
- `timeslice`

The `relevancy` parameter from the `term` or `group_by` aggregation is ignored. The `pair` aggregation type calculates relevancy values by using combinations of document sets from the results of the two aggregations.

Only one pair aggregation can be used per query request, and it cannot be combined with any other aggregations.

Example

For example, you might specify `term(model_name)` as the first aggregation and `term(component_name)` as the second. Each of the aggregations returns the following values as keys of aggregated document sets:

- `term(model_name): Accord, CR-V`
- `term(component_name): engine, brake, radiator`

The calculated relevancy values of combinations of each of the document sets might look like this:

- Accord x engine
- Accord x brake
- Accord x radiator
- CR-V x engine
- CR-V x brake
- CR-V x radiator

The response defines a two-dimensional array of aggregation results, which can be represented in a table.

Car model	Component: engine	Component: brake	Component: radiator
Accord	Accord x engine	Accord x brake	Accord x radiator
CR-V	CR-V x engine	CR-V x brake	CR-V x radiator

Table 4. Pair aggregation example

Each array of columns and rows of the table is sorted in the same order of the results of the first and second aggregations. For example, if you specify the `term` aggregation as the first argument, the resulting column arrays are sorted by frequency of terms. If you use the `timeslice` aggregation as the second argument, the row arrays are sorted by date or time.

sum

Adds the values of the specified field across all matching documents.

Syntax

```
sum(field)
```

Example

Product	Price
I Series	200
J Series	450
X Series	325

Table 6. Sample product prices

When the `sum` aggregation type is applied to a set of documents in which the `price` field contains the values that are shown in Table 6, the result is `975`.

```
sum(price)=975
```

This aggregation type performs a terminal operation. When combined with other aggregations, the output is not used as input for the next aggregation. The output is returned in a discrete group.

term

Indicates the frequency of a term or set of terms in a set of queried documents.

Syntax

```
term(field:{field_name})
```

You can optionally specify the following parameters:

- `count`: Specifies the maximum number of terms to return.
- `name`: You can optionally include a custom name. Not returned if relevancy information is included in the request.
- `relevancy`: Boolean value that indicates whether to include relevancy information in the result. You can use relevancy to get a score that indicates the level of relevancy between the term and keywords in the query. This parameter is `false` by default. If set to true, the following fields are returned also:
 - `total_matching_documents`: Number of documents in the collection where the term is mentioned in the specified field.
 - `estimated_matching_results`: Number of documents that are estimated to have the term in the specified field in the set of documents that are returned by the query.

Example

The following example returns the text from the recognized entities in the document, and specifies to return a maximum of 10 terms.

For example:

```
term(enriched_text.entities.text,count:10)
```

When `relevancy` is set to `true`, a relevancy score is shown in the results. Relevancy measures the level of uniqueness of the frequency count compared to other documents that match your query. If the relevancy shows 2.0, it means that the number of times that the two data points intersect is 2 times larger than expected.

For more examples, see [Grouping documents](#) and [Combining aggregation types](#).

timeslice

A specialized histogram that uses dates to create interval segments.

Syntax

The syntax is `timeslice({field},{interval},{time_zone})`.

- The field that you specify must have a `date` data type. For more information about date field, see [How dates are handled](#).
- Valid interval values are `1second` or `{n}seconds`, `1minute` or `{n}minutes`, `1hour` or `{n}hours`, `1day` or `{n}days`, `1week` or `{n}weeks`, `1month` or `{n}months`, and `1year` or `{n}years` where `{n}` is a number.
- You can optionally include a custom name by including a `name` parameter.

Example

The following example shows the number of matches for each day value.

```
timeslice(field:DATEA,interval:1day)
```

The results look as follows.

```
"aggregations": [
  {
    "type": "timeslice",
    "field": "DATEA",
    "interval": "1d",
    "results": [
      {
        "key": 1262304000000,
        "key_as_string": "2010-01-01T00:00:00.000Z",
        "matching_results": 5
      },
      {
        "key": 1262390400000,
        "key_as_string": "2010-01-02T00:00:00.000Z",
        "matching_results": 18
      },
      {
        "key": 1262476800000,
        "key_as_string": "2010-01-03T00:00:00.000Z",
        "matching_results": 38
      },
      {
        "key": 1262563200000,
        "key_as_string": "2010-01-04T00:00:00.000Z",
        "matching_results": 66
      }
    ]
  }
]
```

top_hits

Returns the documents ranked by the score of the query or enrichment. Can be used with any query parameter or aggregation.

Syntax

```
{aggregation}.top_hits({n})
```

Example

The following example returns the top hit for the term `halt` per city.

```
{
  "query": "halt",
  "aggregation": "term(CITY).top_hits(1)"
}
```

The response contains the top query results for the term `halt` grouped by cities mentioned in documents where the term is most mentioned. Ten results are returned by default. For each of the 10 cities, the document with the top score is returned as the `hit` object. The content for each `hit` in the `hits` array matches the content in each `result` in the `results` array. Only the order of the results is different.

```
"aggregations": [
  {
    "type": "term",
    "field": "CITY",
    "results": [
      {
        "key": "LOS ALTOS",
        "matching_results": 3,
        "hit": {
          "id": "12345678901234567890123456789012"
        }
      }
    ]
  }
]
```

```

"aggregations": [
  {
    "type": "top_hits",
    "size": 1,
    "hits": {
      "matching_results": 3,
      "hits": [
        {
          "document_id": "2bed19a9069442fd82542827ebe260d5_7015",
          ...
        }
      ]
    }
  },
  {
    "key": "ANDOVER",
    "matching_results": 2,
    "aggregations": [
      {
        "type": "top_hits",
        "size": 1,
        "hits": {
          "matching_results": 2,
          "hits": [
            {
              "document_id": "2bed19a9069442fd82542827ebe260d5_18329",
              ...
            }
          ]
        }
      }
    ]
  },
  ...
  {
    "key": "ACTON",
    "matching_results": 1,
    "aggregations": []
  }
  ...
]

```

This aggregation type performs a terminal operation. When combined with other aggregations, the output is not used as input for the next aggregation. The output is returned in a discrete group.

trend

Detects sharp and unexpected changes in the frequency of a keyword value in a specified time period based on the past frequency changes of the keyword value.

Syntax

```
trend(facet:{aggregation},time_segments:{aggregation})
```

The first (`facet`) aggregation must be one of the following types of aggregations:

- `term`
- `group_by`

The `relevancy` parameter from the `term` or `group_by` aggregation is ignored.

The second (`time_segments`) aggregation must be an aggregation of type `timeslice`.

You can alternatively include the following parameters:

- `show_estimated_matching_results:true`: Indicates whether to include the `estimated_matching_results` information in the result. This field contains the number of documents that are estimated to have the term in the specified field or meet the conditions in the specified aggregation for the specified time interval in the set of documents that are returned by the query.
- `show_total_matching_documents:true`: Indicates whether to include the `total_matching_documents` information in the result. This field contains the number of documents in the collection where the term is mentioned in the specified field or the condition is met.

Only one trend aggregation can be used per query request, and it cannot be combined with any other aggregations.

Example

The following example calculates the *trend indicator* or *trend index* by using combinations of results from the following aggregations:

- term(flavor): vanilla, chocolate, mint
- timeslice(date, 1month): Jan 2020, Feb 2020, Mar 2020, Apr 2020, May 2020, Jun 2020

```
trend( facet: aggregation(<parameter>...), time_segments: timeslice(<parameter>...)),
show_estimated_matching_results: <true_or_false>, show_total_matching_documents: <true_or_false> )
```

The resulting matrix can be represented in a table.

Month in 2020	Flavor: vanilla	Flavor: chocolate	Flavor: mint
Jan	vanilla x Jan	chocolate x Jan	mint x Jan
Feb	vanilla x Feb	chocolate x Feb	mint x Feb
Mar	vanilla x Mar	chocolate x Mar	mint x Mar
Apr	vanilla x Apr	chocolate x Apr	mint x Apr
May	vanilla x May	chocolate x May	mint x May
Jun	vanilla x Jun	chocolate x Jun	mint x Jun

Table 5. Trend aggregation example

In the following sample response, the key information is the **trend_indicator** value. The trend indicator measures the increase ratio of the frequency of a given facet value for a given time interval compared to the expected average frequency. The expected average frequency is calculated based on the changes in the past time interval frequencies of the given facet value, using a weighted arithmetic mean.

If the standardized residual value is less than -2, the observed frequency is less than the expected frequency. If it is greater than 2, the observed frequency is greater than the expected frequency. If the standardized residual is greater or less than the expected frequency by 3 or more, then something unusual is happening and suggests that there might be an anomaly that is worth investigating.

For example, the expected number of feedback submissions for the **vanilla** flavor in May is calculated from the number of feedback submissions that were received previously (from Jan to Apr). The result is **5.341**. The actual number of feedback submissions in May is **10**. The results indicate that the vanilla flavor got about twice the number of feedback submissions as expected. The standardized residual value is **2.016**, which is greater than expected, but not unusually so.

```
{
  "aggregations": [
    {
      "type": "trend",
      "facet": "term(flavor)",
      "time_segments": "timeslice(date, 1month)",
      "show_estimated_matching_results": true,
      "show_total_matching_documents": true,
      "results": [
        {
          "aggregations": [
            {
              "type": "term",
              "field": "flavor",
              "results": [
                {
                  "key": "vanilla",
                  "matching_results": 36,
                  "aggregations": [
                    {
                      "type": "timeslice",
                      "field": "date",
                      "results": [
                        {
                          "key": 1577836800000,
                          "key_as_string": "2020-01-01T00:00:00.000Z",
                          "matching_results": 4,
                          "trend_indicator": 0.0,
                          "total_matching_documents": 7,
                          "estimated_matching_results": 0.0
                        },
                        {
                          "key": 1588291200000,
                          "key_as_string": "2020-05-01T00:00:00.000Z",
                          "matching_results": 10,
                          "trend_indicator": 2.016,
                          "total_matching_documents": 10,
                          "estimated_matching_results": 5.341
                        }
                      ]
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

```

        "matching_results": 10,
        "trend_indicator": 2.016106745,
        "total_matching_documents": 12,
        "estimated_matching_results": 5.340760209
    },
    {
        "key": 1590969600000,
        "key_as_string": "2020-06-01T00:00:00.000Z",
        "matching_results": 5,
        "trend_indicator": -0.763212711,
        "total_matching_documents": 11,
        "estimated_matching_results": 7.022515985
    }
]
},
{
    "key": "chocolate",
    "matching_results": 10,
    "aggregations": [...]
},
{
    "key": "mint",
    "matching_results": 25,
    "aggregations": [...]
}
...
}

```

topic

Detects how much the frequency of a keyword value deviates from the expected average for the specified time period. This aggregation type does not use data from previous time periods. It calculates an index by using the averages of frequency counts of other keyword values for the specified time period.

Syntax

```
topic(facet:{aggregation},time_segments:{aggregation})
```

The first (`facet`) aggregation must be one of the following types of aggregations:

- `term`
- `group_by`

The `relevancy` parameter from the `term` or `group_by` aggregation is ignored.

The second (`time_segments`) aggregation must be an aggregation of type `timeslice`.

You can alternatively include the following parameters:

- `show_estimated_matching_results:true`: Indicates whether to include the `estimated_matching_results` information in the result. This field contains the number of documents that are estimated to have the term in the specified field or meet the conditions in the specified aggregation for the specified time interval in the set of documents that are returned by the query.
- `show_total_matching_documents:true`: Indicates whether to include the `total_matching_documents` information in the result. This field contains the number of documents in the collection where the term is mentioned in the specified field or the condition is met.

Only one topic aggregation can be used per query request, and it cannot be combined with any other aggregations.

Example

```
{
    "query: like",
    "aggregation": "topic( facet: term(flavor), time_segments: timeslice(date, 1month), show_estimated_matching_results: true,
show_total_matching_documents: true )"
}
```

With the same data set and aggregation as is used in the term aggregation example, the results might look as follows.

Notice that the `topic_indicator` values are different from the `trend_indicator` values that are returned by the `trend` aggregation. While both are calculated from the actual and expected frequencies, they differ because their expected frequencies are computed differently. In the `trend` aggregation, the expected frequency of the feedback submissions for vanilla-flavored ice cream in May is computed from the number of feedback submissions that were received for vanilla previously (from Jan to Apr) and the total number of feedback submissions received for all of the flavors in May. However, in the `topic` aggregation, the expected frequency of feedback submissions for vanilla-flavored ice cream in May is calculated from the number of feedback submissions that were received for vanilla and the total number of feedback submissions received for all of the flavors in May. In this example, the expected frequency

result is **12.169**, the actual frequency is **10**, and the **topic_indicator** is **-0.621777032**.

```
{  
  "aggregations": [  
    {  
      "type": "topic",  
      "facet": "term(flavor)",  
      "time_segments": "timeslice(date, 1month)",  
      "show_estimated_matching_results": true,  
      "show_total_matching_documents": true,  
      "results": [  
        {  
          "aggregations": [  
            {  
              "type": "term",  
              "field": "flavor",  
              "results": [  
                {  
                  "key": "vanilla",  
                  "matching_results": 36,  
                  "aggregations": [  
                    {  
                      "type": "timeslice",  
                      "field": "date",  
                      "results": [  
                        {  
                          "key": 1577836800000,  
                          "key_as_string": "2020-01-01T00:00:00.000Z",  
                          "matching_results": 4,  
                          "topic_indicator": -0.027972712,  
                          "total_matching_documents": 7,  
                          "estimated_matching_results": 4.056338028  
                        },  
                        {  
                          "key": 1588291200000,  
                          "key_as_string": "2020-05-01T00:00:00.000Z",  
                          "matching_results": 10,  
                          "topic_indicator": -0.621777032,  
                          "total_matching_documents": 12,  
                          "estimated_matching_results": 12.16901408  
                        },  
                        {  
                          "key": 1590969600000,  
                          "key_as_string": "2020-06-01T00:00:00.000Z",  
                          "matching_results": 5,  
                          "topic_indicator": -0.787665504,  
                          "total_matching_documents": 11,  
                          "estimated_matching_results": 7.098591549  
                        }  
                      ]  
                    ]  
                  ]  
                }  
              ]  
            }  
          ]  
        }  
      ]  
    }  
  ]  
}
```

unique_count

Returns a count of the unique instances of the specified field in the collection.

Syntax

```
unique_count(field)
```

Example

The following aggregation requests the number of unique enrichment types that are recognized in the query.

```
unique_count(enriched_text.keyword.type)
```

The result indicates that there are 17 matching results. In those 17 documents, 14 entity types are mentioned.

```
{
  "matching_results": 17,
  "retrieval_details": {
    "document_retrieval_strategy": "untrained"
  },
  "aggregations": [
    {
      "type": "unique_count",
      "field": "enriched_text.entities.type",
      "value": 14.0
    }
  ],
  "results": []
}
```

This aggregation type performs a terminal operation. When combined with other aggregations, the output is not used as input for the next aggregation. The output is returned in a discrete group.

In the following example, the aggregation parameter requests for the results to show the first 45 most-frequently mentioned entities. Per entity, it indicates how many documents mention the term and how many times in total that the term occurs.

```
term(enriched_text.entities.text,count:45).unique_count(enriched_text.entities.type)
```

The results include several aggregations such as the following group for the term **PostgreSQL**. The aggregation indicates that the term appears in 4 documents and is mentioned 12 times.

```
{
  "key": "PostgreSQL",
  "matching_results": 4,
  "aggregations": [
    {
      "type": "unique_count",
      "field": "enriched_text.entities.type",
      "value": 12.0
    }
  ]
}
```

Curations API

The Curations feature is beta functionality.

Use curations to specify the exact document to return in response to a specific natural language query. Curations can guarantee that frequent or important questions always return the most valuable document. The **confidence_score** for a curated query is always **1.00000**.

This beta feature is only available from the API and is applied only to natural language queries, not queries that are specified by using the Discovery Query Language. Beta features are not available from the SDKs.

You can define up to 1,000 curations. For more information, see [Create curation](#) in the API reference.

This example shows how a curation is added with the API. When querying with the same or similar **natural_language_query** the document with the **document_id** of **document_id1234** is returned.

```
{
  "natural_language_query": "curations in watson discovery",
  "curated_results": [
    {
      "document_id": "document_id1234",
      "collection_id": "collection_id1234"
    }
  ]
}
```

The natural language query that is submitted by the customer must be an exact match for the query that is specified in the curation. Both queries, the one submitted by the user at run time and the one that is submitted by the curation API and then stored in the index, undergo query analysis. The query analyzer lemmatizes text, removes stop words, and adds query expansions.

You can optionally specify a hard-coded response to the query by including a snippet. A snippet is a response that you author and that is returned when the associated document is returned for the specified natural language query.

```
{
  "curations": [
    {

```

```
"curation_id": "c1175536f509405bc68a9f76235fa7bbb6f9af2f",
"natural_language_query": "What is a project",
"curated_results": [
  {
    "collection_id": "47477591-b520-6039-0000-017ea213e837",
    "document_id": "web_crawl_123a2a56-8c26-5acb-9544-c4702ac899a4",
    "snippet": "A project is a convenient way to collect and manage the resources in your application. You can assign a project type and connect your data to the project by creating a collection."
  }
]
```

If `passages.per_document` is `true`, the text snippet that you specify is returned as the top passage in the `passage_text` field instead of the original passage that is chosen by search. Only one text snippet can be specified per document. If `passages.max_per_document` is greater than `1`, the snippet is returned first, followed by the passages that are chosen by search. Query filters are applied to curation results.

Analyzing data with the Content Mining application

Analyzing your data with the Content Mining application

Use the Discovery Content Mining application to analyze your data. The application shows subsets of your information in visualizations that can help you to find patterns, trends, and anomalies.



Note: Only users of installed deployments (IBM Cloud Pak for Data) or Enterprise and Premium plan managed deployments can use the Content Mining application.

Overview video



View video: [IBM Watson Discovery Content Mining](#)

Video transcript

Watson Discovery Content Mining Project presented by Stuart Strolin. (Music intro) The purpose of this video is to familiarize you with the content mining project in Watson Discovery.

Content mining is one of the primary use cases for Watson Discovery and is used for analyzing and exploring both structured and unstructured data to find insights and extract hidden meaning. It is used by both the citizen analyst and the data scientist.

The content mining project can be used for all types of analysis because the user interface is not specific to a particular industry or set of data.

In this scenario, you are an analyst for a fictitious automobile company. Operational reports have alerted the company to an unusual accident rate for one of their cars. Your job is to find out why.

Using the content mining project, you begin your analysis by looking at the unstructured data from the national motor vehicle incident reports. You are presented with an interface that allows you to select the car model and begin your analysis (on the Collections page). In this case, you are interested in the Hill Walker. You could type that information into the search section at the start of the page. But it's easier just to click on the item. You can add as many search terms and conditions as you like. But in reality, you want to let the application guide your analysis.

What you see now is the navigation view (in Guided mode). It keeps track of your analysis and provides options for next steps. It also provides a count of the number of documents that match your current state of analysis. In this small collection, the number of documents relating to the Hill Walker is only 51. In a production data set, the number would usually be much larger. Analyzing trends and anomalies is often a good way to start as it allows you to see if anything seems out of the ordinary.

Immediately, you notice that the Hill Walker has problems in December and January. You decide to investigate further by narrowing this initial exploration to just the month of December.

Notice how the navigation view at the top always keep you informed of where you are in your analysis. Next, you select **Analyze cause and characteristics** because you are interested in why things are happening.

You notice that words like 'snow' and 'brake' are highlighted together (in the Part of Speech section), so you add these to your analysis.

The Content Miner project has narrowed your investigation to a small number of complaints that can be easily read. (clicks Show Documents)

The common theme here is that there is an unexpected problem with the way the brakes are working in snowy conditions. You now have the information you need to ask the engineering department to perform a detailed inspection of the braking system and determine why it is not working as expected in snowy conditions.

In this demonstration, you saw how a citizen analyst using Watson Discovery and content mining can easily discover hidden meaning in unstructured text. (list of features, functionality, and use cases)

What will you do with Watson Discovery? (Music outro)

How it works

To analyze your data, you use **facets**. Facets give you a way to slice your data and visualize a subset of information so it is easier to

comprehend.

From the data analysis page for your collection, you can choose for the data to be shown in one of the following views:

Facets

Shows facets that are derived from annotations that are added to your documents by enrichments that are applied to your documents. Enrichments can include built-in Natural Language Processing enrichments, such as **Part of Speech** or **Entities**. They can also include custom enrichments that you add, such as dictionaries, regular expression patterns, and machine learning models.

Metadata facets

Shows facets that are derived from your data. When you add files to a collection, Discovery analyzes and indexes the data. Annotations are added to identify content types and are shown as metadata facets. The best metadata facets result when you ingest structured data, such as records from a CSV file.

Custom

Shows only the facets that you choose to add to the view. You can add a mix of enrichment-derived and content-derived facets to your custom view.

When you create a **Content Mining** project type, the **Part of Speech** facet is applied to your data automatically. This facet is a great place to start because it is valid for all data, no matter the subject. The output gives you a quick look at the terminology that is most common in the data.

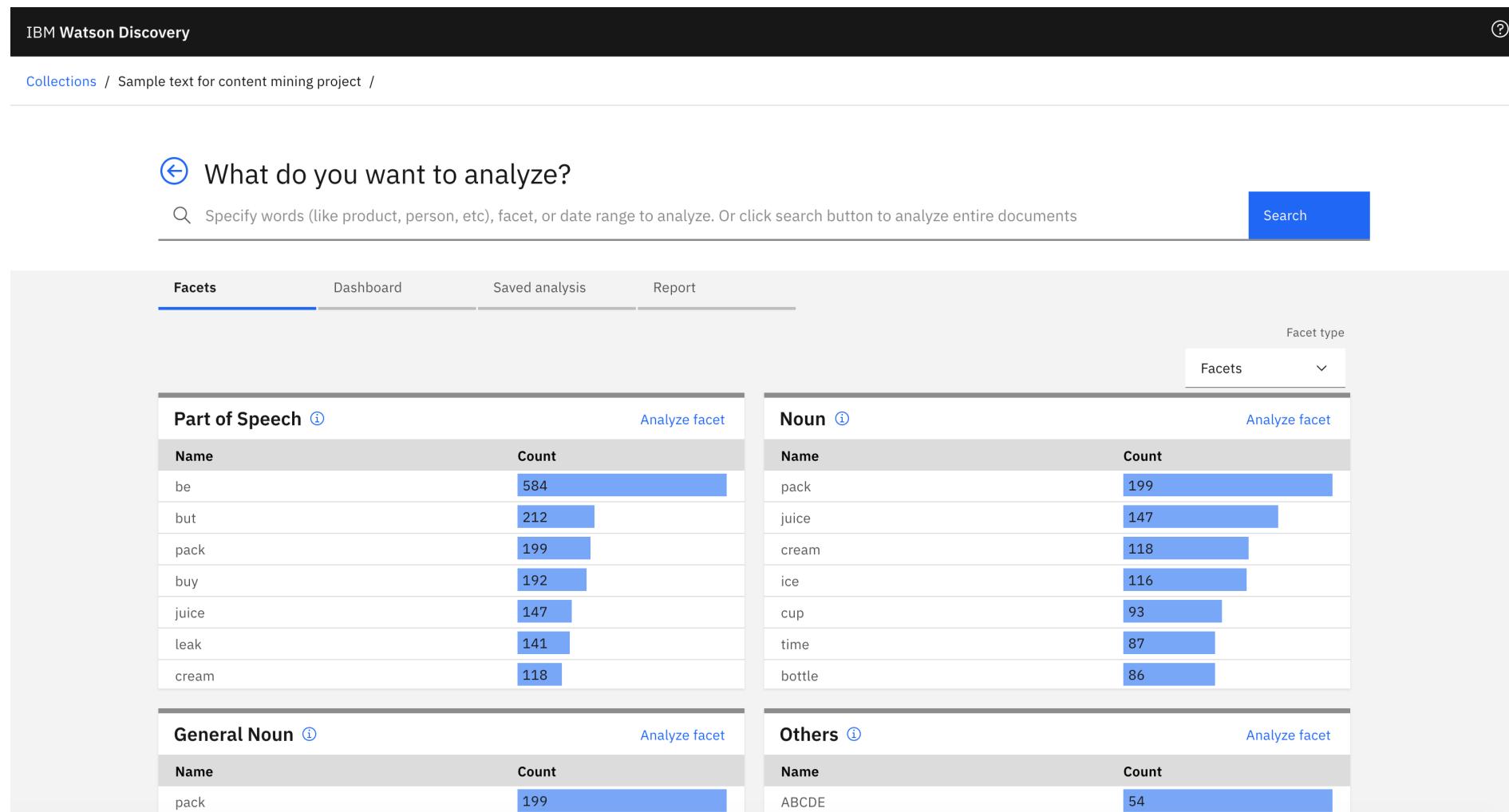


Figure 1. Watson Discovery Content Mining application home page

From this starting point, you can determine other ways to filter the data that might be useful.

If your data consists of traffic reports, for example, the **Part of Speech** facet might show that high frequency keywords include terms such as **engine**, **brake**, **fire**, **smoke**, and **spark**. Given this common terminology, you can create dictionaries to help you categorize and filter the data. The keywords from the example might lead you to create the following dictionaries:

- **component** dictionary for terms such as engine and brake
- **phenomenon** dictionary for terms such as fire, smoke, and spark

When you apply the dictionary enrichment to your data, it generates **annotations**. You can think of annotations as tags that you add to words or phrases, where the tag categorizes or identifies the meaning of the word or phrase. The resulting annotations function as new facets that you can use to filter and dissect your data further.

With your new **component** and **phenomenon** facets, for example, you can look for correlations between components and phenomena that are involved in traffic incidents.

[Learn about the ways that you can analyze your data.](#)

Digging deeper

To dig even deeper into your data, apply or create AI models that can find different types of information in your documents. You can apply built-in natural language processing models, such as the **Entities** enrichment that can recognize mentions of commonly known things, such as business or location names and other types of proper nouns. Or you can apply a custom model that recognizes terms and categories that are unique to your data.

[Extend your analysis by adding your own facets.](#)

Getting started

Before you can use the application, you must create a Discovery Content Mining project. After the project is created and data is uploaded, you can open the Content Mining application.

For more information, see [Creating projects](#).

Of course, you can't get out useful insights if you don't put the right type of information in. Be sure to include consistent data. If you want to find trends over time, your data must include data points that specify a date.

Data that is submitted in CSV file format is optimal. For a sample of a CSV file that provides interesting analysis capabilities, see [Analyzing CSV files](#).

Data analysis methods

Use tools from the Content Mining application to analyze your data.

You can analyze your data in the following ways:

- [Look for relevant keywords](#)
- [Find trends](#)
- [Identify anomalies in cyclical patterns](#)
- [Find characteristic words](#)
- [Analyze relationships](#)
- [Analyze relationships between many facets](#)

As you review the results of your analysis, you can flag documents that you want to research further later. For more information, see [Flagging documents](#).

When you find important insights, you can take a snapshot of the view, and then add it to a report to share with others. For more information, see [Creating a report](#).

Start your analysis

Use the content mining application to analyze documents in your collection based on the document text and any annotations or enrichments that are stored in the documents.

To start your analysis, complete the following steps:

1. Enter a search term, click a facet with which to filter the documents, or leave the search field blank to return all of your documents.
2. Click **Search**.

The guided mode view of the results shows suggested next steps that you can take to analyze your data further. If you don't want to see suggestions, you can switch to **Expert mode**. In Expert mode, the **Documents** view that lists the search results is returned whenever you submit a search.

The tasks in this topic describe how to use the application in guided mode.

Look for relevant keywords

To analyze keyword relevance, complete the following steps:

1. From the initial search page, submit a keyword search to filter the documents.
2. From the search results page in guided mode, click **Analyze cause or characteristics**.

After the characteristic words pane, a pane with relevancy information for each facet type is displayed.

The figure shows three facets in the IBM Watson Discovery interface:

- Characteristic Words**: A list of words with their counts and relevancy scores. The top words include: check, oil, stall, restart, shut, run, RPM, RPMS, ECM, ALTIMA, cool, extinguish, CX-7, EXCURSION, EXPEDITION, KAWASAKI MOTORS CORP., U.S.A., LAND ROVER, KAWASAKI, IT'S TERRIBLE, PROBLEM WAS FOUND, ASHEVILLE, SYSTEM WARNING, VEHICLE STALLED, COBALT WHICH HAS THE EXACT SAME PLATFORM AND STEERING HAS THE SAME ISSUES, help, plug, 1G3WX52H5YF, 1N4AL11D73C, 1G8AZ55F17Z.
- CITY**: A table showing cities with their document counts and relevancy scores. The data is as follows:

Name	Count	Relevancy
ASHEVILLE	14	2.20
ALEXANDRIA	27	1.33
NASHVILLE	20	1.22
DAYTON	15	1.19
MANCHESTER	18	1.18
ALBANY	18	1.15
SACRAMENTO	25	1.13
NEWARK	17	1.02

- MAKETXT**: A table showing vehicle models with their document counts and relevancy scores. The data is as follows:

Name	Count	Relevancy
KAWASAKI	30	2.54
LAND ROVER	36	2.02
NISSAN	588	1.38
MAZDA	164	1.35
YAMAHA	28	1.35
JAGUAR	29	1.33
OLDSMOBILE	30	1.30
SUBARU	83	1.28

Figure 1. Facet relevancy

Each relevancy pane shows a list of the keywords that occur in the documents that match the facet type.

The **Count** column shows the number of documents in the current result set that contain the keyword. The **Relevancy** column shows the level of uniqueness of the frequency count compared to other documents that match your query. High relevancy values are shown in shades of color with increasing intensity. The color begins at yellow, then increases to orange, and then to red.

Find trends

Use **Trends** analysis to find trends in your data. For example, you might see that a new product release aligns with an uptick in customer interest. Or that a new customer care approach is followed by an increase in customer satisfaction.

Important: Your documents must contain at least one date field for trend information to be available.

To find trends, complete the following steps:

- From the initial search page, enter a keyword or select a facet with number values to filter the documents.
- Click **Find trends and anomaly** from the list of suggested next steps that is displayed in the guided mode view.

The resulting bar graph shows the number of documents that mention the term or facet value that you specified in the search query over time.

The screenshot shows the IBM Watson Discovery interface with the following components:

- Header:** IBM Watson Discovery, Collections / Traffic CSV / Guided mode /
- Toolbar:** Includes icons for back, forward, search, and document actions.
- Central Panel:** A tree diagram showing "Traffic CSV" and "All documents (5,000)". A node labeled "General Noun problem (1,878)" is highlighted with a blue circle and has a dropdown menu with four options:
 - Analyze cause or characteristics
 - Analyze trends and anomaly
 - Show documents
 - Show analysis dashboard
- Dashboard View:** A "General Noun - Trends" section showing a timeline from 01/01/2 to 01/10. It lists terms like stop (203), light (367), day (236), speed (267), and wheel (171) with corresponding trend charts.
- Document View:** A list of documents with their titles and dates. The first document is titled "2bed19a9069442fd82542827ebe260d" and dated 1/4/2010, 7:00:00 PM. The text of the document includes: "MANY WEB SITES I VISITED, MANY CUSTOMERS HAVE HAD THE SAME PROBLEM WITH NO ACTION TO REPAIR IT FOR NO CHARGE. *TR".

Figure 2. Facet trend graph

The time series chart is rendered as a heat map. Each cell color indicates a level of relevancy.

3. You can click a facet to investigate it more closely. The facet is shown in a bar graph.

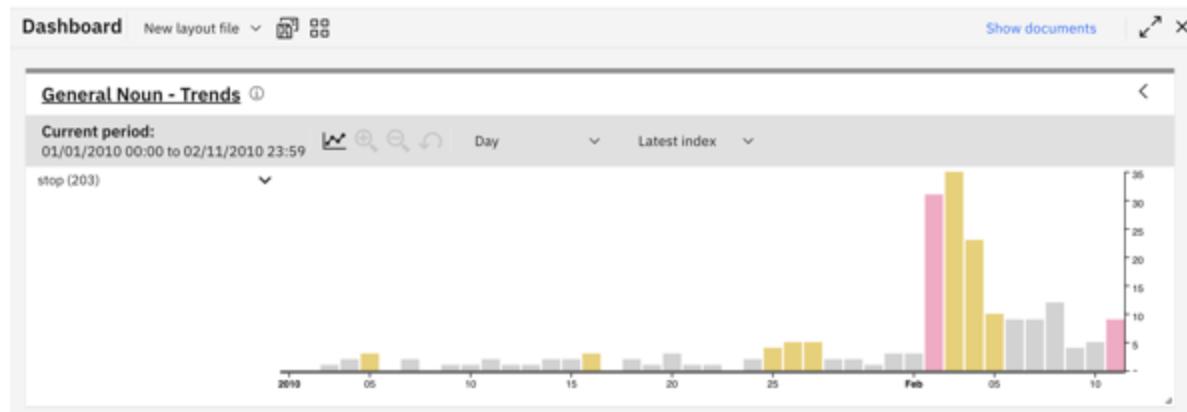


Figure 3. Facet trend detail in bar graph

Each individual bar graph highlights trends in your data that deviate from the normal distribution by displaying **increase indicators**.

Increase indicators measure how much the frequency of a facet value on a specific date or in a particular time interval deviates from the expected average frequency. The average is calculated based on the changes in the past time interval frequencies.

You can click individual items in a visualization or click and drag the cursor to select contiguous items.

The cyclical data is calculated from the current time zone setting of your collection. If you want to change the time zone that is used by the graph, see [Change the time zone](#).

Identify anomalies in cyclical patterns

Use **Topic** analysis to find anomalies in seasonal, monthly, or even daily patterns that are present in your data.



Important: Your documents must contain at least one date or time field for topic information to be available.

Topic analysis focuses on how much the frequency of a keyword deviates from the expected average frequency in a specific time period. The expected average uses all of the averages of the frequency counts for other keywords in the same time period. This method of analysis is useful for identifying patterns that occur cyclically and highlights any unexpected changes that might occur in these cyclical patterns.

To find anomalies, complete the following steps:

1. From the initial search page, enter a keyword or select a facet with number values to filter the documents.
2. From the search results page in guided mode, click **Analyze cause or characteristics**.
3. From the **Facet analysis** pane, select **Topic**.
4. Adjust the following values to suit your analysis:
 - Number of results
 - Date facet
 - Time scale
 - Date range
5. Choose a target facet or subfacet, and then click **Analyze**.

The resulting time series graph shows changes in the frequency of keyword mentions over time.

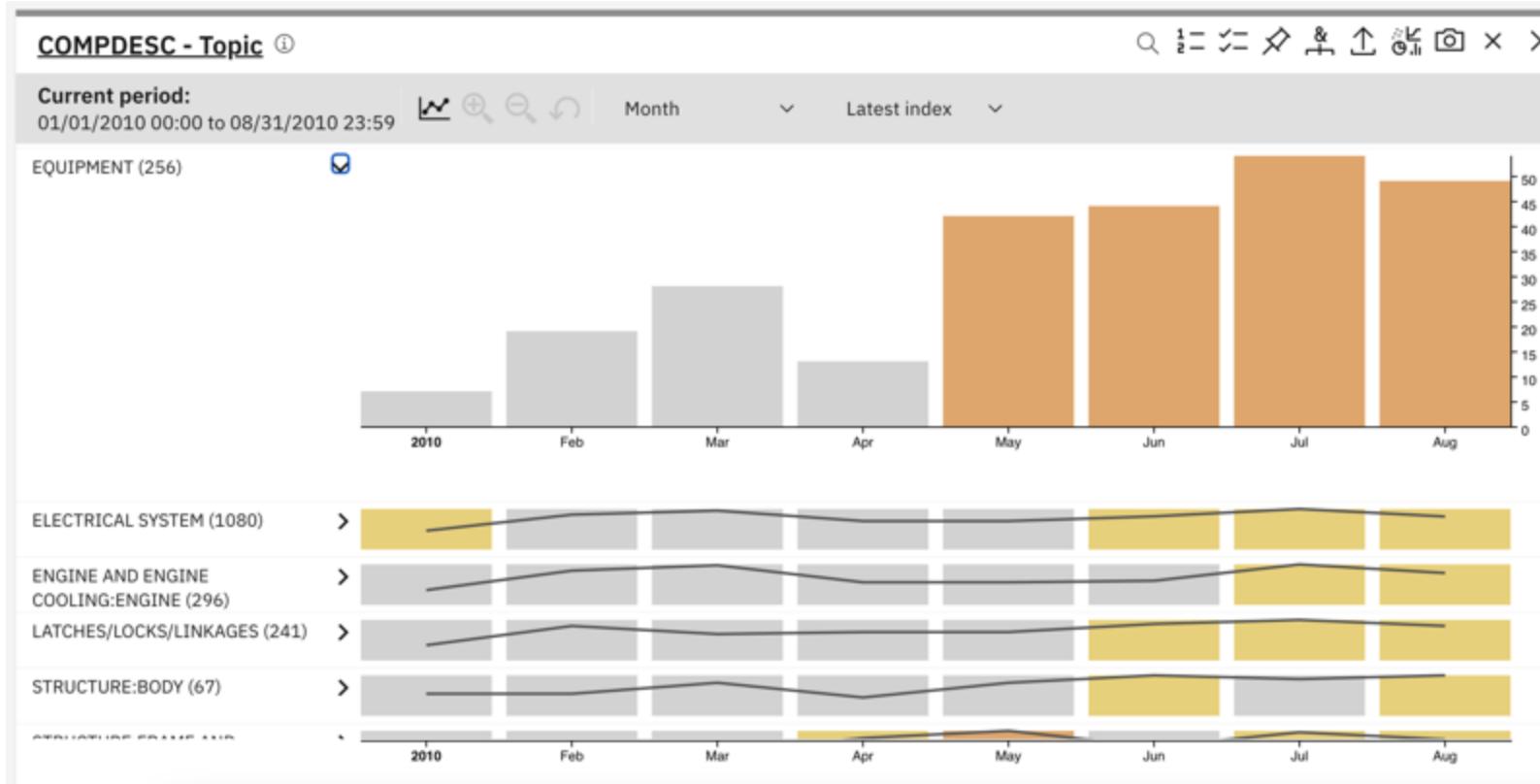


Figure 4. Topic analysis time series view

Color coding is used to highlight when the number of mentions deviates from the expected frequency. The higher the deviation, the more intense the color, from yellow to orange to red. The average is calculated based on the frequency of occurrence of other keywords in the same time period.

The cyclical data is calculated from the current time zone setting of your collection. If you want to change the time zone that is used by the graph, see [Change the time zone](#).

Find significant terms

Find characteristic words from your data set. The characteristic words view is a word cloud that shows terms that are mentioned frequently in the documents you are analyzing.

You can click a word from the word cloud to add it to the existing query and filter the current document set to include only documents that also mention the specified word.

To find significant terms, complete the following steps:

- From the search results page in guided mode, click **Analyze cause or characteristics**.

The characteristic words view is displayed.

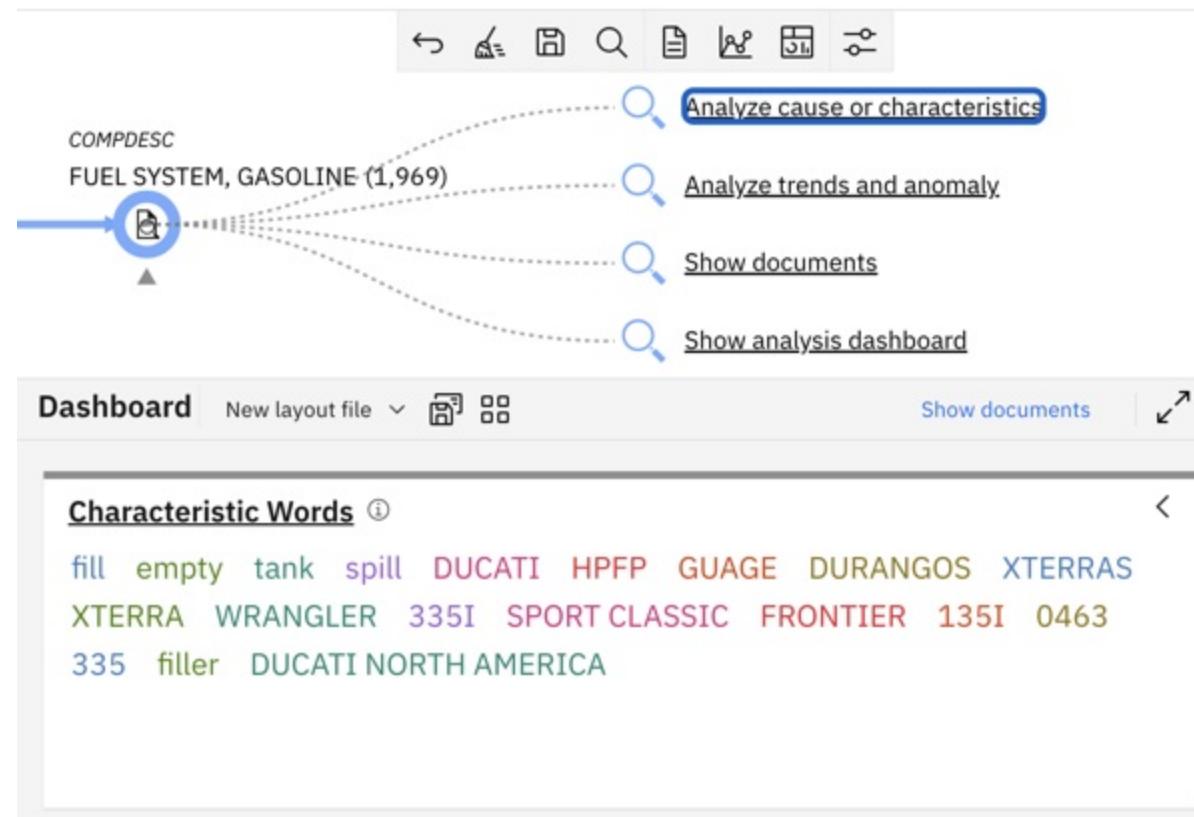


Figure 5. Characteristic word cloud



Note: The different font colors help to distinguish the words from one another; they have no statistical meaning.

- Click a word in the cloud to limit the document set to include only documents that mention the word.

Analyze relationships between two facets

Use **Pairs** analysis to see how two facets are related to one another.

To compare two facets, complete the following steps:

1. From the **Facet analysis** pane, select **Pairs**.
2. Find the first facet that you want to compare in the list. Click either the X- or Y-axis icon that is associated with the facet to indicate where you want the facet values to be displayed in a two-dimensional graph.
3. Find the second facet, and then click the remaining axis icon. For example, if you selected the X-axis icon previously, select the Y-axis icon for the second facet.

Data from the two facets is displayed in a graph.

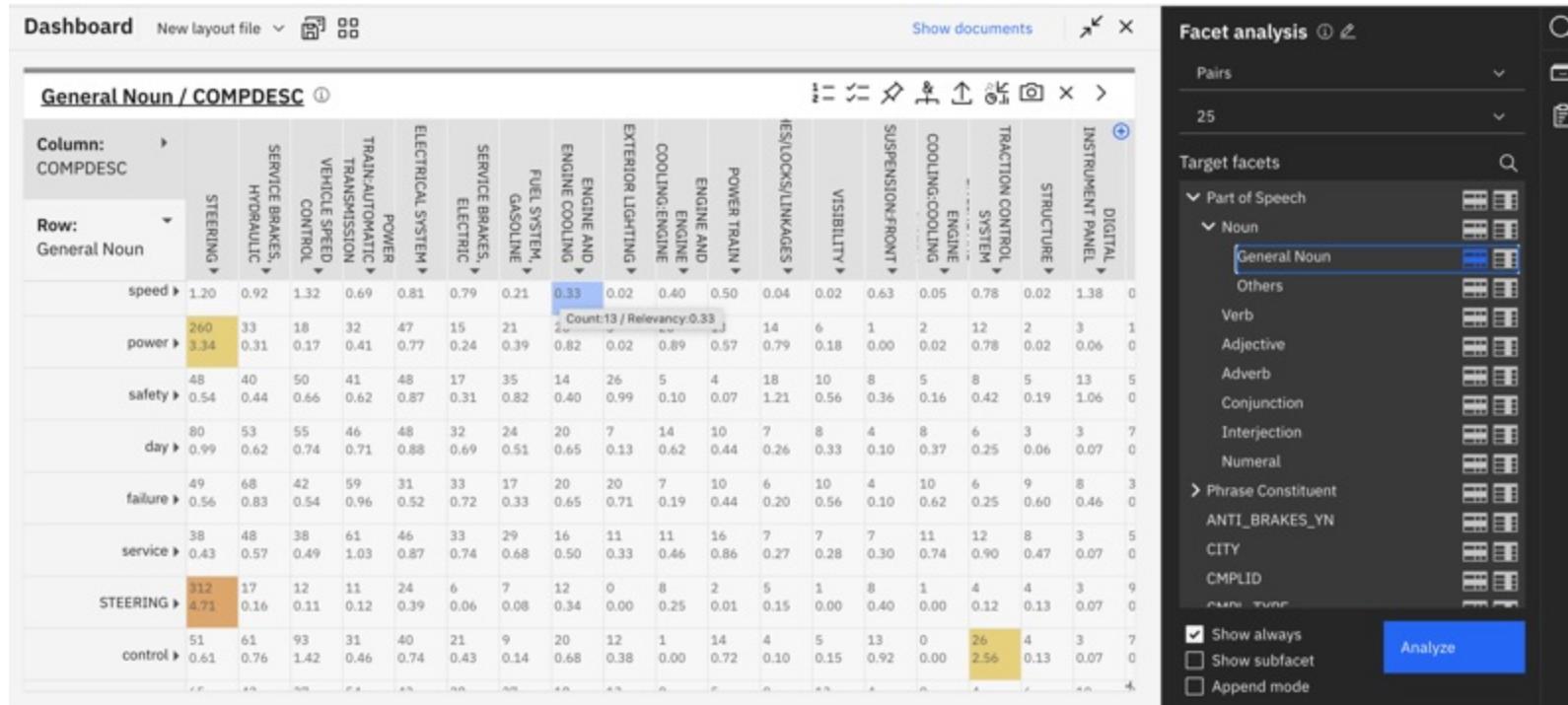


Figure 6. Facet comparison graph

The graph shows two numbers. The first number is a frequency count and the second number is a relevancy value. The frequency count measures how many times the two data points are found together in a document. Relevancy measures the level of uniqueness of the frequency count compared to other documents that match your query. If the relevancy shows 2.0, it means that the number of times that the two data points intersect is 2 times larger than expected. To help you identify anomalies that might require more in-depth analysis, high relevancy values are shown in shades of color with increasing intensity, from yellow to orange to red.

Analyze relationships between many facets

Use **Connections** analysis to see how multiple facets are related to each other.

To compare two or more facets, complete the following steps:

1. From the **Facet analysis** pane, select **Connections**.
2. Select the root facet that you want to compare to other facets first.
3. Select up to 4 more facets from the list, and then click **Analyze**.

Pair analysis is done between the first facet and each other facet in turn.

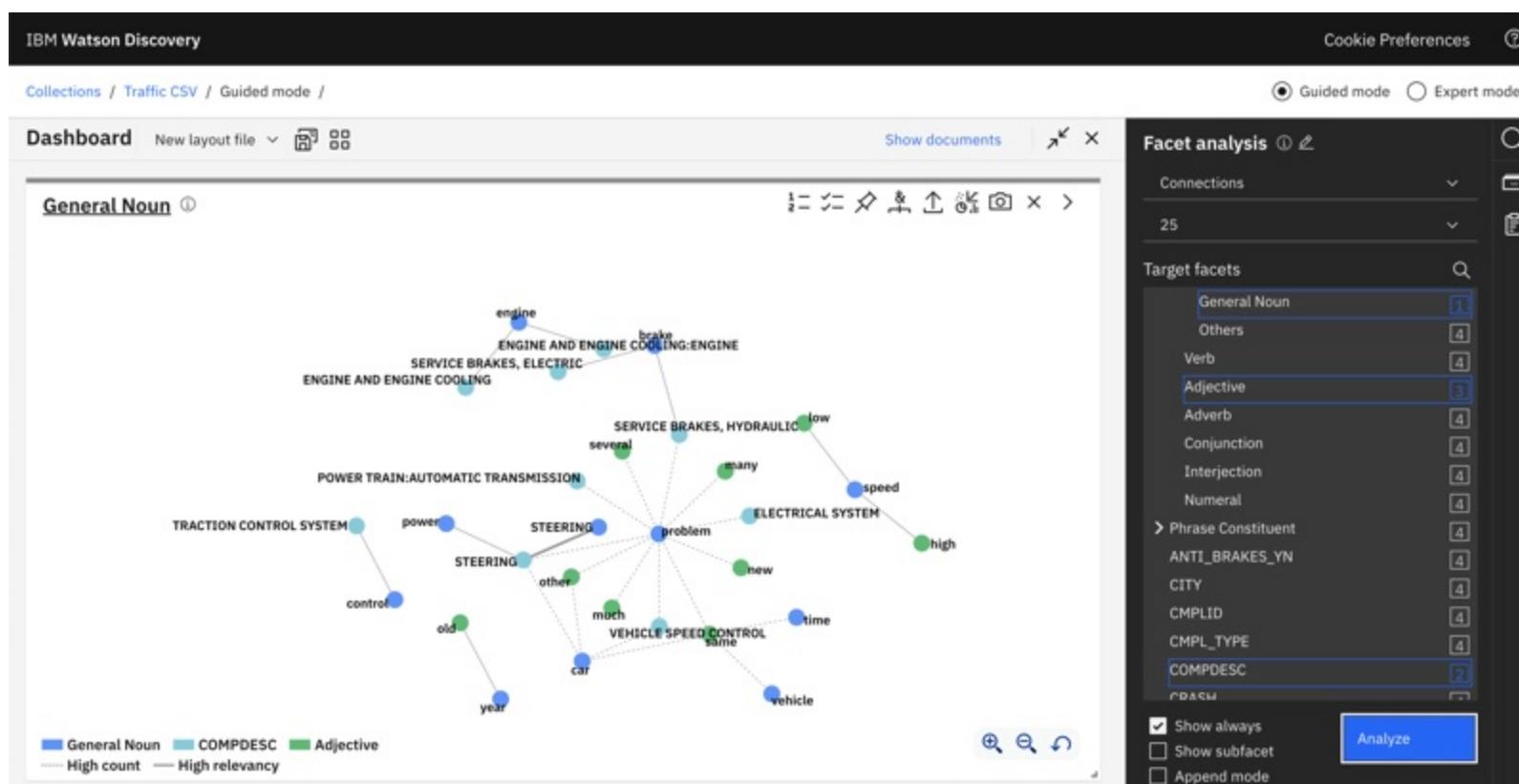


Figure 7. Facet network graph

The resulting network graph shows only highly relevant and high-frequency pairs. Each node represents a facet value. The node color reflects the facet type. A solid-line connection between nodes identifies highly relevant pairs. A dotted-line connection identifies high-frequency pairs.

Changing number ranges

If the scale of a graph is not optimized for your data, you can change it. For example, to plot vehicle speeds, you might want a range that increments by tens or twenties rather than by thousands.

To change the scale of a graph for a facet, complete the following steps:

1. Click **Collections** link in the page header.
2. In the tile for your collection, click the *Open and close list of options* icon, and then choose **Edit collection**.
3. In the **Facet** tab, find the facet for which you want to change the number range.
4. In the Range field, click **Edit**.
5. Define each range that you want to use as a JSON object. You can add or remove objects to change the number of data points in the range.

For example, the JSON objects that identify the ranges for vehicle speeds might look as follows:

```
[  
  {  
    "query": "[1, 20)",  
    "label": "1 - 19"  
  },  
  {  
    "query": "[20, 40)",  
    "label": "20 - 39"  
  },  
  {  
    "query": "[40, 60)",  
    "label": "40 - 59"  
  },  
  {  
    "query": "[60, 80)",  
    "label": "60 - 79"  
  },  
  {  
    "query": "[80, 100000)",  
    "label": "80+"  
  }  
]
```

6. Click **Apply**.
7. Click **Save**, and then click **Close**.
8. Click your collection tile to return to the collection and continue your analysis.

The changes to the number ranges for vehicle speeds introduce more opportunities for relationships or anomalies in the data to be highlighted.

STATE / VEH_SPEED ⓘ					
Column:	VEH_SPEED				
Row:	STATE	1 - 19	20 - 39	40 - 59	60 - 79
AZ ▶	0.70	0.36	0.74	0.54	0.00
IN ▶	25 0.41	34 0.63	34 1.00	11 0.40	0 0.00
CO ▶	31 0.55	35 0.66	16 0.39	17 0.72	0 0.00
WI ▶	33 0.59	38 0.73	18 0.46	16 0.67	1 0.02
CT ▶	35 0.65	49 1.02	9 0.18	11 0.41	0 0.00
MN ▶	55 1.15	38 0.76	14 0.34	10 0.37	0 0.00
TN ▶	27 0.52	29 0.59	21 0.63	14 0.63	5 2.04
SC ▶	20 0.49	19 0.47	16 0.60	10 0.54	0 0.00
AL ▶	7 0.10	15 0.36	20 0.85	10 0.57	0 0.00

Figure 8. Results after changed number range

Showing results in a map visualization

Facets that represent geographical locations can be shown in a map visualization. For example, if you have a collection with a US states facet, you might want to display data per state from a visualization that enables users to select each state from a map.

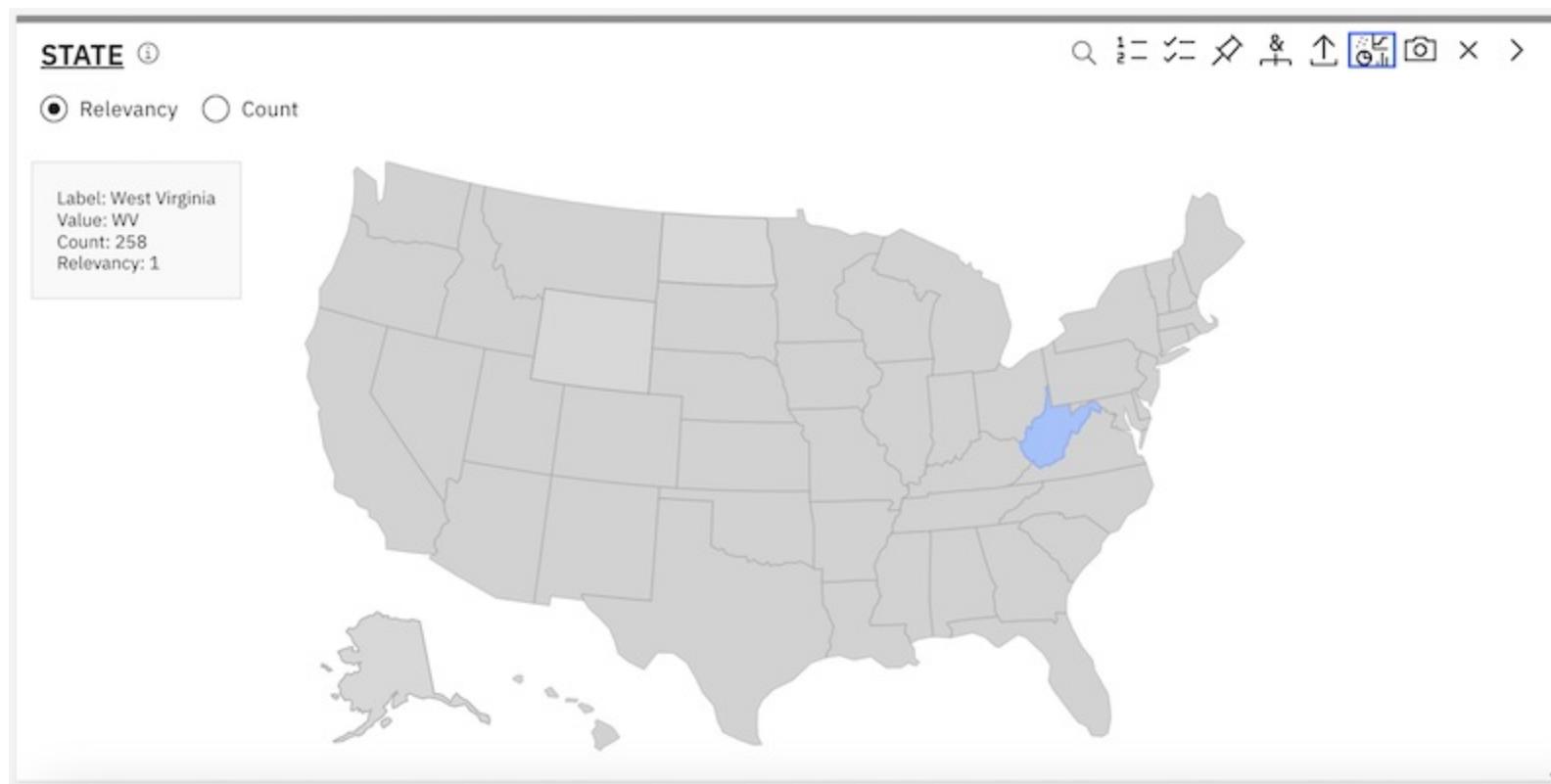


Figure 9. Results shown in a map visualization

A US Map is available by default. You can add a custom map that is built in GeoJSON format. For more information, see [RFC7946](#).

To use a map that you define, complete the following steps to import the map definition:

1. From the Content Mining application home page, click **Collections** from the breadcrumb in the page header.
2. Click the **Settings** icon at the start of the page.
3. Click **Manage customization resources**.
4. Click **Add resource**.
5. Name the resource, and then click **Next**.
6. Add your map file, and then click **Save**.

To make the map that you added available as a visualization option for a facet, you must edit the facet.

1. Click **Home** from the breadcrumb in the page header.
2. Right-click the overflow menu for your collection, and then choose **Edit Collection**
3. Open the **Facet** tab, and then find the facet with which you want to associate the map visualization.
4. Change the **Visualization type** value to **Map**, and then pick the map that you added from the list in the **Resource** field.
5. Click **Save**, and then click **Close**.

Flag documents of interest

Use document flags to assign a custom flag to a document or a group of documents for classification, export, or further analysis.

Flagging documents is a useful way to highlight documents that you want to examine further later.

Before you can flag documents, you must create flags for your collection. For more information, see [Add document flags](#).

To apply flags, complete the following steps:

1. From the analysis view of your collection, create a query that returns a set of documents with specific characteristics.
2. From the documents view, click the **Document flags** icon.
3. Select a flag.
4. You can choose to apply the flag to all query results or to selected documents, and then click **Apply**.



Note: You can't set a document flag more than 50 times per collection. Whether you flag one document that you select individually or flag a query, which might return many documents, each action counts as setting a flag one time.

A flagged document set dynamically changes as the collection is updated. Flagged document sets are stored as queries in the index. Each flag has a query that represents the document set that it is associated with. For example, after you create the document flag and you search for the term **ice cream** and apply a red flag to all of the documents that have this word, **ice cream** is stored as the query that represents the flag. Then, if you search for the term **coffee** and apply the red flag to all of the documents that have that word, the internal flag query changes to **(ice cream) OR coffee**. Therefore, if new documents that contain the word **coffee** are ingested, the red flag is applied to those documents automatically.

Viewing flagged documents

To view the documents to which a flag is applied, complete the following steps:

1. In the **Facet analysis** panel, scroll down to the **Document flags** facet.
2. Select the facet, and then click **Analyze** to open the **Document flags** dashboard.
3. Click one of the flags, click **Analyze more**, and then click **Show documents**.

Removing document flags from a Document Flags query

To remove a document flag, complete the following steps:

1. From the **What do you want to analyze?** page, submit an empty query by clicking **Search**.
The empty query returns all of the documents in your collection.
2. Click **Show documents**.
3. Click the **Document flags** icon on the toolbar, clear the checkbox of the document flag, and then click **Apply**.
The document flags are removed from your documents.

Adding facets

Add more facets that you can use to filter your data.

When you apply custom enrichments to your collection, annotations are added to its documents. The annotations feed into new facets that you can use to sort your data.

The following table describes the types of facets that you can create from annotations.

Information to recognize	Annotator type
Commonly understood terms, such as organization or people names.	Built-in Natural Language Processing models

Phrases that express an opinion and evaluate whether the opinion is positive or negative.

[Phrase sentiment](#)

Alternative words that share a meaning with terms in a finite list.

[Dictionary](#)

Terms that match a syntactical pattern

[Regular expression](#)

Custom terms by the context in which they are used.

[Machine learning model](#)

Documents that fit into categories that you define.

[Document classifier](#)

Custom facet types

Grouping facets

To organize your facets, you can group them in folders.

Grouping facets does not combine the data from the facets. It merely makes the facets easier to find because they are organized in named folders.

To associate facets such that you can combine data from multiple facets, the facets must have a facet and subfacet relationship. Such hierarchical relationships must be defined at the time that the facet enrichment or annotation is created and applied to the collection.

To group facets, complete the following steps:

1. From the initial search page, submit a search.
2. From the **Facet analysis** pane, click the **Edit** icon.
3. Name the group, and then select the facets that you want to group together.

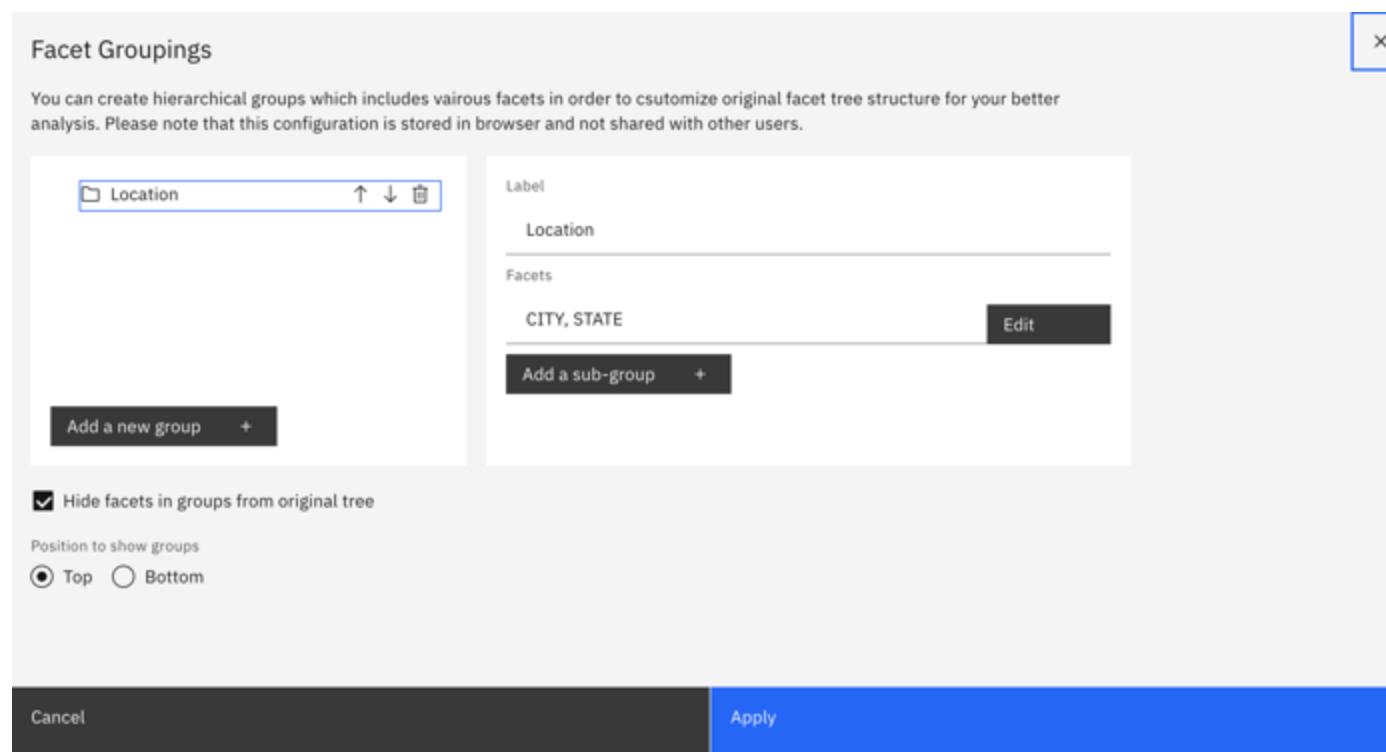


Figure 1. Facet grouping dialog

4. Click **Apply**.
5. The facets that you grouped are now available from a folder with the group name.

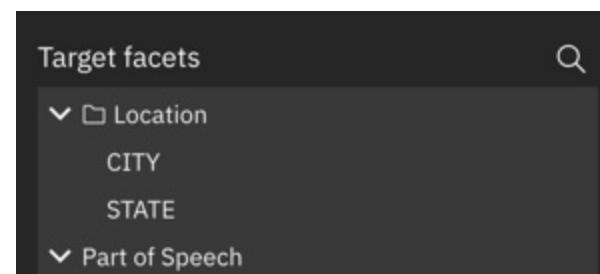


Figure 2. Facet folder from the facet list

Creating custom annotators

You can create a dictionary, regular expression, or machine learning annotator to generate new facets that can help you to analyze your data.

Before you begin, have the following data ready.

Annotator type	Description	Data
Dictionary	Assigns facets to terms that match dictionary entries that you define or upload.	You can optionally upload a file of dictionary terms.
Machine learning	Assigns facets to mentions that are recognized by a machine learning model that you upload.	A compressed file of a machine learning model is required.
Regular expression	Assigns facets to text that matches Java regular expression patterns that you define or upload.	You can optionally upload a JSON file that contains regular expression patterns.

Custom annotator prerequisite data

To create a custom annotator, complete the following steps:

- From the analysis view of your collection, click the **Collections** link in the breadcrumb to open the *Create a collection for your analytics solutions* page of the Content Mining application.
- To create an annotator, click **collection**, and then select **custom annotator** from the list.

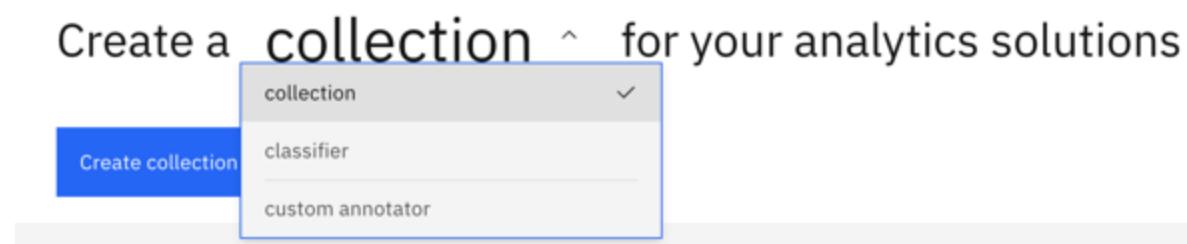


Figure 1. Collection menu

- Click **Create custom annotator**.
- Name your annotator, and then optionally add a description.
- Choose the annotator type, and then click **Next**.
- Follow the on-screen instructions.

For more information about how to configure each annotator type, see one of the following sections:

- [Dictionary](#)
- [Machine learning model](#)
- [Regular expressions](#)

Dictionary configuration

You can import an existing dictionary by uploading it or you can create a dictionary by adding terms one at a time.

If you plan to import a dictionary, the dictionary terms must be defined in a CSV file. Specify each term and its synonyms in a separate line. Use the following syntax to specify each term:

```
{term},{synonym},{synonym},...
```

To add a dictionary, complete the following steps:

- Do one of the following things:
 - To import the dictionary terms:
 - Click **Import**, and then browse for the file with your dictionary terms.
 - Click **Import**.
 - To define the dictionary terms:
 - Click **Add**.
 - Click **Word list** to add the dictionary terms.
 - Click **Add**, and then add the term in the **Base word** field and any synonyms that you want to define for the term in the **Other words** field. Separate multiple synonyms with commas. Click **OK**.
 - Repeat the previous step to add more dictionary terms.
 - After you finish adding dictionary terms, click **Basic settings**.
- Name the dictionary.

3. If you plan to define terms with a part of speech other than a noun, specify the part of speech.

4. Decide how you want to handle case.

When case is ignored, the terms **Sat**, **SAT**, and **sat** are all labeled as occurrences of the **Sat** dictionary term.

When you deselect the **Ignore case** checkbox to create a case-sensitive dictionary, the surface form of the term with uppercase match is used. Annotations are added for the term exactly as written and for variations of the term in which the letters are uppercase.

For example, a **sat** entry in the dictionary results in annotations for **sat**, **Sat**, or **SAT** mentions when they occur in text. For a **Sat** entry in the dictionary, annotations are added for occurrences of **Sat** and **SAT**, but not for **sat**.

5. Identify the facet name to use for this dictionary.

The facet name that you specify for the annotator is the facet name that is displayed from the collection search view.

You can create a hierarchical facet by including a period (.) in the facet name. For example, you might create one dictionary with the facet path **Food.Vegetables** and others with the facet paths **Food.Fruits** and **Food.Proteins**. Add more facet groups with more periods. For example, you can add **Food.Proteins.Nuts** and **Food.Proteins.Meats** to categorize proteins even further.

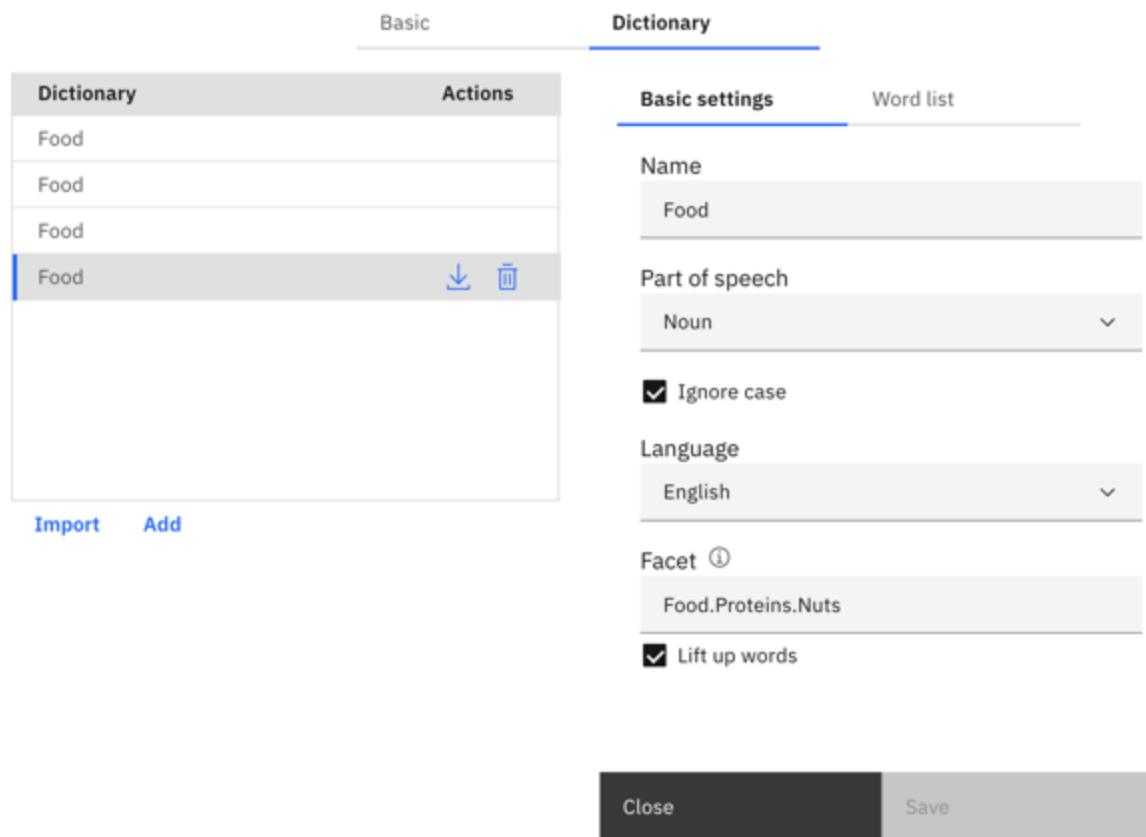


Figure 2. Adding a dictionary

6. If you want documents that are returned for a subfacet to be included when a user filters on the root facet, select **Lift up words**.

For example, you might enable **Lift up words** for **Food.Fruits** and **Food.Proteins** but not **Food.Vegetables**. As a result, when a user clicks the Food facet, the returned documents include documents that mention terms included in the Fruits and Meats dictionaries, such as *apples* and *beef*.

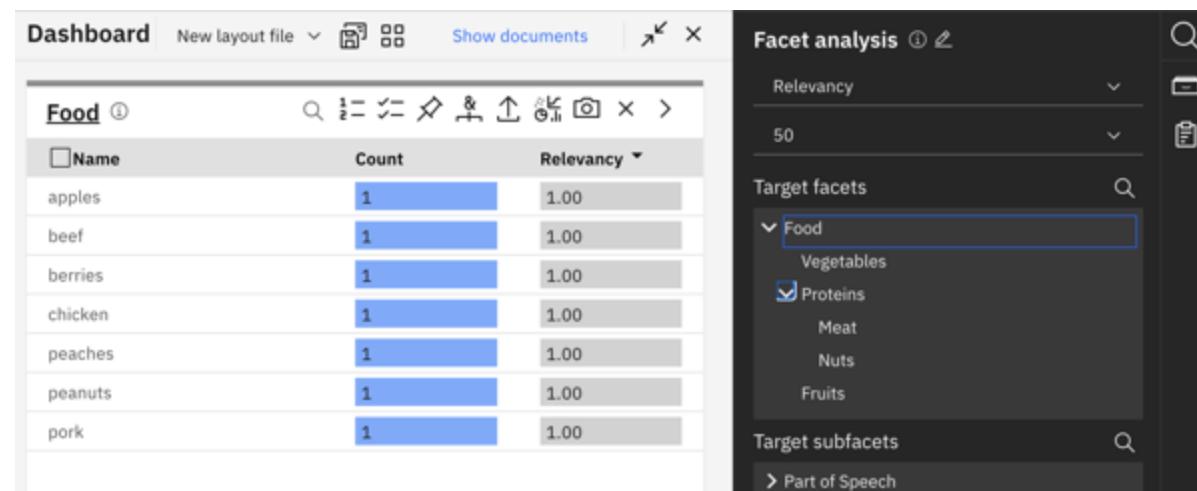


Figure 3. Dictionary enrichment application

However, a user must click the **Food>Vegetables** facet explicitly to get documents that mention terms in the Vegetables dictionary, such as *lettuce*, to be returned.

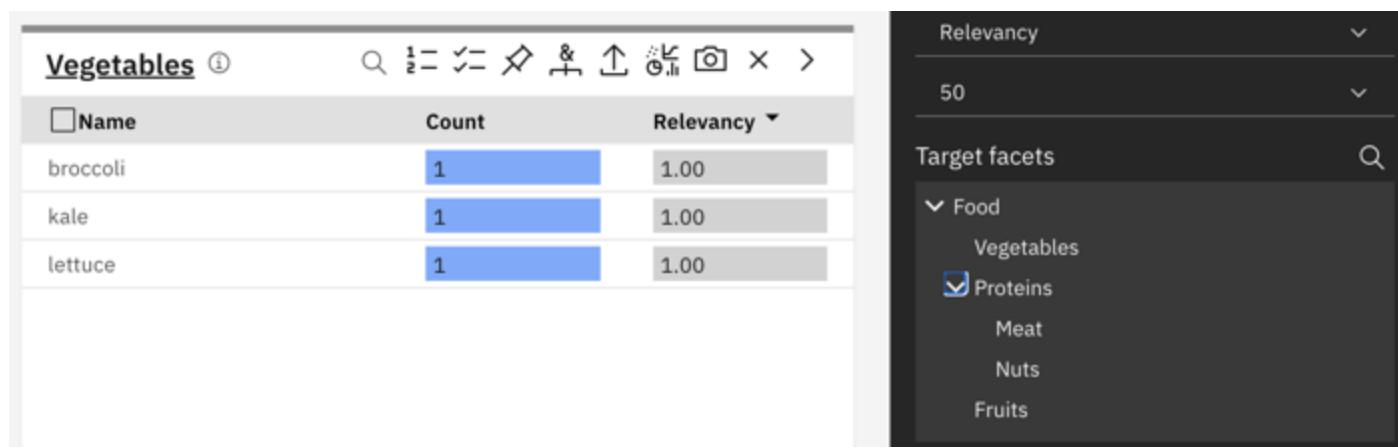


Figure 4. Subfacets

7. Repeat previous steps to add more dictionaries.

8. Click **Save**.

From the custom annotator page, you can see dictionaries that were created in other projects, including non-Content Mining projects. Dictionaries from other project types show the enrichment name as the annotator name. The *Ignore case* and *Lift up words* settings are disabled and the dictionary is named **custom dict**.

Dictionary limits

Plan	Number of dictionaries per service instance	Number of base words per dictionary	Number of terms for which suggestions can be generated
Cloud Pak for Data	Unlimited	Unlimited	1,000
Premium	100	10,000	1,000
Enterprise	100	10,000	1,000
Dictionary plan limits			

Totals include enrichments that you create in this Content Mining project and in other projects in the same service instance.

Machine learning configuration

You can import an existing machine learning model.

To use Discovery to create a model, see [Entity extractor](#).

To import a model, complete the following steps:

1. Click **Select file**, and then browse for the machine learning model file.
2. In the **Facet path** field, specify the root facet name to use for the model.

The facet name that you specify for the annotator is the facet name that is displayed from the collection search view.

3. Click **Save**.

Machine learning model limits

Plan	ML models per service instance
Cloud Pak for Data	Unlimited
Premium	10
Enterprise	10
ML model plan limits	

Totals include enrichments that you create in this Content Mining project and in other projects in the same service instance.

Regular expressions configuration

You can import existing patterns by uploading them in a JSON file or you can add patterns.

To add patterns, complete the following steps:

1. Add the regular expression pattern to the **New pattern** field, and then click **Add**.
2. Specify a name for the pattern, and then identify the facet name to use for this pattern.

The facet name that you specify for the annotator is the facet name that is displayed from the collection search view.

3. **Optional:** Specify a facet value. You can specify a value from the options that are described in the table.

Facet value	Description
\$0	Displays the matched text as-is.
\$n	If your regular expression pattern contains groups, you can specify a group number to return the matched text from the pattern group only. For example, if your regular expression consists of 3 groups that define a US phone number pattern, such as <code>(\d{3})-(\d{3})-(\d{4})</code> , and you want to return only the area code portion of the phone number, you can specify <code>\$1</code> . If the matched text is 212-555-1234 , then the facet value is displayed as 212 . Only specify a group as the facet value for patterns that you know will return matches.
{prefix-text}:\$0	Adds hardcoded text in front of the facet name. You might want to use this option if you want to distinguish facets that are generated by this regular expression from facets that are similar but generated in some other way. For example, <code>MyRegex:\$0</code> results in a facet named <code>MyRegex:212-555-1234</code> .

Regular expression facet value options

4. Click **Save**.

To import patterns, complete the following steps:

1. Define the patterns that you want to add in a JSON file.

The pattern definition must use the following syntax:

```
[  
  {  
    "name": "US Phone number",  
    "description": "US mobile phone number",  
    "pattern": "(\\d{3})-(\\d{3})-(\\d{4})",  
    "facetPath": ".regex.usphonenumber",  
    "facetValue": "$0"  
  }  
]
```

Keep the following notes in mind:

- The patterns must be defined in an array, even if you plan to define only one pattern.
- Escape any backslash (\) characters with a backslash.
- For more information about the facet value options, see the *Regular expression facet value options* table.

2. Click **Import**, and then choose the JSON file where the patterns are defined.

3. Click **Save**.

Regular expression limits

Plan	Regex enrichments per service instance	Regex patterns per service instance
Cloud Pak for Data	Unlimited	Unlimited
Premium	100	50
Enterprise	100	50

Regular expression plan limits

Totals include enrichments that you create in this Content Mining project and in other projects in the same service instance.

Applying the annotator

After the annotator is created, you must apply it to your collection.

1. From the **Create a custom annotator for your analytics solutions** page of the Content Mining application, click **custom annotator**, and then select **collection** from the list.
2. In the tile for your collection, click the **options** icon, and choose **Edit collection**.
3. Click the **Enrichment** tab, and then select the annotator that you created.
You might need to scroll to find it.
4. Click **Save**, and then confirm the action.

Give the index time to rebuild.

Filtering documents with your facet

1. Click the collection tile to open your collection in the data analysis page.
2. Do one of the following things:
 - Your custom facets are listed in the **Facets** view. Scroll and click **Load more** repeatedly until your facets are displayed.
 - Submit an empty search to return all documents. In the **Facet analysis** pane, select the facet that you created.
 - To access your custom facets more quickly, add them to a custom view. Select **Custom** as the view, and then click **Edit**. Select one or more facets to add to the view, and then click **Save**.

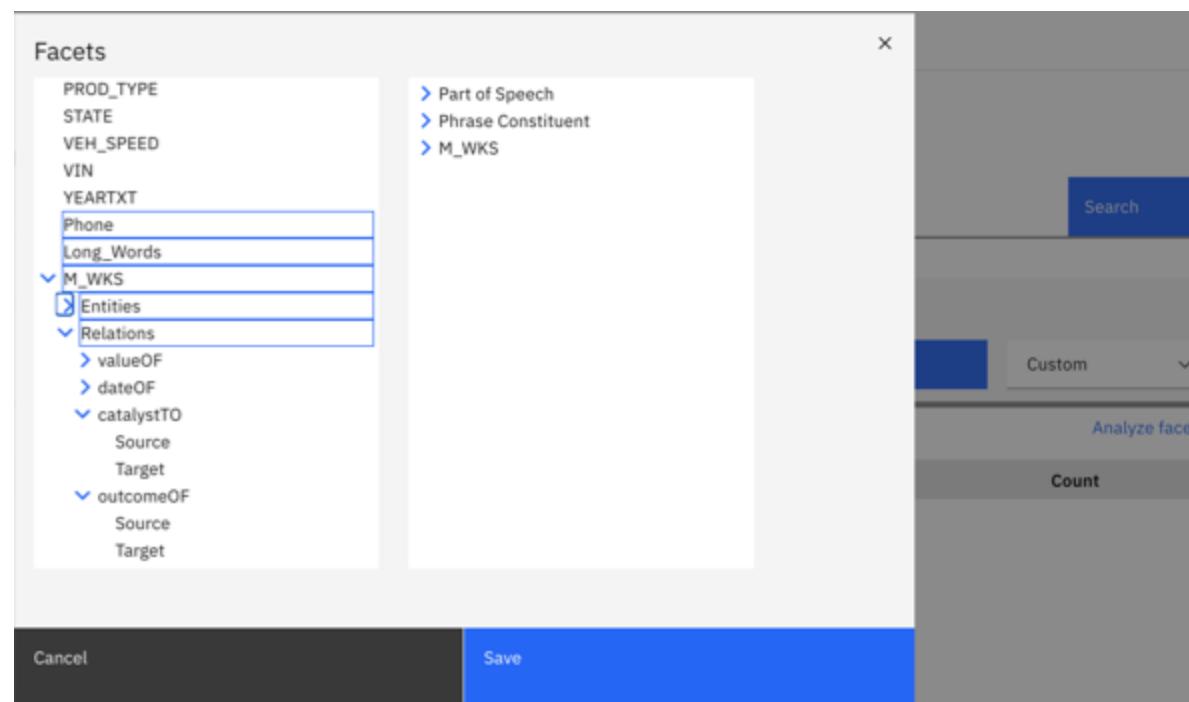


Figure 1. Collection menu

Classifying documents

A document classifier machine learning model analyzes documents and tags them with the appropriate label from a set of labels that you define.

Classifying documents is useful when you want to sort many documents into groups programmatically. For example, you might have a collection that contains customer comments about products that you sell. If you can automatically sort the feedback into classes, you can isolate urgent issues that customers mention and tackle them first. Based on previous feedback, you might define classes such as the following labels:

- Not functioning correctly
- Features not as advertised
- Difficult to use
- Missing parts
- Parts shipped don't match parts list in assembly instructions

To create a document classifier, you build a machine learning model that can recognize which class best captures the point of customer feedback that is specified in natural language. You pair them with class labels that represent real scenarios that make sense for your business.

What's the difference between a document classifier and a text classifier?

A document classifier can classify documents based on words and phrases extracted from the body text fields with information from their part of speech and the other enrichments that are applied to the body text taken into account. The information from the other non-body fields are also used. A text classifier can classify documents based on words and phrases extracted from the

body text with their part of speech information taken into account. For more information about how to create a text classifier, see [Classifier](#).

Before you begin

To train the document classifier model, you must provide sample documents that are labeled appropriately. Prepare the following files:

Training data

Required. CSV file that is used to train the document classifier machine learning model. The file can contain key data points per column. The data points can vary, but the file must include the following columns:

- Natural language text that you want to classify or label.
- Label or class name that categorizes the idea that is expressed in the document text. You can apply more than one label to a text sample. Separate multiple label values with a semicolon.

Test data

Optional. CSV file that is used to test the document classifier machine learning model after it is trained. If you don't specify a separate file for testing, a subset of the training data content is used for testing purposes.

Target data

Required. CSV file with the data that you want to classify.



Important: All of the CSV files (training, test, and target) must have the same column names. The data in the columns must have the same data types, such as string, number, and so on.

You can use a CSV file that you uploaded at the time that you created the Content Mining project or you can create a new collection.

For more information, see the following topics:

- [Adding collections](#)
- [Analyzing CSV files](#)

Document classifier training data sample

The following table shows an example of the type of content that might be stored in CSV files that are used to train a document classifier.

Claim_id	Date	Product_line	Product	Client_segments	Client_location	Client age	Feedback	Label
0	2016/1/1	tea	lemon tea	Not Member	Manhattan	20	The straw was peeled off from the juice pack.	package_container
1	2016/1/2	ice cream	vanilla ice cream	Silver Member	Queens	20	I got some ice cream for my children, but there was something like a piece of thread inside the cup.	contamination_tampering

Table 1. Sample data for CSV files

Note that the two required fields are present in the sample. The required fields have the following names:

- **Feedback**: Natural language text to label.
- **Label**: Label to apply to the feedback.

Opening the Content Mining application

If you didn't do so, create the project and add a collection to it. If you already created the project and collection, you can skip this procedure and [create the document classifier](#).

1. In Discovery, create a Content Mining project.
2. Choose to upload data to create the collection. Name your collection, and click **Next**.
3. Upload the CSV file that contains your training data.

The training data file must contain the following information at a minimum:

- A column that contains sample text that you want to classify. For example, the sample text might be a product review.
- A column that contains a class or category label that is assigned to the sample text.

4. After collection processing is complete, click **Launch application** to open the Content Mining application.

The facet details are displayed for the collection.

Creating a document classifier

To create a document classifier, complete the following steps:

1. From the Content Mining application, click the **Collections** link in the breadcrumb to open the *Create a collection* page.
The status of index creation is displayed. Wait for the collection to be fully indexed before you continue with this procedure.
2. To create a classifier, click **collection**, and then choose **classifier** from the list.

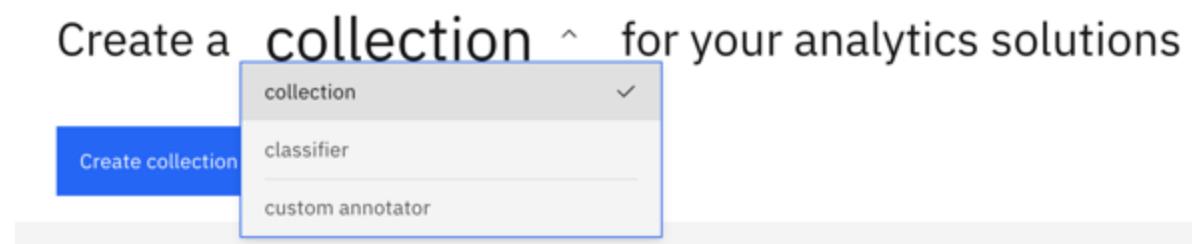


Figure 1. Collection menu

3. Click **Create classifier**.
4. Name your classifier.

When you deploy the model as an enrichment later, the enrichment is given a name with the format `{classifier name} - {model name}`. For example, if your classifier is named `Product reviews` and the model is named `v0.1`, then the enrichment name is `Product reviews - v0.1`.

Optionally, add a description and identify the language of your training data by selecting it from the **Language** field.

5. Click **Next**.
6. On the **Training data** page, select the file that you uploaded previously from the list, and then click **Next**.

Alternatively, you can upload a CSV file that contains your training data.

The **Fields** page is displayed. It shows details about the fields that are generated from the file that you added. Typically, each column in a CSV file is converted into a field and is assigned a name that is copied from the column header.

7. Deselect any metadata fields that you want to exclude from the data set for your document classifier to learn from, and then click **Next**.

Any fields that you include are used as additional features in the classification. All of the fields are selected by default. You might need to scroll horizontally to review all of the fields.

8. On the **Classifier** page, specify the fields to use for machine learning training and prediction.

Answer field

Select the field from your training data file with the classification label. From the earlier example, the **Label** field is the best

choice.

Predicted field

The name of the facet that is generated for the predicted class values. By default, the facet name has the syntax **<Answer field value>_predicted**. For example, **Label_predicted**.

Test dataset

Specifies the data set to use to test the classifier model. By default, the training data CSV file that you uploaded and configured is split into three data sets that are used for training, validation, and test respectively. However, you can optionally specify a separate data set to use for testing the model.

Train federated model

Creates more than one model, based on values from a specific field in the data set. For example, if the document has a **Product** field, you can configure the classifier to create a separate classifier model for each product name value that is specified in the field. By default, the classifier creates one machine learning classifier model.



Note: You don't need to specify the field that contains the text to be classified. The system detects this field automatically. You can check which field the analyzable text is extracted from and change it or augment it by changing index type of another field. For more information, see [Identifying the text field](#).

Click **Next**.

9. If you want to apply an enrichment to the text in your training data, select at least one field from the **Target fields** list where you want to apply enrichments.

Typically, you want to choose the field that contains the body of text that you want to classify. From the earlier example, the **Feedback** field is the best choice.

Next, select any annotators that you want to apply to enrich the text in the target field or fields, and then click **Next**.

The **Part of speech** annotator is selected by default.

10. On the **Confirm** page, review your classifier configuration settings. To make changes, use the **Back** button. Otherwise, click **Save**.

An **Overview** page is displayed.

11. Click **New model** to create and train your machine learning model.

12. You can optionally change the name of the model and add a description.

You can change the default ratio values that are specified for the following data sets:

- Training dataset: Updates the weights of the training model.
- Validation set: Monitors the accuracy of the training model during training. The accuracy result is used to draw a training loss graph.
- Test dataset: Calculates the score of the trained model.

13. Click **Create**.

It might take several minutes for model training to complete.

Deploying the document classifier model

After the model is trained, deploy the model as an enrichment.

1. Click the overflow menu icon in the **Actions** column, and then click **Deploy model**. Specify the name and other details, and then click **Deploy**.
2. Do one of the following things:
 - To apply the document classifier to a collection in your Content Mining project, see [Enriching your collection](#).
 - To apply the document classifier to a collection in a different project, complete the following steps:
 1. In Discovery, create or open the collection that has the documents that you want to classify.



Note: The data in the collection where you apply the enrichment must have the same fields as the collection that you used to train the model.

- In the **Enrichments** tab, locate your classifier in the **Name** column. From the **Fields to enrich** field, choose the same text field that was used to train the model. (This field is determined by the system and is indexed as the **Analyzable text content** field. For more information, see [Identifying the text field](#).)
- Click **Apply changes and reprocess**.

Results of classification

After the enrichment is applied to a collection, a facet is generated that you can use to find the predicted classes. In this example, the predicted field is named `label_answer_predicted`.

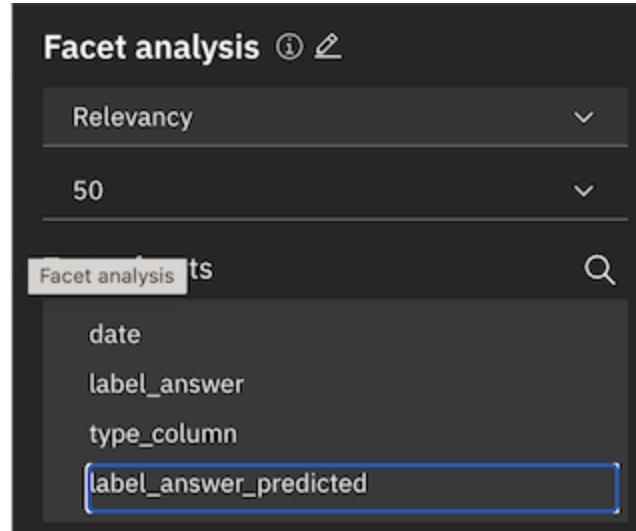


Figure 2. A Label_answer_predicted facet is generated

Use the generated facet to filter documents by classification and analyze subsets of documents. Doing so helps you to find patterns and discover other insights. You can export these target documents to share with team members or to analyze further. For more information, see [Exporting data](#).

When the document classifier classifies a document, it stores the classification in the `document_level_enrichment.classes.class_name` field.

For example, the following JSON excerpt shows a document that was classified with the `package_container` class.

```

▼ "body" : [ 1 item
  0 : "I found a dark clump in the bottle."
]

"client_age" : 30
"client_location" : "Manhattan"
▼ "document_level_enrichment" : { 1 item
  ▼ "classes" : [ 1 item
    ▼ 0 : { 3 items
      "confidence" : 0.9924432635307312
      "classifier_name" : "Product-review_v0.1"
      "class_name" : "package_container"
    }
  ]
}

```

Figure 2. Document classifier enrichment syntax

Document classifier limits

The number of document classifiers and labels that you can create per service instance depends on your Discovery plan type.

Limit	Enterprise	Premium	Cloud Pak for Data
-------	------------	---------	--------------------

Number of document classifiers per service instance	20	20	Unlimited
Number of labeled data rows	20,000	20,000	20,000
Maximum size in MB of training data after enrichment	1,024	1,024	1,024
Number of labels	1,000	1,000	1,000
Number of target fields	50	50	50

Document classifier plan limits

Detecting phrases that express sentiment

Analyze a document to find phrases that express an opinion or reaction and assess whether the sentiment expressed is positive, neutral, or negative. For English and Japanese, you can also detect specific sentiment targets. The Content Mining application marks these extractions as annotations.

For example, if a product feedback form contains the following sentence, you want to find it and indicate that it is a **positive** statement.

I love my XYZ blender...

What's the difference between phrase and document sentiment?

Document sentiment is a built-in Natural Language Processing enrichment that is available for all project types. Document sentiment evaluates the overall sentiment that is expressed in a document to determine whether it is positive, neutral, or negative. Phrase sentiment does the same. However, phrase sentiment can detect and assess multiple opinions in a single document and, in English and Japanese documents, can find specific phrases. For more information about the document sentiment enrichment, see [Sentiment](#).

Complete the following steps to enable phrase sentiment analysis:

- From the analysis view of your collection, click the **Collections** breadcrumb link in the page header.
- In the tile for your collection, click the *open and close list of options* icon, and then choose **Edit collection**.
- Click the **Enrichment** tab, and then select the **Sentiment of phrases** annotator.
- Click **Save**, and then click **OK** to verify the change.

The collection is reindexed. Wait for processing to be completed.

- Click **Close** to return to the **Collections** page, and then click your collection tile.
- In the **What do you want to analyze?** field, enter a term to search for in your documents or select one or more facets, and then click **Search** to filter the documents.

The search results are displayed in the mining graph. The **Facet analysis** pane is displayed also. By default, **Relevancy** analysis is shown.

- In the drop-down menu from the **Facet analysis** pane, select **Sentiment**.
- In **Target facets** from the **Facet analysis** pane, expand the **Sentiment Analysis** option to see facets that are available for analysis in your documents.
- Click a facet to explore.

For example, if you click **Positive Expression**, you can see the following information:

- Positive expressions that were identified in your documents
- Sentiment percentage
- Side-by-side comparison of positive and negative expressions
- Number of instances of the expression
- Expression relevancy

- Click one or more options in the facet list, or select one or both facet lists, and then click **Analyze more**.

View the phrase, expression, or target in the **Documents** or **Trends** views.

 **Note:** Text from the body field of the document is analyzed. For more information about which field is used for the body text, see [Identifying the text field](#).

Adding collections

You can add a collection directly to the Content Mining application.

You might want to add a collection from within the Content Mining application to make data available for use as training data for a document classifier, for example.

The collection can contain an uploaded CSV file only. For information about file guidelines, see [Analyzing CSV files](#).

The collection that you create is not added to your existing Content Mining project. A new Content Mining project is created to store the collection. The project that is generated is given the name that you specify for the collection.

To add a collection, complete the following steps:

1. From the analysis view of your collection, click the **Collections** link in the breadcrumb to open the *Create a collection for your analytics solutions* page of the Content Mining application.
2. Click **Create collection**.
3. Drag your CSV file to the **Import your files** dialog, or click Open to browse for the file. When the button is available, click **Next**.
4. You can optionally customize the columns that you want to include or exclude from the collection, and adjust the data types of the fields from the **Fields** page. Click **Next**.
5. From the **Enrichments** page, you can optionally apply or remove any enrichments from the collection, and then click **Next**.

The **Part of Speech** enrichment is applied automatically.

6. On the **Facets** page, you can optionally customize the data that is displayed for facets. Click **Next**.
7. Click **Save** to save and index the collection.

Editing your collection

You can change the characteristics of your collection from the Content Mining application.

You can change the following characteristics:

- [Change the time zone of your collection](#)
- [Add document flags that you can use to tag documents of interest in your collection](#)
- [Change or augment the field that is designated as the source for the text body of your documents](#)
- [Group text body fields](#)
- [Add, remove, or change the enrichments that are applied to the collection](#)

Edit a collection

1. From the analysis view of your collection, click the **Collections** link in the page header.
2. In the tile for your collection, click the **Open and close list of options** icon, and then choose **Edit collection**.
3. Use the appropriate tab to change characteristics of the collection.
4. When you are done making changes, click **Save**.

The following message is displayed:

You need to clear index to make these changes.
After clearing index, fully build the index to
analyze using this collection.

You can ignore the message. The index is rebuilt automatically when you click **OK**.

5. Click **OK** to verify the change.
6. Click **Close** to return to the **Collections** page.

 **Tip:** Wait for the index to be rebuilt before you continue your analysis. From the **Collections** page, you can see the progress

of the index rebuild.

7. Click your collection tile to return to the data analysis page.

Change the time zone

To change the time zone that is used by the trend graph, you must edit the default time zone for the collection.

1. Complete the steps in [Editing a collection](#) to get the collection into edit mode.
2. In the **Edit** tab, change the value of the **Time zone** field, and then click **Save**.

Add document flags

To add document flags, complete the following steps:

1. Complete the steps in [Editing a collection](#) to get the collection into edit mode.
2. Click the **Document flags** tab, and then click **Add flag**.
3. In the **Document flag** dialog box, name the flag, add a description, choose a flag color, and then click **Add**.
4. Repeat the previous steps to add more flags.
5. From the **Document flags** view, select **Enabled** so that the flags appear in your documents, and then click **Save** to make them available in your collection.

For more information about how to flag documents, see [Flag documents of interest](#).

Identify the text field

When you analyze data with the Content Mining application, Discovery determines which field contains the **body** of the text to be analyzed. It does so by looking for the field with the highest average word count.

You can check which field is designated as the main text body field, and change it or augment it by changing the index type of another field.

1. Complete the steps in [Editing a collection](#) to get the collection into edit mode.
2. Click the **Fields** tab. Check the **Index type** column to find the field designated with the **Analyzable text content** index type.

You can change the field or set more than one text field to be an **Analyzable text content** index type.

3. Click **Save**.

If you select multiple fields to analyze, you cannot see the facet analysis for only one field. To view the analysis for multiple fields, you must group them.

Group multiple text fields

1. Complete the steps in [Editing a collection](#) to get the collection into edit mode.
2. Click the **Contextual view** tab, and then click **Add view**.
3. Complete the following fields:
 - **Name**: The name or label of your grouped view.
 - **Id**: The alphanumeric ID that Discovery uses when you submit a text query. For example, **ans1**.
 - **Fields**: The text fields that have the **Analyzable text content** setting applied. Select one or multiple text fields that you want to group for facet analysis.
4. Click **Add**.

Repeat this task if you want to add more text fields that you want to group for facet analysis.

5. Click **Save**.

Now you can return to the data analysis page for your collection. From the **Facet analysis** panel, you can click **Contextual view selection** to see the text fields that you grouped. You can select one of the text fields to view the facet analysis for that field.

Enrich your collection

Discovery provides built-in natural language processing models, such as the **Entities** enrichment that can recognize mentions of commonly known things, such as business or location names and other types of proper nouns. You can apply these built-in NLP enrichments to your collection.

You can also apply a document classifier enrichment that you created in the Content Mining application to your collection.

Alternatively, you can apply enrichments that were built in other projects in the same service instance to the collection in your content mining project. For example, you can apply a dictionary or text classifier that was built in another project in the same service instance to your collection.

To apply enrichments to your collection, complete the following steps:

1. Complete the steps in [Editing a collection](#) to get the collection into edit mode.
2. Click the **Enrichment** tab, and then select the enrichments that you want to apply to your collection.
3. Click **Save**.

Analyzing CSV files

You can add the data that you want to analyze as a comma-separated value (CSV) formatted file.

The content mining project works well with CSV files. When your CSV file is ingested, each row in the spreadsheet is stored as a separate document in the collection index. Each column becomes a root-level field in the document.

Follow these guidelines when you create a CSV file for use in the project:

- Add each record that you want to analyze as a row in the spreadsheet.
- Include a column for each significant data point.
- Specify column headers.

The root-level field that is added to the document is given the column header name. If no header exists, hardcoded names, such as **column_0** and **column_1**, are applied to the columns. Specify column names to ensure that the resulting document fields have meaningful names.

- If you want to find trends over time, be sure that each record has some date information that can be used to plot the information on a timeline.

Discovery recognizes the following date formats automatically:

```
yyyy-MM-dd'T'HH:mm:ssZ  
yyyy-MM-dd'T'HH:mm:ssXXX  
yyyy-MM-dd'T'HH:mm:ss.SSSZ  
yyyy-MM-dd'T'HH:mm:ss.SSSX  
yyyy-MM-dd  
M/d/yy  
yyyyMMdd  
yyyy/MM/dd
```

If you store dates in other formats, you can add the format to the list of supported formats.

From the Discovery user interface, open the **Manage collection** page. Click your collection tile. From the **Manage fields** page for the collection, add a format to the **Date formats** field. Specify a date format that is supported by the Java [SimpleDateFormat](#) class.

For example, if your records store only year values for dates, add **yyyy** to the supported date formats list. You can then set the data type for the field that contains a year value to **Date**, and reprocess your collection. As a result, an occurrence of **2019** in the date field is stored as **2019-01-01T05:00:00Z** in the index.

Sample CSV file

The following image shows an excerpt from a CSV file with data that is well suited for analysis with the Content Mining application. The data comes from 2010 traffic records that are published by the National Highway Traffic Safety Administration (NHTSA). Each record includes car make, model, and year information, the date of the traffic incident, and text from the driver's statement, along with other useful data points.

MAKETXT	MODELTXT	YEARTXT	CRASH	FAILDATE	FIRE	COMPODESC	CITY	STATE	DATEA	LDATE	MILES	CDESCR	
2	TOYOTA	SIENNA	2010	N	20100101	N	ENGINE AND ENGINE COOLING	AVON	IN	20100101	20100101	3500	ENGINE SPEED CONTROL , IT DOESN'T GO UP SMOOTHLY FROM
3	FORD	EXPLORER	2002	N	20100101	N	LATCHES/LOCKS/LINKAGES	LOUISVILLE	KY	20100101	20100101	176000	2002 FORD EXPLORER DOOR LOCKS WILL NOT FUNCTION PROPI
4	FORD	FREESTAR	2005	N	20100101	N	POWER TRAIN/AUTOMATIC TRANSMISSION	NILES	MI	20100101	20100101	55000	WE WERE IN MY WIFE'S 2005 FORD FREESTAR DRIVING HOME FR
5	MERCDES BENZ	E430	2000	N	20100101	N	AIR BAGS:FRONTAL	VIRGINIA BEACH	VA	20100101	20100101	ON E-CLASS MERCEDES, PASSENGER SEAT HAS FUNCTION TO C	
6	CHEVROLET	IMPALA	2007	N	20100101	N	POWER TRAIN/AUTOMATIC TRANSMISSION	TEMPE	AZ	20100101	20100101	40000	TRANSMISSION 'SLIPS' THEN ENGAGES HARD. HAS PROGRESS
7	JEEP	LIBERTY	2002	N	20100102	N	WHEELS	AF	UT	20100102	20100102	FRONT LUG NUTS LOOSEN ON 2002 JEEP LIBERTY. THIRD TIME C	
8	JEEP	GRAND CHEROKEE	2002	N	20100101	N	ELECTRICAL SYSTEM:IGNITION	MINNEAPOLIS	MN	20100102	20100102	98400	KEY WON'T TURN IN IGNITION, STEERING WHEEL LOCKED, CAR
9	JEEP	GRAND CHEROKEE	2002	N	20100101	N	STEERING	MINNEAPOLIS	MN	20100102	20100102	98400	KEY WON'T TURN IN IGNITION, STEERING WHEEL LOCKED, CAR
10	CHEVROLET	HHR	2007	N	20100102	N	SERVICE BRAKES, HYDRAULIC	MOORESVILLE	NC	20100102	20100102	30000	WITH 61,000 MILES ON MY 2007 CHEVY HHR, I AM HAVING TO RE

Figure 1. Sample CSV file

For more information about the sample data, see <https://www.nhtsa.gov/data/traffic-records>.

Creating a report

If you discover insights as you analyze your data, you can save and share them with others by creating a report. A report consists of snapshots and notes about the analysis.

Take a snapshot

1. Click the camera icon from the dashboard toolbar.



Note: You can also take a snapshot of the document preview. When you select one or more documents, only the selected documents are stored and displayed in the report. When no selection is made, all documents in the current page are stored and displayed in the report.

2. Thumbnails of the snapshot are displayed in the **Report** pane, which is a temporary store for snapshots.

This store is cleared when the browser is refreshed or another collection is opened.

3. From the menu icon of the snapshot's thumbnail, you can enter comments, or delete the snapshot. You can also edit comments later.

4. Choose thumbnails that you want to add to a new report, and then click **Create**.

Create a report

To create a report, complete the following steps:

1. From the **Report** pane, click **Create**.
2. On the **Basic** tab, name and date the report.
3. **Optional:** On the **Comments** tab, edit the title of the analysis result, and enter a comment.
4. Review the preview on the **Preview** tab.
5. When you're done editing, click **Save**.

Your report is added to the **Report** tab on the application launch page. From the **Actions** menu, you can copy the link for your report to share it with others.

Exporting data

If you discover insights as you analyze your data, you can export the data to share with others or analyze further in another business insights tool, for example.

You can export your data as a CSV file or you can generate a separate JSON file for each record.

IBM Cloud Pak for Data For IBM Cloud Pak for Data only, you cannot export secured collections. For more information, see [Supporting document-level security](#).

To export your data, complete the following steps:

1. Submit a search to find the documents of interest.
2. Click **Show documents** to open the **Documents** view, and then click the **Export** icon in the toolbar.
3. Complete the appropriate steps for the format in which you want to export the data.
 - If you want to export the data in JSON format, complete the following steps:
 1. Choose **Export JSON** to generate one JSON file for each record.
 2. **Optional:** You can change the following values:
 - Name. The file is named `export_document_{today's_date}` by default.
 - Encoding. **UTF-8** is used by default.
 - Choose whether to include fields and facets. They are excluded by default.
 - If you want to export the data in CSV format, complete the following steps:
 1. **Optional:** To customize the CSV output, choose **Export CSV with advanced options**.

You can define the format of the following elements:

 - Text content field: This is the main body field (or fields, if you configured more than one field with analyzable text). You can choose to exclude it from the export. It is exported as a column for a fact table by default.
 - All other fields: You can choose to export them as columns for fact tables or export them as dimension tables.

They are excluded from the export by default.

- Facets: You can choose to export the facets as separate CSV files that can be used as dimension tables. They are excluded from the export by default.

After customizing the CSV format, click **Save**, and then click the **Export** icon from the toolbar again.



Note: If you use the same web browser to export data in CSV format again later, your saved settings are applied automatically.

2. Choose **Export CSV**.
3. **Optional:** You can change the following values:
 - Name. The file is named `export_document_{today's_date}` by default.
 - Encoding. `UTF-8` is used by default.
 - Date and time format. `Unix epoch time` is used by default.
4. Click **Export**.

Analyzing data on demand with the Analyze API

Use the Analyze API to process text documents through the enrichment pipeline of the Discovery service without storing any data from the source documents.



Note: The Analyze API is supported by Enterprise plan deployments and installed deployments only.

This approach is ideal for business automation purposes. For example, if you want to classify emails, you can use the Analyze API to synchronously call Discovery to get a classification of the email. Then, you can use the output of that classification in your business logic.



Important: The Analyze API supports JSON documents only.

When you analyze a document with the API, you indicate how you want the document to be processed by specifying the collection to associate with the analysis. The document is not stored in the collection. Instead, the configuration settings of the collection are applied to the document. For example, if you want to find entity references in a document, run the Analyze API against a collection where the **Entities** enrichment is applied. The resulting document analysis identifies any entity mentions in the document.

Submit a request for analysis against only one collection that is configured with the enrichments that you want to use to analyze your document on demand. Remember, the documents in the collection are not significant. It is the enrichments that are defined for the collection that matter. If you submit requests to several collections, then several models are initiated at the same time, which can cause request failures.

The following enrichments are supported in the Analyze API:

- [Advanced rules models](#)
- [Contracts](#)
- [Custom entities](#)
- [Dictionary](#)
- [Document classifier](#)
- [Entities \(NLP\)](#)
- [Keywords \(NLP\)](#)
- [Machine learning and Watson Explorer Content Analytics Studio models](#)
- [Regular expressions](#)
- [Patterns \(Enterprise plan only\)](#)
- [Sentiment of documents](#)
- [Table understanding](#)^[1]
- [Text classifier](#)

For the complete list of the enrichments that are supported in each language, see [Language support](#).

For more information, see the Discovery [API reference](#).

Analysis example

The data that you submit for analysis must be in JSON format. The text must be specified as a string; it cannot be specified as an array. For example, the following JSON file contains a quotation in the **Quote** field that you want to analyze to find any keyword mentions in the text.

```
{  
  "Author": "Jane Austen",  
  "Book": "Pride and Prejudice",  
  "Quote": "From this day you must be a stranger to one of your parents. Your mother will never see you again if you do not marry Mr. Collins, and I will never see you again if you do.",  
  "Year": "1813/01/01",  
  "Subject": "Parental love",  
  "Speaker": "Mr. Bennett",  
  "url": "https://www.gutenberg.org/files/1342/1342-h/1342-h.htm#link2HCH0020"  
}
```

You know the name of a collection in your project where the **Keywords** enrichment is configured to be applied to documents in the collection. You can use the API to [list your collections](#) to find the ID associated with the collection that you look for by name.

After you get the collection ID, include it in the POST request that you submit to apply the configuration settings from the collection to your JSON file. For example, the following request submits the JSON snippet in a file that is named **favorites2.json** for keyword analysis.

```
curl --location --request POST \
```

```
'https://my-cloud-pak-for-data-cluster/discovery/zen-wd/instances/{instance-id}/api/v2/\nprojects/{project-id}/collections/{collection-id}/analyze?version=2020-08-30' \
--header 'Authorization: Bearer ...' \
--form 'file=@"/quotations/favorites2.json"'
```

The result contains a list of keywords that were recognized in the quotation.

```
{
  "result": {
    "enriched_Quote": [
      {
        "keywords": [
          {
            "text": "day",
            "mentions": [
              {
                "text": "day",
                "location": {
                  "begin": 10,
                  "end": 13
                }
              }
            ],
            "relevance": 0.673739
          },
          {
            "text": "stranger",
            "mentions": [
              {
                "text": "stranger",
                "location": {
                  "begin": 28,
                  "end": 36
                }
              }
            ],
            "relevance": 0.596757
          },
          {
            "text": "parents",
            "mentions": [
              {
                "text": "parents",
                "location": {
                  "begin": 52,
                  "end": 59
                }
              }
            ],
            "relevance": 0.568336
          },
          {
            "text": "mother",
            "mentions": [
              {
                "text": "mother",
                "location": {
                  "begin": 66,
                  "end": 72
                }
              }
            ],
            "relevance": 0.755562
          },
          {
            "text": "Mr. Collins",
            "mentions": [
              {
                "text": "Mr. Collins",
                "location": {
                  "begin": 118,
                  "end": 129
                }
              }
            ],
            "relevance": 0.945891
          }
        ]
      },
      "url": "https://www.gutenberg.org/files/1342/1342-h/1342-h.htm#link2HCH0020",
    ]
  }
}
```

```

"Subject": "Parental love",
"Year": "1813/01/01",
"Book": "Pride and Prejudice",
"Author": "Jane Austen",
"Quote": [
    "From this day you must be a stranger to one of your parents. Your mother will never see you again if you do not marry Mr. Collins, and I will never see you again if you do."
],
"metadata": {
    "name": "favorites2.json"
},
"Speaker": "Mr. Bennett"
},
"notices": []
}

```

You cannot submit an array of objects as input. For example, you might want to analyze multiple quotations, so your source might look as follows:

```

{
  "quotations": [
    {
      "Author": "Jane Austen",
      "Book": "Sense and Sensibility",
      "Quote": "Is there a felicity in the world superior to this?",
      "Year": "1811/01/01",
      "Subject": "Nature",
      "Speaker": "Marianne Dashwood",
      "url": "https://www.gutenberg.org/files/1342/1342-h/1342-h.htm#link2HCH0059"
    },
    {
      "Author": "Jane Austen",
      "Book": "Persuasion",
      "Quote": "A man does not recover from such a devotion of the heart to such a woman. He ought not; he does not.",
      "Subject": "Romantic love",
      "Year": "1818/01/01",
      "Speaker": "Captain Wentworth",
      "url": "https://www.gutenberg.org/files/105/105-h/105-h.htm#chap20"
    }
  ]
}

```

If so, break each object into a separate file and analyze each file individually.

Analyzing a text snippet

You can submit text for analysis when you specify the text in JSON format by using syntax like this:

```
{
  "text": "The text that you want to analyze."
}
```

The following example request shows how to analyze text that you specify in the request, not that you pass in a physical file.

```
curl --location --request POST \
'https://my-cloud-pak-for-data-cluster/discovery/zen-wd/instances/{instance-id}/api/v2/ \
projects/{project-id}/collections/{collection-id}/analyze?version=2020-08-30' \
--header 'Authorization: Bearer ...' \
--form 'file={"text": "ISO 9000 is a standard."}'
```

The response might look as follows.

```
{
  "result": [
    {
      "enriched_text": [
        {
          "entities": [
            {
              "text": "ISO 9000",
              "type": "my_iso_pattern",
              "mentions": [
                {
                  "text": "ISO 9000",
                  "confidence": 1.0,
                  "location": {
                    "begin": 0,
                    "end": 8
                  }
                }
              ],
              "score": 1.0
            }
          ],
          "text": "ISO 9000 is a standard."
        }
      ],
      "text": "ISO 9000 is a standard."
    }
  ]
}
```

```

    "model_name" : "My ISO Pattern"
  },
  "text" : "9000",
  "type" : "Number",
  "mentions" : [ {
    "text" : "9000",
    "confidence" : 0.8,
    "location" : {
      "begin" : 4,
      "end" : 8
    }
  }],
  "model_name" : "natural_language_understanding"
},
"metadata" : { },
"text" : [ "ISO 9000 is a standard." ]
},
"notices" : [ ]
}

```

Analyzing HTML content

You can analyze HTML when you submit the html in JSON format by using syntax like this:

```
{
  "html": "<p>My html content.</p>"
}
```

The following example request shows how to analyze text that you specify in the request, not that you pass in a physical file.

The collection to which the request is made uses the following enrichments, which means these enrichments are applied to the content that you submit with the API request:

- Entities
- Keywords
- Table Understanding

Request example

The body of the request contains **form-data** with the name **file**. The value is the JSON content to be analyzed.

```
curl --location --request POST \
'https://cpd-abc.example.com/discovery/abc-wd/instances/1671204318684041/api/v2/projects/d457fc9-a4ce-4637-a340-33123b5cbe2c/collections/2d47dbcc-64c7-84e9-0000-01851bb9d998/analyze?version=2020-08-30' \
--header 'Authorization: Bearer ...' \
--form 'file={
  "html": "<html><head>This is my html file</head><body><p>My file contains a table.</p><table><tbody><tr><th>Holiday</th><th>Popular greeting</th></tr><tr><td>Christmas</td><td>Merry Christma!s</td></tr></tbody></table></body></html>",
  "text": "This is a sentence that contains key words, such as George Washington and Boston, MA."
}'
```

Results

The results show the output of the Entities, Keywords, and Table Understanding enrichments on the **text** and **html** fields that were submitted.

```
{
  "result": {
    "text": [
      "This is a sentence that contains key words, such as George Washington and Boston, MA."
    ],
    "enriched_text": [
      {
        "keywords": [
          {
            "text": "George Washington",
            "mentions": [
              {
                "text": "George Washington",
                "location": {
                  "begin": 52,
                  "end": 69
                }
              }
            ]
          }
        ]
      }
    ]
  }
}
```

```

        }
    ],
    "relevance": 0.952591
},
{
    "text": "Boston",
    "mentions": [
        {
            "text": "Boston",
            "location": {
                "begin": 74,
                "end": 80
            }
        }
    ],
    "relevance": 0.578079
},
{
    "text": "MA",
    "mentions": [
        {
            "text": "MA",
            "location": {
                "begin": 82,
                "end": 84
            }
        }
    ],
    "relevance": 0.146905
}
],
"entities": [
{
    "text": "George Washington",
    "type": "Location",
    "mentions": [
        {
            "text": "George Washington",
            "confidence": 0.54922265,
            "location": {
                "begin": 52,
                "end": 69
            }
        }
    ],
    "model_name": "natural_language_understanding"
},
{
    "text": "Boston, MA",
    "type": "Location",
    "mentions": [
        {
            "text": "Boston, MA",
            "confidence": 0.66049105,
            "location": {
                "begin": 74,
                "end": 84
            }
        }
    ],
    "model_name": "natural_language_understanding"
}
]
},
"metadata": {},
"enriched_html": [
{
    "tables": [
        {
            "body_cells": [
                {}
            ],
            "location": {
                "begin": 99,
                "end": 183
            },
            "row_headers": [],
            "key_value_pairs": [],
            "section_title": {},
            "contexts": [],
            "text": "Holiday Popular greeting Christmas Merry Christmas!",
            "table_headers": []
        }
    ]
}
]

```

```

        "title": {},
        "column_headers": []
    }
],
"html": [
    "<html><head>This is my html file</head><body><p>My file contains a table.</p><table><tbody><tr>
<th>Holiday</th><th>Popular greeting</th></tr><tr><td>Christmas</td><td>Merry Christmas!</td></tr></tbody></table>
</body></html>"
],
"notices": []
}

```

Analyze API limits

The following table shows the file size and usage limits for the Analyze API.

Deployment type	File size limit	Concurrent collections limit	Concurrent queries per collection limit
Cloud Pak for Data installed deployment	Unlimited	Unlimited	Unlimited
Enterprise plan managed deployment	50 KB	5	5

Limits that are applied to the Analyze API usage

Use of the Analyze API from Discovery Cartridge for IBM Cloud Pak for Data affects license usage. For more information, see the [license information](#).

Monitoring usage IBM Cloud Pak for Data

You can monitor the usage of the Analyze API from the **API usage** page.



Note: The **API usage** page is available from installed deployments only. For Enterprise plans, analyze method call information is combined with query method call information and is reported as part of the query metrics.

To access the **API usage** page, open the **Projects** page, select **Data usage**, then **API usage**.

Start date

The start date of the API call-monitoring period.

End date

The end date of the API call-monitoring period.

Thirty-day call total

Number of calls to the Analyze API in the 30-day time interval that is indicated by the **Start date** and **End date**. The time interval is determined by calculating the consecutive time period with the highest number of API calls. The 30-day window updates as the time interval with the highest number of API call changes.



Note: The **API usage** is not displayed until some time after API usage monitoring begins. A delay in displaying the final total number of the **30-day call total** might occur, even if the 30-day period that is listed includes the current date.

- For the table understanding enrichment to produce any results, the input must contain a `<table>` HTML element to analyze. ↗

Searching Discovery data from Watson Assistant

Your Discovery project can provide answers to questions that stump your assistant. Instead of answering with "I don't know", your assistant can say, "I'm not sure, but I searched my knowledge base and found these answers which might help."

For more information about how to search a Discovery project from an assistant, read the appropriate Watson Assistant documentation for your situation.

- From the new experience user interface, see [Search trigger](#).
- From an actions skill in the classic user interface, see [Configuring the search for an answer](#).
- From a dialog skill, see [Adding a search skill response type](#).

 **Tip:** If you use the built-in web chat, you can use answer finding by enabling the **Emphasize the answer** feature. Answer finding highlights the word or phrase in the search result that is determined to be the exact answer to the customer's question.

For a more detailed look at the steps to take to connect to a Discovery project from Watson Assistant, take a tutorial that walks you through them. For more information, see [Power your assistant with answers from web resources](#).

Alternatively, you can add a generative language service named NeuralSeek between the Watson Discovery and Watson Assistant services. For more information, see [Use NeuralSeek to return polished answers from existing help content](#).

How the assistant calls Discovery

When a user asks your assistant a question that triggers a search, the following API request is sent to Discovery if **Emphasize the answer** is enabled.



Note: The **Emphasize the answer** feature is available from instances that are managed by IBM Cloud only.

```
{  
  "aggregation": "",  
  "sort": "",  
  "count": 10,  
  "return": [],  
  "filter": <custom_filter_specified_in_assistant>  
  "passages": {  
    "enabled": "true",  
    "fields": [  
      <search_config_body_field_specified_in_assistant>  
    ],  
    "characters": 325,  
    "per_document": true,  
    "max_per_document": 3,  
    "find_answers": true,  
    "max_answers_per_passage": 1  
  },  
  "highlight": false,  
  "spellingSuggestions": false,  
  "table_results": {  
    "enabled": false  
  },  
  "suggested_refinements": {  
    "enabled": false  
  }  
}
```

When **Emphasize the answer** is used (`"find_answers": true`), Discovery rescores and reorders the documents to ensure that documents with the highest-quality answers are returned first.

Choosing a project type

If the **Conversational Search** project type isn't providing the best answers and you want to understand why, switch to using a **Document Retrieval** project type.

Most often, the **Conversational Search** project type is the right choice. You get great results from the start, and when you enable extra features like **Emphasize the answer**, the answers are clear and concise. However, for advanced use cases, or if you want to be able to troubleshoot issues, a **Document Retrieval** project type might be a better fit.

To help you choose the right Discovery project type, review the project type differences that are described in the following table.

Function	Conversational Search	Document Retrieval
----------	-----------------------	--------------------

Enrichment support	Only the Part of Speech enrichment is applied.	The Part of Speech and Entities enrichments are applied. The Entities enrichment is helpful for identifying important information and introduces more ways to filter query results.
Testing queries from the Improve and customize page in Discovery	You see only one of the responses that are returned from the chatbot. You cannot see all of the available responses and cannot analyze individual query results.	You can filter query results by enrichment-based facets. You can review details about fields that are indexed in the source documents that are returned for a query. Access to more information makes it easier to troubleshoot unexpected results.
Search triggers	Returns answers from the text field automatically. If answers are stored in another field, you must change the configuration.	You can apply a Smart Document Understanding (SDU) model or enrichments to your collections and retrieve useful information from fields other than text when search is triggered from the assistant.

Project type details

For both project types, the best way to test is to trigger search from the Watson Assistant preview. When you configure search support for an assistant, you can fine-tune the experience in ways that aren't available in Discovery.

And settings that are available from the **Search results** tool for a **Document Retrieval** project type are replaced by configuration settings that you specify in Watson Assistant. For example, the query response title and body are defined in Watson Assistant. And a passage length of 325 characters is applied to responses regardless of what you specify in the **Max characters in a passage** field.

The way that you deploy search support in your chatbot is the same regardless of the project type. You enable search support in your assistant and then publish your assistant.



Note: If you decide that you want to use a **Document Retrieval** project type, you must create it *before* you add the search function to your virtual assistant. Otherwise, when you add search support to your assistant, a **Conversational Search** project is created for you automatically. When you have a service instance that contains a **Document Retrieval** project already, the Watson Assistant user interface shows the existing instance, and you can choose to use it.

Deploying the built-in UI components

For Document Retrieval and custom projects, a set of user interface components are available for your use.

Work with a developer to use the pre-built UI components that are provided by IBM to deploy an application.

For more information about building your own app, see the [Building custom applications with the API](#).

Several built-in UI components are available.

Search bar

A search box that uses a natural language understanding query to fetch the most relevant results.

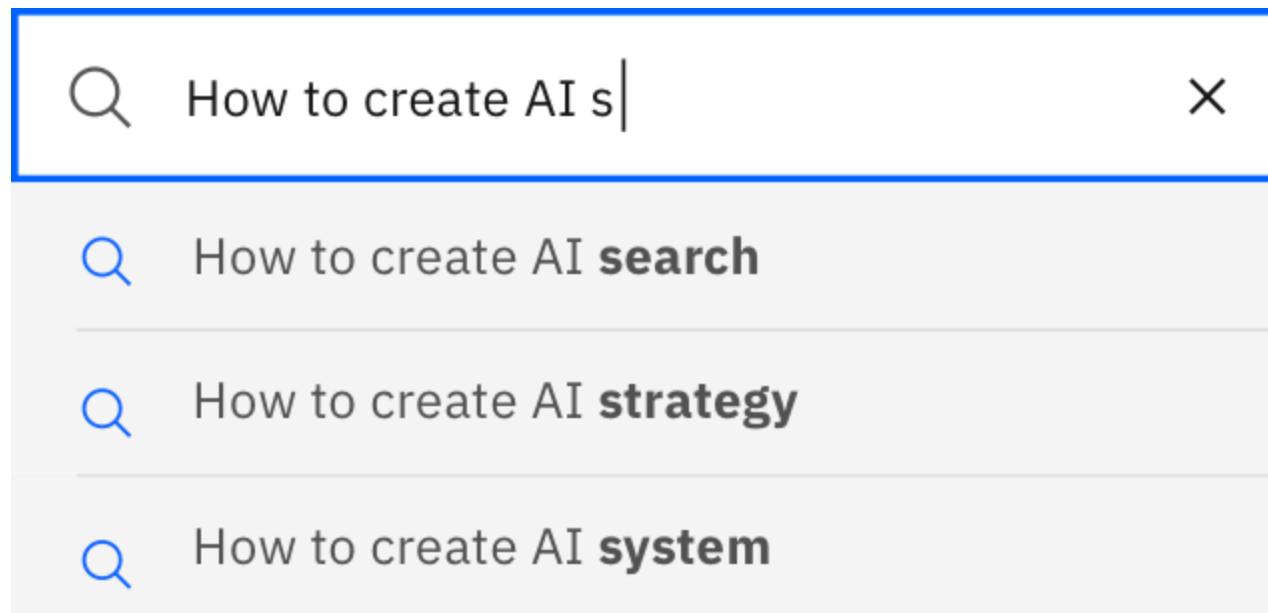


Figure 1. Search bar type ahead

[Try it](#)

Search results

A set of results that rank the most relevant passages and tables to a query.

A screenshot of a search results list. It displays two search results, each in a separate card. The first result is a passage from a document titled "IBM_Analytics_Machine_Learning.pdf": "Machine-learning techniques are required to improve the accuracy of predictive models. Depending on the nature of the business problem being addressed, there are different approaches based on the type and volume of the data." Below the passage are two buttons: "View in document" and "IBM Docs". The second result is a passage from a document titled "IBM_Research_DL.pdf": "Deep learning is a specific method of machine learning that incorporates neural networks in successive layers to learn from data in an iterative manner. Deep learning is especially useful when you're trying to learn patterns from unstructured data." Below the passage are two buttons: "View in document" and "IBM Docs".

Figure 2. Search results list

[Try it](#)

Facets

Refine your results with facets that help users filter the search results by specific categories and domains.

Machine Learning Terms

- Neural network
- Reinforced learning
- CIFAR-10
- MNIST
- Recommender systems

"Neural-network training can be slow and energy weight data for the network between conventional Analogue non-volatile memory can accelerate the backpropagation by performing parallelized multi|

[View passage in document](#)

Equivalent-accuracy accelerated neural.pdf

"Neural-network training can be slow and energy weight data for the network between conventional Analogue non-volatile memory can accelerate the backpropagation by performing parallelized multi|

[View passage in document](#)

Figure 3. Facets

[Try it](#)

Rich document preview

Displays your results in a document preview. This view highlights result passages within the text of the original document. It also shows any enrichment mentions that are detected in the document. The rich preview is available with source documents where an SDU model is applied, such as PDF, Microsoft PowerPoint, and Microsoft Word files.

Identified elements ⓘ

Based on the enrichments that are applied to the project, the following items were found. Select items below to show them highlighted in the document.

[Clear all ×](#)

Entities v2 ^

- Organization (492)
- Date (285)
- Number (131)
- Ordinal (49)
- Money (14)
- Percent (8)
- JobTitle (6)
- Location (4)
- Person (1)

[Show less](#)

5 / 144 pages

Our commitment to science and innovation
While we are focused on meeting the needs of clients today, we continue to shape the technologies of tomorrow. That is why IBM Research continues to advance the fundamental science of computing, driving innovation and pioneering a new era of accelerated discovery.

IBM continues to lead the development of quantum computing. This year, we delivered operational quantum computers to Japan and Germany, deployed the world's first 127-qubit processor, and are now on our way to a 1,000-qubit processor by the end of 2023. We also forged a series of long-term partnerships with universities, governments, and hospitals to develop quantum applications that will accelerate the discovery of everything from medicine to materials.

In 2021, we unveiled not one, but two, major breakthroughs in semiconductor design. First, the world's first 2-nanometer chip technology, which will allow 50 billion transistors to fit on a chip the size of a fingernail. The chips are expected to achieve 45% higher performance than today's 7-nanometer chips. Second, in collaboration with our Albany Research Alliance partner Samsung, IBM Research introduced a completely new approach to semiconductor design called Vertical-Transport Nanosheet Field Effect Transistor, or VTFET, which could help keep Moore's law alive for years to come.

Responsible stewardship for the digital age
At IBM, we have always understood that our responsibilities extend far beyond the bottom line. That is why we embrace our leadership role in defining good tech in the digital age.

Among the most pressing challenges facing our society today is closing the STEM skills gap, which holds back both technological and socioeconomic progress. To address this issue, IBM regularly engages at the highest levels of government



Arvind Krishna
Chairman and
Chief Executive Officer

Figure 4. Rich document view

[Try it](#)

Deploying a project

To deploy your project, complete the following steps:

1. To use the API, you need to know the project ID for your project. Go to the [Integrate and Deploy > API Information](#) page.
2. From the [Integrate and Deploy > UI Components](#) page, find links to resources that a developer can use to get started.
 - [GitHub](#)
 - [Storybook](#)

Getting started with the GitHub sample app

From resources available in GitHub, you can run a script to start a sample app with prebuilt UI components. In fact, the sample app looks a lot like the [Improve and customize](#) page of the product because the product itself uses these UI components.

The script requires some prerequisite software to function. After you start the script, it checks whether you have the necessary

software installed on your system. If not, it lets you know what software you need to install. Install the following packages if they are not installed already:

- [Node.js](#)
- [Yarn](#)

The script needs information about your service instance and project to use the data and search settings that you configured for your project and apply them to the sample app. You must collect the following information so that you can share it with the script when it asks you for the information later:

Service credentials

The following information is used by the sample app script to construct an endpoint where it can send API requests and to authenticate with your service instance:

- URL
- API key

To get this information, complete the appropriate steps for the type of deployment you are using:

- IBM Cloud From the [IBM Cloud Resource list](#), expand the **AI/Machine Learning** section, and then find the service instance that you created earlier. Click the instance to open its overview page. From the **Credentials** section, copy the URL and API key values and store them somewhere where you can access them later, such as a local text file.
- IBM Cloud Pak for Data From the IBM Cloud Pak for Data web client main menu, expand **Services**, and then click **Instances**. Find your instance, and then click it to open its summary page. Scroll to the **Access information** section of the page, and then copy the **URL** and bearer token. Store the values somewhere where you can access them later, such as a local text file. (The bearer token serves as the apikey for installed deployments.)

Project ID

The unique identifier for the project you created in this tutorial.

You can copy the project ID from the **API Information** tab of the **Integrate and deploy** page.

To run the script that starts the sample app, complete the following steps:

1. Do one of the following things:
 - If you downloaded the repo, extract the files from the archive to a working directory on your system. Open a command terminal window, and then change to the directory where you downloaded the repository files.
 - If you cloned the repo, open a terminal window from the directory to which you cloned the repository.
2. Enter the following command to start the script:

```
./runExampleApp.sh
```

Give the script time to set up the necessary resources to run the application.

If any required prerequisite software packages are missing, the script lets you know what packages you need to install before you can use the script successfully.

3. When prompted to specify the **authType**, enter the type of authentication you use. The type differs based on how your service instance is deployed:
 - IBM Cloud Enter **iam**
 - IBM Cloud Pak for Data Enter **CP4D**.

The **iam** value indicates that you are using Identity and Access Management, which is a service that is used by IBM Cloud to authenticate its managed services. For installed instances that are deployed on IBM Cloud Pak for Data, **CP4D** is specified instead.

For the next three prompts, enter the information that you copied and saved earlier.

- url
- apikey
- project_id

When the script is done, it asks if you want to start the sample app now. Enter **y** for yes. A new web browser window or tab is

displayed and the sample app is rendered in the page. The URL for the sample app is `http://localhost:3000/`, which means that the app is running locally and cannot be accessed by anyone who is using a different computer.

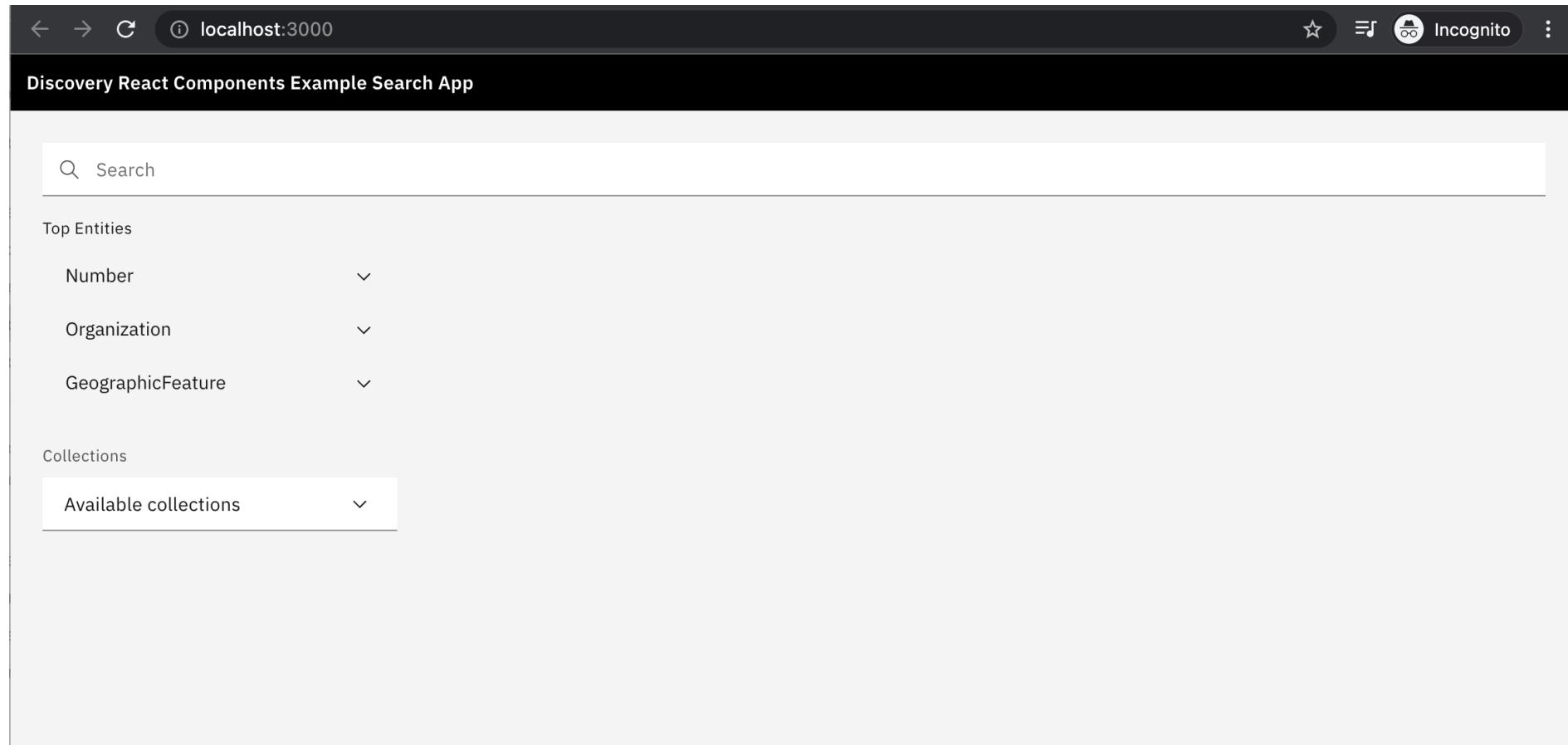


Figure 5. Sample app user interface

The sample app gives you a preview of your search project. Use it to test your search project and make any necessary adjustments.

When you're done testing with the sample app, you can stop it by returning to the terminal window where you ran the initial script, and pressing `Ctrl + C`.

Deploying other project types

To analyze data from a Content Mining project, click **Launch application** from the *Improve and customize* page. For more information, see [Analyzing your data](#).

To analyze a contract from a Document Retrieval for Contracts project, submit a query. For more information about understanding contract information, see [Understanding contracts](#).

To deploy a Conversational Search project, connect this project to an assistant that is built with Watson Assistant. The general steps to follow include:

1. Create an assistant.

You can use a Watson Assistant Trial plan for testing purposes.

2. Add a search skill to your assistant, and then connect it to this project.
3. Deploy your assistant.

For more information about building a Watson Assistant search skill, see the appropriate documentation for your deployment:

- IBM Cloud From the new experience, see [Adding a search integration](#).
- IBM Cloud From the classic experience, see [Embedding existing help content](#).
- IBM Cloud Pak for Data [Creating a search skill](#).

Choosing a deployment solution

Discovery is available both as a service that is hosted by IBM Cloud and as a service that you install on IBM Cloud Pak for Data. Learn about these deployment solutions and how they differ.

The product user interface and APIs are mostly equivalent regardless of whether you use the managed or installed version of the service. The few differences between the two solutions include:

- How you deploy and set up the service
- The underlying technology that is used to crawl data sources
- Limits for things like the maximum number of documents and enrichments or file sizes
- Who to contact for support and how you share product feedback

Although the documentation that describes how to use the product is the same (what you're reading now), more documentation about how to install and administer the service in IBM Cloud Pak for Data is available from the [IBM Cloud Pak for Data product documentation](#) that is hosted in IBM Documentation.

To keep up with product changes, check the following topics periodically:

- IBM Cloud Pak for Data [Release notes for IBM Cloud Pak for Data service instances](#)
- IBM Cloud [Release notes for IBM Cloud service instances](#)

Comparing features

The following table describes the feature support differences between the two deployment types.

Feature	IBM Cloud	IBM Cloud Pak for Data
Crawl the local file system, Window file system, databases, LDAP directories, FileNet P8, and HCL Notes		
Schedule crawls with more precision		
Apply document-level security to crawled collections		
Enable JavaScript execution for web pages that you want to crawl		
Crawl IBM Cloud Object Storage		
Preview .pdf files that are crawled from external data sources		
Build custom crawlers		
Use App Connect to crawl other external data sources		
Apply answer finding to search queries		
Optical Character Recognition v2		
Patterns enrichment		
App switcher menu where you can get service instance information and usage statistics		
Import and apply UIMA text analysis models created in Watson Explorer Content Analytics Studio		
Monitor usage with activity tracker events		



Feature support details

Installing

Installation overview

Find information about how to install IBM Watson® Discovery Cartridge for IBM Cloud Pak® for Data.

IBM Cloud Pak for Data



Note: This information applies only to installed deployments.

Full installation instructions

- [4.7.x](#)
- [4.6.x](#)
- [4.5.x](#)
- [4.0.x](#)
- [2.2.0, 2.2.1](#)

Federal Information Security Management Act (FISMA) support is available for Discovery for Cloud Pak for Data offerings purchased on or after August 30, 2019. IBM Watson® Discovery is FISMA High Ready.

Support matrix

You install IBM Cloud Pak for Data, and then install the Discovery service.

- The 4.6.2 release is the last supported release on Red Hat OpenShift Container Platform 4.8.
- The 4.5.3 release is the last supported release on Red Hat OpenShift Container Platform 4.6.29 or later.

Discovery version	IBM Cloud Pak for Data version	Red Hat OpenShift version
4.7.1	4.7.1	4.12
4.7.1	4.7.1	4.10
4.7.0	4.7.0	4.12
4.7.0	4.7.0	4.10
4.6.5	4.6.6	4.12
4.6.5	4.6.6	4.10
4.6.5	4.6.5	4.12
4.6.5	4.6.5	4.10
4.6.3	4.6.4	4.12
4.6.3	4.6.4	4.10
4.6.3	4.6.3	4.10
4.6.2	4.6.2	4.10
4.6.2	4.6.2	4.8
4.6.2	4.6.1	4.10
4.6.2	4.6.1	4.8

4.6.0	4.6.0	4.10
4.6.0	4.6.0	4.8
4.5.3	4.5.3	4.10
4.5.3	4.5.3	4.8
4.5.3	4.5.3	4.6.29 or later
4.5.1	4.5.1	4.10
4.5.1	4.5.1	4.8
4.5.1	4.5.1	4.6.29 or later
4.5.0	4.5.0	4.10
4.5.0	4.5.0	4.8
4.5.0	4.5.0	4.6.29 or later
4.0.9	4.0.9	4.8
4.0.9	4.0.9	4.6.29 or later
4.0.8	4.0.8	4.8
4.0.8	4.0.8	4.6.29 or later
4.0.7	4.0.7	4.8
4.0.7	4.0.7	4.6.29 or later
4.0.6	4.0.6	4.8
4.0.6	4.0.6	4.6.29 or later
4.0.5	4.0.5	4.8
4.0.5	4.0.5	4.6.29 or later
4.0.4	4.0.4	4.8
4.0.4	4.0.4	4.6.29 or later
4.0.3	4.0.3	4.8
4.0.3	4.0.3	4.6.29 or later
4.0.2	4.0.2	4.8
4.0.2	4.0.2	4.6
4.0.0	4.0.0	4.6

2.2.1	3.5.0	4.5, 4.6
2.2.1	3.5.0	3.11.188
2.2.1	3.0.1	4.5, 4.6
2.2.1	3.0.1	3.11.188
2.2.0	3.5.0	4.5
2.2.0	3.5.0	3.11.188
2.2.0	3.0.1	4.5
2.2.0	3.0.1	3.11
2.1.4	3.0.1	3.11.188
2.1.4	2.5	3.11
2.1.3	3.0.1	3.11.188
2.1.3	2.5	3.11

Support matrix

The **3.11.188** version more precisely means 3.11.188 or a later 3.11 version.

Upgrading the service

Learn how to upgrade the version of your installed service deployment.

IBM Cloud Pak for Data



Note: This information applies only to installed deployments.

Upgrade your deployment

The steps to follow to upgrade your Discovery service instance are described in the IBM Cloud Pak for Data documentation. The following in-place upgrades are supported:

- From one 4.7.x release to a later 4.7.y release. For more information, see [Upgrading Watson Discovery to a newer 4.7 refresh](#).
- From a 4.6.x release to the latest 4.7 release. For more information, see [Upgrading Watson Discovery](#).
- From a 4.0.x, 4.5.x, or earlier 4.6.x release to the latest 4.6 release. For more information, see [Upgrading Watson Discovery](#).
- From a 4.0.x release or from one 4.5.x release to a later 4.5.y release. For more information, see [Upgrading Watson Discovery](#).
- From one 4.0.x release to a later 4.0.y release. For more information, see [Upgrading Watson Discovery to a newer 4.0 refresh](#).

You cannot do an in-place upgrade from releases prior to version 4.x. For information about how to move from a 2.x deployment to IBM Watson® Discovery 4.5, see [Backing up and restoring data in IBM Cloud Pak for Data](#).

Troubleshooting IBM Watson® Discovery Cartridge for IBM Cloud Pak® for Data deployments

Learn ways to troubleshoot and address issues that you might encounter while using the product.

IBM Cloud Pak for Data



Note: This information applies only to instances of IBM Watson® Discovery that are installed on IBM Cloud Pak® for Data. For troubleshooting tips about adding data to both installed and managed deployments, see [Troubleshooting ingestion](#).

The information in this topic suggests steps you can take to investigate issues that might occur. For information about known issues

and their workarounds per version, see [Known issues](#).

Minio pods enter a reboot loop during installation or upgrade

- **Error:** `Cannot find volume "export" to mount into container "ibm-minio"` is displayed during an installation or upgrade of Discovery. When you check the status of the Minio pods by using the command, `oc get pods -l release=wd-minio -o wide`, and then check the `Minio operator` logs by using the commands, `oc get pods -A | grep ibm-minio-operator`, and then `oc logs -n <namespace> ibm-minio-operator-XXXXX`, you see an error similar to the following one in the logs:

```
ibm-minio/templates/minio-create-bucket-job.yaml failed: jobs.batch "wd-minio-discovery-create-bucket" already exists) and failed rollback: failed to replace object"
```

- **Cause:** A job that creates a storage bucket for Minio and then is deleted after it completes, is not being deleted properly.
- **Solution:** Complete the following steps to check whether an incomplete `create-bucket` job for Minio exists. If so, delete the incomplete job so that the job can be recreated and can then run successfully.

1. Check for the Minio job by using the following command:

```
oc get jobs | grep 'wd-minio-discovery-create-bucket'
```

2. If an existing job is listed in the response, delete the job by using the following command:

```
oc delete job $(oc get jobs -o yaml | grep 'wd-minio-discovery-create-bucket')
```

3. Verify that all of the Minio pods start successfully by using the following command:

```
oc get pods -l release=wd-minio -o wide
```

A `No space left on device` message is displayed in the log

If the a `wd-ibm-elasticsearch-es-server-client` pod restarts repeatedly and then reports the `Crashloopbackoff` state with the message `No space left on device` written to the log for the pod, then the issue might be a lack of memory on the pod. Follow the steps to [troubleshoot out of memory issues](#) or contact IBM Support.

A `java.lang.OutOfMemoryError: Java heap space` message is displayed in the log

When indexing a large set of documents that have multiple enrichments applied to them, the worker node can run out of space. To address the issue, first determine which pod is out of memory by completing the following steps:

If the document status cannot be promoted to `Processing`, check the status of the `inlet`, `outlet`, and `converter` pods.

1. Run the following command:

```
$ oc get pod -l 'tenant=wd,run in (inlet,outlet,converter)'
```

2. If any of the pods are not showing a `Running` status, restart the failing pod by using the following command:

```
$ oc delete pod <pod_name>
```

3. Otherwise, open the `Manage collections>{collection name}>Activity` page. Check the `Warnings and errors at a glance` section for the message, `OutOfMemory happened during conversion. Please reconsider size of documents.` If shown, use the following command to raise the memory size for the converter:

```
$ oc patch wd wd --type=merge \
--patch='{"spec": {"ingestion": {"converter": {"maxHeapMemory": "10240m", "resources": {"limits": {"memory": "10Gi"} } } } }'
```



Note: Adjust the value of `maxHeapMemory` and the container memory according to your cluster resources.

4. After the `converter` pod restarts successfully, click `Reprocess` from the Activity tab.

If documents are stuck in `Processing` status and cannot be promoted to the `Available` status for a collection, complete the following steps:

1. Check the status of Hadoop by using the following command:

```
$ oc get pod -l 'tenant=wd,run in (hdp-rm,hdp-worker)'
```

where `l` is a lowercase L for list.

2. If any of the pods are not showing a `Running` status, restart the failing pod by using the following command:

```
$ oc delete pod <pod_name>
```

3. Check whether any of the Hadoop worker nodes has insufficient memory by using the following command to look for the `OOM when allocating` message:

```
$ oc logs -l tenant=wd,run=hdp-worker -c logger --tail=-1 \  
| grep "OOM when allocating"
```

4. If a match is found, use the following command to patch the resource:

```
$ oc patch wd wd --type=merge \  
--patch='{"spec":{"orchestrator":{"docproc":{"pythonAnalyzerMaxMemory":"8g"}}}}'
```

 **Note:** The maximum allowed value for `pythonAnalyzerMaxMemory` is `12g`. The default value is `6g`. Increase the value gradually, such as in increments of 2g at a time according to your cluster resources.

5. Check whether any of the Hadoop worker nodes has insufficient memory by using the following command to look for the `OutOfMemoryError` message:

```
$ oc logs -l tenant=wd,run=hdp-worker -c logger --tail=-1 \  
| grep "OutOfMemoryError"
```

6. If a match is found, check the current environment variable values and the Hadoop worker node memory resources by using following commands:

- To check the `"DOCPROC_MAX_MEMORY"` variable in the orchestrator container:

```
$ oc exec `oc get po -l run=orchestrator -o 'jsonpath={.items[0].metadata.name}'` env \  
| grep DOCPROC_MAX_MEMORY
```

- To check the `"YARN_NODEMANAGER_RESOURCE_MEMORY_MB"` variable in the Hadoop worker node container:

```
$ oc exec `oc get po -l run=hdp-worker -o 'jsonpath={.items[0].metadata.name}'` \  
-c hdp-worker -- env | grep YARN_NODEMANAGER_RESOURCE_MEMORY_MB
```

- To check the memory resource of the `hdp-worker` container:

```
$ oc get po -l run=hdp-worker -o 'jsonpath=requests are \  
.items[*].spec.containers[?(.name=="hdp-worker")].resources.requests.memory, \  
limits are {.items[*].spec.containers[?(.name=="hdp-worker")].resources.limits.memory}'
```

7. Patch the environment variable resources gradually by using the following command:

```
$ oc patch wd `oc get wd -o 'jsonpath={.items[0].metadata.name}'` \  
--type=merge --patch='{"spec":{"orchestrator":{"docproc":{"maxMemory":"4g"}}, \  
"hdp":{"worker":{"nm":{"memoryMB":12000}, "resources":{"limits":{"memory":"20Gi"}, \  
"requests":{"memory":"20Gi"}}}}}'
```

The default values for the resources are as follows:

- `docproc.maxMemory`: 2g

Increase in increments of 2g at a time.

- `nm.memoryMB`: 10,240

Start from 12,000 and increase in increments of 2,000 at a time.

- `memory requests/limits`: 13Gi/18Gi

Increase in increments of 2Gi at a time.

8. Check whether the Hadoop pods restart successfully by using the following command:

```
$ oc get pods -l 'tenant=wd,run in (orchestrator,hdp-worker)'
```

9. Confirm that the new configurations were applied after you patched the cluster.

10. If the pod is not restarted, check whether the resource got updated by using the following command:

```
$ oc get wd wd -o yaml
```

11. Check the status of Elasticsearch by checking the client and data nodes separately.

12. On the client node, run the following command to check whether an out-of-memory exception occurred:

```
$ oc logs -l tenant=wd,ibm-es-data=False,ibm-es-master=False \
-c elasticsearch --tail=-1 | grep "OutOfMemoryError"
```

13. If an error message is found, excluding INFO messages, increase the memory resource by using the following command:

```
$ oc patch wd wd --type=merge \
--patch='{"spec": {"elasticsearch": {"clientNode": {"maxHeap": "896m", "resources": {"limits": {"memory": "1792Mi"} }}}}'
```

14. After you run this command, the Elasticsearch client pod is restarted about 20 minutes later. Monitor the "AGE" of the pod by using the following command:

```
$ oc get pod -l tenant=wd,ibm-es-data=False,ibm-es-master=False
```

15. After the pod is restarted successfully, check the new value of **ES_JAVA_OPTS** and the container memory limit by using the following command:

```
$ oc describe $(oc get po -l tenant=wd,ibm-es-data=False,ibm-es-master=False -o name)
```

16. On the data node, run the following command to check whether an out-of-memory exception occurred:

```
$ oc logs -l tenant=wd,ibm-es-data=True,ibm-es-master=False \
-c elasticsearch --tail=-1 | grep "OutOfMemoryError"
```

17. If an error message is found, excluding INFO messages, increase the memory resource by using the following command:

```
$ oc patch wd wd --type=merge \
--patch='{"spec": {"elasticsearch": {"dataNode": {"maxHeap": "6g", "resources": {"limits": {"memory": "10Gi"}, "requests": {"memory": "8Gi"} }}}}'
```

18. After you run this command, the Elasticsearch client pod is restarted about 20 minutes later. Monitor the "AGE" of the pod by using the following command:

```
$ oc get pod -l tenant=wd,ibm-es-data=True,ibm-es-master=False
```

19. After the pod is restarted successfully, check the new value of **ES_JAVA_OPTS** and the container memory requests/limit by using the following command:

```
$ oc describe $(oc get po -l tenant=wd,ibm-es-data=True,ibm-es-master=False -o name)
```



Note: For both nodes (client and data), set the **resources.limits.memory** equal to **2 * maxHeap**.

If the pod cannot be restarted after 30 mins by applying the **oc patch** command, collect logs to share with IBM Support by using the following command:

```
$ oc logs -l control-plane=ibm-es-controller-manager --tail=-1
```

For both issues, where document status cannot be promoted to **Processing** and where documents are stuck in **Processing** status, if you are using Portworx storage, you can check whether the Elasticsearch disk is full.

1. Run the following command to check whether the Elasticsearch disk is full:

```
$ oc logs -l tenant=wd,run=elastic,ibm-es-master=True \
-c elasticsearch --tail=100000|grep 'disk watermark'
```

2. If the log shows a message such as **watermark exceeded on x-data-1**, it means the disk on the node that is specified is full and you need to increase the disk size by using the following command:

```
$ oc patch pvc $(oc get pvc -l tenant=wd,run=elastic,ibm-es-data=True,ibm-es-master=False \
-o jsonpath='{.items[N].metadata.name}') \
-p '{"spec": {"resources": {"requests": {"storage": "60Gi"} }}}'
```

where **N** denotes the data node number that was reported from the log.

For example, if the log mentions **data-1** in the node name, then the command to use is:

```
$ oc patch pvc $(oc get pvc -l tenant=wd,run=elastic,ibm-es-data=True,ibm-es-master=False \
-o jsonpath='{.items[1].metadata.name}') -p '{"spec": {"resources": {"requests": {"storage": "60Gi"} }}}'
```

Setting the shard limit in Discovery for Cloud Pak for Data

In Discovery version 2.2.0, there is a limit to the number of shards that can stay open on a cluster. In development instances, the limit is 1,000 open shards, and in production instances, the limit is two data nodes, which is equal to 2,000 open shards, or 1,000 open shards per data node. After you reach either limit, you cannot create any more projects and collections on your cluster, and if you try to create a new project and collection, you receive an error message.

This limit is due to the fact that, when you install Discovery version 2.2.0, Elasticsearch version 7.8.0 automatically runs on your clusters. Because this version of Elasticsearch runs on your clusters, a new cluster stability configuration becomes available that limits the number of open shards to 1,000 for each Elasticsearch data node.

If you are unable to create new projects and collections and you receive errors, first check the status of your Elasticsearch cluster and the number of shards on that cluster. Consider increasing the number of data nodes on your cluster to support more shards. This method is optimal for maximizing performance. However, an increased number of nodes uses more memory. If the number of shards reaches the limit, you can also increase the limit in a data node. For more information about increasing the shard limit in a node, see [Increasing the shard limit](#).



Note: This limit of 1,000 shards does not apply to versions of Discovery that are earlier than 2.2.0.

Increasing the shard limit

1. Log in to your Discovery cluster.
2. Access your data node.
3. Enter the following command:

```
$ oc exec -it $(oc get pod \
-l app=elastic,ibm-es-data=True -o jsonpath='{.items[0].metadata.name}') -- bash
```

4. Enter the following command, replacing the **<>** and the content inside with your port number:

```
$ curl -X POST http://localhost:<port_number>/_cluster/health?pretty
```

If you do not know what your port number is, enter the following command to find it:

```
$ oc get pod -l app=elastic,ibm-es-data=True -o json \
| jq .items[].spec.containers[].ports[0].containerPort | head -n 1`
```

The curl **POST** command returns a value for **active_primary_shards**. If you have one data node that has a value larger than 1,000 or if you have two data nodes that have a value larger than 2,000, you must increase the shard limit to create new projects and collections in your cluster.



Important: If you increase this limit, the cluster becomes less stable because it contains an increased number of shards.

5. Enter the following command to increase the number of shards, replacing **<port_number>** with your port number and **<total_shards_per_node>** and **<max_shards_per_node>** with the new shard limit that you want to assign to a node:

```
$ curl -X POST http://localhost:<port_number>/_cluster/settings \
```

```
-d '{"persistent": {"cluster.routing.allocation.total_shards_per_node":<total_shards_per_node>, \
"cluster.max_shards_per_node":<max_shards_per_node>} }' \
-XPUT -H 'Content-Type:application/json'
```

After you increase the shard limit, you can create more projects and collections on your cluster.

Clearing a lock state

IBM Cloud Pak for Data **Installed only**: When the `gateway` pod restarts, it runs a database validation plug-in that checks for changes and applies the latest change sets to the shared database. If the pod is restarted while this check is in process, the plug-in might remain in a lock state, preventing the service from starting. Manual database intervention might be needed to clear the lock.

If the Discovery API does not come online or if the `gateway-0` pod looks like it is in a constant crash loop, you can try checking the Liberty server logs for the API service located here: `/opt/ibm/wlp/output/wdapi/logs/messages.log`

The logs would indicate if Liquibase is failing and unable to run. If the system is locked, you might see something similar to the following:

```
$ [11/7/19 5:07:51:491 UTC] 0000002f liquibase.executor.jvm.JdbcExecutor I SELECT LOCKED FROM
public.databasechangeloglock WHERE ID=1
[11/7/19 5:07:51:593 UTC] 0000002f liquibase.lockservice.StandardLockService I Waiting for changelog lock....
[11/7/19 5:08:01:601 UTC] 0000002f liquibase.executor.jvm.JdbcExecutor I SELECT LOCKED FROM
public.databasechangeloglock WHERE ID=1
[11/7/19 5:08:02:091 UTC] 0000002f liquibase.lockservice.StandardLockService I Waiting for changelog lock....
[11/7/19 5:08:12:097 UTC] 0000002f liquibase.executor.jvm.JdbcExecutor I SELECT LOCKED FROM
public.databasechangeloglock WHERE ID=1
[11/7/19 5:08:12:197 UTC] 0000002f liquibase.lockservice.StandardLockService I Waiting for changelog lock....
[11/7/19 5:08:22:203 UTC] 0000002f liquibase.executor.jvm.JdbcExecutor I SELECT ID,LOCKED,LOCKGRANTED,LOCKEDBY
FROM public.databasechangeloglock WHERE ID=1
[11/7/19 5:08:22:613 UTC] 0000002f com.ibm.ws.logging.internal.impl.IncidentImpl I FFDC1015I: An FFDC Incident
has been created: "org.jboss.weld.exceptions.DeploymentException: WELD-000049: Unable to invoke public void
liquibase.integration.cdi.CDILiquibase.onStartUp() on liquibase.integration.cdi.CDILiquibase@7f02a07
com.ibm.ws.container.service.state.internal.ApplicationStateManager 31" at ffdc\19.11.07\_05.08.22.0.log
```

It is possible to manually unlock the plug-in. If you have Discovery 2.1.4 or earlier, enter the following command on the postgres database that the `gateway-0` pod is looking at:

```
$ psql dadmin
UPDATE DATABASECHANGELOGLOCK SET LOCKED=False, LOCKGRANTED=null, LOCKEDBY=null where ID=1;
```

If you have Discovery 2.2.0 or later, enter the following command on the postgres database that the `gateway-0` pod is looking at:

```
$ oc exec -it wd-discovery-postgres-0 -- bash -c 'env PGPASSWORD="$PG_PASSWORD" psql
"postgresql://$PG_USER@$STKEEPER_CLUSTER_NAME-proxy-service:$STKEEPER_PG_PORT/dadmin" -c "UPDATE
DATABASECHANGELOGLOCK SET LOCKED=False, LOCKGRANTED=null, LOCKEDBY=null where ID=1"'
```

If you can then restart the `gateway` pod, everything should resume normally.

Environment variable settings for Smart Document Understanding

There are two environment variables that need to be adjusted for Smart Document Understanding in IBM Watson® Discovery version 2.1.0. This was resolved in version 2.1.1, see [2.1.1 release, 24 Jan 2020](#).

```
$ SDU_PYTHON_REST_RESPONSE_TIMEOUT_MS
SDU_YOLO_TIMEOUT_SEC
```

Both of these should be set to their respective `hour` value. These values must be set in the `<release-name>-watson-discovery-hdp` ConfigMap.

The values should be:

`SDU_PYTHON_REST_RESPONSE_TIMEOUT_MS` should be set to `3600000`

`SDU_YOLO_TIMEOUT_SEC` should be set to `3600`

These values should be set when the software is installed or reinstalled.

Troubleshooting error messages

If you receive error messages that are related to timeouts and insufficient memory for your enrichments, you can enter the following

commands to change timeout and memory settings to potentially resolve these error messages:

- Notice message: **<enrichment_name>: Document enrichment timed out**

Suggested action: Increase the document processing timeout. Enter the following command to increase the default timeout from 10 to 20 minutes:

```
$ oc patch wd wd --type=merge \
--patch='{"spec": {"orchestrator": {"docproc": {"defaultTimeoutSeconds": 1200} } } }'
```

- Notice message: **<enrichment_name>: Document enrichment failed due to lack of memory**

Suggested action: Increase the orchestrator container memory limit. Enter the following command to increase the memory limit from 4 Gi to 6 Gi:

```
$ oc patch wd wd --type=merge \
--patch='{"spec": {"orchestrator": {"resources": {"limits": {"memory": "6Gi"} } } } }'
```

- Notice message: **Indexing request timed out**

Suggested action: Increase the timeout for pushing documents to Elasticsearch. Enter the following command to increase the default timeout from 10 to 20 minutes:

```
$ oc patch wd wd --type=merge \
--patch='{"spec": {"shared": {"elastic": {"publishTimeoutSeconds": 1200} } } } }'
```

Known issues

Known issues are listed by the release in which they were identified.

- For the list of release notes, see [Release notes](#).
- For troubleshooting information, see [Troubleshooting](#).

IBM Cloud Pak for Data



Note: The known issues that are described in this topic apply to installed deployments only.



Important: Known issues are cumulative. Issues from previous releases persist in later releases unless otherwise noted.

4.7.x releases

See [Limitations and known issues in Watson Discovery](#)

4.6.x releases

See [Limitations and known issues in Watson Discovery](#)

4.5.x releases

See [Limitations and known issues in Watson Discovery](#)

4.0.x releases

For more information about known issues, see the [IBM Cloud Pak for Data documentation](#).

4.0.9, 25 May 2022

- Discovery generates a partial failure status message for the IBM Cloud Pak for Data Red Hat OpenShift APIs for Data Protection (OADP) backup and restore utility.
 - **Error:** When you check the status of the OADP backup utility after using it to backup a cluster where Discovery is installed, a **Phase: PartiallyFailed** message is displayed. One or more Discovery components are included in the **Failed** list.
 - **Cause:** Discovery cannot be backed up and restored by using the OADP backup and restore utility. When the Discovery service is present, and an administrator backs up an entire IBM Cloud Pak for Data instance, a status message is displayed that indicates a partial failure. This status is displayed because the persistent volume claims (PVCs) for Discovery are not backed up. However, the message does not impact the back up of the rest of the services.
 - **Solution:** No action is required to resolve the status message. You can remove the persistent volume claims that are

associated with the Discovery service separately. After using the scripts to back up your Discovery service data, you can follow the step that is documented in the uninstall instructions for the Discovery service to delete the PVCs. For more information about how to remove the PVC associated with Discovery, see [Uninstalling the Discovery service](#).

4.0.8, 27 April 2022

- The wd-discovery-multi-tenant-migration job fails if anyone besides a system administrator performs the upgrade.
 - Error:** When you upgrade with a user ID other than admin, the migration job fails.
 - Cause:** The migration script assumes that the script is run by a user with the admin user ID.
 - Solution:** Apply a patch that allows the migration to be successful. Complete the following steps:
 - From the Cloud Pak for Data web client, get the user ID of the owner of the instance that you want to upgrade.
 - Download the `wd-migration-uid-patch.zip` patch file from the [Watson Developer Cloud GitHub](#) repository.
 - Extract the `wd-migration-uid-patch.yaml` file from the archive file, and then open it in a text editor.
 - Replace the `<user_id>` variable with the user ID of the owner of the instance that you want to upgrade.
 - Run the following command in a terminal that is logged in to the cluster:

```
oc create -f wd-migration-uid-patch.yaml
```

- Delete the previous migration job by using the following command:

```
oc delete job wd-discovery-multi-tenant-migration
```

After the job is deleted, the migration job restarts and the migration resumes.

The issue is fixed with the 4.0.9 release.

- Discovery generates a partial failure status message for the IBM Cloud Pak for Data OpenShift® APIs for Data Protection (OADP) backup and restore utility.
 - Error:** When you check the status of the OADP backup utility after using it to backup a cluster where Discovery is installed, a **Phase: PartiallyFailed** message is displayed. One or more Discovery components are included in the **Failed** list.
 - Cause:** Discovery cannot be backed up and restored by using the OADP backup and restore utility. When the Discovery service is present, and an administrator backs up an entire IBM Cloud Pak for Data instance, a status message is displayed that indicates a partial failure. This status is displayed because the persistent volume claims (PVCs) for Discovery are not backed up. However, the message does not impact the back up of the rest of the services.
 - Solution:** No action is required to resolve the status message. You can remove the persistent volume claims that are associated with the Discovery service separately. After using the scripts to back up your Discovery service data, you can follow the step that is documented in the uninstall instructions for the Discovery service to delete the PVCs. For more information about how to remove the PVC associated with Discovery, see [Uninstalling the Discovery service](#).

4.0.7, 30 March 2022

- Discovery generates an error in the IBM Cloud Pak for Data OpenShift® APIs for Data Protection (OADP) backup and restore utility.
 - Error:** The utility does not complete successfully and the following message is written to the log:
`preBackupViaConfigHookRule on backupconfig/watson-discovery in namespace cpd (status=error)`
 - Cause:** Discovery cannot be backed up and restored by using the OADP backup and restore utility. When the Discovery service is present, and an administrator attempts to backup an entire IBM Cloud Pak for Data instance, Discovery prevents the utility from completing successfully.
 - Solution:** Apply a patch that stops Discovery from preventing the utility from completing successfully.
 - Download the `wd-aux-br-patch.zip` file from the [Watson Developer Cloud Github](#) repository.
 - Extract the `wd-aux-br-patch.yaml` file from the ZIP file.
 - Run the following command in a terminal that is logged in to the cluster:

```
$ oc create -f wd-aux-br-patch.yaml
```

The issue is fixed with the 4.0.8 release. (You still cannot back up the Discovery service by using the OADP utility, but the OADP utility can back up other services when Discovery is installed.)

- Deployed** status of resources fluctuates after the 4.0.7 upgrade is completed.

- **Error:** When you check the status by submitting the `oc get WatsonDiscovery` command, the ready status of the resources toggles between showing `23/23` and `20/23` components as being ready for use.
- **Cause:** The readiness state of the resources is not reported consistently after a migration.
- **Solution:** Typically, the instance is ready for use despite the ready state instability. To manually refresh the status information, run the following commands in a terminal that is logged in to the cluster:

```
$ oc proxy &
curl -ksS -X PATCH -H "Accept: application/json, */*" -H "Content-Type: application/merge-patch+json"
http://127.0.0.1:8001/apis/discovery.watson.ibm.com/v1/namespaces/<namespace>/watsondiscoveries/wd/status
--data '{"status": null}'
```

This issue is fixed with the 4.0.8 release.

- The wd-discovery-multi-tenant-migration job fails if anyone besides a system administrator performs the upgrade.
 - **Error:** When you upgrade with a user ID other than admin, the migration job fails.
 - **Cause:** The migration script assumes that the script is run by a user with the admin user ID.
 - **Solution:** Apply a patch that allows the migration to be successful. Complete the following steps:
 1. From the Cloud Pak for Data web client, get the user ID of the owner of the instance that you want to upgrade.
 2. Download the `wd-migration-uid-patch.zip` patch file from the [Watson Developer Cloud GitHub](#) repository.
 3. Extract the `wd-migration-uid-patch.yaml` file from the archive file, and then open it in a text editor.
 4. Replace the `<user_id>` variable with the user ID of the owner of the instance that you want to upgrade.
 5. Run the following command in a terminal that is logged in to the cluster:

```
oc create -f wd-migration-uid-patch.yaml
```

6. Delete the previous migration job by using the following command:

```
oc delete job wd-discovery-multi-tenant-migration
```

After the job is deleted, the migration job restarts and the migration resumes.

The issue is fixed with the 4.0.9 release.

4.0.6, 1 March 2022

- Upgrade to 4.0.6 fails if no Discovery instance is provisioned in the existing cluster before you begin the upgrade process.
 - **Error:** The 4.0.6 upgrade process assumes that a Discovery instance is provisioned in the existing cluster. For example, if you are upgrading from 4.0.5 to 4.0.6, you must have an instance provisioned in the 4.0.5 cluster before you begin the migration.
 - **Cause:** The current code returns an error when no instance exists because it cannot find a document index to migrate.
 - **Solution:** Verify that an instance of Discovery has been provisioned in the existing IBM Cloud Pak for Data cluster before you start the upgrade to 4.0.6. If you tried to upgrade to 4.0.6, but no instances were provisioned and the migration failed, remove the existing installation and install 4.0.6 from scratch.
- **Deployed** status of resources fluctuates after the 4.0.6 upgrade is completed.
 - **Error:** When you check the status by submitting the `oc get WatsonDiscovery` command, the ready status of the resources toggles between showing `23/23` and `20/23` components as being ready for use.
 - **Cause:** The readiness state of the resources is not reported consistently after a migration.
 - **Solution:** Typically, the instance is ready for use despite the ready state instability. The ready state settles after approximately 5 hours. You can wait for the readiness state to consistently show `23/23` or you can manually refresh the status information by running the following commands in a terminal that is logged into the cluster:

```
$ oc proxy &
curl -ksS -X PATCH -H "Accept: application/json, */*" -H "Content-Type: application/merge-patch+json"
http://127.0.0.1:8001/apis/discovery.watson.ibm.com/v1/namespaces/<namespace>/watsondiscoveries/wd/status
--data '{"status": null}'
```

This issue is fixed with the 4.0.8 release.

- Discovery generates an error in the IBM Cloud Pak for Data OpenShift® APIs for Data Protection (OADP) backup and restore utility.

- **Error:** The utility does not complete successfully and the following message is written to the log:
`preBackupViaConfigHookRule on backupconfig/watson-discovery in namespace cpd (status=error)`
- **Cause:** Discovery cannot be backed up and restored by using the OADP backup and restore utility. When the Discovery service is present, and an administrator attempts to backup an entire IBM Cloud Pak for Data instance, Discovery prevents the utility from completing successfully.
- **Solution:** Apply a patch that stops Discovery from preventing the utility from completing successfully.
 1. Download the `wd-aux-br-patch.zip` file from the [Watson Developer Cloud Github](#) repository.
 2. Extract the `wd-aux-br-patch.yaml` file from the ZIP file.
 3. Run the following command in a terminal that is logged in to the cluster:

```
$ oc create -f wd-aux-br-patch.yaml
```

This issue was fixed with the 4.0.8 release. (You still cannot back up the Discovery service by using the OADP utility, but the OADP utility can back up other services when Discovery is installed.)

- The wd-discovery-multi-tenant-migration job fails if anyone besides a system administrator performs the upgrade.
 - **Error:** When you upgrade with a user ID other than admin, the migration job fails.
 - **Cause:** The migration script assumes that the script is run by a user with the admin user ID.
 - **Solution:** Apply a patch that allows the migration to be successful. Complete the following steps:
 1. From the Cloud Pak for Data web client, get the user ID of the owner of the instance that you want to upgrade.
 2. Download the `wd-migration-uid-patch.zip` patch file from the [Watson Developer Cloud GitHub](#) repository.
 3. Extract the `wd-migration-uid-patch.yaml` file from the archive file, and then open it in a text editor.
 4. Replace the `<user_id>` variable with the user ID of the owner of the instance that you want to upgrade.
 5. Run the following command in a terminal that is logged in to the cluster:

```
oc create -f wd-migration-uid-patch.yaml
```

6. Delete the previous migration job by using the following command:

```
oc delete job wd-discovery-multi-tenant-migration
```

After the job is deleted, the migration job restarts and the migration resumes.

The issue is fixed with the 4.0.9 release.

4.0.5, 26 January 2022

- Discovery generates an error in the IBM Cloud Pak for Data OpenShift® APIs for Data Protection (OADP) backup and restore utility.
 - **Error:** The utility does not complete successfully and the following message is written to the log:
`preBackupViaConfigHookRule on backupconfig/watson-discovery in namespace cpd (status=error)`
 - **Cause:** Discovery cannot be backed up and restored by using the OADP backup and restore utility. When the Discovery service is present, and an administrator attempts to backup an entire IBM Cloud Pak for Data instance, Discovery prevents the utility from completing successfully.
 - **Solution:** Apply a patch that stops Discovery from preventing the utility from completing successfully.
 1. Download the `wd-aux-br-patch.zip` file from the [Watson Developer Cloud Github](#) repository.
 2. Extract the `wd-aux-br-patch.yaml` file from the ZIP file.
 3. Run the following command in a terminal that is logged in to the cluster:

```
$ oc create -f wd-aux-br-patch.yaml
```

This issue was fixed with the 4.0.8 release. (You still cannot back up the Discovery service by using the OADP utility, but the OADP utility can back up other services when Discovery is installed.)

4.0.4, 20 December 2021

- Discovery generates an error in the IBM Cloud Pak for Data OpenShift® APIs for Data Protection (OADP) backup and restore utility.

- **Error:** The utility does not complete successfully and the following message is written to the log:
preBackupViaConfigHookRule on backupconfig/watson-discovery in namespace cpd (status=error)
- **Cause:** Discovery cannot be backed up and restored by using the OADP backup and restore utility. When the Discovery service is present, and an administrator attempts to backup an entire IBM Cloud Pak for Data instance, Discovery prevents the utility from completing successfully.
- **Solution:** Apply a patch that stops Discovery from preventing the utility from completing successfully.

1. Download the **wd-aux-br-patch.zip** file from the [Watson Developer Cloud Github](#) repository.
2. Extract the **wd-aux-br-patch.yaml** file from the ZIP file.
3. Run the following command in a terminal that is logged in to the cluster:

```
$ oc create -f wd-aux-br-patch.yaml
```

This issue was fixed with the 4.0.8 release. (You still cannot back up the Discovery service by using the OADP utility, but the OADP utility can back up other services when Discovery is installed.)

4.0.3, 18 November 2021

- The guided tours are not available in this release.
 - Discovery generates an error in the IBM Cloud Pak for Data OpenShift® APIs for Data Protection (OADP) backup and restore utility.
 - **Error:** The utility does not complete successfully and the following message is written to the log:
preBackupViaConfigHookRule on backupconfig/watson-discovery in namespace cpd (status=error)
 - **Cause:** Discovery cannot be backed up and restored by using the OADP backup and restore utility. When the Discovery service is present, and an administrator attempts to backup an entire IBM Cloud Pak for Data instance, Discovery prevents the utility from completing successfully.
 - **Solution:** Apply a patch that stops Discovery from preventing the utility from completing successfully.
1. Download the **wd-aux-br-patch.zip** file from the [Watson Developer Cloud Github](#) repository.
 2. Extract the **wd-aux-br-patch.yaml** file from the ZIP file.
 3. Run the following command in a terminal that is logged in to the cluster:

```
$ oc create -f wd-aux-br-patch.yaml
```

This issue was fixed with the 4.0.8 release. (You still cannot back up the Discovery service by using the OADP utility, but the OADP utility can back up other services when Discovery is installed.)

4.0.0, 13 July 2021

- Machine learning model enrichments that you apply by using the Analyze API can fail.
 - **Error:** **[WKSML_MODEL_NAME]: Enrichment of a document failed**
 - **Cause:** There is a known issue in Watson Knowledge Studio that can cause a timeout in enrichment processing.
 - **Solution:** When you use the Analyze API to apply a Watson Knowledge Studio model enrichment to a collection, keep the size of the input document under 50 KB.

2.2.1 issues that were fixed in subsequent releases

- [Fixed in version 4] If you add an IBM Watson® Knowledge Studio machine learning enrichment to a collection, the ingestion process might run very slowly but will eventually complete. If ingestion processes slowly, you might see the following error message in **Warnings and errors**:

```
$ [WKSML_MODEL_NAME]: Document analysis timed out
```

For additional timeout details, you can check your Knowledge Studio machine learning logs, which might look similar to the following:

```
{
  "message": "Analysis failed due to:
org.apache.uima.analysis_engine.AnalysisEngineProcessException
at c.i.n.b.SIREAnnotator.process(_:454)
```

```
...",
  "level": "SEVERE",
}
```

Documents that time out during processing are indexed without Knowledge Studio enrichment results.

2.2.1, 26 February 2021

- Deployment timing issue:
 - **Error:** After installing patch 7, when you try to provision a service instance, a **404 Not Found** error is displayed. The following message might be logged for the **nginx** pods: `open()` `"/usr/local/openresty/nginx/html/watson/common/discovery/auth"` failed (2: No such file or directory)
 - **Solution:** Restart the **zen-watcher** pod.
- If you perform an air-gapped installation that pulls container images from an external container registry, you might experience the following issue:
 - **Error:** Some Discovery pods might report an **ImagePullBackoff** error.
 - **Cause:** The wrong image pull secret is being used.
 - **Solution:** Complete the following steps during the installation:
 - Start installing Watson Discovery.
 - After `watson-discovery-operator` module completes, check if a `WatsonDiscovery` custom resource is created by running the following command:

```
$ oc get WatsonDiscovery wd
```

- After the custom resource is created, run the following commands to point the correct image pull secret to pull images from the external registry:

```
pull_secret=$(oc get secrets | grep 'docker-pull-*-watson-discovery-registry-registry' | cut -d '' -f 1)
cat << EOS > discovery-patch.yaml
spec:
  shared:
    imagePullSecret: $pull_secret
EOS
oc patch wd wd --type=merge --patch "$(cat discovery-patch.yaml)"
```

- If the **RabbitMQ** pods are still in `ImagePullBackoff` state, remove the `RabbitMQ` CR to enable the `rabbitmq-operator` to re-create the RabbitMQ clusters. You can use the following command:

```
$ oc delete IbmRabbitmq wd-rabbitmq
```

- In IBM Watson® Discovery, the **Content Mining** project only supports one collection per project. If you create more than one **Content Mining** collection, you might experience errors. If you experience errors, delete additional **Content Mining** collections so that each **Content Mining** project has only one associated collection.
- If you are preparing your Discovery for Cloud Pak for Data clusters for an in-place upgrade of your instance from 2.2.0 to 2.2.1, occasionally, the `cpd-cli adm` command fails, showing the following error message: `Error from server (UnsupportedMediaType): error when applying patch`. If you receive this error message, enter `oc delete scc cpd-zensys-scc cpd-user-scc cpd-noperm-scc edb-operator-scc admin-discovery-scc` to delete the related resources, and re-enter the `cpd-cli adm` command.
- If you are upgrading your Discovery for Cloud Pak for Data instance from 2.2.0 to 2.2.1, occasionally, the `cpd-cli upgrade` command completes before rolling updates complete. For information about verifying that your upgrade completed successfully, see [Verifying that your upgrade completed successfully](#).
- Model-train images are not updated after upgrading from Discovery 2.2.0 to 2.2.1. To work around this issue, delete the deployments that the model-train operator creates, and wait for the operator to recreate the deployments. Enter the following command to delete the deployments:

```
$ oc delete deploy -l 'app.kubernetes.io/managed-by=ibm-modeltrain'
```

After you run this command, the model-train operator creates new deployments.

- If you upgrade Discovery for Cloud Pak for Data from 2.2.0 to 2.2.1, you might receive the following error message:

```
$ [ERROR] [2021-03-04 05:12:44-0657] Exiting due to error (Storage class is immutable. Module ibm-watson-gateway-operator x86_64 from Assembly portworx-shared-gp3 was installed with ibm-watson-gateway-operator x86_64, but new install/upgrade command is requesting portworx-db-gp3-sc. If you installed the assembly with a different storage class, please upgrade it individually.). Please check /ibm/cpd-cli-workspace/logs/CPD-2021-03-04T05-12-04.log for details
[ERROR] 2021-03-04T05:12:44.659615Z Execution error: exit status 1
```

This error message is generated because the storage class that was used for installation is different than the one that was used during the upgrade. This discrepancy results from a different add-on installing the dependency operators because the storage class dependency operators of the different add-on were recorded as the ones that were used for installation. To work around this issue, you must upgrade the following subassemblies individually:

- Upgrade the Watson gateway operator:

```
$ ./cpd-cli upgrade \
--repo ./repo.yaml \
--assembly ibm-watson-gateway-operator \
--arch Cluster_architecture \
--namespace <Project> \
--transfer-image-to <Registry_location> \
--cluster-pull-prefix <Registry_from_cluster> \
--ask-pull-registry-credentials \
--ask-push-registry-credentials
```

- Upgrade Minio operator:

```
$ ./cpd-cli upgrade \
--repo ./repo.yaml \
--assembly ibm-minio-operator \
--namespace <Project> \
--transfer-image-to <Registry_location> \
--cluster-pull-prefix <Registry_from_cluster> \
--ask-pull-registry-credentials \
--ask-push-registry-credentials
```

- Upgrade RabbitMQ operator:

```
$ ./cpd-cli upgrade \
--repo ./repo.yaml \
--assembly ibm-rabbitmq-operator \
--namespace <Project> \
--transfer-image-to <Registry_location> \
--cluster-pull-prefix <Registry_from_cluster> \
--ask-pull-registry-credentials \
--ask-push-registry-credentials
```

- Upgrade etcd operator:

```
$ ./cpd-cli upgrade \
--repo ./repo.yaml \
--assembly ibm-etcd-operator \
--namespace <Project> \
--transfer-image-to <Registry_location> \
--cluster-pull-prefix <Registry_from_cluster> \
--ask-pull-registry-credentials \
--ask-push-registry-credentials
```

- Upgrade model train classic operator:

```
$ ./cpd-cli upgrade \
--repo ./repo.yaml \
--assembly modeltrain-classic \
--arch Cluster_architecture \
--namespace <Project> \
--transfer-image-to <Registry_location> \
--cluster-pull-prefix <Registry_from_cluster> \
--ask-pull-registry-credentials \
--ask-push-registry-credentials
```

- Upgrade Elasticsearch operator:

```
$ ./cpd-cli upgrade \
--repo ./repo.yaml \
```

```
--assembly ibm-cloudpakopen-elasticsearch-operator \
--namespace <Project> \
--transfer-image-to <Registry_location> \
--cluster-pull-prefix <Registry_from_cluster> \
--ask-pull-registry-credentials \
--ask-push-registry-credentials
```

where `<Project>` is the namespace where your Discovery for Cloud Pak for Data 2.2.0 instance is installed, where `<Registry_location>` is the location of the images that you pushed to the registry server, and where `<Registry_from_cluster>` is the location from which pods on the cluster can pull images.

- When you install on IBM Cloud Pak for Data 3.5, you might encounter the following issue:
 - Error:** If you try to provision the Discovery service on a cluster where Planning Analytics is running, some of the Discovery pods don't start and installation fails. The logs for the pod show messages such as, `java.lang.NumberFormatException: For input string`.
 - Cause:** An environment variable named `COUCHDB_PORT` is added to the Kubernetes cluster by the couchdb service that is installed with Planning Analytics. Discovery does not use couchdb, and therefore does not specify a value for this environment variable. However, some pods attempt to parse the variable, which results in the error.
 - Solution:** [Install patch cpd-watson-discovery-2.2.1-patch-1](#), which fixes this issue.

Also, see the issues in all previous releases.

2.2, 8 December 2020

- When a small CSV file (generally a CSV with 99 lines or fewer) is uploaded, the header and/or first row may not be ingested correctly. If this happens, in the tooling, navigate to the CSV Settings tab and update the settings. After reprocessing, navigate to the **Manage fields** tab and update the field types if needed.
- If you have set up your collections using a custom crawler built with the [IBM Cloud Pak for Data custom connector](#), and then remove the custom crawler deployment, the Processing Settings page will not display the crawler configuration. This is because the underlying crawler is not available. To work around this issue, confirm that the custom crawler is deployed when there are collections using it.
- When using a [IBM Cloud Pak for Data custom connector](#) with Discovery for IBM Cloud Pak for Data 2.2, the script `scripts/manage_custom_crawler.sh` used to deploy and remove the deployment of the custom crawler fails. To work around this issue, replace line 37 `podname="gateway"` with `podname="wd-discovery-gateway"` in `scripts/manage_custom_crawler.sh`, and then rerun the deploy command.
- When you create a custom enrichment in the tooling, you must choose a field the enrichment should be applied to and click **Apply**. If no field is selected, then the **Apply and reprocess** button will be disabled for enrichments changes until the new enrichment has a field.
- If you apply the [Contracts](#) enrichment or the [Understanding tables](#) enrichment to a collection, you might receive the following error message when that collection is ingesting documents: `The number of nested documents has exceeded the allowed limit of [X]`. Contact the [IBM Support Center](#) to adjust the limit.
- When text is enriched with a custom dictionary, the output of `entities.type` should be the full facet path for the Dictionary enrichment. However, in this release, the full facet path will not be displayed. To work around this, reprocess the collection. For example, if the facet path is `sample1.sample2`, it will look like this before reprocessing:

```
{
  "result" : {
    "enriched_text" : [
      {
        "entities" : [
          {
            "text" : "capital",
            "type" : "sample2",
            ...
            "model_name" : "Dictionary:.sample1.sample2"
            ...
          }
        ]
      }
    ]
  }
}
```

And this after:

```
{
  "result" : {
    "enriched_text" : [
      {
        "entities" : [
          {
            "text" : "capital",
            "type" : "sample1.sample2",
            ...
            "model_name" : "Dictionary:.sample1.sample2"
          }
        ]
      }
    ]
  }
}
```

- When a CSV file is uploaded with the converter settings set to `auto_detection=true`, the **CSV settings** tab in the tooling will display the incorrect settings. If you update the settings on the **CSV settings** tab, `auto_detection` will no longer be set to `true`.
- In Office documents ('.doc', '.docx', '.odf', '.xls', '.xlsx', '.ods', '.ppt', '.pptx', '.odp') converted using a Smart Document Understanding (SDU) custom model, the `publicationdate` may not display in `extracted_metadata` field in the JSON response. It will instead appear in the `html` field of the JSON response. The `publicationdate` in the `html` field will be the date the document was ingested and not the document's original publication date.
- The Analyze API uses an in-memory cache to hold the enrichment models associated with the collection used to run the documents. If the collection contains many large enrichments or multiple of these collections are used at the same time, the cache may run out of memory. When this happens, the Analyze API returns null results (see example) and the stateless api rest proxy will display this message in its log: `RESOURCE_EXHAUSTED: stateless.Analysis/analyze: RESOURCE_EXHAUSTED`.

```
{
  "result": null,
  "notices": null
}
```

To work around this issue:

- Review the enrichments used in the collection and remove those that are not necessary for your application. In particular, remove the **Part of Speech** enrichment.
 - Reduce the number of collections used concurrently with the Analyze API.
 - Increase the cache memory:
 - Increase the memory limit of `container model-runtime` in deployment `core-discovery-stateless-api-model-runtime` to **10** GB or more
 - Edit the environment variable `CAPACITY_MB` in deployment `core-discovery-stateless-api-model-runtime`, set it to **10240** or more
- If the model runtime container is restarted but the model mesh runtime container is not, the Analyze API can run into problems.
 - Error:** The Analyze API call returns 500 error on a specific collection and the log contains the following entry:

```
"message": "error occurred in analyzer
java.lang.NullPointerException
at c.i.e.a.a.s.r.ModelManager$2.analyze(ModelManager.java:112)
```

- Cause:** The model runtime container and model mesh runtime container are out of sync.
- Solution:** Delete the `wd-stateless-api-model-runtime` pods to restart both the model mesh and model runtime containers.

Also see the issues identified in all previous releases.

2.1.4, 2 September 2020:

- When configuring a Web crawl using FORM authentication, if you specify a URL without a trailing slash, for example: `https://webcrawlurl.com`, the web crawl will only crawl the login page. To work around this issue, add a trailing slash to the URL, for example: `https://webcrawlurl.com/`.
- The [Guided Tours](#) do not run on Firefox. For the list of other supported browsers, see [Browser support](#).
- Ingesting documents into a collection that uses a custom [Advanced Rules model](#) built in Watson Knowledge Studio may fail if multiple extractors in the model internally use the same names for one or more output views.
- If you delete a large number of documents, then immediately ingest a large number of documents, it may take longer for all the documents to become available.
- The [Classifier](#) enrichment doesn't work when FIPS (Federal Information Processing Standards) is enabled.

Also see the issues identified in all previous releases.

2.1.4 issues that were fixed in subsequent releases

- [Fixed in version 2.2] In the deployed Content Mining application, if you include the tilde (~) symbol in a search query to enable fuzzy matching or include an asterisk (*) symbol to represent a wildcard, the search customizations function properly, but the matching string is not highlighted in the query result.
- [Fixed in version 2.2] A conversion error may occur when the `Include in index` field on the **Manage fields** tab in the tooling is changed. The document will not be indexed if this error occurs. To work around the issue:
 - `oc edit sts core-discovery-converter`
 - Edit between `containers` and `- name: INGESTION_POD_NAME` as follows:

```

containers:
  - command:
    - bash
    - -C
    - |
      FILE=/opt/ibm/wex/zing/bin/converter.sh &&
      sed -i "/choreo_2.11-9.1.1.jar/d" $FILE &&
      sed -i "/disco-doc-conversion-commons_2.11-1.0.4.jar/d" $FILE &&
      sed -i "/jackson-module-scala_2.11-2.10.4.jar/d" $FILE &&
      sed -i "/macro-compat_2.11-1.1.1.jar/d" $FILE &&
      sed -i "/pureconfig-core_2.11-0.12.2.jar/d" $FILE &&
      sed -i "/pureconfig-generic-base_2.11-0.12.2.jar/d" $FILE &&
      sed -i "/pureconfig-generic_2.11-0.12.2.jar/d" $FILE &&
      sed -i "/pureconfig-macros_2.11-0.12.2.jar/d" $FILE &&
      sed -i "/pureconfig_2.11-0.12.2.jar/d" $FILE &&
      sed -i "/scala-guice_2.11-4.1.1.jar/d" $FILE &&
      sed -i "/scala-logging_2.11-3.7.2.jar/d" $FILE &&
      sed -i "/scalactic_2.11-3.0.5.jar/d" $FILE &&
      sed -i "/scalaj-http_2.11-2.3.0.jar/d" $FILE &&
      sed -i "/service-commons_2.11-22.1.0.jar/d" $FILE &&
      sed -i "/shapeless_2.11-2.3.3.jar/d" $FILE &&
      /opt/ibm/wex/zing/bin/entrypoint.sh /opt/ibm/wex/zing/bin/controller.sh
    env:
      - name: INGESTION POD NAME

```

Added lines from `- command:` to `/opt/ibm/wex/zing/bin/entrypoint.sh` `/opt/ibm/wex/zing/bin/controller.sh`
and removed `-` before `env:`

- Save the changes. It will restart the `converter` pod.

2.1.3, 19 June 2020:

- Entity Subtypes** in IBM Watson® Knowledge Studio Machine Learning models are not supported in Discovery for Cloud Pak for Data 2.1.3 or later. For instructions on converting existing models, contact the [Support center](#).
- You cannot upload CSV files that include a space in the file name (for example: `file 1.csv`) to a Content Mining project. Rename the file to work around the issue.
- When performing Project level relevancy training, if you have multiple collections, and two or more of those collections contains a duplicate `document_id`, then project level relevancy training will fail. Example of duplicate `document_ids`: `Collection A` contains a document with the id of `1234`, and `Collection B` also contains a document with the id of `1234`.
- Only the first facet using a field with the prefix `extracted_metadata` is saved correctly after creation. Others with that prefix will appear but after a screen refresh will be gone. This only happens once per project, so the workaround is to refresh and add the facet again.
- IBM Cloud Pak for Data During installation on IBM Cloud Pak® for Data 2.5.0.0, some Kubernetes Jobs may incorrectly report their status as `OOMKilled`, causing the install to timeout. To resolve this, once a Job returns `OOMKilled` verify the logs of the Pod associated with that Job. There should be no obvious error messages in the logs and the resources are reported in the logs as created. Manually verify these resources exist in the namespace and then delete the Job. This will cause the install to continue.
- Some documents may show two `html` fields when applying an enrichment. Both `html` fields shown are the same and operate as such.
- When creating a data source in Firefox, you may not see the entire list of options, including the **More processing settings** settings. To work around the issue, zoom out, increase the browser height, or use another supported browser.
- When customizing the display of search results, the changes made sometimes do not save after clicking the `Apply` button. To work around this issue, refresh the browser and try to make the changes again.
- When setting up a data source or web crawler for your collection, if you enter an incorrect configuration, then try to update it on the **Processing settings** page, the data source update or crawl may not start when you click the `Apply changes and reprocess` button. You can confirm this issue by opening the **Activity** page for your collection to see if processing has started. If you see that processing has not started for your data source, click the `Recrawl` button, then the `Apply changes and reprocess` button. If you see that processing has not started for your web crawl, click the `Stop` button, then the `Recrawl` button.
- IBM Cloud Pak for Data When running Helm tests on the `core` deployment using `helm test core`, the `core-discovery-api-post-install-test` will return a `FAILED` status. This is due to a bug within the `test` pod's image. The test result can be ignored as the failure is not related to anything within the deployment.
- By default, Optical Character Recognition (OCR) is set to `off` when you create any **Project type** with the tooling. However, if you create a Project using the API, OCR is set to `on`. To work around this issue, open the Tooling and change the **Project setting** to `off`.
- When Optical Character Recognition (OCR) is set to `on` for a Collection AND no trained Smart Document Understanding (SDU)

model is applied, PNG, TIFF, and JPG files will not be processed for text recognition. Images embedded in PDF, Word, PowerPoint, and Excel documents will not be processed - only the non-image portion of these documents will be processed for text recognition. To work around this issue, import or train an SDU model and reprocess the collection. This will allow text to be extracted from the images.

- After you create a Search Skill in Watson Assistant and are directed to the Watson Discovery tooling, the screen is blank. This happens because the URL is missing the Discovery instance ID. To work around this issue:

1. From the IBM Cloud Pak for Data web client menu, choose **My Instances**. For example: <https://mycluster.com/zen/#/myInstances>.
2. Select the Discovery instance you are using and click **Launch Tool**.
3. Once the tooling is loaded, the URL should have the following structure: <https://mycluster.com/discovery/core/instances/00000000-0000-0000-0001-597165341876/projects>
4. Copy the entire path, excluding `/projects`. For example: <https://mycluster.com/discovery/core/instances/00000000-0000-0000-0001-597165341876>
5. Go back to the browser tab that is displaying the blank Discovery screen. That URL structure will look like this: https://mycluster.com/discovery/core/collections/new?redirect_uri=...
6. Replace <https://mycluster.com/discovery/core> with the URL you copied previously, so the new URL should look like this: https://mycluster.com/discovery/core/instances/00000000-0000-0000-0001-597165341876/collections/new?redirect_uri=...
7. Press enter to open updated URL. You should now be on the Watson Discovery **Manage collections** page.

Also see the issues identified in all previous releases.

2.1.2, 31 March 2020

- When using passage retrieval with Korean, Polish, Japanese, Slovak or Chinese you may encounter much slower response times in this version. To resolve this, either disable passage retrieval, or upload a custom stopword list with words that are common in your documents (for example, prepositions and pronouns). See [Defining stopwords](#) for example stopword lists in several languages. Also see [Stopwords ISO](#) on GitHub.
- [Update: fixed in version 2.1.3] In versions 2.1.2, 2.1.1, and 2.1.0, PNG, TIFF, and JPG individual image files are not scanned, and no text is extracted from those files. PNG, TIFF, and JPEG images embedded in PDF, Word, PowerPoint, and Excel files are also not scanned, and no text is extracted from those image files.
- Smart Document Understanding does not support `.doc`, `.docx`, `.odf`, `.xls`, `.xlsx`, `.ods`, `.ppt`, `.pptx`, and `.odp` conversion when FIPS (Federal Information Processing Standards) is enabled.
- In a Content Mining application, any document flags set will disappear if the index is rebuilt for that collection.
- Beginning with the 2.1.2 release, uploading and managing relevancy training data using the v1 APIs will not train a relevancy training model. The v1 APIs have been superseded by the [Projects relevancy training v2 APIs](#). If your training data needs to be preserved, it can be listed using the v1 API, then added to a project with the v2 API.
- Multiple [Regular expressions](#) cannot be applied to a collection at the same time.
- IBM Cloud Pak for Data There were two small changes to the installation instructions README included with the download of IBM Watson® Discovery for IBM Cloud Pak® for Data. For the updated version of the README, see the [Discovery Helm chart README.md](#).
 - A change to the description of the `--cluster-pull-prefix PREFIX` argument.
 - The language extension pack name has been updated from `ibm-watson-discovery-pack1-2.1.2.tar.xz` to `ibm-wat-dis-pack1-prod-2.1.2.tar.xz`.

Also see the issues identified in all previous releases.

2.1.1, 24 January 2020

- When creating a [dictionary](#), suggested dictionary terms are normalized to lowercase by default (for example, Watson Assistant will be normalized to watson assistant). To ensure matching on uppercase terms, they should be explicitly included as part of the **Other terms** list or as the **Base term**.
- When backing up and restoring data, training data does not restore successfully. If the documents in your collection were added by crawl using a connector or web crawl, your training data can be separately retrieved for backup from an existing project and uploaded to a new restored project. For more information, see [List training queries](#) and [Create training queries](#)) in the API reference.
- When crawling SharePoint Online or SharePoint OnPrem documents, JSON documents may not be indexed correctly and the **title** returned may be **errored**. This is because SharePoint web services use the **ows_FileRef** property to retrieve JSON files, which will return an error page. To fix this issue, contact your SharePoint Administrator and Microsoft Support.
- If you migrate a collection created in version 2.0.1 to either version 2.1.0 or 2.1.1, that collection will not have a **Project type** assigned and the collection will not be available to be queried. To assign a **Project type**, open the **Projects** page by selecting **My Projects**. Name your project and choose one of the Project types: **Document Retrieval**, **Conversational Search**, **Content Mining**, or **Custom**.

Also see the issues identified in all previous releases.

2.1.1 issues that were fixed in subsequent releases

- [Fixed in version 2.1.2] When installing Discovery for Cloud Pak for Data on OpenShift, the `ranker-rest` service might intermittently fail to startup, due to an incompatible jar in the `classpath`. To fix the issue:

1. Open the `ranker-rest` editor with this command: `kubectl edit deployment {release-name}-{watson-discovery}-ranker-rest`
2. In the editor, search for the `ranker-rest image` (for example: `{docker-registry}/{namespace}/discovery-ranker-rest-service:20200113-150050-2-d1527c2`)
3. Add the following command below `{docker-registry}/{namespace}/discovery-ranker-rest-service:20200113-150050-2-d1527c2`:

```
$ command: ["/tini"]
args: ["-s", "-v", "--", "java", "-Dkaryon.ssl=true", "-Dkaryon.port=9081", "-Dkaryon.ssl.port=9090", "-Dkaryon.ssl.certificate=/opt/bluegoat/karyon/ssl/karyon-cert.pem", "-Dkaryon.ssl.privatekey=/opt/bluegoat/karyon/ssl/karyon-private-key.pem", "-Djavax.net.ssl.trustStore=/opt/bluegoat/karyon/ssl/keystore.jks", "-Djavax.net.ssl.keyStore=/opt/bluegoat/karyon/ssl/keystore.jks", "-Dlog4j.debug=false", "-Dlitelinks.threadcontexts=log4j_mdc", "-Dwatson.ssl.truststore.path=/opt/bluegoat/karyon/ssl/litelinks-truststore.jks", "-Dwatson.ssl.truststore.password=watson15qa", "-Dlitelinks.delay_client_close=false", "-Drxnetty.http.maxcontentlength=314572800", "-cp", "lib/logback-classic-1.2.3.jar:*.jar", "com.ibm.watson.raas.rest.Runner"]
```

2.1.0, 27 November 2019

- When you apply an enrichment to a collection, the enrichment language must match the collection language, or it will fail. The tooling displays all the collections, regardless of language.
- On the Manage Fields tab, you can edit system-generated fields. The following fields should not be edited by changing the field type or turning off indexing: `document_id`, `extracted_metadata`, `metadata`.
- When you delete a Collection and select the option `Don't delete underlying data`, any incomplete document ingestion crawls will continue running in the background, which will impact the new crawl start times, until the existing crawls are completed.
- IBM Cloud Pak for Data Discovery can fail to start up correctly due to components getting into a lock state. Manual database intervention may be needed to clear the lock. For more information on identifying and resolving this issue, see [Clearing a lock state](#).
- If you upload a document with the Upload Data function, delete that document, and then try to upload either the same document or another document with the same document ID, the upload will fail and the message `Error during creating a document` will be displayed.
- Documents that produce an `html` field when processed can not be used with relevancy training. `html` is produced for documents processed with Smart Document Understanding or Content Intelligence. The `html` field must be removed before relevancy training can complete successfully.
- If the `Part of Speech` enrichment is not turned on: Dynamic facets will not be created, Dictionary suggestions cannot be used, Content Miner "extracted facets" will not generate.
- [Update: fixed in version 2.1.1] Discovery for Content Intelligence and Table Understanding enrichments are configured out of the box to be applied on a field named `html`. When a user uploads a JSON document without a root-level field named `html`, these enrichments will not yield results in the index. To run the enrichments on this kind of JSON documents, users must re-configure the enrichments to run on an existing field (or fields) in the JSON document.
- When viewing the Content Miner deploy page, sometimes the full application URL is not displayed for copying. To fix, refresh the page.
- [Update: fixed in version 2.1.2] Deprovisioning a IBM Watson® Discovery for IBM Cloud Pak® for Data Instance will not delete the underlying data. Delete the collections and documents manually.
- [Update: fixed in version 2.1.3] On the Improvement tools panel, the enrichment `Sentiment of phrases` is listed, but is not currently available.
- In Content Mining projects, the `dates` fields may not be parsed properly for display in facets.
- The Dynamic facets toggle should not appear in Content Mining projects.
- A minimum of 50-100 documents should be ingested to see valid dynamic facets generated.
- If you click `Stop` to stop a crawler and the converter processes slowly or has errors, you might see a status of the crawler running.
- The total size limit of all non-HTML fields in uploaded and crawled documents is 1MB, which is equivalent to 1,048,576 bytes, and the total size limit of all HTML fields in these documents is 5MB. If you exceed either limit, you receive an error message stating `The document has fields/HTML fields that exceed the 1 MB/5 MB limit.`, and the document is not ingested. For assistance on increasing either size limit, contact the [IBM Support Center](#).

Also see the issues identified in all previous releases.

2.0.1, 30 August 2019

- After you create a Machine Learning enrichment using a IBM Watson® Knowledge Studio model, two identically named enrichments may display on the `Enrich fields` page. This will not affect the enrichments, but it is best to use only one of them to select and apply the enrichment to one or more fields.
- If a web crawl appears to be stuck processing at a fixed number of documents, and the message displayed on the `Logs` page is `The ingestion job <jobid> is terminated incorrectly`, contact IBM support for assistance restarting the crawl.
- If one or more of your collections is trained, the training data from one of those collection may display on the `Train` page of an untrained collection. Refresh the page to clear that training data.

- The following types of documents will not be processed if they do not have the proper file extension: .docx, .pptx, .xlsx.

Also see the issues identified in the previous release.

2.0.1 issues that were fixed in subsequent releases

- [Fixed in version 2.1.2] When you upload documents to a collection with existing documents, a **Documents uploaded!** message displays on the **Activity** page, but no further processing status displays until the number of documents increases.

General Availability (GA) release, 28 June 2019

- If you are working in the Discovery for Cloud Pak for Data tooling, and your IBM Cloud Pak® for Data session expires, you will receive a blank page. To return to the tooling, refresh the browser and log back in.
- All JSON files ingested into Discovery should include the .json file extension.
- When querying on the **collection_id** of a trained collection, the **training_status.notices** value may occasionally display as **0** instead of the correct value.
- Not all query limitations are enforced in this release. See [query limitations](#) for the complete list of banned fields.
- In JSON source documents, you should not duplicate the following system-generated fields: **document_id**, **parent_document_id**, **filename**, and **title**. This will cause the duplicate fields to nest within arrays and break certain features, such as ranker training.
- Do not include a root-level **metadata** property in your JSON documents. If you upload a JSON document that already contains a root-level **metadata** property, then the **metadata** property of the indexed document will be converted to an array in the index.
- Do not use metadata for column names in your CSV files. If you upload a CSV file that uses metadata for the column names in the header, then the **metadata** property of the indexed document will be converted to an array in the index.
- CSV files must use commas (,) or semicolons (;) as delimiters; other delimiters are not supported. If your CSV file includes values containing either commas or semicolons, you should surround those values in double quotation marks so they are not separated. If header rows are present, the values within them are processed in the same manner as values in all other rows. The last row of CSV files will not be processed if not followed by a CRLF (carriage return).
- Currently, unique collection names are not enforced. Using duplicate collection names is not recommended and should be avoided

Advanced development

Configuring a Cloud Pak for Data custom connector

Building a Cloud Pak for Data custom connector

Discovery provides connectors to many popular data sources, as described in [Configuring Cloud Pak for Data data sources](#). If you need to connect to a different data source, you can write and deploy a *custom connector*.

IBM Cloud Pak for Data **IBM Cloud Pak for Data only**



Note: This information applies only to installed deployments.



Note: Any custom code that is used with IBM Watson® Discovery is the responsibility of the developer and is not covered by IBM support.

Example code and configuration files for a basic custom connector are included.

Related topics:

- [Developing custom Cloud Pak for Data connector code](#)
- [Assembling, compiling, and packaging a custom Cloud Pak for Data connector](#)
- [Installing and uninstalling a custom Cloud Pak for Data connector](#)
- [Using a custom Cloud Pak for Data connector from the Discovery user interface](#)

Custom connector requirements

A custom connector is a component that uses the SDK and crawler framework that is documented here to connect to and crawl a specific data source. Custom connectors have the same general requirements as provided connectors. For more information, see [Data source requirements](#).

Before you implement a custom connector, you need to know the following information about the data source:

- The data source's network location (server name or address, including port, or URL, including port)
- The data source's authentication method and security credentials
- The path or paths on the data source that the connector needs to crawl
- The connection method or protocol that the data source supports

Designing a custom connector

A custom connector needs the following capabilities:

- Configuring a crawler.
 - Configuring all settings that are required to connect to the data source.
 - Discovering a **crawl space** on the data source. At least one crawl space is required.
- Crawling documents.
 - Crawling the documents on each data set.
 - Adding Access Control List (ACL) information to each document.
- Retrieving ACL information for the username that authenticates to the data source.

These capabilities can be implemented by using the interfaces and methods that are described in [Developing custom connector code](#).

Custom connector limitations

Observe the following notes and warnings when you implement a custom connector.

- Custom connectors do **not** support the following features:
 - Synchronization settings
 - Filtering documents based on user access at query time. (At crawl and index time, only documents that the current user has the right to access are returned.)
 - The **required** and **hidden** validation settings. They are ignored when the connector is displayed in Discovery
 - The use of `<condition />` tags in the definition file. These tags are currently ignored.
- When you use the example connector code in the current release, Discovery does not collapse and group authentication settings for the custom connector's properties. For example, even when the `{connector_name}_DATASOURCE_SETTINGS_USE_KEY_LABEL` toggle is set to **Off**, the user interface shows the fields for `{connector_name}_DATASOURCE_SETTINGS_KEY_LABEL` and

{connector_name}_DATATSOURCE_SETTINGS_PASSPHRASE_LABEL

- The `list` parameter type is not supported.
 - If a custom connector fails to connect to its source for any reason, it issues a generic error message such as `Failed to create connector` or `Timed out`, or a `500` HTTP error. Specific failure information is not currently provided.
- See the [Release notes](#) for more possible issues.

Developing custom Cloud Pak for Data connector code

The custom connector example includes a Java package named `com.ibm.es.ama.custom.crawler`. The package includes the following Java interfaces that you can use when you write your own custom connector.

IBM Cloud Pak for Data **IBM Cloud Pak for Data only**



Note: This information applies only to installed deployments.

Interfaces and Javadoc

The interfaces that are listed in this document are available in the JAR package file that is included with the custom connector compressed file. After you download and expand the `custom-crawler-docs.zip` file as described in [Downloading the custom-crawler-docs.zip file in Discovery 2.2.1 and later](#) and [Downloading the custom-crawler-docs.zip file in Discovery 2.2.0 and earlier](#), the interface JAR file is available as `wexlib/ama-zing-custom-crawler-{version_numbers}.jar` from the root level of the expanded compressed file. Javadoc for the JAR file is available as `wexlib/ama-zing-custom-crawler-{version_numbers}-javadoc.jar` at the same level.

Initialization interface

CustomCrawler

Use the `com.ibm.es.ama.custom.crawler.CustomCrawler` interface to start or stop a custom crawler or to crawl documents from a path. The interface has the following methods.

Method	Description
<code>init</code>	Start a custom crawler
<code>term</code>	Stop a custom crawler
<code>crawl</code>	Crawl documents from a specified path

CustomCrawler methods

Configuration interfaces

CustomCrawlerConfiguration

Use the `com.ibm.es.ama.custom.crawler.CustomCrawlerConfiguration` interface to validate the configuration and to discover available crawl spaces on the data source. The interface has the following methods.

Method	Description
<code>validate</code>	Validate configuration
<code>getFieldsFor</code>	List known fields and their types
<code>discoverySubspaces</code>	Discover crawl spaces on the data source

CustomCrawlerConfiguration methods

ConfigProvider

Use the `com.ibm.es.ama.custom.crawler.CustomCrawlerConfiguration.ConfigProvider` interface to map the settings of the data source and to list the crawl-space settings on the data source. The interface has the following methods:

Method	Description
<code>get</code>	Get a map of the settings in a section
<code>getCrawlSpaceSettings</code>	Get a list of crawl-space settings
ConfigProvider methods	

SubspaceConsumer

Use the `com.ibm.es.ama.custom.crawler.CustomCrawlerConfiguration.SubspaceConsumer` interface to add a path to a crawl space. The interface has the following method:

Method	Description
<code>add</code>	Add a path to the crawl space
SubspaceConsumer methods	

Crawler interface

RecordKeeper

Use the `com.ibm.es.ama.custom.crawler.CustomCrawler.RecordKeeper` interface to keep records of crawls and to publish crawled documents. The interface has the following methods:

Method	Description
<code>canContinue</code>	Boolean that lists whether the crawler can continue. The custom crawler must poll this value periodically and terminate if it returns <code>false</code> .
<code>check</code>	Get metadata fields from the last crawled document
<code>upsert</code>	Publish a document for further processing
<code>delete</code>	Delete a document
RecordKeeper methods	

Security interface

CustomCrawlerSecurityHandler

Use the `com.ibm.es.ama.custom.crawler.CustomCrawlerSecurityHandler` interface to implement security for your custom crawler. The interface has the following methods:

Method	Description
<code>term</code>	Terminate a security handler
<code>getUserAndGroups</code>	Get the ACLs of a given user
CustomCrawlerSecurityHandler methods	

⚠️ Important: When the `getUserAndGroups` logic of a connector is updated, it can take up to 10 minutes after the connector is redeployed for the change to take effect.

Custom connector example

The example connector is a Secure File Transfer Protocol (SFTP) connector that crawls files that are located on an SFTP server.

The example connector includes three components:

- Java source code for the connector

- An XML definition file that defines the parameters that the connector uses to connect to and crawl the data source
- A properties file that defines optional behaviors for the connector

Requirements

The Java source code for the example connector has the following dependencies:

- Java SE Development Kit (JDK) 1.8 or higher.
- The `custom-crawler-docs.zip` file from an installed Discovery instance as described at [Downloading the custom-crawler-docs.zip file in Discovery 2.2.1 and later](#) and [Downloading the custom-crawler-docs.zip file in Discovery 2.2.0 and earlier](#).
- The `JSch` Java package, as described [Downloaded JSch](#). You can download the package in [ZIP format](#) or [JAR format](#).

Downloading the `custom-crawler-docs.zip` file in Discovery 2.2.1 and later

In Discovery version 2.2.1 and later, perform the following steps to download the `custom-crawler-docs.zip` file to your local machine. You need root access to an installed Discovery instance:

1. Log in to your Discovery cluster.
2. Enter the following command to obtain your `crawler` pod name:

```
$ oc get pods | grep crawler
```

You might see output that looks like this:

```
wd-discovery-crawler-57985fc5cf-rxk89      1/1      Running      0      85m
```

3. Enter the following command to obtain the `custom-crawler-docs.zip` file, replacing `{crawler-pod-name}` with the `crawler` pod name that you obtained in step 2:

```
$ oc exec {crawler-pod-name} -- ls -l /opt/ibm/wex/zing/resources/ \
| grep custom-crawler-docs
```

You might see output that is similar to the following:

```
-rw-r--r--. 1 dadmin dadmin 59451 Jan 19 03:50 custom-crawler-docs-${build-version}.zip
```

4. Enter the following command to copy the `custom-crawler-docs.zip` file to the host server, replacing `{build-version}` with the build version number in step 3:

```
$ oc cp {crawler-pod-name}:/opt/ibm/wex/zing/resources/custom-crawler-docs-${build-version}.zip custom-crawler-docs.zip
```

5. Enter the following command to expand the `custom-crawler-docs.zip` file:

```
$ unzip custom-crawler-docs.zip -d custom-crawler-docs-primary
```



Note: If necessary, copy the `custom-crawler-docs.zip` file to the development server.



Note: If your local machine does not have the `unzip` utility, try using the `gunzip` command instead, or see the documentation of the operating system of your local machine for other alternatives to expand compressed files.



Tip: If you are using a version of Discovery that is earlier than 2.1.2 and you want to access the `custom-crawler-docs.zip` file, enter the following command: `scp root@{instance_name}:/root/bob/sdk/custom-crawler-docs.zip {local_directory}`

For information about downloading the `custom-crawler-docs.zip` file to Discovery 2.2.0 and earlier, see [Downloading the custom-crawler-docs.zip file to Discovery 2.2.0 and earlier](#).

Downloading the `custom-crawler-docs.zip` file to Discovery 2.2.0 and earlier

In Discovery version 2.2.0 and earlier, perform the following steps to download the `custom-crawler-docs.zip` file to your local machine. You need root access to an installed Discovery instance:

1. Obtain the entitlement key by navigating to your [container software library](#).

2. Enter the following command to log in to the Docker registry where your Discovery images are available. Include your entitlement key in the following command:

```
$ docker login cp.icr.io -u cp -p {entitlement_key}
```

3. Enter the following command to pull the `custom-crawler-sdk` image:

```
$ docker pull cp.icr.io/cp/watson-discovery/custom-crawler-sdk:2.1.3
```

4. Enter the following command to run the `custom-crawler-sdk` image:

```
$ docker run cp.icr.io/cp/watson-discovery/custom-crawler-sdk:2.1.3
```

5. Enter the following command to copy `custom-crawler-docs.zip` from the container where the image is running:

```
$ docker cp {container_name}:/crawler/custom-crawler-docs.zip .
```

To find the image, enter `docker ps -a | grep custom-crawler-sdk`.

6. Expand the `custom-crawler-docs.zip` file:

```
$ cd {local_directory}
```

where `{local_directory}` is the directory on your local machine to which you downloaded the `custom-crawler-docs.zip` file.

```
$ unzip custom-crawler-docs.zip
```



Note: If your local machine does not have the `unzip` utility, try using the `gunzip` command instead, or see the documentation of the operating system of your local machine for other alternatives to expand compressed files.



Tip: If you are using a version of Discovery that is earlier than 2.1.2 and you want to access the `custom-crawler-docs.zip` file, enter the following command: `scp root@{instance_name}:/root/bob/sdk/custom-crawler-docs.zip {local_directory}`.

For information about downloading the `custom-crawler-docs.zip` file on Discovery version 2.2.1 and later, see [Downloading the `custom-crawler-docs.zip` file in Discovery 2.2.1 and later](#).

Understanding the `custom-crawler-docs.zip` file

The `custom-crawler-docs.zip` file expands into a directory named `custom-crawler-docs-primary` that includes the following contents:

```
custom-crawler-docs-primary
├── README.md
├── build.gradle
├── config
│   ├── README.md
│   ├── messages.properties
│   └── template.xml
├── scripts
│   └── manage_custom_crawler.sh
├── settings.gradle
└── src
    └── main
        └── java
            └── com
                └── ibm
                    └── es
                        └── ama
                            └── custom
                                └── crawler
                                    └── sample
                                        └── sftp
                                            └── SftpCrawler.java
                                            └── SftpSecurityHandler.java
└── wexlib
    └── META-INF
```

```
└── MANIFEST.MF
├── README.md
└── ama-zing-custom-crawler-{version_numbers}-javadoc.jar
    └── ama-zing-custom-crawler-{version_numbers}.jar
```

15 directories, 12 files

Downloading JSch

JSch is a Java implementation of the Secure Shell protocol version 2 (SSH2) protocol and, by extension, **sftp**. It is derived from the [Java Cryptography Extension \(JCE\)](#). You can find specifications for SSH2 at [www.openssh.com/specs.html](#).

The current version of JSch is 0.1.55 and is supported by the example connector.

Download JSch to your development directory (**{local_directory}**). You can download the package in [ZIP format](#) or [JAR format](#). If you download the package in .zip format, extract it as described in the previous section.

Files for the example connector

The example custom connector includes three files that get built together:

- Java source files that are named **SftpCrawler.java** and **SftpSecurityHandler.java**
- An XML definitions file named **template.xml**
- A properties file named **message.properties**

You can locate and examine these files by referencing the directory tree listing in [Understanding the custom-crawler-docs.zip file](#).

For more information

For detailed documentation of all of the interfaces and methods that are available in the **com.ibm.es.ama.custom.crawler** package, see the Javadoc, which is available as indicated in [Interfaces and Javadoc](#).

Assembling and compiling a custom Cloud Pak for Data connector

You package a number of component files together to create a custom connector.

IBM Cloud Pak for Data [IBM Cloud Pak for Data only](#)



Note: This information applies only to installed deployments.

Custom connector components

A custom connector package is a compressed file that contains the following components:

Path	Description
config/template.xml	A configuration template
config/messages.properties	A properties file for UI messages
lib/*.jar	JAR files required by the custom connector, not including the connector code that you write

Connector components

Configuration template

The configuration template is an XML file that is divided into sections. Each section contains related settings. The XML snippets are taken from the example **template.xml** file whose location is listed in [Understanding the custom-crawler-docs.zip file](#).

Declaration settings

Declared settings are represented by the **<declare />** element. The element has the following attributes:

Attribute name	Description
type	Data type; one of string , long , boolean , list of strings, or enum

name	The name of the setting
initial-value	The initial value of the setting
enum-value	A list of enum values separated by vertical bars ()
required	Indicates that the setting is required
hidden	Indicates whether to hide the setting from the UI. Specify a value of true to hide the setting.

Declare element attributes

 **Note:** In the current release, the **required** and **hidden** attributes are not applied in the Discovery product user interface.

Declaration setting examples

To declare an **enum** type, use code similar to the following snippet:

```
<declare type="enum" name="type" enum-values="PROXY|BASIC|NTLM" initial-value="BASIC"/>
```

To declare a hidden **string** with an initial value, use code similar to the following snippet:

```
<declare type="string" name="custom_config_class" hidden="true" initial-value="com.example.ExampleCrawlerConfig" />
```

To declare a required **long**, use code similar to the following snippet:

```
<declare type="long" name="port" required="required" initial-value="22"/>
```

Conditional settings

Conditional settings are represented by the **<condition />** element. A conditional setting is displayed only if the condition is satisfied. The element has the following attributes:

Attribute name	Description
name	The name of the setting
enable	Enable the setting if the value of the name attribute equals the value of the enable attribute
in	Enable the setting if the value of the name attribute is included in a specified list of values
Condition element attributes	

 **Note:** In the current release, conditional settings are not applied in the Discovery product user interface.

Conditional setting examples

To enable a section by using a **boolean** condition, use code similar to the following snippet:

```
<declare type="boolean" name="use_key" initial-value="true" />
<!-- Enable setting only if use_key is true -->
<condition name="use_key" enabled="true">
    <declare type="string" name="key" hidden="false" />
</condition>
```

To enable a section by using an **enum** condition, use code similar to the following snippet:

```
<declare type="enum" name="type" enum-values="PROXY|BASIC|NTLM" initial-value="BASIC"/>
<!-- Enable setting for BASIC, NTLM or PROXY -->
<condition name="type" in="BASIC|NTLM|PROXY">
</condition>
<!-- Enable setting for PROXY -->
<condition name="type" in="PROXY">
```

Template sections

Each section includes one `<declare />` element for each of its settings.

XPath expression	Description
<code>/function/@name</code>	The name (type) of the crawler. Not a display name for the UI. Cannot contain spaces.
<code>/function/prototype/proto-section</code>	A section of the configuration.

Template sections

Section: general_settings

The XPath expression is `/function/prototype/proto-section[@section="general_settings"]`. It includes common settings for all crawlers, including the following settings:

```
<declare type="string" name="crawler_name" />
<declare type="string" name="description" />
<declare type="long" name="fetch_interval" initial-value="0" />
<declare type="long" name="number_of_max_threads" initial-value="10" />
<declare type="long" name="number_of_max_documents" initial-value="2000000000" />
<declare type="long" name="max_page_length" initial-value="32768" />
```

The custom crawler is initialized with the following settings in the `general_settings` section. For information about the interfaces, see [Developing custom connector code](#).

Name	Value
<code>custom_config_class</code>	The name of a class that implements the <code>com.ibm.es.ama.custom.crawler.CustomCrawlerConfiguration</code> interface
<code>custom_crawler_class</code>	The name of a class that implements the <code>com.ibm.es.ama.custom.crawler.CustomCrawler</code> interface
<code>custom_security_class</code>	The name of a class that implements the <code>com.ibm.es.ama.custom.crawler.CustomCrawlerSecurityHandler</code> interface
<code>document_level_security_supported</code>	Specifies whether document-level security is enabled (<code>true</code>) or disabled (<code>false</code>)

General settings section defaults

To specify the interfaces, use code similar to the following snippet:

```
<!-- Configuration class -->
<declare type="string" name="custom_config_class" hidden="true" initial-
value="com.ibm.es.ama.custom.crawler.sample.sftp.SftpCrawler" />
<!-- Crawler class -->
<declare type="string" name="custom_crawler_class" hidden="true" initial-
value="com.ibm.es.ama.custom.crawler.sample.sftp.SftpCrawler" />
<!-- Document level security class -->
<declare type="string" name="custom_security_class" hidden="true" initial-
value="com.ibm.es.ama.custom.crawler.sample.sftp.SftpCrawler" />
<!-- Document level security is enabled or not -->
<declare type="boolean" name="document_level_security_supported" initial-value="true" hidden="true"/>
```



Note: If you built a custom connector with an SDK package that was bundled with version 2.2.1 or earlier, `document_level_security_supported` must be disabled (set to `false`). Document-level security is not supported in 2.2.1 and earlier releases. However, the **Enable Document Level Security** option is displayed in Discovery even when document-level security is not supported. Do not select this option when you create a new collection.

To hide the **Enable Document Level Security** option from Discovery if the custom connector was built with an SDK package that was bundled with version 2.2.1 or earlier, complete the following steps:

1. Change the `document_level_security_supported` parameter in the `config/template.xml` file to read as follows:

```
<declare type="boolean" name="document_level_security_supported" hidden="true" initial-value="false"/>
```

2. Rebuild the connector package, and then upload it again.

Section: datasource_settings

The XPath expression is `/function/prototype/proto-section[@section="datasource_settings"]`. It includes settings specific to the data source.

```
<!-- Data source settings change on each server -->
<proto-section section="datasource_settings">
  <!-- Sample: SFTP server settings -->
  <declare type="string" name="host" required="required" initial-value="localhost"/>
  <declare type="long" name="port" required="required" initial-value="22"/>
  <declare type="string" name="user" required="required" />
  <!-- Sample: Use key file or password -->
  <declare type="boolean" name="use_key" initial-value="true" />
  <!-- Sample: If use key, input key and passphrase -->
  <condition name="use_key" enabled="true">
    <declare type="string" name="key" hidden="false" />
    <declare type="password" name="passphrase" hidden="false" />
  </condition>
  <!-- Sample: If use password, input password -->
  <condition name="use_key" enabled="false">
    <declare type="password" name="secret_key" hidden="false" />
  </condition>
</proto-section>
```

Section: crawlspace_settings

The XPath expression is `/function/prototype/proto-section[@section="crawlspace_settings"]`. The section contains only one `<declare />` element to specify the path. The value of the path is provided by the connector code.

```
<!-- Do not modify, must be here -->
<proto-section section="crawlspace_settings" cardinality="multiple">
  <declare type="string" name="path" hidden="true" />
</proto-section>
```

Properties file

For an example of a properties file, see the example `messages.properties` file whose location is listed in [Understanding the custom-crawler-docs.zip file](#).

JAR files

The JAR files for any interfaces used by your custom connector code, including the `ama-zing-custom-crawler-{version_numbers}.jar` file whose location is listed in [Understanding the custom-crawler-docs.zip file](#). The `ama-zing-custom-crawler-{version_numbers}.jar` file includes the `com.ibm.es.ama.custom.crawler` Java package that is described in [Developing custom connector code](#).

Compiling and packaging the custom connector

After you write the source code and configuration files for your custom connector, you need to compile and package it.

Prerequisites

To compile a custom connector, you need to have the following items on your local system. See [Custom connector example](#) for details.

- Java SDK 1.8 or higher
- [Gradle](#)
- The `custom-crawler-docs.zip` file from an installed Discovery instance
- The JSch package
- The following files for the example custom connector:
 - Java source code (`SftpCrawler.java` and `SftpSecurityHandler.java`)
 - XML definition file (`template.xml`)
 - Properties file (`messages.properties`)

 **Important:** Do not change the names or paths of the example custom connector files. Doing so can result in problems, including build failures.

Compiling and packaging the source code

1. Ensure you are in the custom connector development directory on your local system:

```
$ cd {local_directory}
```

2. Use Gradle to compile your Java source code and to create a compressed file that includes all of the required components for the custom connector:

```
$ gradle build packageCustomCrawler
```

Gradle creates a file in `{local_directory}/build/distributions/{built_connector_zip_file}`, where the name of the `{built_connector_zip_file}` is based on the `rootProject.name` value of `settings.gradle`. For example, if the line reads as follows, Gradle generates a file that is named `{local_directory}/build/distributions/my-sftp-connector.zip`.

```
rootProject.name = 'my-sftp-connector'
```

Next step

Proceed to [Installing and uninstalling a custom connector](#) to install the custom connector to your Discovery instance.

Installing a custom Cloud Pak for Data connector

After you have compiled and packaged your custom connector, you need to install it to your Discovery instance.

IBM Cloud Pak for Data [IBM Cloud Pak for Data only](#)



Note: This information applies only to installed deployments.

Discovery provides a script named `manage_custom_crawler.sh` for installing and uninstalling custom connectors. The script is located in the `scripts` directory of the expanded `custom-crawler-docs.zip` file as described in [Understanding the custom-crawler-docs.zip file](#).

Installing a connector

You can install your custom connector to your Discovery instance by performing the following steps.

1. Ensure that you have completed all steps to create a custom connector up to and including the steps listed in [Compiling and packaging the example connector](#).
2. Run the following command from the directory on your local machine where you created and compiled your custom connector:

```
$ bash scripts/manage_custom_crawler.sh --endpoint {endpoint} --token {access token} deploy -n {crawler name} -f {built_connector_zip_file}
```

where you specify values for the following variables:

- endpoint: URL for your service instance. You can get this value from the **Access information** section of the service instance overview page in the IBM Cloud Pak for Data administrative console.
- access token: Bearer token that is required to access the endpoint. You can get this value from the same page as the endpoint.
- crawler name: (Optional) Name that you specified for the crawler.
- `{built_connector_zip_file}` is the name of the file you created in [Compiling and packaging the example connector](#).

For example:

```
$ bash scripts/manage_custom_crawler.sh --endpoint https://mycpd.wd40.example.com/discovery/zen40-wd/instances/1638165624521059/api --token eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI6ImVKcV9HY29NcHF5WUFJcVByZ0x0cERRZDNQcmRiTWo5TGg0X09W0EU4MlkifP.eyJlaXQi0iI-tKyznA_wjk_G698fbx1Zl73KZKyEWctKtyX7IJ1Px5DPdophcqS9i3bPJowHy-ioVp6DML02mscZImhvZPra-e6gwUdhSB64KArmMClo1-kZG20EclNh6-oxR447Bjdsgp7IYpkmynmw0K6vPIqmzwEhr9gAK1vWL0oVd4EoiYNuxZaSFL5byJ0mnQxXzM14w3lKQHZ91WYVKc4JnuJiSVsdpGqVz1JNFmT8D9FBqJQ4-USKCbJmMPXicU8cDtJIIfheBejwenfvejUTz5rgZgymYWrGvw3G2o0x_L1Yg-Q deploy -n awesome_crawler -f awesome_crawler.zip
```

Instead of specifying an access token for authentication, you can specify username and password parameters. For more information, see [Understanding the manage_custom_crawler.sh script](#).

When the custom crawler is deployed, a resource ID is assigned to the connector.

Verifying an installed connector

Verify that the connector has been deployed to the Discovery instance by logging into the Discovery tooling and ensuring that your connector is displayed as an option on the [Configure collection](#) page.

Using an installed connector on Discovery

To use the installed custom connector, follow the steps listed in [Creating a collection](#). The custom connector appears in the list of connectors provided at [Configuring Cloud Pak for Data data sources](#). For more information, see [Using a custom connector with the Discovery tooling](#).

Uninstalling a connector

To uninstall a custom connector from a Discovery instance, complete the following steps:

1. Optional: If you don't know the resource ID, run the following command to list the custom connectors. The resource IDs of the connectors are returned.

```
$ scripts/manage_custom_crawler.sh --endpoint {endpoint} --token {token} list
```

2. Run the following command from the directory where you extracted your custom connector ZIP file to uninstall the connector:

```
$ scripts/manage_custom_crawler.sh --endpoint {endpoint} --token {token} undeploy --id {crawler_resource_id}
```

where `{crawler-resource-id}` is the ID that is generated for the crawler when it is deployed.

```
$ scripts/manage_custom_crawler.sh --endpoint {endpoint} --token {token} undeploy --id {crawler_resource_id}
```

Instead of specifying an access token for authentication, you can specify username and password parameters. For more information, see [Understanding the manage_custom_crawler.sh script](#).

Understanding the `manage_custom_crawler.sh` script

The `manage_custom_crawler.sh` script has the following internal documentation:

Watson Discovery Custom Crawler Manager

This script will help you deploy, manage, and undeploy your custom crawler for Watson Discovery.

Subcommands:

deploy	Add a new Custom Crawler to your Watson Discovery instance.
undeploy	Undeploy your Custom Crawler by name.
list	List all Custom Crawlers for your Watson Discovery instance.

Options:

-e --endpoint	The endpoint URL for your cluster and add-on service instance (<code>https://cpd_cluster_host}:{port}/discovery/{release}/instances/{instance_id}/api`)</code>
-t --token	The authorization token of your Cloud Pak instance
-u --user	The user name of your Cloud Pak instance
-p --password	The user password of your Cloud Pak instance If the password is not specified, the command line prompts to input
-n --name	The name of the custom crawler to upload (deploy only)
-f --file	The path of the custom crawler package to upload (deploy only)
-i --id	The crawler_resource_id value to delete the custom crawler (undeploy only)
--help	Show this message.

4.0.5 and earlier releases only

Installing a connector in 4.0.5 and earlier releases

You can install your custom connector to your Discovery instance by performing the following steps.

1. Ensure that you have completed all steps to create a custom connector up to and including the steps listed in [Compiling and packaging the example connector](#).
2. Run the following command from the directory on your local machine where you created and compiled your custom connector:

```
$ bash scripts/manage_custom_crawler.sh deploy -z {built_connector_zip_file}
```

where `{built_connector_zip_file}` is the name of the file you packaged in [Compiling and packaging the example connector](#).

 **Important:** If your Discovery instance is running on Red Hat OpenShift, specify the `-o` or `--openshift` parameter with the script.

For example:

```
$ bash scripts/manage_custom_crawler.sh deploy -z myCrawler.zip -o true
```

Uninstalling a connector in 4.0.5 and earlier releases

To uninstall a custom connector from a Discovery instance, run the following command at the root of the unzipped `custom-crawler-docs.zip` directory:

```
$ bash scripts/manage_custom_crawler.sh undeploy -n {built_connector_name}
```

where `{build_connector_name}` is the name, not the zip file, of the installed connector.

 **Important:** If your IBM Watson® Discovery instance is running on Red Hat OpenShift, specify the `-o` or `--openshift` parameter with the script.

```
$ bash scripts/manage_custom_crawler.sh undeploy -n {built_connector_name} -o true
```

Understanding the `manage_custom_crawler.sh` script in 4.0.5 and earlier releases

The `manage_custom_crawler.sh` script has the following internal documentation:

```
Usage: ${BASH_SOURCE[0]} [--pathToZip PATH] [--properties PROPERTIES] [--xml XML]

Watson Discovery Custom Crawler Manager

This script will help you deploy, manage, and undeploy your custom crawler for
Watson Discovery.

Subcommands:
deploy          Add a new Custom Crawler to your Watson Discovery instance.
properties      Generate the properties file for your crawler.
undeploy        Undeploy your Custom Crawler by name.
list            List all Custom Crawlers for your Watson Discovery instance.

Options:
-d --discovery   The name of the Watson Discovery instance
-z --zipfile      The path to the zip file to be uploaded.
                  For deploy only.
-x --xml          The path to the XML file to be uploaded.
                  For deploy only.
-n --name         The name of the Custom Crawler to undeploy.
-m --messages     The path to the properties file, used when doing a two part deploy.
                  For properties only.
-o --openshift    Set flag to true if this is an OpenShift Cluster
--help           Show this message.
```

Using a custom Cloud Pak for Data connector from the Discovery user interface

After you build and deploy a custom connector, you can configure and run it in the Discovery user interface to create a collection.

IBM Cloud Pak for Data [IBM Cloud Pak for Data only](#)



Note: This information applies only to installed deployments.

You create and manage a collection as described in [Creating and managing collections](#). You can use a successfully deployed custom connector during this process as follows. Follow these instructions to use a custom connector instead of one of the pre-built connectors that are listed in [Configuring Cloud Pak for Data data sources](#).

1. After you create a project, look for your custom connector to connect to a data source.
2. Select the custom connector and then click **Next**.

The **Configure collection** page opens.



Note: The following steps apply specifically to the example custom connector that is included with the [custom-crawler-docs.zip](#) file.

3. Enter values for the following fields on the **Configure collection** page. If a field is already populated with a value, verify and change the value if needed. A prepopulated value indicates that a value was specified in the custom connector's [template.xml](#) or [message.properties](#) file.

General

Complete the following fields

- Collection name
- Collection language
- Crawl schedule

Crawler properties

Complete the following fields

- Crawler name
- Crawler description
- Time to wait between retrieval requests (milliseconds)

The default value is [0](#).

- Maximum number of active crawler threads

The default value is [10](#).

- Maximum number of documents to crawl

The default value is [2000000000](#).

- Maximum document size (KB)

The default value is [32768](#).

Data source properties

Complete the following fields

- Host name

The default value is [localhost](#).

- Port

The default value is [22](#).

- User name

- Use key file (or input password)

The default value is [On](#).

- Key file location

- passphrase

- Password

Crawl Space Properties

If the custom crawler supports document-level security and the [document_level_security_supported](#) value in the [template.xml](#) is set to [true](#), then an **Enable Document Level Security** switch is displayed in a **Security** section of the data source connection setup page. To enable document-level security, set the **Enable Document Level Security** switch to **On**. If the switch is set to Off, then the collection that is created cannot support document-level security even if the custom crawler can support document-level security.

4. Click **Finish** to create the collection.

Configuring a Cloud Pak for Data custom crawler plug-in

Building a Cloud Pak for Data custom crawler plug-in

Discovery features the option to build your own crawler plug-in with a Java SDK. By using crawler plug-ins, you can now quickly develop relevant solutions for your use cases. You can download the SDK from your installed Discovery cluster. For more information, see [Obtaining the crawler plug-in SDK package](#).

IBM Cloud Pak for Data **IBM Cloud Pak for Data only**



Note: This information applies only to installed deployments.



Note: Any custom code that you use with IBM Watson® Discovery is the responsibility of the developer; IBM Support does not cover any custom code that the developer creates.

The crawler plug-ins support the following functions:

- Update the metadata list of a crawled document
- Update the content of a crawled document
- Exclude a crawled document
- Reference crawler configurations, masking password values
- Show notice messages in the Discovery user interface
- Output log messages to the **crawler** pod console

However, the **crawler** plug-ins cannot support the following functions:

- Split a crawled document into multiple documents
- Combine content from multiple documents into a single document
- Modify access control lists

Crawler plug-in requirements

Make sure that the following items are installed on the development server that you plan to use to develop a **crawler** plug-in by using this SDK:

- Java SE Development Kit (JDK) 1.8 or higher
- [Gradle](#)
- cURL
- sed (stream editor)

Obtaining the crawler plug-in SDK package

1. Log in to your Discovery cluster.
2. Enter the following command to obtain your **crawler** pod name:

```
$ oc get pods | grep crawler
```

The following example shows sample output.

```
wd-discovery-crawler-57985fc5cf-rxk89      1/1      Running      0          85m
```

3. Enter the following command to obtain the SDK package name, replacing **{crawler-pod-name}** with the **crawler** pod name that you obtained in step 2:

```
$ oc exec {crawler-pod-name} -- ls -l /opt/ibm/wex/zing/resources/ | grep wd-crawler-plugin-sdk
```

The following example shows sample output.

```
-rw-r--r--. 1 dadmin dadmin 35575 Oct  1 16:51 wd-crawler-plugin-sdk-${build-version}.zip
```

4. Enter the following command to copy the SDK package to the host server, replacing **{build-version}** with the build version number from the previous step:

```
$ oc cp {crawler-pod-name}:/opt/ibm/wex/zing/resources/wd-crawler-plugin-sdk-${build-version}.zip wd-crawler-plugin-sdk.zip
```

5. If necessary, copy the SDK package to the development server.

Building a crawler plug-in package

1. Extract the SDK compressed file.
2. Implement the plug-in logic in `src/`. Ensure that the dependency is written in `build.gradle`.
3. Enter `gradle packageCrawlerPlugin` to create the plug-in package. The package is generated as `build/distributed/wd-crawler-plugin-sample.zip`.

Developing and implementing a Cloud Pak for Data custom crawler plug-in

The `crawler` plug-in includes a file that is called `com.ibm.es.ama.plugin.CrawlerPlugin`. This file is the [initialization interface](#) that has methods you can use when you work with your `crawler` plug-in.

IBM Cloud Pak for Data [IBM Cloud Pak for Data only](#)



Note: This information applies only to installed deployments.

Interfaces and Javadoc

The interface library is stored in the `lib/ama-zing-crawler-plugin-${build-version}.jar` directory of the SDK directory. The Javadoc for the JAR file is available in the `lib/ama-zing-crawler-plugin-${build-version}-javadoc.jar` file in the same directory.

Initialization interface

Use the `com.ibm.es.ama.plugin.CrawlerPlugin` interface to manage the `crawler` plug-in. The interface has the following methods:

Method	Description
<code>init</code>	Start a crawler plug-in
<code>term</code>	Stop a crawler plug-in
<code>updateDocument</code>	Update crawled documents
Supported methods	

Dependency management

The file `build.gradle` manages the Java dependencies.

Crawler plug-in sample

A sample `crawler` plug-in is available that illustrates how to add, update, and delete metadata. The plug-in example also updates and deletes documents that are crawled by the local file system connector. The Java source code file is named `src/main/java/com/ibm/es/ama/plugin/sample/SampleCrawlerPlugin.java`.

Logging messages

The custom `crawler` plug-in supports the `java.util.logging.Logger` package for logging messages.

Any log messages that you add must meet the following requirements:

- The log level must be `INFO` or higher.
- The logger name must start with `com.ibm.es.ama`.

Messages are written to the log file of the `crawler` pod where the plug-in is running. A logging sample is available in the `crawler` plug-in sample.

Assembling and compiling a custom crawler plug-in

After you write the source code for your crawler plug-in, you must assemble and compile it.

IBM Cloud Pak for Data [IBM Cloud Pak for Data only](#)



Note: This information applies only to installed deployments.

Prerequisites

You must have the following items to compile a crawler plug-in:

- Java SE Development Kit 1.8 or higher
- [Gradle](#)
- cURL
- sed (stream editor)
- Crawler plug-in SDK package, see [Obtaining the crawler plug-in SDK package](#)

Assembling and compiling the crawler plug-in

1. Specify the class name of the crawler plug-in by opening the `config/template.xml` file and modifying the `initial-value` of the `crawler_plugin_class` element.
2. Ensure that you are in the crawler plug-in SDK directory on your development server.
3. Enter `gradle packageCrawlerPlugin` to use Gradle to compile your Java source code and to create a compressed file that includes all of the required components for the crawler plug-in.
4. Confirm that you have access to the crawler plug-in package, which is in the `build/distributions/wd-crawler-plugin-sample.zip` file.

Managing custom crawler plug-ins on your Watson Discovery cluster

You can use the `scripts/manage_crawler_plugin.sh` script to perform common plug-in management actions. The `scripts/manage_crawler_plugin.sh` script is located in the [crawler plug-in SDK package](#). When you use the script in a command, you must have the endpoint URL of your Discovery cluster and the username and password of your IBM Cloud Pak® for Data instance.

IBM Cloud Pak for Data [IBM Cloud Pak for Data only](#)



Note: This information applies only to installed deployments.

Commands and options for managing your crawler plug-ins

You can enter `scripts/manage_crawler_plugin.sh --help` to view the following help messages in the script:

```
Usage: scripts/manage_crawler_plugin.sh --endpoint endpoint --user username [--password password] command
Watson Discovery Crawler Plug-in Manager
This script will help you deploy, undeploy, and list your crawler plug-ins for Watson Discovery.

Commands:
deploy      Add a new crawler plug-in to your Watson Discovery instance
undeploy    Undeploy your crawler plug-in by ID
list        List all crawler plug-ins for your Watson Discovery instance (default)

Options:
-e --endpoint  The endpoint URL for your cluster and add-on service instance
                (https://cpd\_cluster\_host}:{port}/discovery/{release}/instances/{instance\_id}/api)
-u --user     The user name of your Cloud Pak instance
-p --password The user password of your Cloud Pak instance
-n --name     If the password is not specified, the command line prompts to input
-f --file     The name of the crawler plug-in to upload (deploy only)
--id         The path of the crawler plug-in package to upload (deploy only)
--id         The crawler_resource_id value to delete the crawler plug-in (undeploy only)
--help       Show this message
```

You can use the following commands to deploy, undeploy, and list your crawler plug-ins. Replace the variable references `{variable}` with the required information:

- Deploy crawler plug-in: `scripts/manage_crawler_plugin.sh --endpoint {endpoint_URL} --user {username} deploy --name {plugin_name}`
- Undeploy crawler plug-in: `scripts/manage_crawler_plugin.sh --endpoint {endpoint_URL} --user {username} undeploy --id {crawler_resource_id}`
- List deployed crawler plug-in: `scripts/manage_crawler_plugin.sh --endpoint {endpoint_URL} --user {username} list`

After you use the `scripts/manage_crawler_plugin.sh` script to deploy a crawler plug-in, you can select the plug-in in the Discovery tooling when you create a collection. For more information about the crawler plug-in settings in Discovery, see [Crawler plug-in settings](#).

Using a Cloud Pak for Data custom crawler plug-in from the Discovery user interface

After you build and deploy a crawler plug-in, you can configure your Discovery collection to use your plug-in to process documents.

IBM Cloud Pak for Data **IBM Cloud Pak for Data only**



Note: This information applies only to installed deployments.

- You can create and manage a collection as described in [Creating and managing collections](#).
- You can select a successfully deployed crawler plug-in when you create and manage a collection.

For more information, see [Crawler plug-in settings](#).

- You can also deploy a crawler plug-in package to a testing environment.

Glossary

Term	Definition
Classifier	A resource that you can train to recognize document types and categorize them in your collection. You can create two types of classifiers, a text classifier and a document classifier . A text classifier can classify documents based on words and phrases that are extracted from the body text with their part of speech information taken into account. A document classifier can classify documents based on words and phrases that are extracted from the body text fields with information from their part of speech and the other enrichments that are applied to the body text taken into account. The information from the other nonbody fields are also used. Learn more
Collection	A set of documents that you can enrich and later search for meaningful information. Learn more .
Content Intelligence	A feature that you can use to enrich documents in a Document Retrieval project such that the project can recognize information that is relevant to business contracts. A Document Retrieval project with this feature enabled is referred to as a Document Retrieval for Contracts project type. Learn more
Data source	An external application or service where valuable knowledge resources are stored. Connect to a service where your data is stored so you can crawl the data without having to move it. Learn more .
Domain-specific	Relates to terms and concepts that have special meaning to an industry or business. In tennis, for example, the term love has a special meaning. It represents a zero score. If you built a tennis-related application, you would teach Discovery that the term love has a meaning in the domain of tennis that is different from the generally understood meaning.
Enrichment	What you add to documents in your collection to identify or tag terms in the document that are significant. For example, when you apply the Entity enrichment, terms that mention city names or famous people are tagged as locations or people of interest.
Facet	A category by which you can filter search query results. Automatically, facets based on entity types are applied to the query results for Document Retrieval projects and facets based on the parts of speech are applied to Content Mining projects. You can define your own facet categories based on document fields, including fields generated by enrichments, or based on dictionaries or patterns. Learn more
Index	As you upload data or connect to data that is stored in an external repository, the data is crawled and ingested. As part of the processing, an index is created to keep track of important information that is recognized from the source. The main difference between a data source and a collection is that content in the data source is crawled, normalized, and indexed as it is added to a collection.
Project	A container for the collections of data that fuel your research or search applications. Learn more .
Regular expression	A regular expression, also known as a regex , is a standardized format for defining search patterns. You can define patterns with special significance to your application. For example, the bill of materials (BOM) numbers for parts that you manufacture might have a standard syntax of two uppercase letters followed by four numbers (GT2345). You can teach Discovery to recognize BOM mentions by adding a regular expression that can recognize and tag occurrences of the pattern in text. Learn more .
Stopword	Words to filter out of queries because they are not useful in a search, such as a , an , and the . You can add words that are common and not useful to your use case as stopwords to improve the relevance of results for natural language queries. Learn more .

Definitions of Discovery terms

Watson SDKs

SDKs abstract much of the complexity associated with application development. By providing programming interfaces in languages that you already know, they can help you get up and running quickly with IBM Watson services.

Supported SDKs

The following Watson SDKs are supported by IBM:

- [Java SDK](#)
- [Node.js SDK](#)
- [Python SDK](#)
- [.NET SDK](#)



Tip: The [API reference](#) for each service includes information and examples for these SDKs.

Community SDKs

The following SDKs are available from the Watson community of developers:

- [ABAP SDK for IBM Watson](#), using SAP NetWeaver
- [Android SDK](#)
- [Go SDK](#)
- [Ruby SDK](#)
- [Salesforce SDK](#)
- [Swift SDK](#)
- [Unity SDK](#)

SDK updates and deprecation

The supported Watson SDKs are updated according to the following guidelines.

Semantic versioning

Supported Watson SDKs adhere to semantic versioning with releases labeled as `{major}.{minor}.{patch}`.

Release frequency

SDKs are released independently and might not update on the same schedule.

- The current releases of the Watson SDKs are updated on a 2- to 6-week schedule. These releases are either minor updates or patches that do not include breaking changes. You can update to any version of the SDK with the same major version number.
- Major updates that might include breaking changes are released approximately every 6 months.

Deprecated release

When a major version is released, support continues on the previous major release for 12 months in a deprecation period. The deprecated release might be updated with bug fixes, but no new features will be added and documentation might not be available.

Obsolete release

After the 12-month deprecation period, a release is obsolete. The release might be functional but is unsupported and not updated. Update to the current release.

Query reference

The full list of Discovery query parameters, operators, and aggregations. You can refer to this information for help when you write queries with the Discovery Query Language.

- For more information, see the Discovery [API reference](#).
- For an overview of query concepts, see the [Query overview](#).

Tip: In the Discovery user interface, you can write and test [natural language queries](#) on the *Improve and customize* page.

Parameters descriptions

Use query parameters to search your collection, identify a result set, and analyze result sets.

Parameter	Description	Example
aggregation	A statistical query of the results set	<code>aggregation=term(enriched_text.entities.type)</code>
filter	An unranked query language search for matching documents.	<code>filter=bees</code>
natural_language_query	A ranked natural language search for matching documents	<code>natural_language_query="How do bees fly"</code>
query	A ranked query language search for matching documents.	<code>query=bees</code>

Search parameters

Parameter	Description	Example
count	The number of <code>result</code> documents to return.	<code>count=15</code>
highlight	Highlight query matches	<code>highlight=true</code>
offset	The number of results to ignore before returning <code>result</code> documents from the results set	<code>offset=100</code>
return	List of fields to return	<code>return=title,url</code>
sort	Field to sort results set by	<code>sort=enriched_text.sentiment.document.score</code>
spelling suggestions	Spelling suggestions returned for natural language queries	<code>spellingSuggestions=true</code>

Structure parameters

Query limitations

You cannot query on field names that contain the following characters:

- Numbers (**0 - 9**) in the suffix of the field name. For example, `extracted-content2`.
- The characters `_`, `+`, and `-` in the prefix of the `field_name`. For example, `+extracted-content`.
- The characters `.`, `,`, and `:` in the `field_name`. For example, `new:extracted-content`.

Operators

Operators are the separators between different parts of a query. The following table lists the available operators.

Operator	Description	Example
<code>.</code>	JSON delimiter	<code>enriched_text.concepts.text</code>

<code>:</code>	Includes	<code>text:computer</code>
<code>::</code>	Exact match	<code>title::"Query building"</code>
<code>:!</code>	Does not include	<code>text:!computer</code>
<code>::!</code>	Not an exact match	<code>text::!winter</code>
<code>\</code>	Escape character	<code>title::"Dorothy said: \"There's no place like home.\\""</code>
<code>" "</code>	Phrase query	<code>enriched_text.concepts.text:"IBM Watson"</code>
<code>() []</code>	Nested groups	<code>filter-entities:(text:Turkey,type:Location)</code>
<code> </code>	or	<code>query-enriched.entities.text:Google IBM</code>
<code>,</code>	and	<code>query-enriched.entities.text:Google,IBM</code>
<code><=, >=, >, <</code>	Numerical comparisons	<code>enriched_text.sentiment.document.score>0.679</code>
<code>^x</code>	Score multiplier	<code>text:IBM^3</code>
<code>*</code>	Wildcard	<code>query-enriched_text.concepts.text:pre*</code>
<code>~n</code>	String variation	<code>query-enriched_text.entities.text:cat~1</code>
<code>:*</code>	Exists	<code>title:*</code>
<code>!*</code>	Does not exist	<code>title!*</code>
Query operators		

Aggregations

Aggregations return a set of data values. The following table lists the available aggregations.

Aggregation	Description	Example
<code>average</code>	Mean value for the specified field in the results set.	<code>average(product.price)</code>
<code>filter</code>	Filter results based on the specified pattern	<code>filter(enriched_text.concepts.text:cloud computing)</code>
<code>histogram</code>	Interval-based distribution	<code>histogram(product.price,interval:1)</code>
<code>max</code>	Maximum value for the specified field in the results set.	<code>max(product.price)</code>
<code>min</code>	Minimum value for the specified field in the results set.	<code>min(product.price)</code>
<code>nested</code>	Restrict aggregation	<code>nested(enriched_text.entities)</code>
<code>sum</code>	Sum of all fields in the results set.	<code>sum(product.price)</code>
<code>term</code>	Count of identical values	<code>term(enriched_text.concepts.text,count:10)</code>
<code>timeslice</code>	Time-based distribution	<code>timeslice(last_modified,2day,America/New York)</code>

<u>top_hits</u>	Top-ranked result documents for the current aggregation	<code>term(enriched_text.concepts.text).top_hits(10)</code>
<u>unique_count</u>	Count of unique values for a field within an aggregation	<code>unique_count(enriched_text.entities.type)</code>

Query aggregations

Language support

When you create a collection, you specify the language of the collection. All of the documents that you add to a collection must be written in the same language.



Note: Discovery is not optimized for multilingual search. Although you can add several collections, each one with documents in a separate language, into one project, the query results from the project will be unpredictable. The results might include irrelevant passages from a document in a language that is different from the language of the user's query.

The following table describes the product features that are supported in each language.

Language	Supported features
Arabic (ar)	Advanced rules models, Built-in entities, Classifier (Document and Text), Custom entities, Dictionary, Document sentiment, Keywords, Machine Learning, Optical character recognition v1, Parts of speech, Regular expressions, Smart Document Understanding, Stemmer, Table Understanding
Bosnian (bs)	Classifier (Document and Text), Custom entities, Dictionary, Parts of speech, Regular expressions
Chinese, simplified (zh-CN)	Advanced rules models, Built-in entities, Classifier (Document and Text), Custom entities, Dictionary, Document sentiment, Keywords, Machine Learning, Optical character recognition v1, Parts of speech, Phrase sentiment, Regular expressions, Smart Document Understanding, Table Understanding
Chinese, traditional (zh-TW)	Advanced rules models, Classifier (Document and Text), Custom entities, Dictionary, Regular expressions, Machine Learning, Optical character recognition v1, Parts of speech, Phrase sentiment, Smart Document Understanding, Table Understanding
Croatian (hr)	Classifier (Document and Text), Custom entities, Dictionary, Regular expressions, Parts of speech
Czech (cs)	Classifier (Document and Text), Custom entities, Dictionary, Optical character recognition v1, Parts of speech, Phrase sentiment, Regular expressions, Smart Document Understanding, Stemmer, Table Understanding
Danish (da)	Classifier (Document and Text), Custom entities, Dictionary, Optical character recognition v1, Parts of speech, Regular expressions, Smart Document Understanding, Stemmer, Table Understanding
Dutch (nl)	Advanced rules models, Built-in entities, Classifier (Document and Text), Custom entities, Dictionary, Document sentiment, Keywords, Machine Learning, Optical character recognition v2, Parts of speech, Phrase sentiment, Regular expressions, Smart Document Understanding, Stemmer, Table Understanding
English (en)	Advanced rules models, Built-in entities, Classifier (Document and Text), Contracts, Custom entities, Dictionary, Document sentiment, Keywords, Machine Learning, Optical character recognition v2, Parts of speech, Phrase sentiment, Regular expressions, Smart Document Understanding, Stemmer, Table Understanding
Finnish (fi)	Classifier (Document and Text), Custom entities, Dictionary, Parts of speech, Regular expressions, Smart Document Understanding, Stemmer, Table Understanding
French (fr)	Advanced rules models, Built-in entities, Classifier (Document and Text), Custom entities, Dictionary, Document sentiment, Keywords, Machine Learning, Optical character recognition v2, Parts of speech, Regular expressions, Smart Document Understanding, Stemmer, Table Understanding
German (de)	Advanced rules models, Built-in entities, Classifier (Document and Text), Custom entities, Dictionary, Document sentiment, Keywords, Machine Learning, Optical character recognition v2, Parts of speech, Regular expressions, Smart Document Understanding, Stemmer, Table Understanding
Hebrew (he)	Classifier (Document and Text), Custom entities, Dictionary, Optical character recognition v2 (Cloud-managed), Parts of speech, Regular expressions, Smart Document Understanding, Table Understanding
Hindi (hi)	Classifier (Document and Text), Custom entities, Dictionary, Parts of speech, Regular expressions, Stemmer

Italian (it)	Advanced rules models, Built-in entities, Classifier (Document and Text), Custom entities, Dictionary, Document sentiment, Keywords, Machine Learning, Optical character recognition v1, Parts of speech, Regular expressions, Smart Document Understanding, Stemmer, Table Understanding
Japanese (ja)	Advanced rules models, Built-in entities, Classifier (Document and Text), Custom entities, Dictionary, Document sentiment, Keywords, Machine Learning, Optical character recognition v1, Parts of speech, Phrase sentiment, Regular expressions, Smart Document Understanding, Table Understanding
Korean (ko)	Advanced rules models, Built-in entities, Classifier (Document and Text), Custom entities, Dictionary, Document sentiment, Keywords, Machine Learning, Optical character recognition v1, Parts of speech, Regular expressions, Smart Document Understanding, Table Understanding
Norwegian (Bokmål) (nb)	Classifier (Document and Text), Custom entities, Dictionary, Optical character recognition v1, Parts of speech, Regular expressions, Smart Document Understanding, Stemmer, Table Understanding
Norwegian (Nynorsk) (nn)	Classifier (Document and Text), Custom entities, Dictionary, Optical character recognition v1, Parts of speech, Regular expressions, Smart Document Understanding, Stemmer, Table Understanding
Polish (pl)	Classifier (Document and Text), Custom entities, Dictionary, Optical character recognition v1, Parts of speech, Regular expressions, Smart Document Understanding, Table Understanding
Portuguese, Brazilian (pt-br)	Advanced rules models, Built-in entities, Classifier (Document and Text), Custom entities, Dictionary, Document sentiment, Keywords, Machine Learning, Optical character recognition v2, Parts of speech, Regular expressions, Smart Document Understanding, Stemmer, Table Understanding
Romanian (ro)	Classifier (Document and Text), Custom entities, Dictionary, Optical character recognition v1, Parts of speech, Phrase sentiment, Regular expressions, Smart Document Understanding, Stemmer, Table Understanding
Russian (ru)	Classifier (Document and Text), Custom entities, Dictionary, Optical character recognition v1, Parts of speech, Phrase sentiment, Regular expressions, Smart Document Understanding, Stemmer, Table Understanding
Serbian (sr) [1]	Classifier (Document and Text), Custom entities, Dictionary, Parts of speech, Regular expressions
Slovak (sk)	Classifier (Document and Text), Custom entities, Dictionary, Optical character recognition v1, Parts of speech, Regular expressions, Smart Document Understanding, Table Understanding
Spanish (es)	Advanced rules models, Built-in entities, Classifier (Document and Text), Custom entities, Dictionary, Document sentiment, Keywords, Machine Learning, Optical character recognition v2, Parts of speech, Phrase sentiment, Regular expressions, Smart Document Understanding, Stemmer, Table Understanding
Swedish (sv)	Classifier (Document and Text), Custom entities, Dictionary, Optical character recognition v1, Parts of speech, Regular expressions, Smart Document Understanding, Stemmer, Table Understanding

Feature support per language



Note: Optical character recognition (OCR) v2 was introduced in Cloud-managed service instances on 2 November 2022. OCR v2 was introduced in IBM Cloud Pak for Data instances with version 4.7.1.

English-only support

The following features are currently supported in English only:

- [Document Retrieval for Contract project type](#)
- IBM Cloud [Patterns \(beta\)](#)

1. Serbian supports Latin script only. [←](#)

IBM Cloud security

Information security

IBM is committed to providing our clients and partners with innovative data privacy, security, and governance solutions.

IBM Cloud **IBM Cloud only**



Note: This information applies only to managed deployments.

Notice: Clients are responsible for ensuring their own compliance with various laws and regulations, including the European Union General Data Protection Regulation. Clients are solely responsible for obtaining advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulations that may affect the clients' business and any actions the clients may need to take to comply with such laws and regulations.

The products, services, and other capabilities that are described herein are not suitable for all client situations and might have restricted availability. IBM does not provide legal, accounting, or auditing advice or represent or warrant that its services or products ensure that clients are in compliance with any law or regulation.

If you need to request GDPR support for IBM Cloud® Watson resources that are created, see [GDPR Subject Access Request](#).

European Union General Data Protection Regulation (GDPR)

IBM is committed to providing our clients and partners with innovative data privacy, security, and governance solutions to assist them on their journey to GDPR compliance.

Learn more about IBM's own GDPR readiness journey and our GDPR capabilities and offerings to support your compliance journey [here](#).

Labeling and deleting data in Discovery

Discovery includes an API to label data per call. For more information about how to label data by using either the API or from the Discovery product user interface, see [Labeling data](#).

Customer data can be deleted by using the API. For more information about deleting customer data, see [Deleting labeled data](#).



Note: Experimental and beta features are not intended for use with a production environment and are not guaranteed to function as expected when you label and delete their associated data. Do not use experimental and beta features when you implement a solution that requires the labeling and deletion of data.

Methods that support labeling data

The following stored information can be deleted by using a `customer_id` if the `customer_id` was specified when the information was originally added by using the associated method:

- Curations (`/v2/projects/{project_id}/curations`) Only available for `natural_language_query` query types.
- Documents (`/v2/projects/{project_id}/collections/{collection_id}/documents`)
- Notices (`/v2/projects/{project_id}/notices`) Only ingestion `notices` are labeled.
- Training data (`/v2/projects/{project_id}/training_data/queries`)
- Dictionaries (Only when created in the Discovery product user interface)
- Exported documents (Only when created in the Content Mining application)
- Reports (Only when created in the Content Mining application)

Exported documents and reports can be viewed in the `Repository` and `Report` pages of the Content Mining application. They are not available by using the API.

Discovery does not log query request data.

For more information about the options for labeling data in Discovery, see [Labeling data](#).

The following stored information is not explicitly labeled and cannot be deleted by specifying the `customer_id`. Personal Data is not supported in these fields.

Any string fields (including but not limited to `name` and `description`) of the following stored items:

- Collections
- Projects

Labeling data

Data can be labeled by using the API, or by using the Discovery product user interface. For more information about labeling with the product user interface, see [Labeling data in the product user interface](#).

 **Important:** You cannot label data that is added by crawling external data sources.

Data is labeled by adding the `customer_id` of your choice to the optional `X-Watson-Metadata` header. Discovery can then delete it by `customer_id`.

You can label data with the API in different ways:

- When you ingest documents by using the `POST /v2/projects/{project_id}/collections/{collection_id}/documents` or `POST /v2/projects/{project_id}/collections/{collection_id}/documents/{ID}` operations, send an optional header `X-Watson-Metadata`. The `X-Watson-Metadata` header must include either of the following items:
 - Semicolon separated `field=value` pairs (for example: `customer_id=123`)
 - The `customer_id` field. By adding the `customer_id` in `X-Watson-Metadata` header, the request indicates that it contains data that belongs to this `customer_id`.

Optionally, you can include the `customer_id` field with the `metadata` multipart form part instead of including the `X-Watson-Metadata` header.

 **Note:** If you specify a `customer_id` in the `metadata` multipart form part and the `X-Watson-Metadata` header for the same document, then the `customer_id` in the `X-Watson-Metadata` header is used.

This example adds the `customer_id` to both the `X-Watson-Metadata` header and the `metadata`:

```
$ curl -k -u "apikey:$API_KEY" \
-H "x-watson-userinfo:instance-id=asdf" \
-H "x-watson-metadata:customer_id=customer_header_123" \
-H "x-watson-discovery-next:true" \
-F "file=@$FILENAME" \
-F "metadata={"customer_id": "new123"}" \
-X POST "$API_URL/v2/projects/$PROJECT_ID/collections/$COLLECTION_ID/documents?version=2020-03-08" \
```

Example output:

```
$ {
  "document_id": "8b152926-e9f5-4f34-940a-c02da7ef3af4",
  "result_metadata": {
    "collection_id": "24265c0b-2a55-3ccf-0000-017334467b6e"
  },
  "metadata": {
    "date": 1594319812384,
    "parent_document_id": "8b152926-e9f5-4f34-940a-c02da7ef3af4",
    "customer_id": "customer_header_123"
  },
  "extracted_metadata": {
    "sha1": "CEC7C1D3423C7D4ED58FC448F52681ECA93CED8A",
    "numPages": "1",
    "filename": "Simple.pdf",
    "author": [
      "Simple Man"
    ],
    "subject": "Simple Metadata",
    "file_type": "pdf",
    "title": "Simple Title",
    "publicationdate": "2016-10-05"
  }
}
```

 **Important:** If your documents are already ingested, you must reingest them to add the `X-Watson-Metadata` header and `customer_id`.

Restrictions:

- The value of the `X-Watson-Metadata` header cannot exceed 4 KB of text.
- The `X-Watson-Metadata` header must contain a semicolon-separated list of `field=value` pairs. The `field` and `value` must not contain semicolons (`;`) or equals signs (`=`).
- `customer_ids` are unique within each Discovery instance. They are NOT unique per project or collection.
- A `customer_id` cannot be more than 256 characters in length.
- If a `customer_id` contains only white space or is empty, it is treated as though the `customer_id` was not provided at all, and no error messages are returned.

Labeling data in the product user interface

Data can be labeled by using the Discovery product user interface, or by using the API. For more information about labeling with the API, see [Labeling data](#).

To label data with the product user interface:

1. Open the **Projects** page by selecting **My Projects**.
2. Select **Data usage and GDPR**.
3. Choose the **GDPR data label** tab.
4. Set the **Label data with customer ID** toggle to **on**. The **Customer ID** field appears.
5. Enter a unique ID for the customer in the **Customer ID** field. Do not include personal data in a **Customer ID**.
6. Click **Save ID**.

After the **Customer ID** (`customer_id`) field is set, all data that is uploaded during the current browser session is labeled with the specified **Customer ID**. (You cannot label data that is added by crawling external data sources.)

Adding a **Customer ID** labels the documents, notices, dictionaries, and training data within that URL domain from that point forward, including each instance under that domain. Any actions, including document uploads, that occurred in the Discovery product user interface before the **Customer ID** field was added are not labeled.



Note: If you switch domains or browsers, empty the browser cache, or start an incognito session after you specify your **Customer ID** by using the Discovery product user interface, the **Customer ID** is not retained, and your data is not labeled. If you must switch domains or browsers, after the switch, open the **GDPR data label** tab, enter the **Customer ID** again, and then click **Save ID**.

If an existing **Customer ID** needs to be changed:

1. Delete the data associated with that **Customer ID**. For instructions, see [Deleting labeled data](#).
2. Follow the instructions to label data with the Discovery product user interface, or by using the API.
3. Upload or crawl the data.

Deleting labeled data

Customer data that is labeled with a `customer_id` can be deleted by using the API. For more information about how to label data by using either the API or from the Discovery product user interface, see [Labeling data](#).



Note: You cannot delete labeled data from the Discovery product user interface.

1. Use the **DELETE /v2/user_data** operation and provide the `customer_id` of the data you want to delete.
 - o **DELETE /v2/user_data** deletes all data that is associated with a particular `customer_id` within that service instance, as specified in [Methods that support labeling data](#). Also, see **Delete labeled data** in the [API reference](#)
2. To ensure all labeled content is correctly removed, run the **DELETE /v2/user_data** operation after the **processing** and **pending** counts for your collections return **0**.

Notes on deleting labeled data:

- Deletions happen asynchronously. You cannot track the progress of deletions.
- If a nonexistent `customer_id` is provided, nothing is deleted, but a **202 - Accepted** response is returned.
- Projects and collections are not labeled with a `customer_id`, even if a **X-Watson-Metadata** header is included in the request to create the project or collection. Only the individual documents within a collection are labeled. Therefore, when data is deleted, individual projects and collections are NOT deleted.

Managing IAM access for Discovery

Share a preview of your search application or build a team to work on a project. You can give team members access to your Discovery service instance through IBM Cloud.

IBM Cloud



Important: The information in this topic applies to managed deployments only.

Access to IBM Cloud service instances is controlled by IBM Cloud® Identity and Access Management (IAM). Every person who accesses Discovery in your account must be assigned an access policy with an IAM role.

Only an owner of the service account can add users. For more information, see [Managing access to resources](#).

Platform roles

A platform role controls a person's ability to access a service instance in IBM Cloud.

To give someone access to your service instance, assign them any platform role other than **Viewer**. A person with the **Viewer** role cannot access the product user interface. All of the other roles allow users to perform all tasks.

Service roles

A service role controls what a person can do in Discovery.

With Discovery, anyone who can access a Discovery instance can perform all actions. The only limitation is with the **Reader** role; anyone with reader-level access cannot submit API POST requests.

Activity Tracker events

As a security officer, auditor, or manager, you can use the Activity Tracker service to track how users and applications interact with the Discovery service in IBM Cloud®.

IBM Cloud



Note: This information applies only to managed deployments.

IBM Cloud Activity Tracker records user-initiated activities that change the state of a service in IBM Cloud. You can use this service to investigate abnormal activity and critical actions and to comply with regulatory audit requirements. In addition, you can be alerted about actions as they happen. The events that are collected comply with the Cloud Auditing Data Federation (CADF) standard. For more information, see the [getting started tutorial for IBM Cloud Activity Tracker](#).

List of events

The following table lists the Discovery actions that generate an event.

Action	Description
<code>discovery.analyze-api.read</code>	Process text by using the Analyze API.
<code>discovery.autocompletion.read</code>	Suggest complete queries based on documents.
<code>discovery.collection-notices.read</code>	Get notices for a collection.
<code>discovery.collection-training-status.read</code>	Get the training status of a single-collection training.
<code>discovery.collections.read</code>	Read collection annotations.
<code>discovery.content-miner-csv.create</code>	Import a CSV file to a Content Mining project.
<code>discovery.content-miner-csv.delete</code>	Delete a CSV file from a Content Mining project.
<code>discovery.content-miner-csv.read</code>	Get a CSV file from a Content Mining project.
<code>discovery.content-miner-export.create</code>	Create a set of exported documents from a Content Mining project.
<code>discovery.content-miner-export.download</code>	Download exported documents from a Content Mining project.
<code>discovery.content-miner-export.search</code>	List sets of exported documents from a Content Mining project.
<code>discovery.content-miner-report.create</code>	Create a report from a Content Mining project.

<code>discovery.content-miner-report.delete</code>	Delete a report from a Content Mining project.
<code>discovery.content-miner-report.read</code>	Get report content from a Content Mining project.
<code>discovery.content-miner-report.update</code>	Update report content from a Content Mining project.
<code>discovery.credential.create</code>	Create a credential.
<code>discovery.credential.delete</code>	Delete a credential.
<code>discovery.credential.read</code>	Get a credential, Salesforce objects, or a list of Cloud Object Storage buckets.
<code>discovery.credential.update</code>	Update a credential.
<code>discovery.curations.create</code>	Create a curated query.
<code>discovery.curations.delete</code>	Delete specified curation.
<code>discovery.curations.read</code>	List currently configured curation queries.
<code>discovery.curations.update</code>	Update existing curated results documents for a specified query.
<code>discovery.dataset.create</code>	Create a data set.
<code>discovery.dataset.update</code>	Update a data set.
<code>discovery.dataset-notices.read</code>	Get notices for a data set.
<code>discovery.dictionary.create</code>	Create a dictionary.
<code>discovery.dictionary.delete</code>	Delete a dictionary.
<code>discovery.dictionary.read</code>	Read a dictionary.
<code>discovery.dictionary.update</code>	Update a dictionary.
<code>discovery.document.add</code>	Add one document.
<code>discovery.document.create</code>	Create a document.
<code>discovery.document.delete</code>	Delete a document by ID.
<code>discovery.document.read</code>	Download a PDF version of a document.
<code>discovery.document.update</code>	Update one document by ingesting new or modified content for a document given a document ID, or by changing the label for a specified document ID.
<code>discovery.document-annotation.read</code>	Read document annotations.
<code>discovery.document-results.read</code>	Search collections for relevant documents.
<code>discovery.entity-extractor.create</code>	Create an entity extractor.

<code>discovery.entity-extractor.delete</code>	Delete an entity extractor.
<code>discovery.entity-extractor.read</code>	Read an entity extractor.
<code>discovery.entity-extractor.update</code>	Update an entity extractor.
<code>discovery.event.create</code>	Add a click event to a query.
<code>discovery.expansions.create</code>	Add synonyms to a collection.
<code>discovery.expansions.delete</code>	Delete synonyms from a collection.
<code>discovery.expansions.read</code>	Get synonyms for a collection.
<code>discovery.fields.read</code>	Get fields for a collection.
<code>discovery.label.create</code>	Create a collection label.
<code>discovery.label.delete</code>	Delete one or multiple documents from one or multiple collections based on a label.
<code>discovery.label.read</code>	Read a collection label.
<code>discovery.label.update</code>	Update a collection label.
<code>discovery.logs.read</code>	Get logs for a collection.
<code>discovery.metric.read</code>	Request a metric.
<code>discovery.model.export</code>	Export a model.
<code>discovery.model.import</code>	Import a model.
<code>discovery.multi-collection-training-status.read</code>	Get the training status of a multi-collection training.
<code>discovery.notices.read</code>	Get notices for a collection.
<code>discovery.page.read</code>	Read document pages.
<code>discovery.page-annotation.add</code>	Add document page annotation.
<code>discovery.page-prediction.read</code>	Read document page predictions.
<code>discovery.page-view.read</code>	Read document page view.
<code>discovery.project.create</code>	Create a project.
<code>discovery.project.delete</code>	Delete a project.
<code>discovery.project-notices.read</code>	Get notices for a project.
<code>discovery.stopwords.create</code>	Add stopwords to a collection.
<code>discovery.stopwords.delete</code>	Delete stopwords from a collection.

<code>discovery.stopwords.read</code>	Get stopwords for a collection.
<code>discovery.table-annotation.read</code>	Read document page table annotations.
<code>discovery.table-cell.read</code>	Read document page table cell.
<code>discovery.user-data.delete</code>	Delete all data associated with a customer ID.

Table 1. Actions that generate events

Viewing events

Events that are generated by an instance of the Discovery service are automatically forwarded to the IBM Cloud Activity Tracker service instance that is available in the same location.

IBM Cloud Activity Tracker can have only one instance per location. To view events, you must access the web UI of the IBM Cloud Activity Tracker service in the same location where your service instance is available.

To open the IBM Cloud dashboard, click the user icon in the page header, and then click [IBM Cloud dashboard](#). 

For more information about how to open IBM Cloud Activity Tracker from the dashboard, see [Navigating to the UI](#).

Public and private network endpoints

IBM Cloud

IBM Cloud® supports both public and private network endpoints for certain plans. Connections to private network endpoints do not require public internet access.

Private network endpoints support routing services over the IBM Cloud private network instead of the public network. A private network endpoint provides a unique IP address that is accessible to you without a VPN connection.

Enabling your account

 **Important:** Private network endpoints are supported for paid plans. Check the plan information for your service to learn about the plans that support private network endpoints.

Your account must be configured before you can use private endpoints. To use private network endpoints, the following account features must be enabled for your account.

- Virtual routing and forwarding (VRF).
- Service endpoints. Enabling service endpoints means that all users in the account can connect to private network endpoints.

To enable VRF, you create a support case. To enable service endpoints, you use the IBM Cloud CLI. For more information about how to enable your account, see [Enabling VRF and service endpoints](#).

Setting a private endpoint

After your account is enabled for VRF and service endpoints, you can add a private network endpoint to a service instance.

A service instance can have a private network endpoint, a public network endpoint, or both.

- Public: A service endpoint on the IBM Cloud public network.
- Private: A service endpoint that is accessible only on the IBM Cloud private network with no access from the public internet.
- Both public and private: Service endpoints that allow access over both networks.

Adding a private network endpoint

You add a private endpoint to a paid service instance from the service details page if you have a Manager or Writer service access role.

1. Go to your [Resource list](#).
2. Click the name of a service instance that is on a paid plan. Lite plans do not support private network endpoints.
3. In the service details page, click the **Manage** tab.
4. Click **Add private network endpoint**.

Viewing your endpoint URL

The service endpoint URLs are different for private and public network endpoints. You can view the URL for an endpoint from the service details page.

1. Go to your [Resource list](#).
2. Click the name of a service instance that has a private network endpoint.
3. In the service details page, click the **Manage** tab, and then click **Private Network Endpoint**.

What to do next

- [Configure your account](#) for VRF and Service endpoints.
- Modify your applications to use the new service endpoint URL.
- Read more about [service endpoints](#).

Virtual Private Endpoints

IBM Cloud

IBM Cloud® Virtual Private Endpoints (VPE) for VPC enables you to connect to supported IBM Cloud® services from your VPC network by using the IP addresses of your choosing, allocated from a subnet within your VPC. See more details [here](#).



Note: This document applies to Watson Assistant, Discovery, Speech to Text, and Text to Speech. Virtual Private Endpoints (VPEs) are available for these services in the Dallas, Washington, Frankfurt, London, Sydney, and Tokyo locations.

Prerequisites

- Have a [Virtual Private Cloud \(VPC\)](#)
- Have [private endpoints enabled](#) for your service instance

Instructions

- Create a VPE Gateway (VPEG) through the [UI](#), [CLI](#), or [API](#).
- Creation of the VPEG is confirmed when an IP address is set in the [details view of the VPEG page in the UI](#), [CLI output of the endpoint-gateway command](#), or [API details call](#).

You can verify by running `nslookup <endpoint>` on the private service endpoint of the Watson service from your VPC, for example:

```
# nslookup api.private.us-south.assistant.watson.cloud.ibm.com
Server: 127.0.0.53
Address: 127.0.0.53#53

Non-authoritative answer:
Name: api.private.us-south.assistant.watson.cloud.ibm.com
Address: 10.240.0.9 <---- your VPE IP address
```

To make requests using the assigned IP address instead of just the private service endpoint as suggested [here](#), you must do your own hostname resolution. For example:

```
$ curl -X POST "https://api.private.us-south.assistant.watson.cloud.ibm.com/v2/assistants" --connect ::10.240.0.9
```

```
$ curl -X POST "https://api.private.us-south.assistant.watson.cloud.ibm.com/v2/assistants" --resolve api.private.us-
south.assistant.watson.cloud.ibm.com:443:10.240.0.9
```

Additional links

- [IBM Cloud VPC](#)
- [VPE FAQ](#)
- [Accessing the VPE after setup](#)
- [Viewing Details of a VPE Gateway](#)
- [VPE Limitations](#)
- [Full VPC CLI reference](#)

Backup and restore

Backing up and restoring data in Cloud Pak for Data

Use the following procedures to back up and restore data in your IBM Watson® Discovery for IBM Cloud Pak® for Data instance.

IBM Cloud Pak for Data



Note: This information applies only to installed deployments.

You use the same set of backup and restore scripts to back up and restore data in any of the supported upgrade paths. The backup script stores the version number of the service with data to back up from the existing deployment. The restore script detects the version of the service that is installed on the new IBM Cloud Pak for Data deployment, and then follows the appropriate steps to restore data to the detected version.

The following table lists the upgrade paths that are supported by the scripts.

Version in use	Version that you can upgrade to
4.7.0	4.7.x
4.6.x	4.7.x
4.5.x	4.7.x except 4.7.0
4.0.x	4.7.x except 4.7.0

Supported upgrade paths

If you are upgrading from 4.5.x to 4.7.x, a simpler way to complete the upgrade is described in the following topics:

- [Upgrading Watson Discovery from Version 4.7](#).
- [Upgrading Watson Discovery from Version 4.6](#).
- [Upgrading Watson Discovery from Version 4.5.x](#).

If you use IBM Cloud Pak for Data Red Hat OpenShift APIs for Data Protection (OADP) backup and restore utility to back up and restore an entire cluster to 4.6, a few extra steps are required. For more information, see [Using OADP to back up a cluster where Discovery is installed](#).

You can do an in-place upgrade from one 4.7.x version to a later 4.7.y version. For more information, see [Upgrading Watson Discovery from Version 4.7.x to a later 4.7 refresh](#).

You can do an in-place upgrade from one 4.6.x version to a later 4.6.y version. For more information, see [Upgrading Watson Discovery from Version 4.6.x to a later 4.6 refresh](#).

You can do an in-place upgrade from one 4.5.x version to a later 4.5.y version. For more information, see [Upgrading Watson Discovery to the latest Version 4.5 refresh](#).

You can do an in-place upgrade from one 4.0.x version to a later 4.0.y version. For more information, see [Upgrading Watson Discovery to a newer 4.0 refresh](#).

Process overview

At a high level, the process includes the following steps:

1. Back up your Discovery data by using the backup script.
2. Install the latest version of IBM Cloud Pak for Data.
3. Install the latest version of the Discovery service on the cluster.
4. Restore the backed-up Discovery data by using the restore script.

Back up and restore limitations

You cannot migrate the following data:

- Dictionary suggestions models. These models are created when you build a dictionary. The dictionary is included in the backup, but the term suggestions model is not. Reprocess the migrated collections to enable dictionary term suggestions.
- You cannot back up and restore curations or migrate them because curations are a beta feature.

You can back up and restore some data by using the backup and restore scripts, but you must back up and restore other data

manually. The following data must be backed up manually:

- Local file system folders and documents that you can crawl by using the Local file system data source.

The following updates are made when your collections are restored:

- Any collection that contains documents that were created by uploading data are automatically recrawled and reindexed when restored. These documents are assigned new document ID numbers in the restored collections.
- Collections that were used in **Content Mining** projects are automatically recrawled and reindexed when restored. Only documents that are added by uploading data are assigned new document ID numbers in the restored collections.

Back up and restore methods

You can back up and restore your instance of Discovery manually or by using scripts.

- [Using the backup scripts](#)
- [Using the restore scripts](#)
- [Backing up data manually](#)
- [Restoring data manually](#)

You must have Administrative access to the Discovery instance on your Discovery cluster (where the data to be backed up is stored) and administrative access to the new instance (where the data will be restored to).

⚠️ Important: The backup and restore scripts complete many operations and can take quite a bit of time to run. To avoid timeout issues, run a tool that prevents timeouts, such as `nohup`.

Using the backup scripts

Because changes to the data stored in IBM Watson® Discovery during a backup can cause the backup to become corrupted and unusable, no in-flight requests are allowed during the backup period.

An in-flight request is any IBM Watson® Discovery action that processes data, including the following actions:

- Source crawl (scheduled or unscheduled)
- Ingesting documents
- Training a trained query model

The amount of storage that is available in the node where you run the backup script must be 3 times as large as the largest backup file in the data store that you plan to back up. If your data store is large, consider using a persistent volume claim instead of relying on the node's ephemeral storage. For more information, see [Configuring jobs to use PVC](#).

Complete the following steps to back up IBM Watson® Discovery data by using the backup scripts:

1. Enter the following command to set the current namespace where your Discovery instance is deployed:

```
$ oc project <namespace>
```

2. Get the backup script from the [GitHub repository](#).

⚠️ Important: You need all of the files in the repository to complete a backup and restore. Follow the instructions in GitHub Help to clone or download a compressed file of the repository.

3. Make each script an executable file by running the following command:

```
$ chmod +x <name-of-script>
```

Replace `<name-of-script>` with the name of the script.

4. Run the `all-backup-restore.sh` script.

```
$ ./all-backup-restore.sh backup [ -f backup_file_name ] [ --pvc ]
```

The `-f backup_file_name` parameter is optional. The name `watson_discovery_<timestamp>.backup` is used if you don't specify a name.

The `--pvc` parameter is optional. For more information about when to use it, see [Configuring jobs to use PVC](#). By default, the backup and restore scripts create a `tmp` directory in the current directory that the script uses for extracting or compressing backup files.

If you run into issues with the backup, rerun the backup command and include the `--use-job` parameter. This parameter instructs the backup script to use a Kubernetes job to back up ElasticSearch and MinIO in addition to Postgres, which uses a Kubernetes job by default. If the size of the data in ElasticSearch and MinIO is large and ephemeral storage is insufficient, include the `--pvc` option. When you do so, the script uses the persistent volume claim that is specified with the `--pvc` option instead of the `emptyDir` ephemeral storage as the temporary working directory for the job.

Extracting files from the backup archive file

The scripts generate an archive file, including the backup files of the services that are listed in Step 1.

1. You can extract files from the archive file by running the following command:

```
$ tar xvf <backup_file_name>
```

Configuring jobs to use PVC

The backup and restore process uses Kubernetes jobs. The jobs use ephemeral volumes that use ephemeral storage. It is a temporary storage mount on the pod that uses local storage of a node. In rare cases, the ephemeral storage is not large enough. You can optionally instruct the job to mount a Persistent Volume Claim (PVC) on its pod to use for storing the backup data. To do so, specify the `--pvc` option when you run the script. The scripts use `emptyDir` of Kubernetes otherwise.

In most cases, you don't need to use a persistent volume. If you choose to use a persistent volume, the volume must be 3 times as large as the largest backup file in the data store. The size of the data store's backup file depends on usage. After you create a backup, you can [extract files from the archive file](#) to check the file sizes.

Also, you must have 2 times as much disk space available on the local system as the size of the data store because the archive of the data is split and then recombined to prevent issues that might otherwise occur when you copy large files from the cluster node to the local system.

Mapping multitenant clusters

When you restore data that was backed up from a version earlier than 4.0.6 to any later release and the backed-up deployment had more than one instance of the service provisioned, an extra step is required. You must create a JSON file that maps the service instance IDs between the backed-up cluster and the cluster where the data is being restored.

This mapping step is not required if the instance IDs did not change between the back up and restore steps. For example, you can skip this step if you are restoring data to the same cluster where it was backed up from or if you are restoring data to a brand new cluster that has no Discovery instances.

To create a mapping, complete the following steps:

1. Extract the mapping template file from the backup archive file.

```
$ tar xf <backup_file_name> tmp/instance_mapping.json -0 > <mapping_file_name>
```

2. Make a list of the names and instance IDs of the service instances that are provisioned to the cluster where the data is being restored.

The instance ID is part of the URL that is specified in the instance summary page. From the IBM Cloud Pak for Data web client main menu, expand Services, and then click Instances. Find your instance, and then click it to open its summary page. Scroll to the **Access information** section of the page, and look for the instance ID in the **URL** field.

For example, https://<host_name>/wd/<namespace>-wd/instances/<instance_id>/api.

Repeat this step to make a note of the instance ID for every instance that is provisioned.

3. Edit the mapping file.

Add the instance IDs for the destination service instances that you listed in the previous step. The following snippet is an example of a mapping file.

```
{
  "instance_mappings": [
    {
      "display_name": "discovery-1",
      "source_instance_id": "1644822491506334",
      "dest_instance_id": "<new_instance_id>"
    },
    {
      "display_name": "discovery-2",
      "source_instance_id": "1644822552830325",
      "dest_instance_id": "<new_instance_id>"
    }
  ]
}
```

```
    ]  
}
```

When you run the restore script, include the optional `--mapping` parameter to apply this mapping file when the data is restored.

Using the restore scripts

 **Important:** If you are restoring data from a version earlier than 4.0.6 and you are restoring a multitenant cluster to a multitenant cluster, you must take an extra step before you begin. For more information, see [Mapping multitenant clusters](#).

Complete the following steps to restore data in IBM Watson® Discovery by using the restore scripts:

1. Enter the following command to set the current namespace where your Discovery instance is deployed:

```
$ oc project <namespace>
```

2. If you haven't already, get the restore script from the [GitHub repository](#).

 **Important:** You need all of the files in the repository to complete a back up and restore. Follow the instructions in GitHub Help to clone or download a compressed file of the repository.

3. Make each script an executable file by running the following command:

```
$ chmod +x <name-of-script>
```

Replace `<name-of-script>` with the name of the script.

4. Restore the data from the backup file on your local system to the new Discovery deployment by running the following command:

```
$ ./all-backup-restore.sh restore -f backup_file_name [--pvc] [--mapping]
```

The `--pvc` parameter is optional. For more information about when to use it, see [Configuring jobs to use PVC](#).

The `--mapping` parameter is optional. For more information about when to use it, see [Mapping multitenant clusters](#).

By default, the backup and restore scripts create a `tmp` directory in the current directory that the script uses for extracting or compressing backup files. If you used the `--use-job` parameter when you backed up the data, specify it again when you restore the data. This parameter instructs the backup script to use a Kubernetes job to back up ElasticSearch and MinIO.

The `gateway`, `ingestion`, `orchestrator`, `hadoop worker`, and `controller` pods automatically restart.

Using OADP to back up a cluster where Discovery is installed

If you plan to back up and restore an entire IBM Cloud Pak for Data instance by using the IBM Cloud Pak for Data Red Hat OpenShift APIs for Data Protection (OADP) backup and restore utility, you must do some additional steps in the right order for the utility to work properly when Discovery is present.

1. Run the Discovery backup script.
2. Use the OADP backup utility to back up the cluster.
3. Delete the project. This process removes the persistent volume claims and persistent volumes that are associated with Discovery.
4. Use the OADP backup utility to restore the cluster.
5. Uninstall Discovery, and then install Discovery again on the restored cluster.



Note: A repeat of the installation is required because the utility does not always reinstall Discovery correctly.

6. Run the Discovery restore script to restore your data.

Backing up data manually

Manually back up data that is not backed up by using the scripts.

To manually back up your data from an instance of Discovery, complete the following steps:

1. Enter the following command to log on to your Discovery cluster:

```
$ oc login https://<OpenShift administrative console URL> \
-u <cluster administrator username> -p <password>
```

2. Enter the following command to switch to the proper namespace:

```
$ oc project <discovery-install namespace>
```

3. Enter `oc get pods|grep crawler`.

4. Enter the following command:

```
$ oc cp <crawler pod>:/mnt <path-to-backup-directory>
```

Restoring data manually

Manually restore data that cannot be restored by using the script.

To manually restore your data from an instance of Discovery, complete the following steps:

1. Enter the following command to log on to your Discovery cluster:

```
$ oc login https://<OpenShift administrative console URL> \
-u <cluster administrator username> -p <password>
```

2. Enter the following command to switch to the proper namespace:

```
$ oc project <discovery-install namespace>
```

3. Enter `oc get pods|grep crawler`.

4. Enter the following command:

```
$ oc cp <path-to-backup-directory> <crawler pod>:/mnt
```

High availability and disaster recovery

IBM Watson® Discovery is highly available in all IBM Cloud® regions where Discovery is offered. However, recovering from potential disasters that affect an entire region requires planning and preparation.

IBM Cloud



Note: This information applies only to managed deployments.

You are responsible for understanding your customization and usage of the service. You are also responsible for being ready to re-create an instance of the service in a new region and to restore your data in any region. See [How do I ensure zero downtime?](#) for more information.

High availability

IBM Watson® Discovery supports high availability with no single point of failure. The service achieves high availability automatically and transparently by using the multi-zone region (MZR) feature provided by IBM Cloud.

IBM Cloud enables multiple zones that do not share a single point of failure within a single location. It also provides automatic load balancing across the zones within a region.

Disaster recovery

Disaster recovery can become an issue if an IBM Cloud region experiences a significant failure that includes the potential loss of data. Because MZR is not available in all regions, wait for IBM to bring a region back online if it becomes unavailable. If underlying data services are compromised by the failure, also wait for IBM to restore those data services.

If a catastrophic failure occurs, IBM might not be able to recover data from database backups. In this case, you need to restore your data to return your service instance to its most recent state. You can restore the data to the same or to a different region.

Your disaster recovery plan includes knowing, preserving, and being prepared to restore all data that is maintained on IBM Cloud.

Backing up your data in Watson Discovery

There are several methods for backing up the data that is stored in IBM Watson® Discovery. Consider including these methods in your disaster recovery plan. Also, consider backing up the following data types:

- Data that you might want a copy of, such as source documents
- Data that Discovery stores and that you want to extract and back up

The following table shows the resources that you can download and re-upload to and from an instance.

Resource	Download/Re-upload from the UI	API support
Uploaded and crawled files		See note.
Relevancy training data	✓	
Expansion list	✓	
Stop words list	✓	
Smart Document Understanding user-trained model	✓	
Smart Document Understanding pretrained model		
Text classifier enrichment	✓	
Dictionary enrichment	✓	
Entity extractor enrichment	✓	
Machine learning enrichment	✓	
Regular expression enrichment	✓	
Pattern enrichment	✓	
Advanced rules enrichment	✓	

Resource recovery support details



Note: You cannot subsequently download files that you add to Discovery because the original files are not stored in Discovery. However, you can retrieve information from the file that is stored in the collection index when the original file is processed. Use the [Query API](#) to submit a query that will return a passage from the file of interest, and then check the response body for data from the file. For example, for some file types, text from the original file is stored in the `text` field.

For information about resources that are created with the Content Mining application, see [Content Mining resources](#).

Ingested documents

Your uploaded documents are converted, enriched, and stored in the search index. If a disaster occurs, the search index is not recoverable. Store a backup of all your source documents in a safe place.

If you also import documents by doing scheduled crawls of external data sources, you might want to retain your data source credentials externally so that you can reestablish the connection to your data sources quickly. For the list of available sources and the credentials that are needed for each one, see [Configuring IBM Cloud data sources](#).

You can get some of the text that was stored in the index when the original document was ingested by using the Query API. For more information, see [Recovering documents](#).

Training data

Refer to this task to back up your training data queries and examples for a trained project. Training data is used for explicit training of your projects and is stored on a per project basis. To extract the training data, use the API to download the queries and the ratings from Discovery. To back up training data queries and examples, complete the following steps:

1. Download your training data by using the [list training queries](#) API.
2. Save your training queries and examples locally.

⚠ Important: The document IDs that you use in your training data point to the documents in your current project. Use the same IDs in your new projects to ensure that the correct documents are referenced. If the IDs do not match, your restored relevancy training will not work.

Expansion lists

If you are using synonyms (query expansions) for query modification, back up your .json expansion list, and store it locally. For more information, see [Implementing synonyms](#).

Stopwords

In the case of stopwords, back up the text file. For more information about stopwords, see [Defining stopwords](#).

Collection information

Tip: This is not required, but it is a best practice to [retrieve the status](#) for each collection regularly and store the information locally. By retaining these statistics, you can later verify that your restoration processes were successful if needed.

Smart Document Understanding models

If you use Smart Document Understanding (SDU), you have models that are associated with your configuration. To avoid loss of this information, [export your models](#), back them up, and store them locally. SDU models have the file extension of **.sdumodel**.

Dictionary enrichments

1. Open your project, and click **Improve and customize**.
2. On the **Improvement tools** panel, click **Teach domain concepts** and then **Dictionaries**.
3. Click the download icon next to your dictionary. Your dictionary then downloads as a .csv file.

Regular expressions enrichments

Back up your regular expressions as a .csv file, and store them locally. Note the regular expressions that you specified to create your enrichments so that you can re-create the enrichments from them. For more information, see [Regular expressions](#).

Machine learning enrichments

Back up your machine learning model .zip or .pear files, and store them locally. For more information, see [Machine learning enrichments and Watson Explorer Content Analytics Studio models](#).

Pattern enrichments

1. Open your project.
2. On the **Improvement tools** panel of the **Improve and customize** page, click **Teach domain concepts** and then **Patterns (Beta)**.
3. Click the download icon next to your pattern. Your pattern model then downloads as a .zip file.

Advanced rules models enrichment

Back up your model files as .zip files, and store them locally. For more information, see [Advanced rules models](#).

Classifier enrichments

Back up your classifier .csv files, and store them locally. For more information, see [Classifier](#).

Entity extractor enrichments

To back up entity extractor models, download the models and store them locally. A model must be fully trained before it can be downloaded. For more information, see [Exporting the entity extractor](#).

Content Mining application resources

You cannot back up certain data types and must manually re-create them. There are several Content Mining custom user resources that the application does not automatically back up. If data loss occurs, you must either manually re-create the following custom user resources in the Content Mining application or upload a locally saved file that contains the resource:

- Custom map

- Searched document export: You can export a searched document in the **Documents** view in the Content Mining application, but you cannot reupload it in the application.
- Facet analysis result export: You can download the results of your facet analysis by clicking the **Export** icon, then **Export results**, and **Export** in the **Analysis export options** dialog box.
- Collection: You can restore a Content Mining collection if you stored the collection locally as a .csv file and then upload it in the application. Otherwise, you must manually re-create the collection.
- Document classifier: You can restore a document classifier if you stored the document classifier locally as a .csv file and then upload it in the application. Otherwise, you must manually re-create the document classifier.
- Custom annotators
 - Dictionary: You can restore a dictionary in the application if you stored the dictionary locally as a .csv file and upload it in the application.
 - Regular expressions: You can restore a regular expression in the application if you stored the regular expression locally as a .csv file and upload it in the application.
 - Machine learning models: You can restore a machine learning model if you stored the model locally as a .zip file and then upload it in the application.
 - PEAR File: You can upload a .pear file if you stored the file locally and then upload it in the application.

You cannot back up the following resources locally and must recreate them in the Content Mining application.

- Saved analysis
- Report
- Dashboard

Restoring your data to a new Watson Discovery instance



Note: Consider using your backups to restore to a new Discovery instance in a different data center, also known as a region or location.

To begin restoration, first start by reviewing your list of collections and associated data sources, as well as your file backups.

- Create your projects and collections. Use the Discovery tooling, or the API. See [Create a project](#) and [Create a collection](#).
- Add back stopwords into the collections. See [Defining stopwords](#).
- If you use custom query expansion, add your query expansions. See [Implementing synonyms](#).
- If you use any custom entity models from IBM Watson® Knowledge Studio for enrichment, reimport that model into your Discovery instance. For details, see [Managing enrichments](#).

After you set up your projects and collections as they were before, begin ingesting your source documents. Depending upon how you ingested your documents previously, you can do so by using your own solution or one of the following methods:

- The [API](#)
- A [connector](#)

Restoring training data

After you restore your projects, you can begin the process of re-creating your relevancy training models. To restore your training data queries and examples, re-create your individual training queries and the examples by using the [create training query](#) API, or you can restore your queries and examples on Discovery. For more information about restoring your training data by using Discovery, see the instructions for accessing the **Train** page in [Improving result relevance with training](#).



Important: For the restore to work properly, note that the document IDs that you use in your training data point to the documents in your current project. Use the same IDs in your new projects to ensure that the correct documents are referenced. If the IDs do not match, your restored relevancy training will not work.

Restoring connections to external data sources

In case of an unanticipated loss of data, you might lose your scheduled crawls of external data sources. See [Configuring IBM Cloud data sources](#) for the list of available sources.

To restore your external data, reestablish your connections to these data sources, and then recrawl them.

To find the data source credentials that you stored, follow the instructions for your chosen data source in [Configuring IBM Cloud data sources](#). These instructions explain how you can reconnect to your data sources and get the data imported into Discovery.

Restoring Smart Document Understanding models

To import a previously exported Smart Document Understanding (SDU) model, see [Importing and exporting models](#). SDU models have

the file extension of **.sdumodel**.

When importing an SDU existing model into a new collection, it is a best practice to create the new collection and add one document, then import the model and upload the remainder of your documents.

Restoring dictionary enrichments

1. Open your project.
2. On the **Improvement tools** panel of the **Improve and customize** page, click **Teach domain concepts, Dictionaries**, and then **Upload**.
3. In the **Apply dictionary** dialog box, enter a name for your .csv file, select a language, specify the facet path, click **Upload**, and select your dictionary .csv file.
4. Click **Create**.

 **Note:** After you upload dictionary .csv files for recovery, you cannot use the dictionary editor to further edit the terms. If you want to use the dictionary editor, create a dictionary, and manually add the dictionary terms.

For information about uploading a dictionary enrichment .csv file, see [Dictionary](#).

Restoring pattern enrichments

You can restore pattern enrichment .zip files as advanced rules models .zip files by completing the following steps:

1. Open your project.
2. On the **Improvement tools** panel of the **Improve and customize** page, click **Teach domain concepts, Advanced rules models**, and then **Upload**.
3. In the **Apply advanced rules model** dialog box, enter a name for your .zip file, select a language, specify a result field, click **Upload**, and select your advanced rules models .zip file.
4. Click **Create**.

 **Note:** After you upload pattern model .zip files for recovery, you cannot use the pattern editor to further edit the .zip files.

For more information about uploading an advanced rules models .zip file, see [Advanced rules models](#).

Restoring entity extractors

To restore an entity extractor model, import the exported .ent file to create a new machine learning model. You cannot open the exported model in the entity extractor tool to continue working with it. However, you can import a finished model and apply it to collections as an entity extractor enrichment.

For more information about how to import an entity extractor model to create a machine learning enrichment, see [Use imported ML models to find custom terms](#).

Deleting a service instance

The procedure that you follow to delete a service instance differs depending on how the instance was deployed.

Deleting a managed service instance IBM Cloud

When you delete a Discovery service instance, the data associated with your service is also removed.

Only the owner or someone with Administrator role-level access to the service instance can delete it.

To delete your Discovery service instance, complete the following steps:

1. From the [IBM Cloud Resource list](#), expand the **AI/Machine Learning** section.
2. Find the service instance that you want to delete.
3. Click the **Actions** menu icon, and then select **Delete**.

The data that is associated with the service instance is retained for 7 days. After the 7-day retention period, it is deleted. If a service instance is deleted by mistake, you can restore it as long as you do so before the 7-day window closes. For more information, see [Using resource reclamations](#) in the IBM Cloud documentation.

For more information about how data is handled by IBM Cloud, see the following information:

- [Data Processing and Protection Datasheet](#)
- [IBM Cloud Data security and privacy terms](#)

For more information about how to delete a deployment of the service from Cloud Pak for Data as a Service, see [Deleting a deployment](#).

Deleting an installed service instance IBM Cloud Pak for Data

Only the owner or an Administrator of the provisioned service instance can delete it.

To deprovision a service instance of Discovery for IBM Cloud Pak for Data, complete the following steps:

1. From the IBM Cloud Pak for Data web client menu, click **Services > Instances**.
2. Find your service instance, and then click the menu icon and choose **Delete**.

For more information about how to uninstall the service, see [Uninstalling Discovery](#).

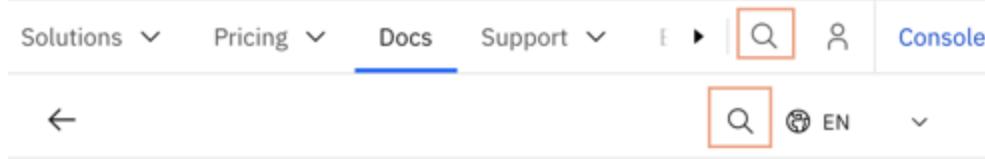
FAQ

Find answers to frequently asked questions.

For more information about Discovery-specific concepts, such as *projects* or *enrichments*, see the [glossary](#).

How do I search the product documentation?

To search the entire IBM Cloud Docs site, enter your search term into the search field in the IBM Cloud website banner. To search for information about the Discovery service only, scroll to the start of the page and enter your search term into the search field in the page header.



About Watson Discovery

IBM Watson™ Discovery is an AI-powered search engine that helps you to extract answers from complex business documents.

Figure 1. Search bar for this product documentation versus all IBM Cloud docs

How does Watson Discovery access my data?

Discovery has built-in connectors that can crawl various data sources, including websites, IBM Cloud Object Storage, Box, Microsoft SharePoint, and Salesforce sites. It even has support for you to build custom connectors. You can schedule crawls so that as the source data changes, the latest version is picked up by your collection automatically. Discovery only ever reads from external data sources; it never writes, updates, or deletes any content in the original data source. For more information, see [Creating collections](#).

Can I upload documents?

Yes, you can upload documents directly to a collection in your project. An upload is a one-time operation that you can use to get started. An alternative approach is to connect to a data source and crawl the source for information. When you crawl data sources, the data can stay where it is and you can set up a schedule by which to crawl the external source to find new and changed information. When you crawl the data, you know that the information in your collection is always up to date. For more information, see [Creating collections](#).

Must all my documents be English?

No. Discovery supports multiple languages. For more information about language support per feature, see [Language support](#).

What types of files can Discovery ingest?

Discovery can ingest most standard business file types, including PDF, Microsoft Word documents, spreadsheets, and presentations. For a complete list, see [Supported file types](#).

How do I know whether I have Discovery v1 or v2?

If you're using Discovery on IBM Cloud Pak® for Data, then you're using Discovery v2.

If you have a service instance that is managed by IBM Cloud, then check what you see when you launch the product. When you open the product user interface in v2, the following page is displayed:

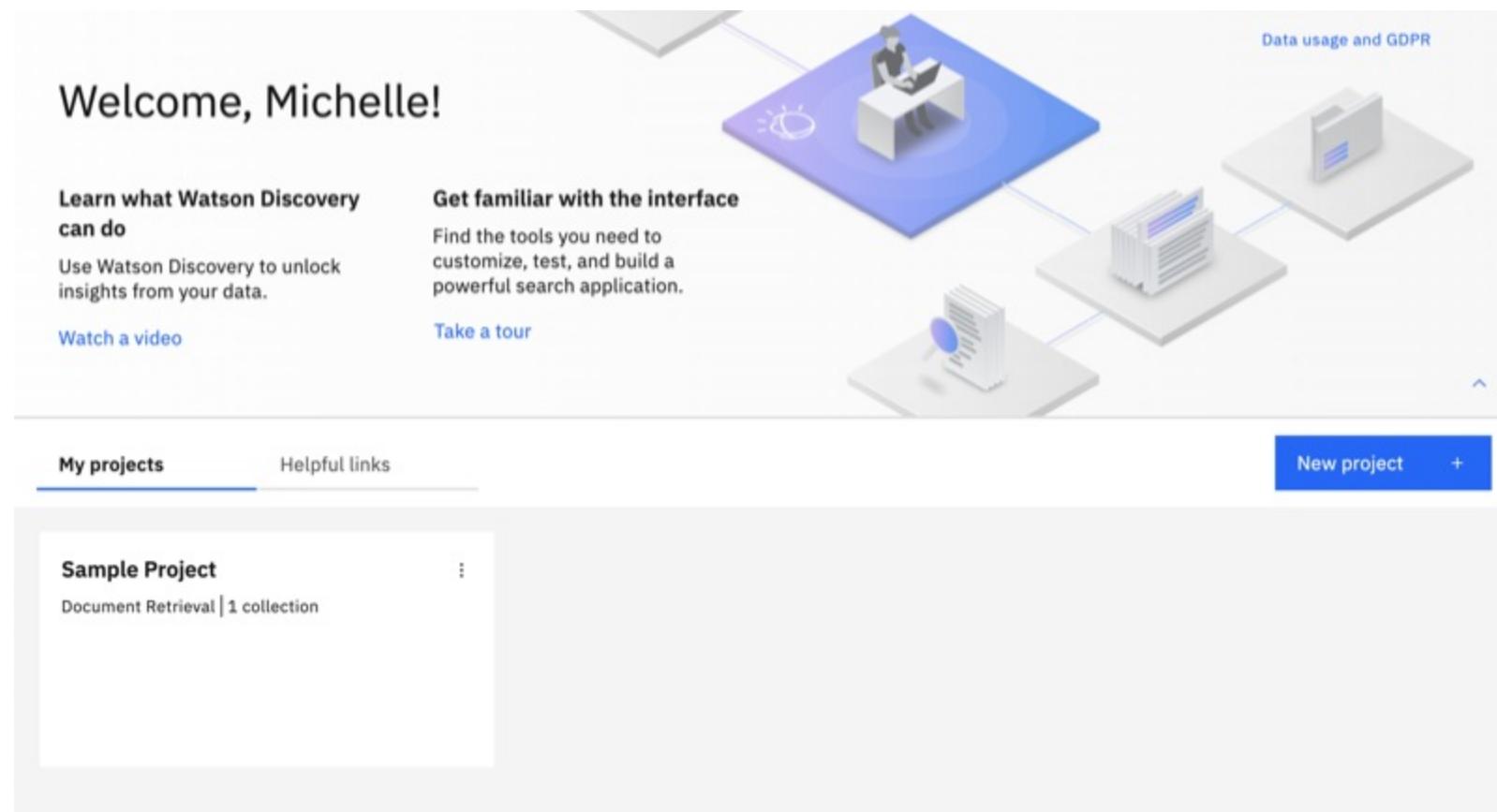


Figure 2. Discovery v2 home page

Can I integrate Watson Discovery with Watson Assistant?

You can integrate Discovery and Watson Assistant to make information that is stored in external data sources available to a virtual assistant. Create a ***Conversational Search*** project in Discovery, and then add the data sources that you want to make available to it. Next, create a ***search integration*** in Watson Assistant, and connect it to your Discovery project and collection.

Can I increase the collection limit for a project?

If you want to add more than 5 collections to your project and you have a Premium plan, you can request an increase to the collection limit by opening a support request. For more information, see [Getting help](#).

Can I find related documents after I add them to a collection

If you want to retain information about the relationship of two or more documents to one another, you can do so. For example, if 3 documents are uploaded from the same folder and their placement in the folder is significant to their meaning, you might want to retain the parent folder information.

When you upload a document, no such information about its relationships to other documents is stored by default. To add the information, you can use the API to add the documents. When you add documents by using the API, you can specify metadata values. You might want to specify a metadata value, such as `"filename": "company_a"`, for each document.

Alternatively, you can copy the document body of each document into a JSON file, where each document is an object in a single array. When the JSON file is ingested, each item in the array is added as a separate document with a separate document ID. Each document shares the same parent ID, which identifies the relationship between them.

You can quickly find documents that share the same parent ID or other common metadata value from the ***Manage data*** page. Customize the view to show the field, such as `extracted_metadata.parent_document_id` or `extracted_metadata.foldername`, that the documents share in common.

Can I customize Discovery to understand my data

Yes. Use the intuitive tools provided with the product to teach Discovery about the unique terminology of your domain. For example, you can teach it to recognize patterns, such as BOM or part numbers that you use, or add dictionaries that recognize your product names and other industry-specific word meanings. For more information, see [Adding domain-specific resources](#).

How does the Smart Document Understanding tool work?

You can use the Smart Document Understanding tool to teach Discovery about sections in your documents with distinct format and structure that you want Discovery to index. You can define a new field, and then annotate documents to train Discovery to understand what type of information is typically stored in the field. For more information, see [Using Smart Document Understanding](#).

What's the best way to add synonyms?

You can use two different methods to define synonyms.

- To define synonyms that are recognized and tagged when a document is ingested and that can be retrieved by search, create a dictionary and add synonyms for the dictionary term entry. A dictionary defines special terms that you want to tag in your documents, such as product names or industry-specific terminology. You can use the dictionary terms later to create facets and to filter documents. For more information, see [Dictionary](#).
- To define synonyms that are applied to the query text that is submitted by users to expand the meaning of the query, add synonyms by using the Synonyms tool on the **Improve relevance** section of the **Improve and customize** page. For more information, see [Expanding the meaning of queries](#).

Can I use Discovery to detect sentiment?

You can use Discovery to detect both phrase and document sentiment. Document sentiment is a built-in Natural Language Processing enrichment that is available for all project types. Document sentiment evaluates the overall sentiment that is expressed in a document to determine whether it is positive, neutral, or negative. Phrase sentiment does the same. However, phrase sentiment can detect and assess multiple opinions in a single document and, in English and Japanese documents, can find specific phrases. For more information about document sentiment, see [Sentiment](#). For more information about phrase sentiment, see [Detecting phrases that express sentiment](#). You cannot detect the sentiment of entities or keywords in v2.

What is a nested field?

When you ingest a file or crawl an external data source, the data that you add to Discovery is processed and added to the collection as a document. Fields from the original file are converted to document fields and are added to the collection's index. Some content is added to root-level index fields and some information is stored in nested fields. Where data gets stored differs by file type. Most of the fields from structured data sources are stored as root-level fields. For files with unstructured data, much of the body of the file is stored in the `text` field in the index. Other information, such as the file name, is stored in nested fields with names like `extracted_metadata.filename`. You can determine whether a field is a nested field by its name. If the field name includes a period, it is a nested field. For more information about how different file types are handled, see [How your data source is processed](#).

Which type of query should I use in my custom app?

When you submit a query, you can choose to submit a natural language query or use the Discovery Query Language to customize the search to target specific fields in the index, for example. For more information about the different types of queries and how to decide which one to use, see [Choosing the right query type](#).

Getting help

Get help to solve issues that you encounter when you use the product.

Use these resources to get answers to your questions:

- Walk through a guided tour to learn about a project type or a feature. Click **Guided tours** from the page header to see a list of available tours.
- For answers to frequently asked questions, see the [FAQ](#).
- Find answers to common questions or ask questions where experts and other community members can answer. Go to the [Watson Discovery Community forum](#).

IBM Cloud Contacting IBM Cloud Support for managed deployments

Managed deployments are deployments that are hosted on IBM Cloud, including IBM Cloud Pak for Data as a Service deployments.

If your service plan covers it, you can get help by opening a case from [IBM Cloud Support](#).

Be ready to share the following information with IBM Support:

Account information

- Account name or customer name.
- Business impact so IBM Support understands the urgency of the issue and can prioritize it.
- Case information for any related cases or a parent case.
- Cloud location where the service instance is hosted (Dallas, Frankfurt, and so on).
- Your service plan (Plus, Premium and so on).

Problem description

- What outcome were you expecting and what happened?
- Message text that is displayed when the error occurs, especially the document ID, if specified.
- Steps to take to reproduce the issue.
- Any screen captures that illustrate the problem.
- When did the problem occur?
- Instance ID. (The instance ID is part of the URL that is specified in the **Credentials** section of the service page on IBM Cloud. You can copy the full URL and provide that.)
- Collect and share the HTTP archive (HAR) file from your browser. The HAR file contains a log of trace information from within a browser session. It records web requests that are made by the browser to the website page including request and response headers, the body, and the time it takes to load the assets.
- If you are using the API, share example API calls, including the version parameter value that was specified, and the API response body.



Note: Do not share code examples. IBM Support cannot debug custom code.

- If the problem is related to a particular project or collection, provide the project ID and collection ID.
 - Project ID. (You can copy the Project ID from the **API Information** tab of the **Integrate and deploy** page in the product user interface.)
 - Collection ID, if you were able to create a collection. (To get the ID, open the **Manage collections** page, and then click the collection to open it. From the web browser location field, scroll to the end of the URL. Look for the **collections/** section, and then copy the ID that is displayed after it. For example, in the URL **/collections/5a525eb7-b175-3820-0000-017d00f0fc1/activity**, the collection ID is **5a525eb7-b175-3820-0000-017d00f0fc1**.)
- If the problem has to do with documents failing to load, provide the following information if known:
 - What kind of documents are being uploaded (such as PDF, Json, CSV). Was optical character recognition (OCR) enabled for the collection?
 - How were the documents loaded into the collection? (using the API, from the product UI, data source connector)
 - Did you identify fields in the collection by using Smart Document Understanding? If so, what type of SDU model was applied to the collection (user-trained or pretrained)?
 - What enrichments were applied to the collection?
- If the problem is related to a particular document (of a small set of documents), provide the **document_id** of the document, if known. You can share example documents if they might be helpful.

- If the problem is related to querying documents, describe the kind of query being used.

IBM Cloud Pak for Data Contacting IBM Support for installed deployments

Installed deployments are deployment that you provision on IBM Cloud Pak for Data.

You can get help by opening a case from IBM Support from [IBM Support](#).

Be ready to share the following information with IBM Support:

Account information

- Account name or customer name.
- Business impact so IBM Support understands the urgency of the issue and can prioritize it.
- Case information for any related cases or a parent case.
- Software versions of both the Discovery service version and IBM Cloud Pak for Data version.
- Relevant details about configuration choices that were made during installation and deployment.

Problem description

- What outcome were you expecting and what happened?
- Message text that is displayed when the error occurs, especially the document ID, if specified.
- Steps to take to reproduce the issue.
- Any screen captures that illustrate the problem.
- When did the problem occur?
- Instance ID. (From the IBM Cloud Pak for Data web client main menu, expand **Services**, and then click **Instances**. Find your instance, and open its summary page. Scroll to the **Access information** section of the page, and then copy the URL. The instance ID is part of the URL. You can provide the full URL to IBM Support.)
- If you are using the API, share example API calls, including the version parameter value that was specified, and the API response body.



Note: Do not share code examples. IBM Support cannot debug custom code.

- Relevant logs, including the Red Hat OpenShift collector logs.

The IBM Support representative can share a script with you that collects relevant logs from your cluster.

- If the problem is related to a particular project or collection, provide the project ID and collection ID.
 - Project ID. (You can copy the Project ID from the **API Information** tab of the **Integrate and deploy** page in the product user interface.)
 - Collection ID, if you were able to create a collection. (To get the ID, open the **Manage collections** page, and then click the collection to open it. From the web browser location field, scroll to the end of the URL. Look for the **collections/** section, and then copy the ID that is displayed after it. For example, in the URL `/collections/5a525eb7-b175-3820-0000-017d00f0fc1/activity`, the collection ID is `5a525eb7-b175-3820-0000-017d00f0fc1`.)
- If the problem has to do with documents failing to load, provide the following information if known:
 - What kind of documents are being uploaded (such as PDF, Json, CSV). Was optical character recognition (OCR) enabled for the collection?
 - How were the documents loaded into the collection? (using the API, from the product UI, data source connector)
 - Did you identify fields in the collection by using Smart Document Understanding? If so, what type of SDU model was applied to the collection (user-trained or pretrained)?
 - What enrichments were applied to the collection?
- If the problem is related to a particular document (of a small set of documents), provide the `document_id` of the document, if known. You can share example documents if they might be helpful.
- If the problem is related to querying documents, describe the kind of query being used.

Feedback

We value your opinion and want to hear it.

Product feedback

How you share product feedback differs depending on the type of deployment you are using.

IBM Cloud

From the page header, click **Share feedback** to open a simple feedback form. You can share your opinion and submit it to the product team for consideration. We appreciate your insights and suggestions.

IBM Cloud Pak for Data

To share ideas or suggest new features for the Discovery service on IBM Cloud Pak for Data, go to the [IBM Data and AI Ideas Portal for Customers](#)

Product documentation feedback

Give us feedback about the product documentation. A **Feedback** button is displayed along the edge of each page. Click it to open a form where you can rate the current topic and share a comment.

Plan information

Discovery pricing plans

Learn more about the IBM Watson® Discovery service plans, so you can pick the plan type that best meets your needs.

Plus

Find targeted answers and insights across many document types.

The first Plus plan that is created includes a 30-day trial at no cost. You must pay for each additional Plus plan (and for use of the first plan after the 30-day trial ends).

 **Important:** If you continue to use a Plus plan service instance after the 30-day trial ends, you are charged for it. To avoid being charged after 30 days, you must delete the Plus plan service instance.

What's included

For subscription accounts, the monthly fee per instance is \$500.

The plan includes the following features:

- 10,000 documents per month (\$50 for every additional 1,000 documents per month)
- 10,000 queries per month (\$20 for every additional 1,000 queries per month)
- Up to 5 queries per second
- Optical Character Recognition(OCR)
- Out of the box data source connectors
- Table Retrieval
- Custom NLP models
- Custom Relevance Model
- Entity extractor

Artifact limits

- Up to 500,000 queries per month
- Up to 500,000 documents
- Up to 20 projects
- Up to 40 collections (Up to 5 collections per project)
- Up to 10 MB document size
- Up to 3 custom models [Learn more](#)
- Up to 40 custom fields for Smart Document Understanding model and model import and export
- Up to 20 custom dictionaries
- Up to 20 custom pattern extraction models
- Up to 20 custom regular expression models
- Up to 20 custom text classification models

Enterprise

Scale and secure your Discovery application with enterprise-grade support and performance, and address more use cases including contract analysis and content mining to explore insights across documents.

What's included

For subscription accounts, the monthly fee per instance is \$5,000.

The plan includes the following features:

- 100,000 documents per month (\$5 for every additional 1,000 documents per month)
- 100,000 query or analyze API calls per month (\$5 for every additional 1,000 calls per month)
- Up to 3 custom models [Learn more](#)
- Up to 10 entity extractor models
- Up to 5 queries per second
- Everything that's available in Plus
- Analyze API
- Content Intelligence
- Content Mining
- Document classification (Text classification is available in all plans. Document classification is available with Enterprise and

higher-level plans only.)

 **Note:** Your bill labels requests that are generated from both query searches and analyze API calls as "Queries".

Artifact limits

- Unlimited queries per month
- Unlimited documents
- Up to 100 projects
- Up to 300 collections (Up to 5 collections for all project types except Content Mining, which supports 1)
- Up to 10 MB document size
- Up to 10 custom Knowledge Studio models
- Up to 10 entity extractor models
- Up to 100 custom fields for Smart Document Understanding model and model import and export
- Up to 100 custom dictionaries
- Up to 100 custom pattern extraction models
- Up to 100 custom regular expression models
- Up to 20 custom text classification models
- Up to 20 custom document classification models

 **Note:** Autocompletion, curation, and notice requests are not billed in any plan type.

How is document pricing calculated?

IBM uses the maximum number of documents per day and then prorates the document price to calculate the cost for a month.

For example, imagine that 300,000 documents are added to a collection in the first 6 days (6/30) of the month, and then another 1 million documents are added over a two-day span. In the first day (1/30) of the two-day span, 700,000 documents are added. All of the documents (1 million + 300,000 = 1,300,000) remain in the collection index for the rest of the month (23/30). The number of documents for the month might be calculated by using an equation like this:

$$300K * (6/30) + 700K * (1/30) + 1300K * (23/30)$$

Document pricing counts the number of indexed documents in each collection. If you reuse data in a second collection, it generates a second set of documents in the index, which are counted separately.

For more information about what counts as a document, see [Document limits](#).

What is a custom model?

Custom models include any of the following model types:

- Discovery entity extractor
- Knowledge Studio machine learning
- Knowledge Studio rule-based

 **Note:** Advanced rules models are not counted as custom models.

Discovery entity extractor models that are trained and published as an enrichment count toward the model limit for the plan. The entity extractor enrichment is what incurs charges, not the workspace or model. The entity extractor enrichment incurs charges whether or not it is applied to a collection.

Premium

Premium plans offer developers and organizations a single tenant instance of one or more Watson services for better isolation and security. These plans offer compute-level isolation on the existing shared platform, as well as end-to-end encrypted data while in transit and at rest.

For more information, or to purchase a Premium plan, contact [Sales](#).

Other plans

Features and limitations are similar between instances that you deploy on installed deployments in IBM Cloud Pak for Data and Premium plan instances that are managed by IBM Cloud.

Additional information

- For more information about how queries are counted, see [Query limits](#).
- For more information about pricing or to create a service instance, see the [IBM Cloud catalog](#).

IBM Cloud resources:

- [How you're charged](#).
- [IBM Cloud Cost estimator](#)
- [IBM Cloud Services terms](#)
- For more information about IBM Cloud security, see [IBM terms](#).

Upgrading

Learn how to upgrade your service plan.

IBM Cloud **IBM Cloud only**

 **Note:** This information applies only to managed deployments. For more information about upgrading an installed deployment that is hosted by IBM Cloud Pak for Data, see [Upgrading the service](#).

Upgrading your plan

You can explore the Discovery [service plan options](#) to decide which plan is best for you.

The page header shows the plan you are using today.

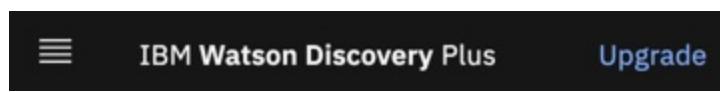


Figure 1. Plus plan is displayed in the page header

You cannot upgrade from any v1 plan to any v2 plan. For example, you cannot upgrade a Lite plan to a Plus, Enterprise, or Premium (v2) plan. And you cannot upgrade an Advanced, Partner, Standard, or Premium (v1) plan to an Enterprise or Premium (v2) plan. To start using v2, create a new Plus, Enterprise, or Premium plan.

For information about upgrading from a Lite to an Advanced v1 plan, see [Upgrading your service](#) in the v1 documentation.

Even though you can use the Plus plan for the first 30 days at no charge, you must have a paid account to create a Plus plan. For more information about creating a paid account, see [Upgrading your account](#).

1. How you upgrade depends on your plan.

- If you decide you want to keep the Plus plan after using the 30-day free trial, no action is required.

After 30 days of using the Plus plan at no cost, you are charged for it.

- If you decide you do **not** want to continue using the Plus plan, delete the Plus plan service instance before the 30-day trial period ends. You can delete the service instance from the [IBM Cloud Resource list](#).

The number of days that are left in your trial is displayed in the page header.

- To upgrade a Plus plan to an Enterprise plan, complete the following steps:

- Open the service page for your Plus plan service instance from the [IBM Cloud Resource list](#).
- Click **Upgrade**.
- Choose the Enterprise plan, and then click **Save**.
- Give the upgrade process time to finish.

The time it takes to convert the plan varies depending on the amount of data in your existing Plus plan service instance. It takes at least 20 minutes and, for instances with large amounts of data, can take more than a day to complete. The IBM Cloud page does not show progress information and doesn't indicate when the plan upgrade process is finished. To check whether the new plan is in effect, you must refresh the service instance overview page, and then check for the new plan name to be displayed in the **Plan** tile.

During the plan upgrade process, you can continue to submit search queries in your existing projects. However, avoid the following actions:

- Adding new projects or collections
- Deleting or changing existing collections, including adding documents, editing fields, and changing enrichment settings.
- If you are creating an Enterprise plan in the same data center location where you have an existing Premium plan, you must create a new resource group for the new plan. You cannot use the same resource group for Enterprise and Premium plans

that are hosted in the same location. For more information, see [Managing resource groups](#).

- You cannot do an in-place upgrade from a Plus or Enterprise plan to a Premium plan.

A Premium plan instance must be provisioned for you. To start the process, contact [Sales](#). You will be asked to provide the following details:

- Customer name
- Customer email
- Planned deployment date
- Data center location, such as Dallas or Frankfurt
- Account ID
- Resource group name
- Resource group ID

The resource group is created by the account holder. For more information, see [Managing resource groups](#).



Important: You cannot directly downgrade from one plan to another. If you want to move from an Enterprise plan to a Plus plan, for example, you must provision a new Plus plan and then move data to it from your existing Enterprise plan. After the data is moved, you can delete the Enterprise plan. For more information about how to back up data that you want to move between service instances, see [High availability and disaster recovery](#).

For more information about plans, see [Discovery pricing plans](#).

Accessibility

IBM strives to provide products with usable access for everyone, regardless of age or ability.

You can interact with all functions of the IBM Watson® Discovery content by using only the keyboard.

For more information about the accessibility compliance of the product, go to the [Product Accessibility Conformance Reports](#) website, and then search for IBM Watson® Discovery.

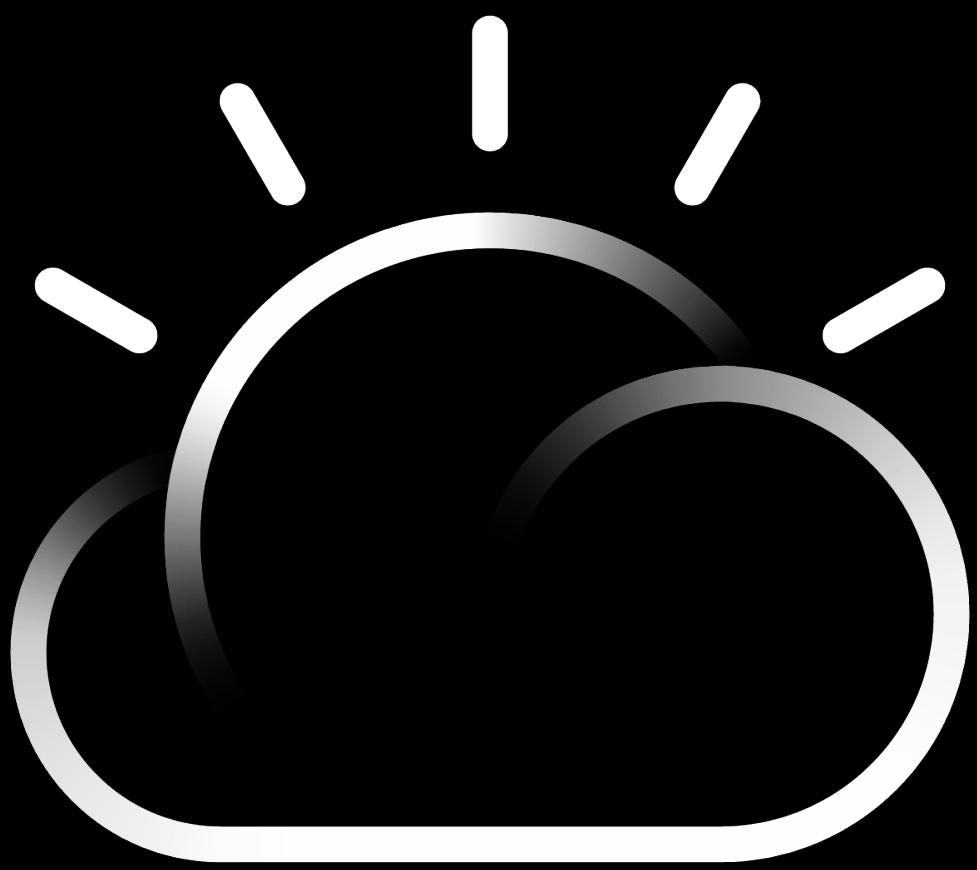
Accessibility features in the product documentation

Accessibility features help people with a physical disability, such as restricted mobility or limited vision, or with other special needs, to use information technology products successfully.

The accessibility features in this product documentation allow users to do the following things:

- Use screen-reader software and digital speech synthesizers to hear what is displayed on the screen. Consult the product documentation of the assistive technology for details on using assistive technologies with HTML-based information.
- Use screen magnifiers to magnify what is displayed on the screen.
- Operate specific or equivalent features by using only the keyboard.

The documentation content is published in the IBM Cloud Docs site. For more information about the accessibility of the site, see [Accessibility features for IBM Cloud](#).



IBM Cloud