IBM Cloud Pak for Business Automation Demos and Labs 2025

Configuration and Usage of IBM Content Assistant on Cloud Pak for Business Automation

V 0.3s (for CP4BA 25.0.0)

Matthias Jung, Ph.D.
matthias.jung@de.ibm.com

NOTICES

This information was developed for products and services offered in the USA.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

TRADEMARKS

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

IT Infrastructure Library is a Registered Trade Mark of AXELOS Limited.

ITIL is a Registered Trade Mark of AXELOS Limited.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

© Copyright International Business Machines Corporation 2021.

This document may not be reproduced in whole or in part without the prior written permission of IBM.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Table of Contents

1 Introduction 4
1.1 IBM Content Assistant 4
1.2 Lab Overview 4
1.3 Lab Setup Instructions 5

2 Exercise: End-user interface 7
2.1 Introduction 7
2.2 Exercise Instructions 7
2.2.1 IBM Content Assistant Chat window 7
2.2.2 Single Document Queries in Browse View 10
2.2.3 Multi Document Queries in Browse View and Searches 12
2.2.4 Single Document Queries from the DaeJa Viewer 16
2.2.5 Running Queries against an Object Store 18

3 Exercise: Administration 20
3.1 Introduction 20
3.2 Exercise Instructions 20
3.2.1 Content Assistant Addon and Configuration 20
3.2.2 Configuring Persistent Storage of Text Extraction 22
3.2.3 Vector Indexing and Summary Generation 28

4 Exercise Application Programming Interface 31
4.1 Introduction 31
4.2 Exercise Instructions 31
4.2.1 Base Query Class 31
4.2.2 Using Base Query Class from GraphQL 33
4.2.3 Queries with One or More Documents 36

5 Troubleshooting 40
5.1 GENAI_QUERY_FAILED 40
5.2 Body has already been consumed. 40
5.3 Subscription contract limits 40

6 Sample Graphql for the GenaiAdhocSummary query class 41

1 Introduction

1.1 IBM Content Assistant

IBM Content Assistant provides the possibility to use Artificial Intelligence to derive information and insights from documents contained in an IBM FileNet Content Engine Object Store. It does so in a secure way, by making sure that any user running queries will get information only from those documents of the Object Store, to which this user has access.

The IBM Content Assistant architecture is composed of two primary components. The server-side component is responsible for transforming documents into vector embeddings and executing semantic search queries against a vector index. It also orchestrates interactions with the underlying Large Language Model (LLM), specifically IBM Watsonx.

Document ingestion involves converting unstructured content into high-dimensional vector representations, enabling efficient similarity search and contextual retrieval. These embeddings are stored in a vector database, which currently operates as a SaaS-managed service.

Future roadmap considerations include support for on-premises deployment of the vector database, allowing customers to maintain full control over data locality and compliance requirements.

In the Cloud Pak for Business Automation environment, integration with IBM Content Assistant Server is facilitated through specialized FileNet Object Store Add-ons. One add-on manages communication with the Content Assistant server, transmitting extracted document text for processing and persisting AI-generated metadata returned by the server.

For each document, the following artifacts may be stored:

- **Indexing metadata**: Status and timestamp indicating if and when the document was ingested into the vector index.
- **AI-generated summary**: A concise abstract generated by the Watsonx LLM.

To improve performance and reduce redundant processing, an additional Object Store Add-on can persistently store the extracted text of documents. This enables reuse of the text, e.g. for indexing with the Content Search Services or Open Search components.

IBM Content Assistant capabilities are exposed via API extensions integrated into the FileNet Content Engine. These are available through both the Java API and a GraphQL interface, supporting flexible integration into custom applications and automation workflows.

For more information, refer to the Overview section in the Knowledge Center for IBM Content Assistant, on https://www.ibm.com/docs/en/content-assistant?topic=assistant-overview-content

1.2 Lab Overview

This lab demonstrates the key use-cases of IBM Content Assistant. The exercises serve as an example. The API exercises are moreover inspired by the needs of Automation projects such as the Client Onboarding application showcased in other labs.

The Content Assistant lab can be performed independently of the other Content labs, i.e. independently of "Setting up FileNet Content Platform Engine for Automation Projects on Cloud Pak for Business Automation",

"Interfacing FileNet Content Platform Engine with GraphQL on Cloud Pak for Business Automation" and "Introduction to Business Automation Navigator in Cloud Pak for Business Automation"

Exercise "End-user interface" demonstrates the user interface of IBM Content Assistant in IBM Content Navigator aka IBM Business Automation Navigator. It shows how IBM Content Assistant presents itself to the end-users and how it can be used to have the Watsonx AI assistant determine facts from documents in the repository, without the need to open the documents and search for the information. Questions on single documents can be raised directly from the Browse tool of IBM Content Navigator or from the open DaeJa Viewer. Further, questions can also be asked on the whole repository, and it is important that ICA will take into account only the documents, to which a given user has access to.

The "**Administration**" exercise provides a deep dive into the administrative aspects of IBM Content Navigator, primarily managed through ACCE (Administrative Console for Content Engine). It focuses on the configuration and behavior of the GenAI Extensions Add-on, which enables vector indexing and AI-based summarization for both newly ingested and existing documents via sweeps and event subscriptions.

The exercise demonstrates how to configure:

- **Automatic Text Extraction and Persistent Storage** using a dedicated Object Store Add-on to cache extracted text for reuse.
- **Automated Vector Indexing** to register documents in the vector database.
- **Automated Summarization** using Watsonx to generate document abstracts.

It also shows where these artifacts—such as indexing metadata, summaries, and extracted text—are stored within the FileNet repository, offering insight into the internal workings of IBM Content Assistant.

The last **Exercise "Application Programming Interface"** demonstrates to the reader the possibility, to create IBM Content Assistant queries through the APIs of FileNet Content Manager. As it would be out of scope to create and run Java programs (for now), the exercise uses the GraphQL interface for some hands-on examples, which can efficiently be adapted to the needs of any automation project.

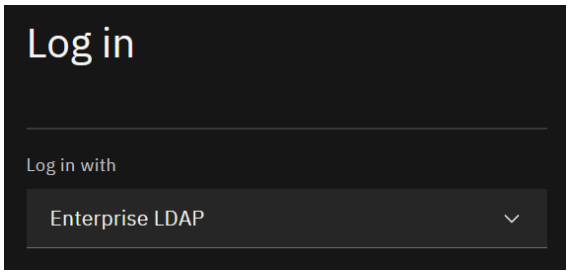
1.3 Lab Setup Instructions

_1. If you are performing this lab as a part of an IBM event, access the document that lists the available systems and URLs along with login instructions. For this lab, you will need to access **GraphiQL**, the **ACCE** and the **IBM Content Navigator ICN desktop.** Further, you will need to have access to a second account named **icademo**, those details should be listed under **ICA Lab Account.**

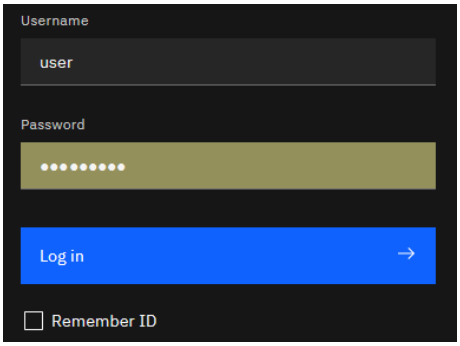
   If you are performing this lab self-paced on a Cluster, which has been enabled for the Client Onboarding labs including the ICA component, you can find the available URLs in the ConfigMap "000-client-onboarding-information" of the "cp4ba" project in the OpenShift Web Console.

_2. The different exercises require different components of the IBM Cloud Pak for Business Automation. Paste the URL used by the lab into your web browser, the first exercise will use the **IBM Content Navigator ICA desktop.**

_3. If you haven't logged on before, the browser will be redirected to a Login page. Select Enterprise LDAP login option



_4. Enter the *Username* and *Password* which were supplied to you, click **Log in**. Only when stated explicitly in the lab instructions, use **ICA Lab Account** details**.**



_5. The chosen component appears in the browser. If you used the **IBM Content Navigator ICN desktop**, it would be IBM Content Navigator showing the Home page with a tile for each feature.

2 Exercise: End-user interface

2.1 Introduction

This exercise demonstrates the user interface of IBM Content Assistant in IBM Content Navigator aka IBM Business Automation Navigator. It shows how IBM Content Assistant presents itself to the end-users and how it can be used to have the Watsonx AI assistant determine facts from documents in the repository, without the need to open the documents and search for the information.

Questions on single documents can be raised directly from the Browse or Search tools of IBM Content Navigator or from the open DaeJa Viewer. IBM Content Assistant makes itself available as a Search Assistant Chatbot in the lower right corner of the Browser window. Through the “Ask Your Repository” tool, questions can also be asked on the whole repository, and it is important that ICA will consider only the documents, to which a given user has access to.

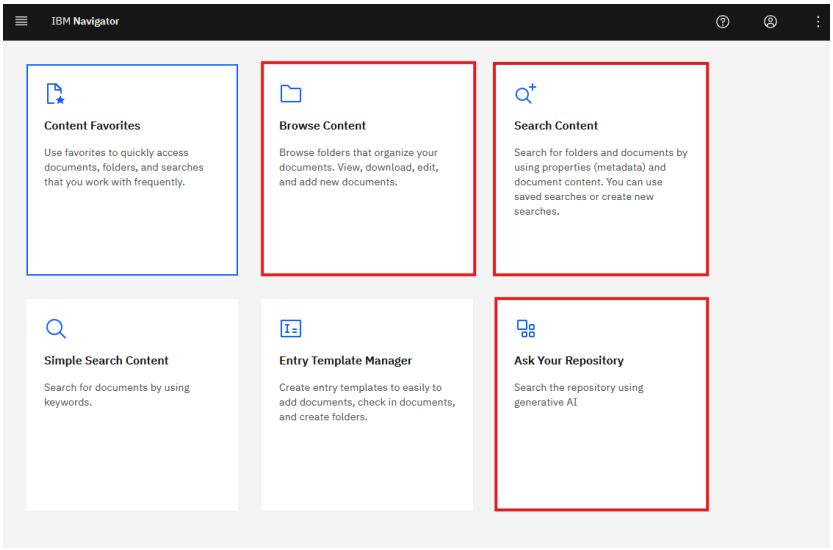
Subsection 2.1.1 introduces the ICA Chat window and its components. The following two subsections focus on executing queries on single and multiple documents. Subsection 2.1.4 demonstrates how to run queries directly from the DaeJa Viewer. This is relevant because IBM Content Navigator is frequently embedded into other applications via bookmark.jsp URLs, which open the DaeJa Viewer for a specific document. In such cases, the IBM Content Assistant Chat Window remains accessible from within the DaeJa Viewer.

The last subsection shows how to ask questions against the complete repository and demonstrates that ICA will only use those documents to derive an answer for a question, to which a given user has access to.

2.2 Exercise Instructions

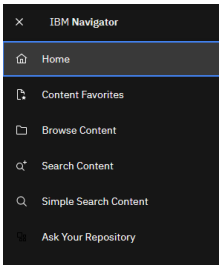
2.2.1 IBM Content Assistant Chat window

_1. After logging in, the IBM Content Navigator Desktop comes up, showing tiles for the Features, which are available on this desktop. For the ICN desktop the following tiles are configured. For IBM Content Assistant foremost the features “Search Content”, “Browse Content” and “Ask Your Repository” are relevant.



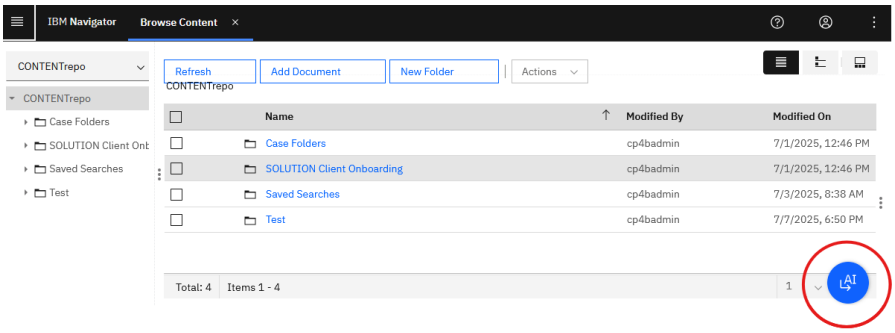
_2. The features are also available through the so-called “Hamburger Menu”, this is the icon at the top left corner of IBM Content Navigator. It is called that way because it looks a bit like a Hamburger burger, with some slices of beef in between. For faster switching, use the Hamburger menu.

Switch to the “Browse Content” feature.

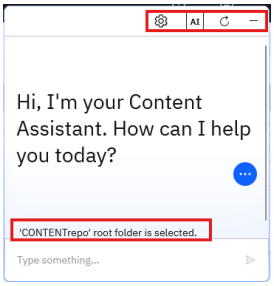


_3. When you select it, the “Browse Content” feature shows the root folder of a FileNet Object Store. In this case this is the only Object Store selected to be available through the desktop, the “CONTENT” Object Store. It is called “CONTENTrepo” repository in IBM Content Navigator.

On the lower right corner, you see a round blue button for invoking an AI assistant. When you use it from the Browse Content Feature, that is the IBM Content Assistant. Other features of Content Navigator might be using other assistants, like the Workflow Assistant when using the Workplace feature.



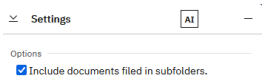
_4. Click on the AI button to bring up IBM Content Assistant Chat



Observe the information "'CONTENTrepo' root folder is selected. It is indicating that on a query all documents in the root folder will be passed as context.

_5. Explore the Chat window by clicking on the toolbar buttons at the top of the chat window.

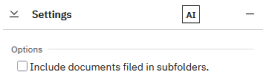
In the "Settings" pane reachable through the wheel icon it is possible to choose, if selecting a folder shall include the containing documents with the search prompt. By default, it is enabled.



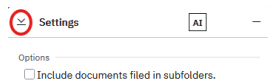
By clicking on the "AI" button, a popup window appears which gives information about what AI model is used in this instance of IBM Content Assistant. Here information that IBM Watsonx is being used, is shown.

The last two buttons start a new chat and minimize the chat window.

_6. Use the "Settings" pane to disable the features to include documents in selected folders into the search query, which is sent to the AI.



Click on the symbol in the upper left corner to return to the chat window.

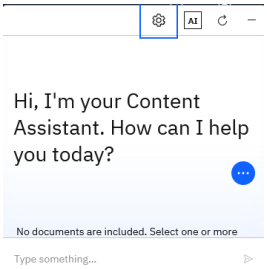


Commented [JR1]: In later steps when the user deselect the option to search into sub folders you can see at this prompt is not here "CONTENTrepo" root folder is selected". I think in this step we should call out what this prompt means so that later it makes more sense.

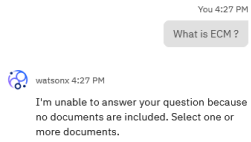
Commented [JR2]: There is an "Include documents filed in subfolders." option that is selected but that is not what I understood from the instructions. We might be better off with similar wording in the lab guide.

Commented [JR3]: It would be good if you showed the symbol and a red square around it. Also you are saying that no documents are selected but I did not know that this was shown now in the chat. The sentence just did not pop up for me. It would be good to highlight the sentence in the image with a red square and tell participants to look at the prompt now saying "No documents are included". All I am saying is that it did not come across as clear for me as to how you came to the conclusion that something had changed.

Notice that now no documents are included with any AI query.



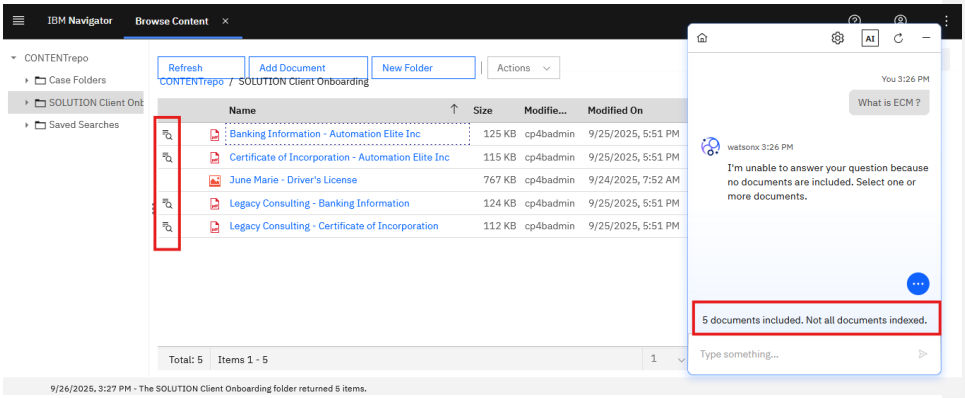
_7. Type in a prompt now. It really doesn't matter what you write, you might be asking "What is ECM ?" IBM Content Assistant will refuse to answer, as no documents are included with the query.



Leave the Chat window open for the next subsection.

2.2.2 Single Document Queries in Browse View

_1. In the navigation area on the left side of IBM Content Navigator, select the "SOLUTION Client Onboarding" folder.



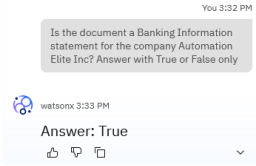
_2. Notice in the Browse view that some documents have an icon on the left side. This icon indicates that the document is vector indexed and can be used in the context of IBM Content Assistant.

_3. Notice further that in the IBM Content Assistant Chat window, now the 5 documents contained in that folder are selected for being included in the search query. If no documents are selected in the browse view, automatically all documents will be included into any search prompt.

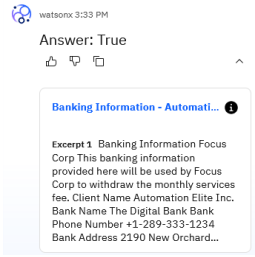
_4. Click on the first document with the title “Banking Information - Automation Elite Inc”. Avoid clicking on the blue link, as this would open the DaeJa Viewer. Notice the change in the IBM Content Assistant window.

_5. Type in (or copy & paste) the question “Is the document a Banking Information statement for the company Automation Elite Inc? Answer with True or False only.”

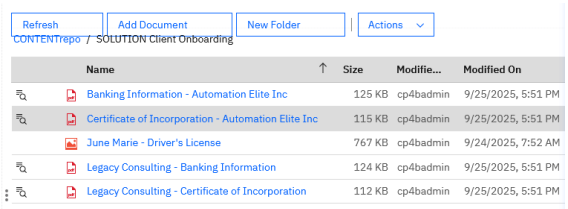
> When postprocessing answers from ICA in a custom application, it’s a good idea to specify what result would be expected.



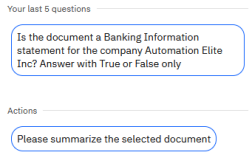
_6. Click the Thumbs up as the answer is correct. Then click the arrow pointing downwards to show the parts of the document from which the answer was derived.



_7. In the Browse View, click on the Certificate of Incorporation of Automation Elite. This will deselect their Banking Information document.

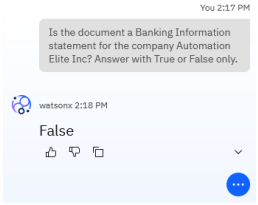


_8. In the ICA Chat window, click the blue circle with the three dots to bring up the suggestions pane. It contains possibly useful further queries or requests, along with a history of your past queries.



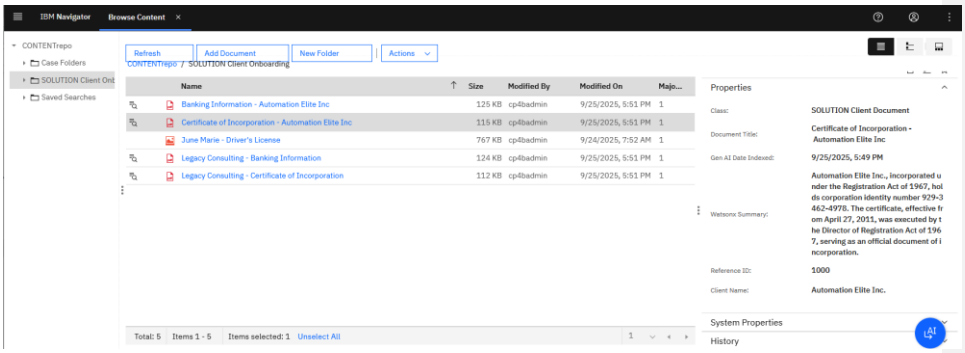
_9. Click on the previously executed query to run it again, but now on the Certificate of Incorporation. The result will now be False. Click the thumbs up again.

Commented [JR4]: I got the following answer instead. "Subscription contract limit for the number of users has exceeded, additional packages can be purchased."

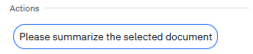


_10. Close the AI Chat window using the icon on the top right corner. Open the properties pane on the right side of the Content Navigator screen to access Document properties. Find a summary of the document generated by IBM Content Assistant.

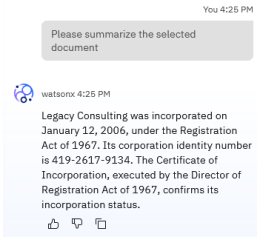
Commented [JR5]: Just a nit. In this step we are showing the Doc properties and the summary for the Legacy Consulting - Certificate of Incorporation document. In the previous step we were working with the Certificate of Incorporation - Automation Elite. In no step did we tell the user to select the Certificate for Legacy Consulting.



_11. An alternative way to access the information would be to open the AI Chat, then click the “Suggestions” button in the ICA Chat window (the blue circle with three dots at the lower end) and invoke the “Summarize the selected document” action.



Clicking the action gives the summary



2.2.3 Multi Document Queries in Browse View and Searches

Often ICA will be used to derive information from multiple documents or even many documents. A multi-document query can include up to 1000 documents. Beyond that, a query would better be run against the complete Object Store.

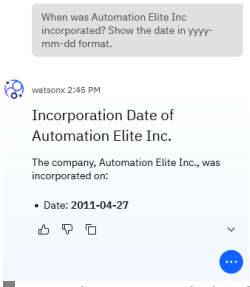
_1. In the Browse View, select both Certificates of Incorporation, by clicking the first, then pressing and holding the Ctrl key[1] and click the second. Remember to avoid the blue links when selecting documents for the AI Chat.

Commented [JR6]: In a Mac it is the Command key instead of the Ctrl key.

	Name	Size	Modifie...	Modified On	Majo...
	Banking Information - Automation Elite Inc	125 KB	cp4badmin	9/25/2025, 5:51 PM	1
	Certificate of Incorporation - Automation Elite Inc	115 KB	cp4badmin	9/25/2025, 5:51 PM	1
	June Marie - Driver's License	767 KB	cp4badmin	9/24/2025, 7:52 AM	1
	Legacy Consulting - Banking Information	124 KB	cp4badmin	9/25/2025, 5:51 PM	1
	Legacy Consulting - Certificate of Incorporation	112 KB	cp4badmin	9/25/2025, 5:51 PM	1

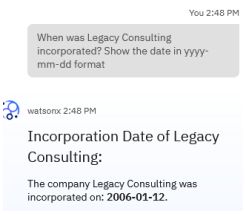
_2. In the Chat Window type in or copy & paste this question: "When was Automation Elite Inc incorporated? Show the date in yyyy-mm-dd format."[2]

Commented [JR7]: When I asked the question as stated in the Lab guide I got the the response as seen in the image I added below.
Not that the date was not formatted as expected.



> Note that ICA needed to find the right document which was relevant to answer the provided question. The format of the result can vary.

_3. Copy and Paste the same question for Legacy Consulting: "When was Legacy Consulting incorporated? Show the date in yyyy-mm-dd format"

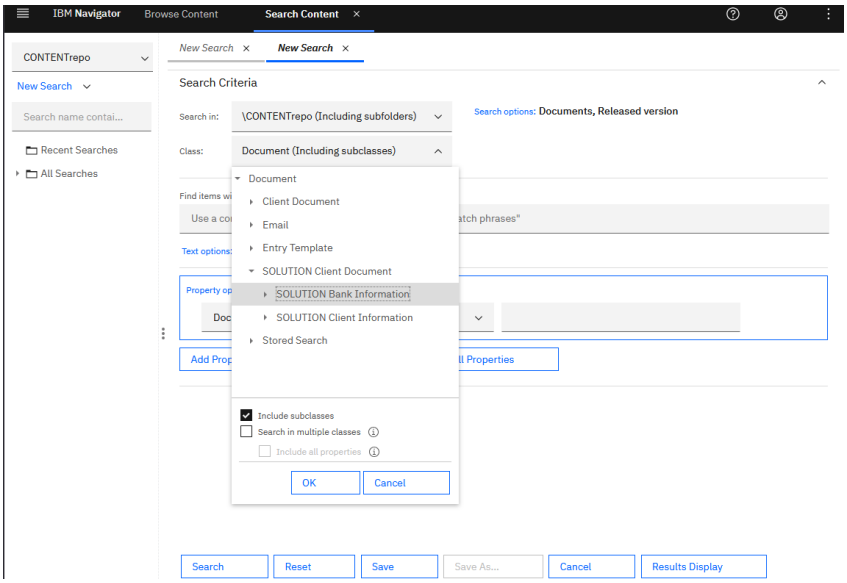


_4. Explore different queries for finding the incorporation dates for all companies, such as: "When were the companies incorporated? Show the company names and incorporation dates as a table." Or "For which companies do we have certificates of incorporation? Show the incorporation dates."

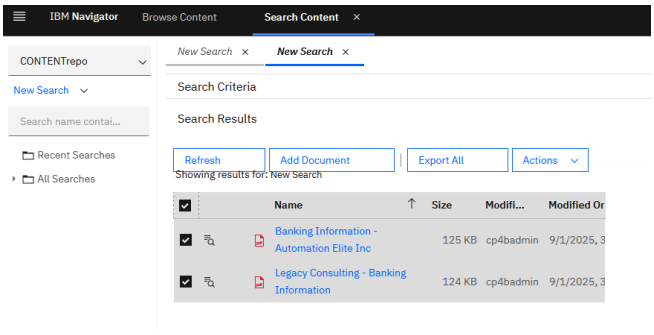
[1] In a Mac it is the Command key instead of the Ctrl key, and on German PCs the key has the label Strg
[2] Answers from the IBM Content Assistant might vary, influenced by many factors. You can experiment with varying the query, until getting the response in the correct format.

_5. IBM Content Assistant can also be used with results of a search. To try it out, In Content Navigator now use the Hamburger Menu on the top left corner to select the "Search Content" feature. Click on "New Search", then select the Solution Bank Information document class, as shown below. Click "OK" to close the class selection window.

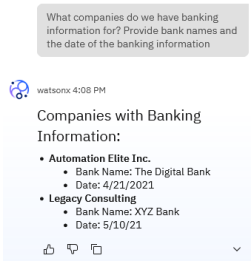
Commented [JR8]: Should probably tell the user to press the OK button.



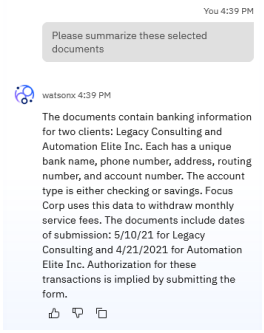
_6. Click the "Search" button to run the search. In the Search Results select all documents. To do that, click the first document, then press and hold the Shift key and click the last document returned from the search.



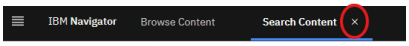
_7. Devise a query which will list the company names along with the names of their banks, such as “For which companies do we have banking information?” or “What companies do we have banking information for? Provide bank names and the date of the banking information.”



_8. Within the ICN Chat window now open the suggestions by clicking on the blue circle with three dots at the bottom right corner. Select the query “Please summarize these selected documents”. IBM Content Assistant will provide an Ad-hoc summary for the selected documents. Find more information in the knowledge on this page: https://www.ibm.com/docs/en/content-assistant?topic=uca-accessing-watsonx-ad-hoc-summary-your-document-selection



_9. Click on the small x icon on the right side of the “Search Content” tab in the Navigator title bar, which will close the “Search Content” tab. The previously active “Browse Content” tab is displayed again.



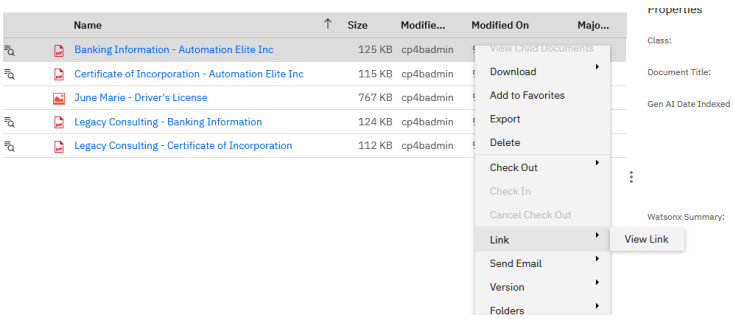
The ICA Chat window can be closed. Leave IBM Content Navigator window open.

2.2.4 Single Document Queries from the DaeJa Viewer

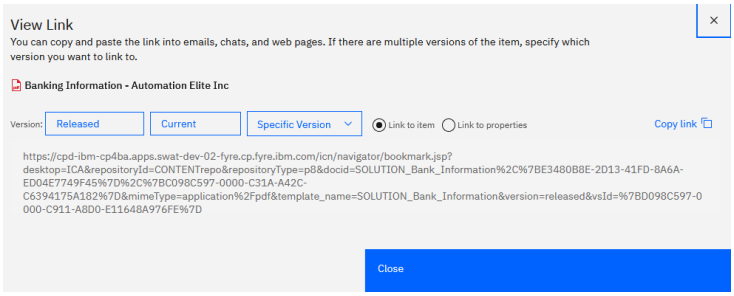
Business Applications including the Client Onboarding Application often embed the DaeJa Viewer. Technically this is done by invoking IBM Content Navigator with some additional parameters, which will make it open a specific document with the DaeJa Viewer, instead of opening the complete Content Navigator window. The good news is, that IBM Content Assistant can also be used from that DaeJa Viewer window.

_1. In the Browse view, select only the Banking Information document for Automation Elite Inc. Then open the Actions menu and select “View link” from the “Link” menu entry.

Commented [JR9]: How did we get from the Search View into now the Browse view? We should include instructions to get to this view again.

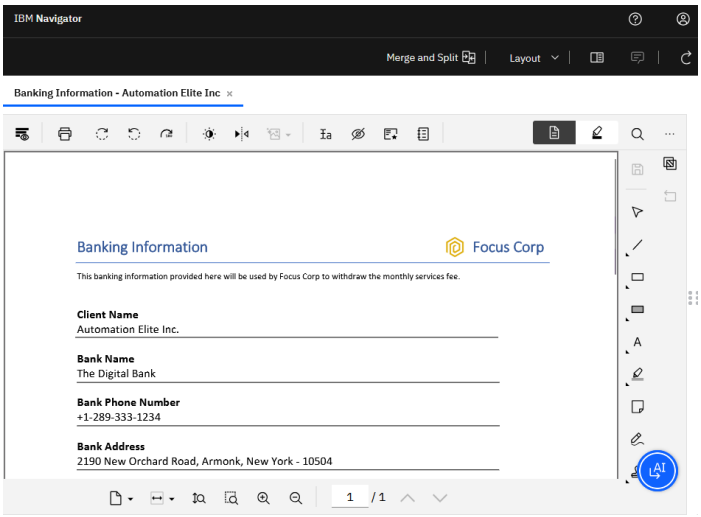


_2. The “View Link” window opens. Don’t change any options and click on “Copy Link”. Then close the window.

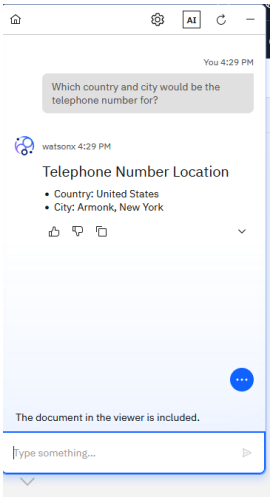


_3. Open a new Tab in your browser and paste the link into the address bar of the browser tab. The DaeJa Viewer will open with the selected document, and the ICA chat window button is showing in lower right corner.

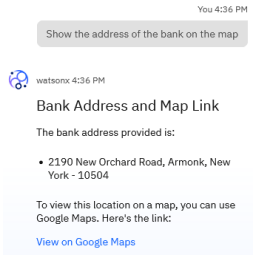
> Of course, you could also have just clicked on the document to open it in the DaeJa Viewer.
> If you get issues with opening the bookmark.jsp URL in a new browser tab, just click the document to open it in DaeJa Viewer.



_4. Use the Chat window to ask a question about the document, for example “Which country and city would be the telephone number for?”. So, it looks as if the phone number and the address are matching together.



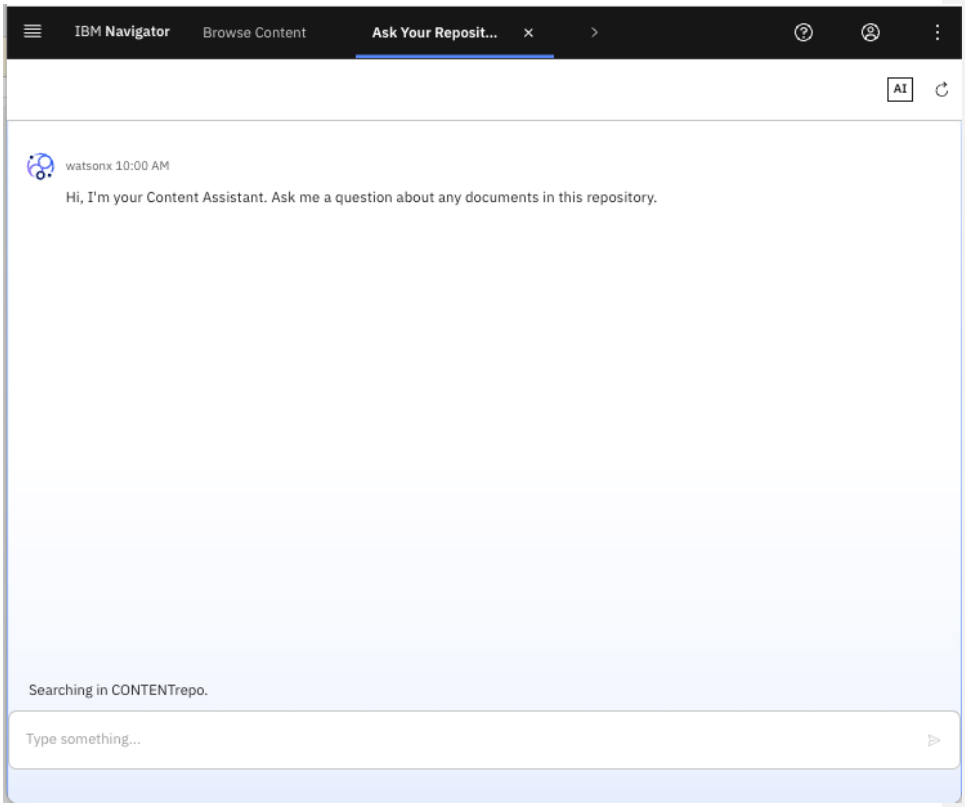
_5. You might be planning a visit to the bank, to discuss topics. Try: “Show the address of the bank on the map”



_6. Right-click on the link and verify that it links to the given address.

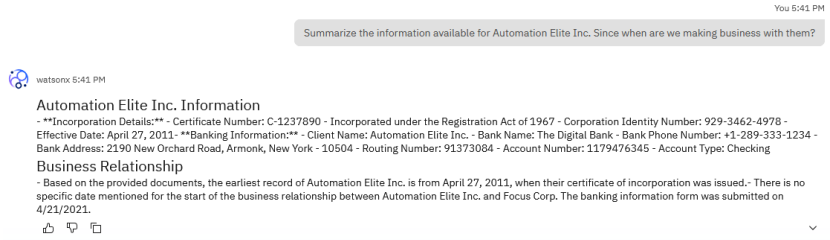
2.2.5 Running Queries against an Object Store

_1. Use the Hamburger Menu in the top left corner to select the feature “Ask Your Repository”. A magnified ICA Chat window will appear, implicitly all documents available to the logged on user are provided with the search.



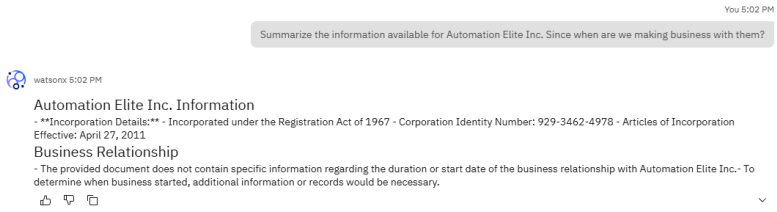
_2. Above the search prompt entry, the Repository being selected for the search is provided. If more than one repository is available on the desktop and enabled for IBM Content Assistant, a “Change…” link would be displayed By clicking on that “Change…” link any of the IBM Content Assistant enabled Repositories of the desktop can be selected. However, the ICN desktop has access to one Repository only.

_3. Keep in mind that information from the whole repository can be used to provide the answer to the query. Run for example this query: “Summarize the information available for Automation Elite Inc. Since when are we making business with them?”



_4. In another browser, or in a private window of the same browser, login to the IBM Content Navigator ICA Desktop again. In this session use the icademo account, which was provided with the access information for the environment. The icademo account has no access to the banking information statements.

_5. On that window, also open the “Ask Your Repository” feature and ask the same question as above. The answer to the icademo user is much shorter, as no banking information is available for that user.



_6. On the two windows, press the button on lower right corner of the answer, which will list the documents used for deriving the information. Compare that for the icademo, only one document (the certificate of incorporation) was used, where your main usrxxx account has access to both documents.

Commented [JR10]: I did not get the Change link. See the image I included below.

Commented [JR11]: Where is the information for this user? I searched this lab guide for this user ID and this is the first time that this ID is mentioned . As a participant I would not know where to get the information.

3 Exercise: Administration

3.1 Introduction

The communication between the IBM Content Assistant Chat window in the IBM Content Navigator, or a custom application, and the IBM Content Assistant Server side is being performed by IBM FileNet Content Platform Engine. The administration and configuration components are being added to an Object Store, by adding an Object Store plugin to each Object Store, which should be configured to be integrated with IBM Content Assistant. For this to work, in the current version of IBM Content Assistant, communication from the FileNet environment to the IBM Content Assistant Server components running on IBM Saas environments must be possible and must be supporting TLS Version 1.2 at least.

Inside an Object Store, some administrative tasks need to be performed to enable the documents of a document class to have their text be extracted and persistently stored, to be indexed in the IBM Content Assistant Vector Index Database, and to have their Summary created and stored in a property of the Document. Not all of these need to be done with every document which should be used with IBM Content Assistant, it might e.g. not be needed to have a summary present for all Vector indexed documents.

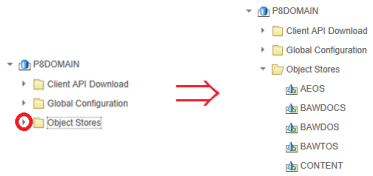
In this exercise, the most important FileNet Components of the IBM Content Assistant will be demonstrated.

3.2 Exercise Instructions

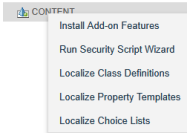
For this exercise, you need to use the Administration Console for Content Engine, ACCE. Login with the usrxx and password account (replacing xx by the number of your user).

3.2.1 Content Assistant Addon and Configuration

_1. After logging in to ACCE, the Domain Property window is shown in the central area, and a navigation bar appears on the left side. The navigation bar contains a list of the object stores, it might be needed to click at the small triangle in front of the “Object Stores” to open the list.



_2. Right-click on the CONTENT Object Store, and select “Install Add-on Features”

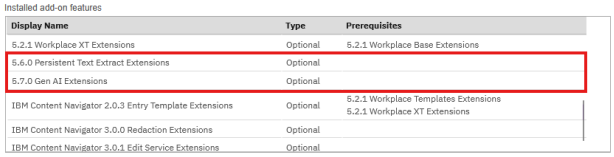


_3. A window opens which shows the installed and the not-yet installed Addons for the CONTENT Object Store. Review the list of installed Addons and find the two that are provided in the context of the IBM Content Assistant.

> FileNet Content Platform Engine 5.7.0, which is used with Cloud Pak for Business Automation 25.0.0 already contains this Addon pre-installed. For earlier versions of FileNet Content Platform Engine, the addons might need to be installed before they can be added to an Object Store. The documentation

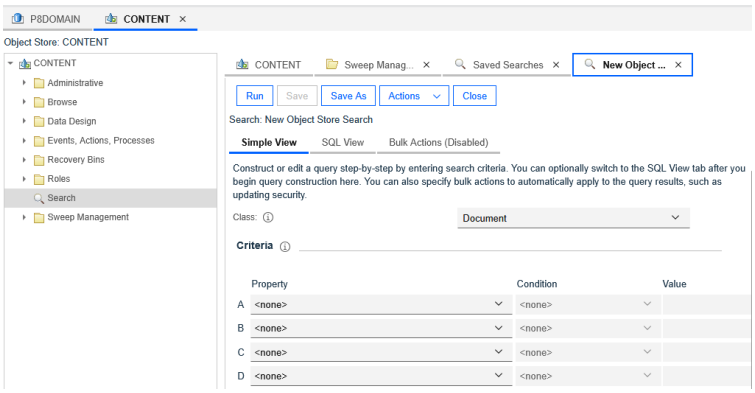
Commented [JR12]: "two ones" is not proper grammar. I would say "and find the two that are provided"

provides details how this is done, see https://www.ibm.com/docs/en/content-assistant?topic=engine-installing-content-assistant-add-object-store.

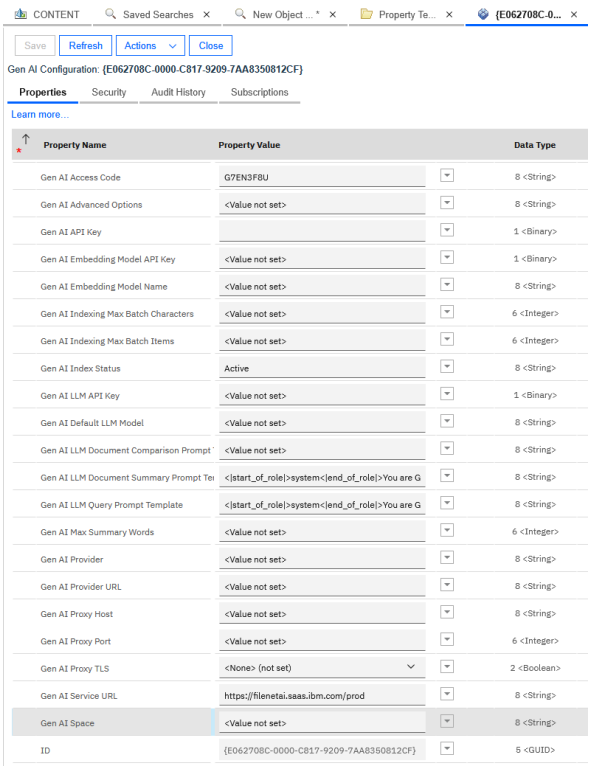


_4. Upon installation of IBM Content Assistant, after adding the Addons to each Object store which shall be used with IBM Content Assistant, IBM Content Assistant is not yet functional. For enabling IBM Content Assistant Object Store, it requires establishing the connection to the Server part of IBM Content Assistant and configuring of an access code.

Click on the CONTENT Object Store to open it. Use the navigation area on the left side to find the “Search” function. After clicking on Search, the “Saved Searches” window opens on the right side. Click on the “New Object Store Search” button in the header row of the “Saved Searches” window. The “New Object Store Search” window opens.



_5. For the Class, type in or select from the Drop-Down List the class “Gen AI Configuration”. No other changes are required. Run the search and dismiss the warning. There is only one result object for this class. Click on the blue ID value to open IBM Content Assistant Configuration object. The properties those name starts with “Gen AI” are store the configuration of IBM Content Assistant.



_6. The property “Gen AI Index Status” reflects the enablement of the Object Store for communication with the IBM Content Assistant Server in the SAAS environment. The value initially is “Inactive”. When the Object Store is successfully enabled, the value changes to “Active”.

_7. The property “Gen AI Access Code” is an access code, which grants one FileNet environment the access to the IBM Content Assistant Server in the SAAS environment. All Object Stores in one environment can use the same value. Object Stores in different environments will need a different Access Code.

_8. The property “Gen AI Service URL” might need to be adapted according to instructions from the administrators of the Server component of IBM Content Assistant. It is the URL of the IBM Content Assistant Server to be used for the environment.

Commented [JR13]: curious why do we mention the dev team here. It sounds like something that is internal for us but it is not something we would put in a lab guide like this one.

_9. Other important properties and their use are documented in the IBM Content Assistant documentation in the Knowledge Center. See https://www.ibm.com/docs/en/content-assistant?topic=assistant-configuring-content-service-parameters-object-store for details. They can be used for changing the large language and embedding models, which are used by IBM Content Assistant, suspend vector indexing, changing the relevancy score for documents, or configuring the most often used default queries.

_10. Don’t make any changes to the configuration and close the IBM Content Assistant Configuration Properties window.

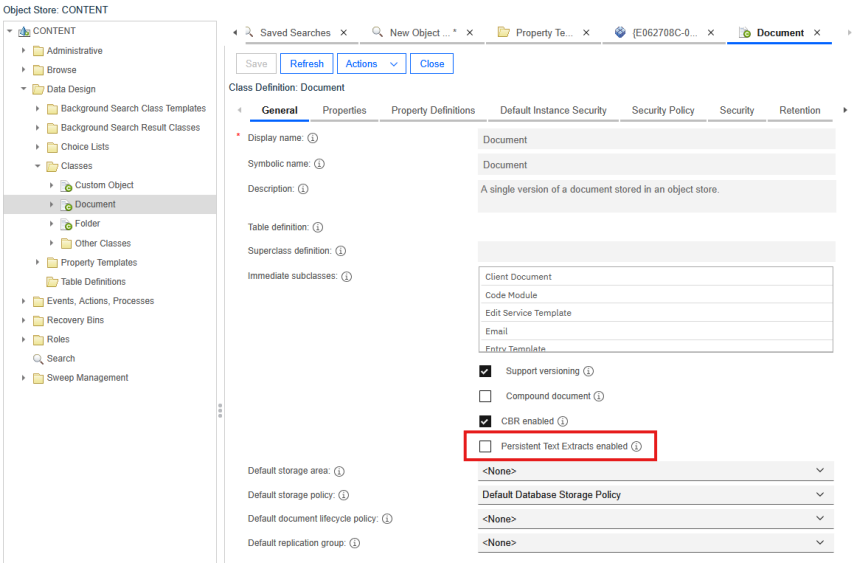
3.2.2 Configuring Persistent Storage of Text Extraction

Text Extraction from Documents was provided by FileNet Content Manager already before IBM Content Assistant was introduced. It was used in the process of CBR indexing documents. Newly introduced in

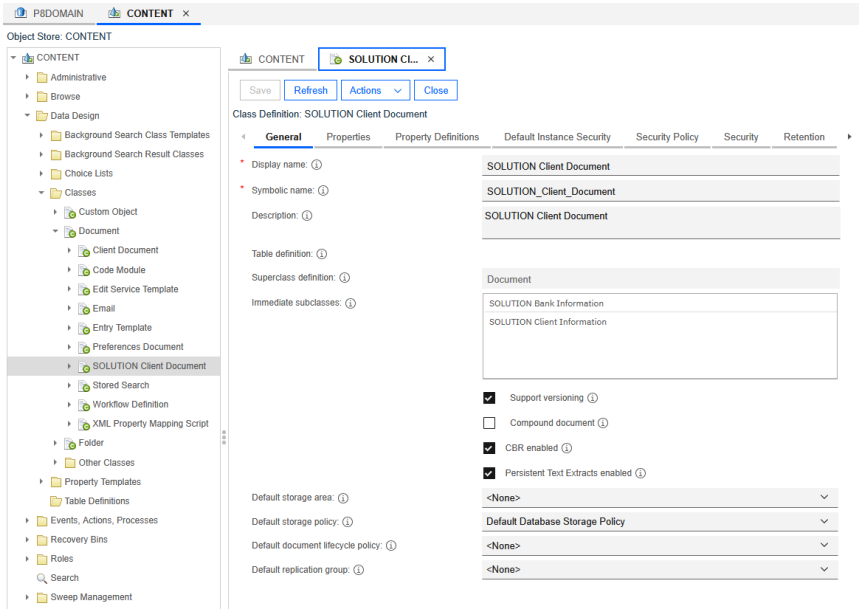
FileNet Content Manager is the feature to persistently store the extracted text, to save time, primarily when documents shall be CBR indexed AND vector indexed at the same time. Consequently, the persistent text extraction is enabled through its own Object Store Addon, as could be seen in previous subsection.

_1. Persistent Storage of Text Extraction is configured at document class level, on the General tab. To see it, navigate in the CONTENT Object store to Data Design -> Classes, and click on the “Document” document class to open its properties. On the General tab, the Object Store Addon for Persistent Storage of Text Extraction has enabled an additional property.

As you see, its not enabled for the “Document” Document class.

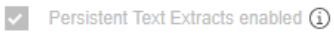


_2. Click on the small triangle in front of the “Document” document class in the navigation pane, to make visible the subclasses of the “Document” document class. Open the “SOLUTION Client Document” document class and display its “General” tab. As you see, its enabled for the “Solution Client Document” document class.

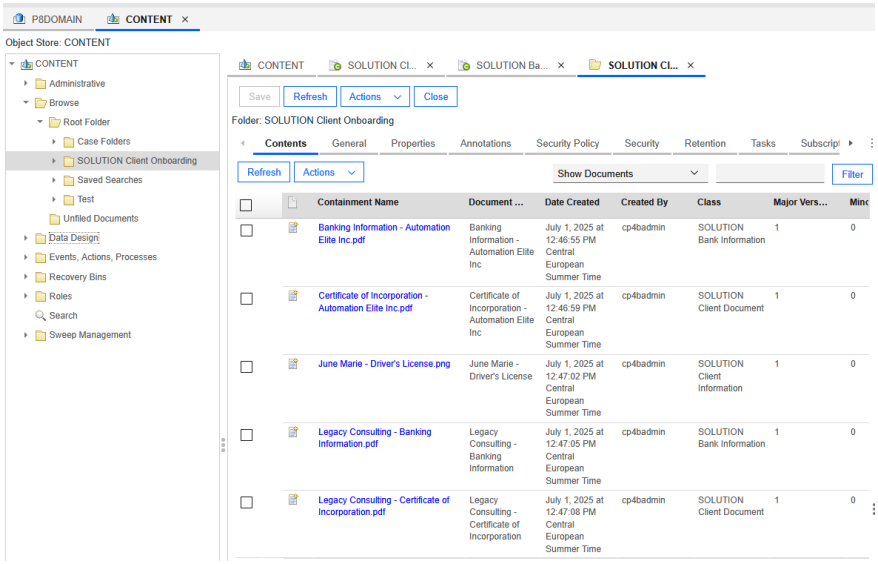


_3. Click the small triangle in front of that class to reveal the subclasses of the “SOLUTION Client Document” document class. Select one of its subclasses, e.g. the “SOLUTION Bank Information” class. There you see that the property is enabled but greyed out. This is indicating that its enabled on a parent document class, and cannot be disabled at this level in the class hierarchy

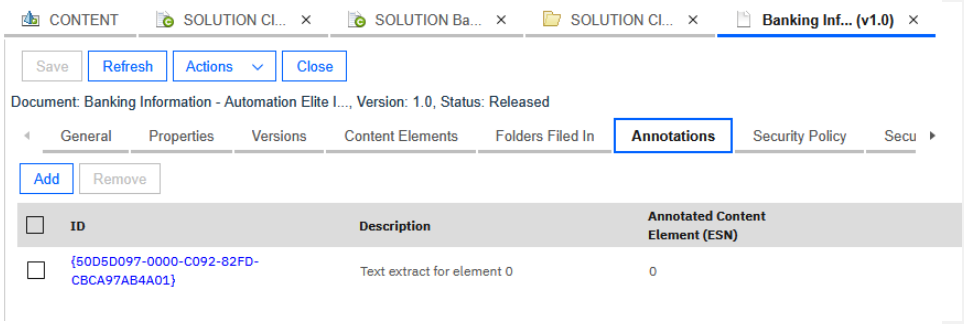
Commented [JR14]: There is a missing step here which is selecting a class under SOLUTION Client Document .



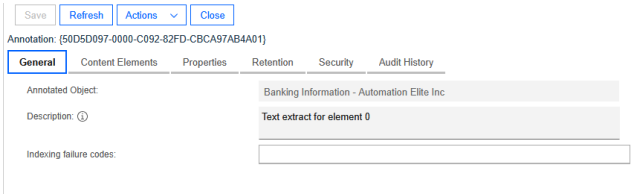
_4. For reviewing, where the extracted text is persistently stored, navigate to the “Browse” -> “Root Folder” -> “SOLUTION Client Onboarding”. Click on the “SOLUTION Client Onboarding” class to display its contents at the right side.



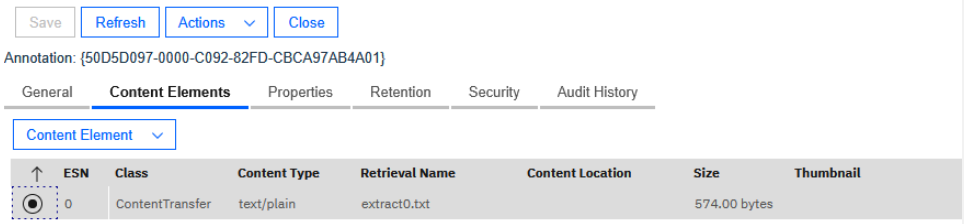
_5. Click on the first document, as it uses the class “SOLUTION Bank Information”, which is a subclass of “SOLUTION Client Document” and has persistent text extraction enabled. Click on the “Annotations” tab.



_6. As you see, FileNet Content Manager has created an annotation object with the extracted text for this document. For displaying its content, click on the blue link to open the Annotation object. As you see, the “General” tab would display any error codes, which occurred during text extraction. In this case it succeeded, and there is no value for that property.

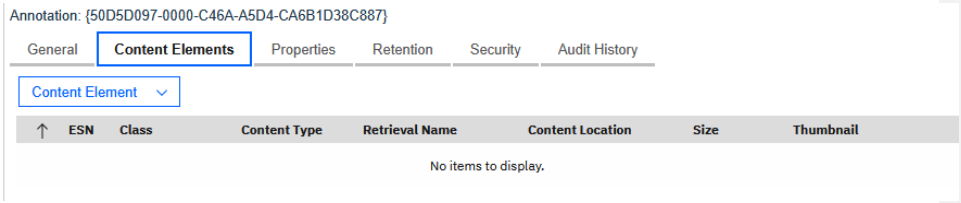


_7. Click on the “Content Elements” tab of the Annotation object. There is one content element. Select it.



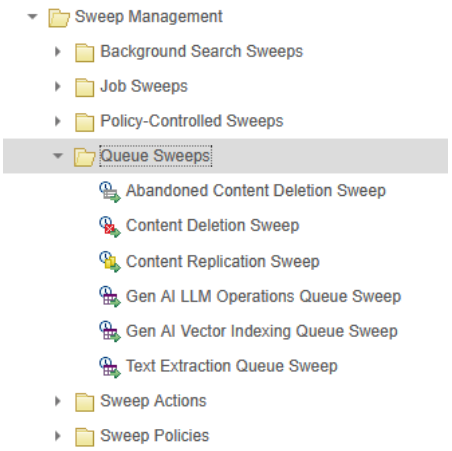
_8. Click on “View/Download in the “Content Element” menu to download the text file with the extracted text. It is downloaded by the browser, and you can open it to display its content. Close it again.

_9. Back in ACCE, switch back to the content of the folder “SOLUTION Client Onboarding”. Find the Annotation object for “June Marie - Driver's License” and open it. Check the “Content Elements” tab.

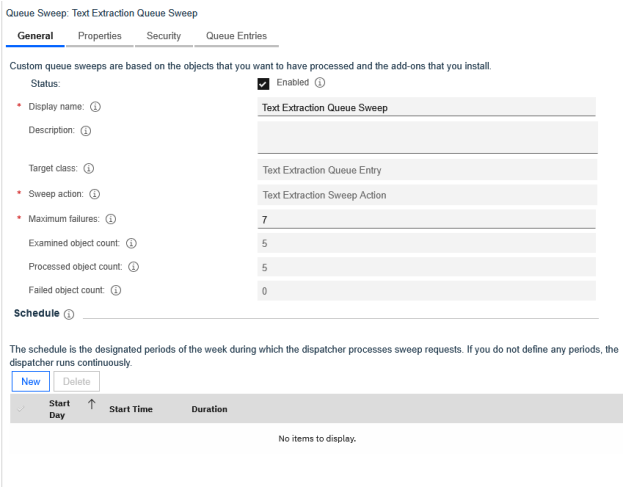


_10. FileNet Content Manager has created an Annotation object without content to indicate that the text extraction has not resulted in any text. By storing the Annotation, text extraction for such documents will not be tried repeatedly, for documents without usable text in it. Close the Annotation object again.

_11. In the Navigation area, navigate to “Sweep Management” and open “Queue Sweeps”. Find 6 Queue Sweeps there.



_12. The first three implement maintenance operations of the Advanced Storage Areas. The two “Gen AI” ones maintain IBM Content Assistant Vector Indexing requests, and IBM Content Assistant LLM Operations. They will be discussed in the next subsection. Click on the last one, “Text Extraction Queue Sweep”.



_13. The “General” tab of this dialog allows to configure a maximum failure count, shows statistic information, and allows to specify a schedule for processing Text Extractions requests. It could be configure, for example that text extraction is done only during the night.

_14. In the “Properties” tab, the most important property is the “Maximum Sweep Workers” property, which allows to increase the parallelism with which text extractions are processed. For the “Persistent Text Extraction” it is set to 2.

_15. In the “Queue Entries” tab, the current queue entries can be seen. There should be none. Close the Text Extraction Queue Sweep Window again. The tab can be used to review failed Text Extraction requests. Close the “Text Extraction Queue Sweep” window.

_16. Use the Navigation area of ACCE to navigate to “Sweep Management” -> “Sweep Actions” and click on “Sweep Actions” to display the defined sweep actions.

> The “Text Extraction Sweep Action”, is a sweep action that can be used to create a Custom Sweep Job, to extract the text of already uploaded documents of a document class, for which “Persistent Storage of Text Extraction” was enabled, but which don’t have their extracted text stored already, as the documents were already stored before enabling persistent storage.
> The “Text Extraction Sweep Action” is for a Custom Sweep Job. It will just create an entry for the documents it is running on, on the “Text Extraction Queue Sweep”.

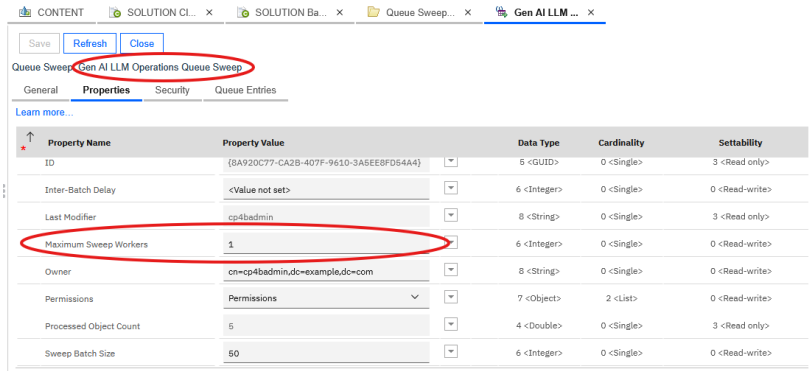
3.2.3 Vector Indexing and Summary Generation

It could be seen in the last subsection already that Vector Indexing and Summary Generation for the IBM Content Assistant functionality are performed using a queue sweep.

_1. Use the Navigation bar to navigate to “Sweep Management” -> “Queue Sweeps” Find the following queue sweeps defined:

- “Gen AI LLM Operations Queue Sweep”. This queue sweep handles all queries to the Large Language Model (LLM). This includes but is not limited to requests for summary generation.
- “Gen AI Vector Indexing Queue Sweep”. This queue sweep handles all indexing requests for documents which should be vector indexed, so that they can be added to requests to the LLM.

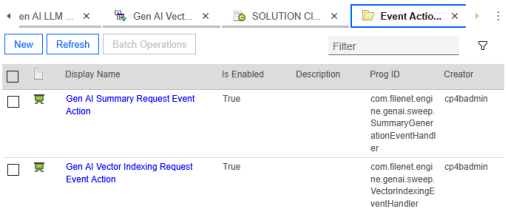
_2. Click on both (one after the other) to bring up a window with the Queue Sweep settings. Navigate to the “Properties” tab and review the setting for the “Maximum Sweep Workers”.



> The “Maximum Sweep Workers” is set to 1, meaning that at most one question of users is handled at any given time, to limit the volume of requests to the LLM. The settings can be increased, when configuring a separate WatsonX.AI entitlement, read more details on https://www.ibm.com/docs/en/content-assistant?topic=assistant-adjusting-number-queue-sweep-workers.

Commented [JR15]: This part is not very clear. I did not understand what you meant here.

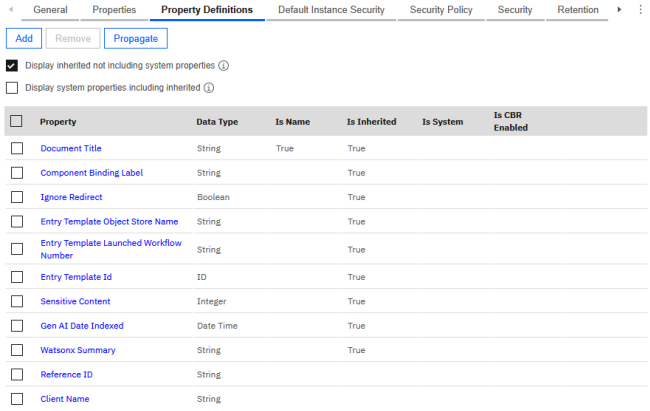
_3. Use the Navigation bar to navigate to “Event, Actions, Processes” -> “Event Actions” and click on “Event Actions”



A document class is configured for Vector indexing and optionally also for Summary Generation, by defining subscriptions to these event actions.

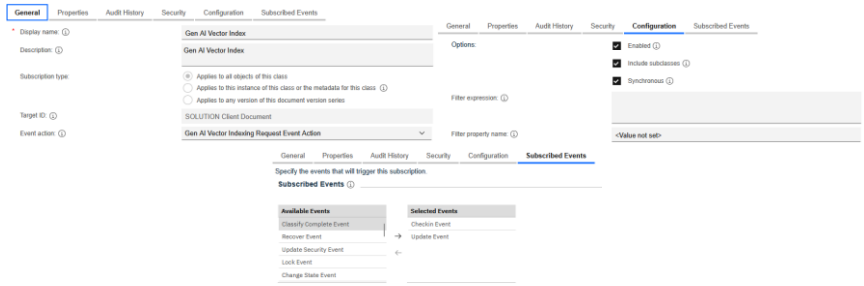
> These event actions are designed to be invoked synchronously. They just create entries in the Queue Sweeps, which were shown in the previous steps. The last subsections has furthermore shown that there are corresponding Sweep Actions, which also enqueue requests to the two Queue sweeps. Through the Subscriptions and Sweep Actions, new and existing documents can be vector indexed and have their summary generated.

_4. Use the Navigation bar to navigate to “Data Design” -> “Classes” -> “Document” and click on “SOLUTION Client Document” document class. The property window opens. Click on the “Property Definitions” tab and enable the box “Display inherited not including system properties”. This will show the properties, which are inherited from the “Document” document class.



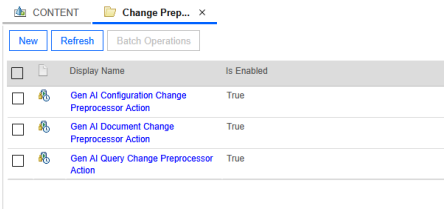
> Find the two properties “Gen AI Date Indexed” and “Watsonx Summary” there. They are added to document classes which are Vector Indexed and Summarized by Watsonx, upon first execution of the corresponding subscriptions. The properties are not designed to have their value provided by the user and are write-protected through a Change Preprocessor.

_5. With the Document class “SOLUTION Client Document” still open, select the “Subscriptions” tab. Click on the “Gen AI Vector Index” subscription to find out that it applies to all instances of the class, executes the “Gen AI Vector Indexing Request Event Action”, subscribes the “Checkin” and the “Update” events, and is executed Synchronously.



> As explained above, this does not mean that Vector indexing is done during the checkin or update of documents. During the checkin or update of documents, this subscription will merely create an entry in the “Gen AI Vector Indexing Queue Sweep”, which is then scheduled for execution in a separate thread.

_6. Use the Navigation bar to navigate to “Events, Actions, Processes” -> “Change Preprocessor Actions” click on “Change Preprocessor Actions”. Take notice of the three Gen AI Change Preprocessors, which were imported through the IBM Content Assistant Addon. Among other things, they protect IBM Content Assistant Metadata in the documents and the Object Store from unauthorized changes.


CONTENT
Change Prep...
New
Refresh
Batch Operations
Display Name
Is Enabled
Gen AI Configuration Change Preprocessor Action
True
Gen AI Document Change Preprocessor Action
True
Gen AI Query Change Preprocessor Action
True

4 Exercise Application Programming Interface

4.1 Introduction

- IBM Content Assistant does not (yet) expose an Application Programming Interface.
- IBM Content Assistant can be used in custom applications to enrich them with AI, optionally adding documents to provide context to queries.

Both statements are true. How can that be?

IBM Content Assistant does not provide an Application Programming Interface by itself. But it can be used through the query classes, which are defined by the IBM Content Assistant Addon in Object Stores, on which it was enabled. Within this exercise, those classes will be shown, and used to run queries against the Watsonx large language model.

Consequently, the first part of the exercise will be performed in ACCE, which is just one place where query objects can be created. Later parts of the exercise demonstrate that the same kind of query objects can be created through GraphQL as well.

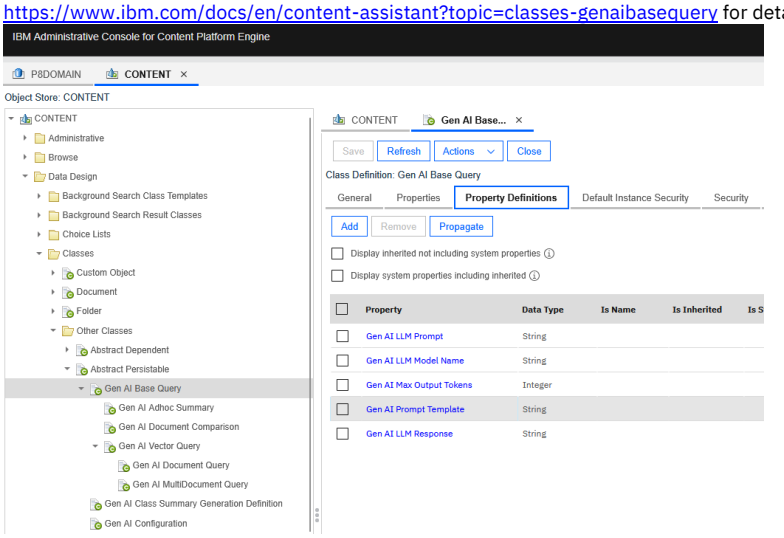
4.2 Exercise Instructions

4.2.1 Base Query Class

The classes, which are designed in an IBM Content Assistant Enabled object store for raising queries are described in the Knowledge Center on page “GenAI query classes” available on https://www.ibm.com/docs/en/content-assistant?topic=applications-genai-query-classes. The documentation lists in subsections the metadata for using them for each of the classes.

_1. Login to ACCE with the user account which was provided to you. Open the “CONTENT” Object Store.

Navigate to “Data Design” -> “Classes” -> “Other Classes” -> “Abstract Persistable”, and click on the “Gen AI Base Query” class. The properties of this class can be found on the Property Definitions tab, see https://www.ibm.com/docs/en/content-assistant?topic=classes-genaibasequery for details.

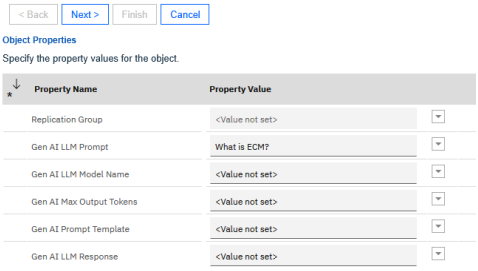


On the left side, the class hierarchy below the base query class can be unfolded. Notice the “Gen AI Document Comparison” class which is yet undocumented and intended for future improvement. The class names are more or less self-documenting. The “Gen AI Vector Query” class is designed for running queries against the object store, i.e. the complete Vector database, hence its name.

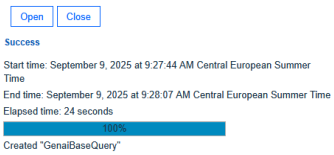
_2. From the “Actions” menu on this class, invoke the “Create instance” function. The Object Properties window opens showing the (still empty) metadata fields for the new query object.

_3. To run a query (in this case against the IBM Watsonx Granite foundation model), fill out the property “Gen AI LLM Prompt”, for example with the query “What is ECM?”.

With the default configuration of IBM Content Assistant, which is using the Watsonx account that is provided through the IBM Content Assistant, there is access only to one LLM Model, namely the IBM Watsonx Granite foundation model , see the section “Changing the Content Assistant large language model for an object store” on https://www.ibm.com/docs/en/content-assistant?topic=ccaspos-changing-content-assistant-large-language-model-object-store for details. If a separate account is configured, and that account has access to other LLMs as well, the LLM to be chosen can be configured in the “Gen AI LLM Model Name” property.



_4. Click on “Next”, then on “Finish”. Notice that the request takes long to complete. Finally, the “Success” page is shown, here after 24 seconds.



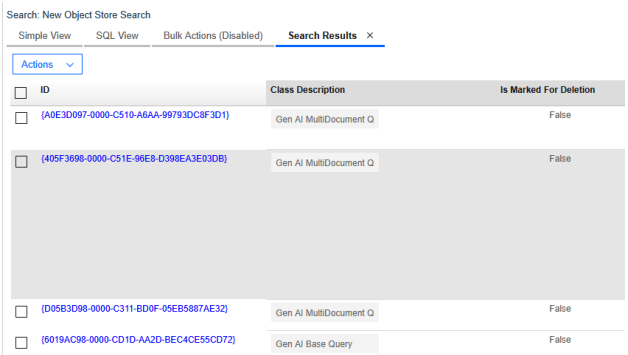
The reason (probably) is that ACCE might save the object with the “Refresh” flag enabled. This will try to read the updated values for the fields after creating the object, which includes trying to read the “Gen AI LLM Response” field. This will implicitly wait for the new entry “Gen AI LLM Operations Queue Sweep” to be processed.
In a custom application, if the call for creating a query shall return immediately, then it would be recommended to unset the “Refresh” flag, see https://www.ibm.com/docs/en/content-assistant?topic=applications-handling-genai-query-objects for more details.

_5. Click on “Open”, which opens the new object with the “Properties” tab. Find the updated field “Gen AI LLM Response” with the answer to the query.

_6. Click on “Search” in the Navigation bar. The “Saved Searches” window opens on the right side. Click on “New Object Store Search”.

_7. In the “New Object Store Search” window, select the “Gen AI Base Query” class, and click on “Run” to run the search. Dismiss the warning dialog.

_8. A list of prior query objects to which the logged-on user has access appears in the Search Results window, for example:



By default, FileNet retains a history of the searches, which were performed by the logged-on users. This history can be seen in the IBM Content Assistant Chat window in IBM Content Navigator too. There are a couple possibilities to fine-tune this behavior according to the needs of customers.

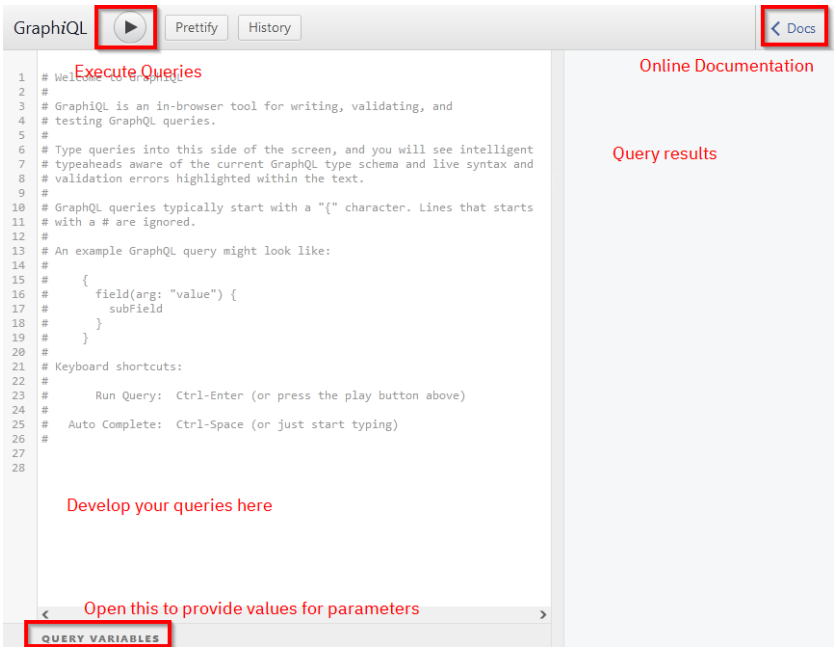
- For example, could the “Default Instance Permission” of the “Gen AI Base Query” class be modified, to only include permissions for the “#CREATOR-OWNER”. Then the history of queries would only be visible to the user who raised them. The first lab of the series contains examples of how this can be done.
- For example, could a Disposal Policy be defined for “Gen AI Base Query” objects, to automatically delete such objects after a defined time span. See “Disposal Policies” in the FileNet Content Manager Knowledge Center for details, available on https://www.ibm.com/docs/en/filenet-p8-platform/5.7.0?topic=development-disposal-policies.

4.2.2 Using Base Query Class from GraphQL

Instead of using ACCE, the queries from this and further subsections will be raised from the Content Services GraphQL GUI, to give examples which can easier be embedded in custom applications.

_1. Login to Content Services GraphQL with the user account which was provided to you.

_2. For a compact introduction, write the queries or mutations into the pane in the upper left corner. Everything what is there can be removed. The field in the lower left corner contains values for variables appearing in the query. By pressing the round button with the triangle in it (Execute Query Button), the query can be executed. Query results are displayed on the right side.



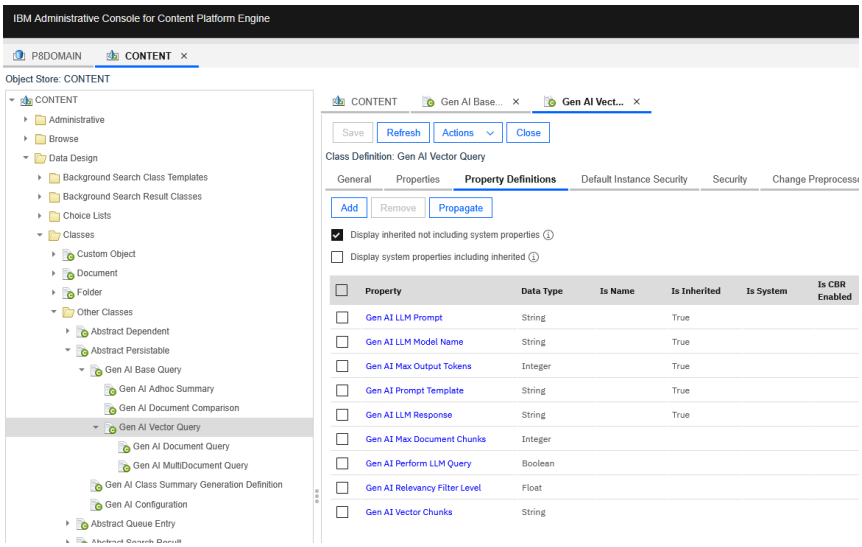
_3. For running a query, new objects for running LLM queries are created in FileNet Content Engine using one of the IBM Content Assistant Query classes. The first example will run the same query as the one from the last subsection. Copy & Paste below query into the field in the upper right corner, then click on the Execute Query Button.

```
mutation basequery {
  whatisecm: createCmAbstractPersistable(
    repositoryIdentifier: "CONTENT",
    classIdentifier: "GenAIBaseQuery",
    cmAbstractPersistableProperties: {
      properties:
       [ { GenaiLLMPrompt: "What is ECM?" }]
    }
  )
  {
    properties(includes: ["GenaiLLMPrompt", "GenaiLLMResponse" ])
      { alias value }
  }
}
```

> See Content Lab 2 “Interfacing FileNet Content Platform Engine with GraphQL on Cloud Pak for Business Automation” for an introduction into FileNet Content Services GraphQL.

In this mutation, the request “createCmAbstractPersistable” can be used to create any subclass of the base class “Abstract Persistable” in FileNet Content Engine. The mutation request is provided with the Object Store, the class name, and the properties of the new object. In this case this is only the GenaiLLMPrompt property. The lower part describes which components from the newly created object shall be contained in the result. In this case its only two properties, and from them only name and value. Last but not least, the label “whatisecm” is configured to be used for the name of the response in the returned JSON information. This way, several such mutations can be sent to GraphQL within one request, if needed, for improved efficiency over wide area network connections.

_4. Queries against the complete vector database can be made in much the same way. Look up the class “Gen AI Vector Query” in ACCE on the Property Definitions tab to get an overview over its properties. Enable “Display inherited not including system properties” to have the display include the inherited properties from the base class.



To use the class, the property “Gen AI LLM Prompt” needs to be provided with the prompt for the LLM to answer, as before. Additionally, the property “Gen AI Perform LLM Query” needs to be set to true. The query can be configured further by passing other configuration properties, see the description of this query class in the IBM Content Assistant Knowledge Center on https://www.ibm.com/docs/en/content-assistant?topic=classes-genaivectorquery for details.

However, on the execution of such queries, only documents available to the logged on user will be used.

_5. Use below sample to perform such a query. Compare the query with the previous query using the GenaiBaseQuery class. There are only few differences, which are typeset here in bold.

```
mutation vectorquery {
  automationelite: createCmAbstractPersistable(
    repositoryIdentifier: "CONTENT",
    classIdentifier: "GenaiVectorQuery",
    cmAbstractPersistableProperties: {
      properties:
       [ { GenaiLLMPrompt: "Summarize the information available for Automation Elite
Inc. Since when are we making business with them?" },
         { GenaiPerformLLMQuery: true }]
    }
  )
  {
    properties(includes: ["GenaiLLMPrompt", "GenaiLLMResponse" ])
      { alias value }
  }
}
```

4.2.3 Queries with One or More Documents

For most other kind of queries, one or more documents need to be passed as references, to be stored in properties of type OBJECT in the IBM Content Assistant query classes. As a blueprint for this kind of queries, the first example creates an LLM query with one document passed to it.

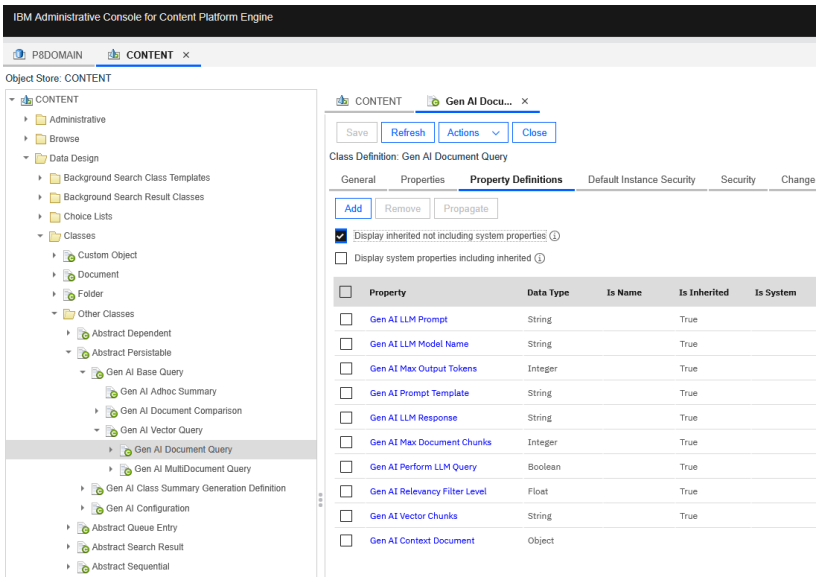
_1. For running this kind of query, information to the referenced document needs to be provided, i.e. the Object ID of the document, and the Object ID or the Name of the Document class. The following GraphQL query lists the documents, which are stored in the Folder “SOLUTION Client Onboarding”. Copy & Paste the query to GraphQL.

```
query solutionDocuments {
  solutiondocuments: folder(
    repositoryIdentifier: "CONTENT",
    identifier: "/SOLUTION Client Onboarding") {
    name
    containedDocuments {
      documents {
        id name className
      }
    }
  }
}
```

_2. Copy the result into an editor window. On a Windows System you might invoke Notepad and copy the result into the Notepad window on a new tab. Use similar tools on Computers with different Operation Systems.

Another solution would be to leave the browser tab with the output from GraphQL open in the browser for future reference and use a new tab for the following queries.

_3. Before using the query, review the class definition of the GenaiDocumentQuery again in ACCE. Go to the Property definitions tab and include display of inherited properties, to get an overview over which parameters can / must be provided. As the class inherits from the GenaiVectorQuery document class, find its properties among the inherited ones. The only new one which needs to be provided is the GenaiContextDocument property.



_4. Copy & Paste the below the parameterized GraphQL query into the query pane of the GraphQL browser window for running a single document query. For providing the value of the GenaiLLMPrompt property, a String parameter is used, for the GenAIContextDocument property, as an Object is expected (see above), the parameter type is ObjectReferenceInput in Graph QL.

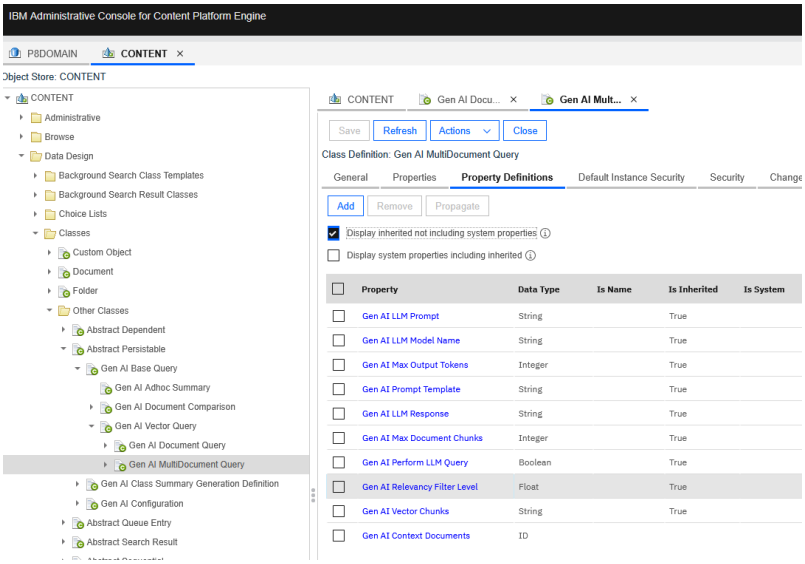
Similar as for the Vector Query, for the Document query the property “GenaiPerformLLMQuery” needs to be set to true. By passing “false” here, developers of custom applications can test their development of an IBM Content Assistant integration in custom applications, without actually executing the queries.

```
mutation documentquery($query: String!, $doc: ObjectReferenceInput!) {
  singleDocQuery: createCmAbstractPersistable(
    repositoryIdentifier: "CONTENT",
    classIdentifier: "GenaiDocumentQuery",
    cmAbstractPersistableProperties: {
      properties:
        [ { GenaiLLMPrompt: $query },
          { GenaiPerformLLMQuery: true },
          { GenaiContextDocument: $doc }
        ]
      }
    )
    {
      properties(includes: ["GenaiLLMPrompt", "GenaiLLMResponse"])
      { alias value }
    }
  }
```

_5. For running the query, in the lower left corner of the Graphql GUI, the parameter values need to be passed using JSON Syntax. In the example below replace DOCID first with the ID of the Banking Information Document for Automation Elite Inc. then with the ID of the Banking Information Document for Legacy Consulting.

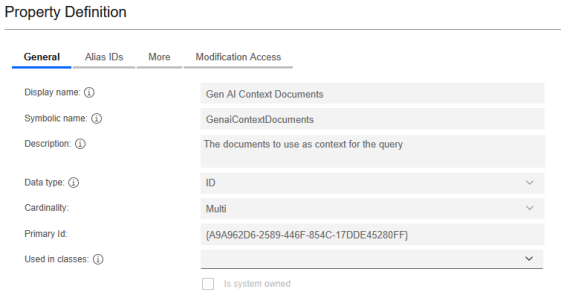
```
{
  "query": "Is the document a Banking Information statement for the company
Automation Elite Inc? Answer with True or False only.",
  "doc": {
    "repositoryIdentifier": "CONTENT",
    "classIdentifier": "SOLUTION_Bank_Information",
    "identifier": "DOCID"
  }
}
```

_6. As the last example in this section, a GenaiMultiDocumentQuery shall be used. Look up the class definition in ACCE first, as before. It looks very similar than the GenaiDocumentQuery class.



_7. Click on the “Gen AI Context Documents” property. A window with details about the “Property Definition” opens. Verify that the Cardinality of the property is given as “Multi” which means that an Array of Document IDs need to be passed for the property value. Close the Property Defintion window.

> Note that a description of the property is also available there. Note further that the symbolic name of that property is “GenaiContextDocuments”. Within applications, always the symbolic names are used. The display names are not only subject to change, but they can also appear translated, when switching the browser to a different language.



_8. Below GraphQL mutation will be creating a GenaiMultiDocumentQuery object. It is again parameterized, as the GenaiDocumentQuery. The type of the second parameter was adapted to provide a array of Document IDs. Copy & Paste the query into the GraphQL query window.

```
mutation multidocumentquery($query: String!, $docs: [ ID ]) {
  multiDocQuery: createCmAbstractPersistable(
    repositoryIdentifier: "CONTENT",
    classIdentifier: "GenaiMultiDocumentQuery",
    cmAbstractPersistableProperties: {
      properties:
           [ { GenaiLLMPrompt: $query },
         { GenaiPerformLLMQuery: true },
         { GenaiContextDocuments: $docs }
        ]
      }
    )
    {
      properties(includes: ["GenaiLLMPrompt", "GenaiLLMResponse"])
      { alias value }
    }
  }
```

_9. The parameter values are again provided using JSON. Replace the two Object IDs from the Banking Information Documents of the two companies.

```
{
  "query": "What companies do we have banking information for? Provide bank
names and the date of the banking information.",
  "docs": [
    "DOCID_AUTOMATION_ELITE",
    "DOCID_LEGACY_CONSULTING"]
}
```

_10. Last but not least, try to create a parameterized GraphQL query for invoking the GenaiAdhocSummary query class, by looking up the class definition in ACCE, and adapting the example for the GenaiMultiDocumentQuery. For that class only the documents need to be supplied, in the same way as for the GenaiMultiDocumentQuery, the values for the GenaiPerformLLMQuery and the GenaiLLMPrompt can be removed.

Find the solution at the end of the booklet, after the Troubleshooting section.

5 Troubleshooting

5.1 GENAI_QUERY_FAILED

When the connection between the FileNet environment and the IBM Content Assistant Server in the SAAS environment is unavailable or overloaded, the following error might be observed in the GraphQL GUI:

```
"errorCode": "FNRJG1005",
"message": "Error when calling the Content Platform Engine API.",
"explanation": "An error occurred when calling the Content Platform Engine API. An
exception occurred in the handler. Message was: GENAI_QUERY_FAILED: Vector query failed.
The response is: {\"message\": \"Endpoint request timed out\"}",
"userResponse": "Contact your system administrator for assistance.",
"statusCode": "500",
"serverErrorMessage": "An exception occurred in the handler. Message was:
GENAI_QUERY_FAILED: Vector query failed. The response is: {\"message\": \"Endpoint request
timed out\"}",
"classification": "DataFetchingException"
```

Retrying the request has proved to be a good strategy to handle this error situation.

5.2 Body has already been consumed.

```
Response.text: Body has already been consumed.
```

This error can happen when the authentication token of the CP4BA environment has expired. To resolve the situation, place the mouse cursor on the address bar of the browser, and hit Return, to reload the page. Pressing F5 key on a Windows machine will have the same result.

If the login page of Cloud Pak for Business Automation appears as a result, then the problem was the Authentication token. Use the provided username and password to login again. Your query and parameter values should still be there after logging in again.

5.3 Subscription contract limits

```
Subscription contract limit for the number of users has exceeded, additional packages can
be purchased.
```

If you get this message, the problem is a limitation configured on the IBM Content Assistant server. Many kinds of limitations are imposed there. Configured usage limits can be increased by the IBM Content Assistant Administrator.

If you get this error, get back to the administrator of the IBM Event, where this lab is a part of, for assistance.

6 Sample Graphql for the GenaiAdhocSummary query class

Use this GraphQL query for example. Compared to the GenaiMultiDocumentQuery, the parameter for the query was removed, as well as the two property definitions, which are not needed.

```
mutation multidocumentquery($docs: [ ID ]) {
  adhocsummary: createCmAbstractPersistable(
    repositoryIdentifier: "CONTENT",
    classIdentifier: "GenaiAdhocSummary",
    cmAbstractPersistableProperties: {
      properties:
            [{ GenaiContextDocuments: $docs }]
     }
    )
    {
          properties(includes: ["GenaiLLMPrompt", "GenaiLLMResponse"])
     { alias value }
    }
  }
```

For the parameters, only remove the prompt, for example:

```
{
  "docs": [
    "DOCID_AUTOMATION_ELITE",
    "DOCID_LEGACY_CONSULTING"]
}
```

Congratulations you have successfully completed the lab “Configuration and Usage of IBM Content Assistant on Cloud Pak for Business Automation”!