IBM Cloud Pak for Business Automation Demos and Labs 2021

Setting up FileNet Content Platform Engine for Automation Projects on Cloud Pak for Business Automation

V 1.3

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.

Table of Contents

1 Introduction

1.1 FileNet Content Platform Engine

IBM FileNet Content Platform Engine is the central part of the FileNet Content Services capability. It is added to a Cloud Pak for Business Automation environment for example by adding the "content" feature to the environment. It is used whenever documents need to be stored, found, access to them controlled or their lifecycle needs to be maintained. FileNet Content Platform Engine can also be installed independently of Cloud Pak for Business Automation, even in environments not running in containerized environments, if needed.

IBM FileNet Content Platform Engine is also a central component when needing to build the information architecture infrastructure of any AI project, because it can store vast amounts of unstructured content in the form of documents, and associated metadata that can serve as a body of knowledge for natural language understanding, providing information for chat bots etc.

1.2 Lab Overview

This lab demonstrates how to setup the FileNet Content Platform Engine component of Cloud Pak for Business Automation to support document storage for an application or solution using components of the Cloud Pak for Business Automation. The exercises serve as an example and are inspired from the Client Onboarding application as showcased in the other labs. The concepts shown are generic, most applications will require one or more of the concepts demonstrated in the exercises

Exercise "Creating the Document Classes" shows the definition of document classes, in this case document classes for the Client Onboarding application. A nested class hierarchy will be created along with some user-defined properties. The differences between the display name and the symbolic name are discussed. The security for the document class, and the default security of the documents created from it are also covered.

Exercise "Security" shows the main security features in FileNet Content Platform Engine, which are direct security and default instance security. Here, a folder will be created which is only visible to the participant. Furthermore the default security for the documents created for the classes defined earlier will be adjusted in such a manner, that the created documents are by default also only visible for the participant.

Exercise "Storage" discusses the important topic of the storage of documents. The introduction gives an overview of the different types of Storage Areas available in FileNet Content Platform Engine. On the exercise instruction part, the available storage areas are reviewed, and a configuration change on the document classes is made to have the created documents stored in a different storage area. The exercise further shows how to migrate documents between storage areas using FileNet Content PlatformEngine Sweep Jobs.

Exercise "Triggering Actions" shows how to trigger automation actions based on events occurring in the FileNet Content Platform Engine. The business requirement is that when the client is providing new documents for the Client Onboarding application, a case in the Workflow capability should be triggered for an end-user to check the new documents. This requires filing of the document in the case folder of a case. The specific case folder needs to be determined by querying for the client onboarding ID.

Exercise "Content Based Retrieval" discusses the required bits and pieces to allow searching documents based on information in their (unstructured) content. This requires using an additional component, the FileNet Content Search Services (CSS), which is setup and configured automatically by the Cloud Pak for Business Automation Operator, when selected.

After performing this lab, it is suggested to continue with the lab on GraphQL titled **"Interfacing FileNet Content Platform Engine with GraphQL on Cloud Pak for Business Automation"**

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 **ACCE**. It would be recommended to store the URL as bookmarks, so you can refer to it easier.

_2. Please download to your machine the Event Subscription (JavaScript) code from the **Lab Data** folder.

_3. Obtain a **user name** for the labs, and the **password**. Note the password, you will require it multiple times.

> In the lab instructions there are sections outlined with a grey bar on the left side, like this one. In sections of this type, additional explanations, for example about concepts and additional features are given. These sections are not needed if you only need to speed through performing the lab exercises. If you want to apply for the badge, though, it might be required to know the concepts introduced in these sections, as well as the descriptions in the instructions.

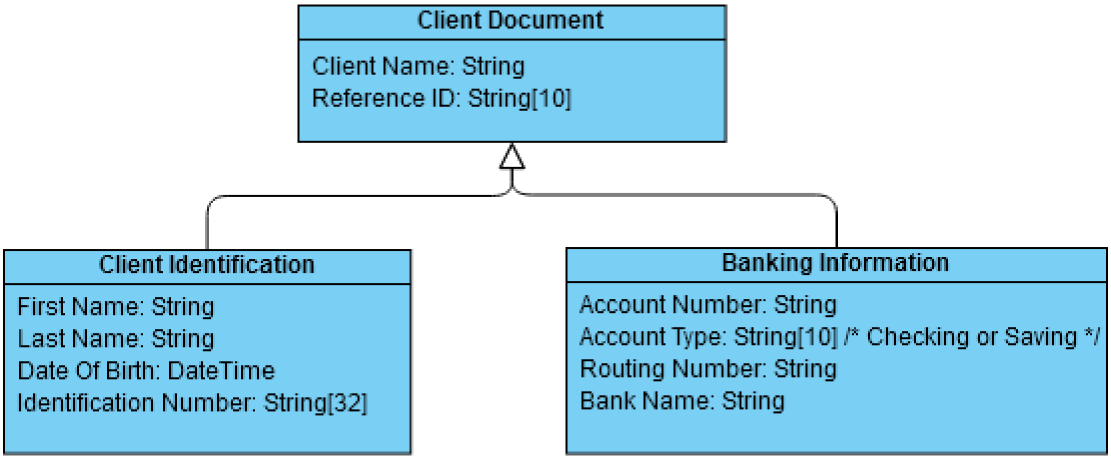
2 Exercise: Creating the Document Classes

2.1 Introduction

Documents and other entities in FileNet Content Platform Engine are referred to as Objects. Documents, or more generally all Objects have common properties, e.g. default security settings, default Storage Area and Storage Policy for defining where the content should be stored, common metadata, i.e. properties, and the like. The common properties of a Document or in general all Objects in FileNet Content Platform Engine are defined in the "Class Definition" of that object. For details on the Class Definitions, please refer the documentation https://www.ibm.com/support/knowledgecenter/SSNW2F_5.5.0/com.ibm.p8.ce.dev.ce.doc/class_concepts.htm.

The "Class Definition" contains references to "Property Definitions", which define details about the properties that are allowed in the instances of the classes. For example, this is where it is defined, that the customer name property has a specific name, is of type String, and may only have a specific maximum string length. The "Property Definitions" are in turn automatically generated from so-called "Property Templates", whenever a "Property" is added to a "Class Definition". So, when you need a "Property" named "Customer Name" in more than one "Class", by re-using the same "Property Template" for each, the metadata for Customer Name can be the same, e.g. it can have same type "String", same string length, same translations of its label to the different languages, and also same column name definition in the Object Store database. For more details on "Property Templates" please see the documentation https://www.ibm.com/support/knowledgecenter/SSNW2F_5.5.0/com.ibm.p8.ce.dev.ce.doc/property_concepts.htm#property_concepts__property_templates.

In the exercise, you will create the "Class Definitions" and the "Property Templates" for a Client Onboarding project. The Client Onboarding project requires a client to provide certain documents, namely an identification document, such as a scan from an ID card (class name Client Information), and a document for proving the banking information (class name Banking Information). In order to be address all documents easily, for example in searches or subscriptions, both classes have a common base class named Client Document. The classes are described in following diagram:



For the Account Type property of the Banking Information Document Class, only two values are allowed, the value "Checking" and the value "Saving". To assist users in choosing only valid values, a Choice List will be defined for this property type.

2.2 Exercise Instructions

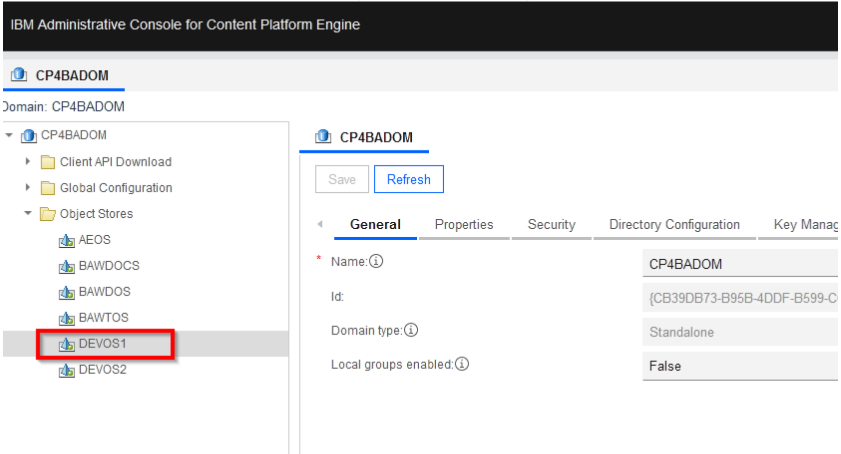
2.2.1 Defining the Choice List

_1. Open a browser and navigate to ACCE, the Administration Console for Content Engine. This is the main administration tool for a FileNet Content Platform Engine environment. Login using the username and the password which you obtained before.

If the language chosen is not English, your browser locale might be configured to a different language. To better follow the descriptions of this guide, its suggested to change the language to English. Click on the persona icon in top right corner, and select "Change language and locale settings". In the dialog select English language and English locale. Press OK.

> In a FileNet Content Platform Engine environment, the document metadata is stored in an Object Store. Each Object Store can be stored in a separate database. During the deployment of a Cloud Pak for Business Automation environment, the Object stores can be created and initialized automatically. For this lab we are using the DEVOS1 Object Store.

_2. Click on the "DEVOS1" Object Store to open it.



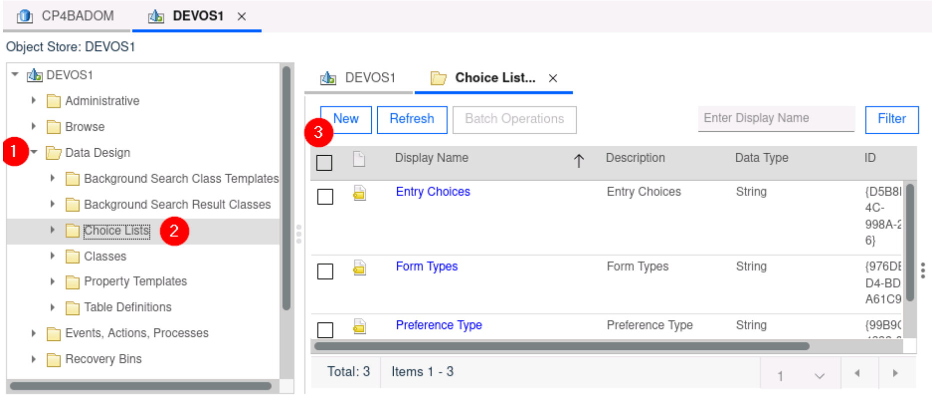
Name: CP4BADOM

Id: {CB39DB73-B95B-4DDF-B599-C

Domain type: Standalone

Local groups enabled: False

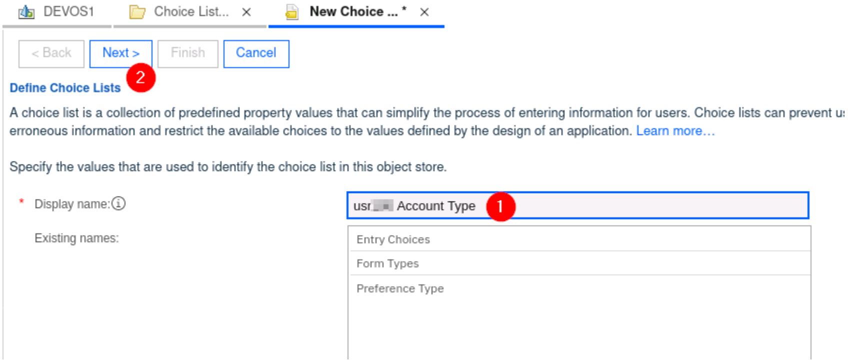
_3. Click on the small triangle in front of "Data Design". In the entries below "Data Design" click on "Choice Lists" to display it on the right side. In the choice list window click the "New" button.



Display Name	Description	Data Type	ID
Entry Choices	Entry Choices	String	{D5B8I 4C-998A-2 6}
Form Types	Form Types	String	{976DI D4-BD A61C9
Preference Type	Preference Type	String	{99B9(

Total: 3 Items 1 - 3

_4. For the name of the new choice list, prefix the name "Account Type" with the name of your user to build a unique name. Then press the "Next" button.



Define Choice Lists

A choice list is a collection of predefined property values that can simplify the process of entering information for users. Choice lists can prevent u erroneous information and restrict the available choices to the values defined by the design of an application. Learn more...

Specify the values that are used to identify the choice list in this object store.

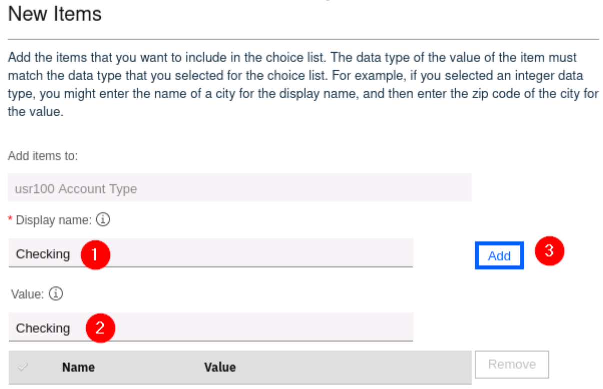
* Display name: usr[redacted] Account Type

Existing names: Entry Choices, Form Types, Preference Type

_5. On the next page choose "String" data type and press "Next" again.

_6. The choice lists allow building hierarchies of values. This will not be needed here. On the "Add Choice List Items" page, therefore just click on "New Items".

_7. In the pop up window, enter first "Checking" as the display name. As soon as you press <Tab> to change to the Value field, "Checking" is copied into it. Press the "Add" button.



New Items

Add the items that you want to include in the choice list. The data type of the value of the item must match the data type that you selected for the choice list. For example, if you selected an integer data type, you might enter the name of a city for the display name, and then enter the zip code of the city for the value.

Add items to: usr100 Account Type

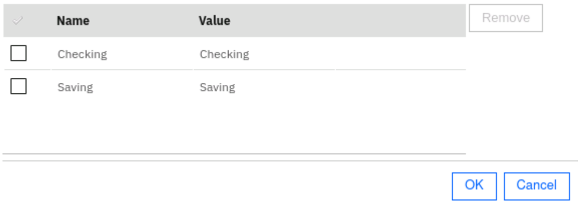
* Display name: Checking

Value: Checking

Name	Value

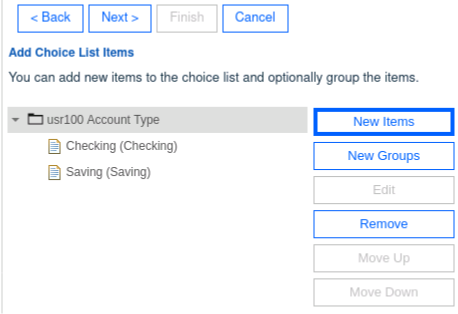
> Having both, "Display name" and "Value" allows to differentiate the content of the user interface from the stored value and is a prerequisite to allow for localization.

_8. In the same way, add the second display name "Saving", also using "Saving" as the value. When both are displayed in the lower part of the dialog correctly, press "OK". If you made an error, you can mark the box of the wrong entry and press "Delete".



	Name	Value
□	Checking	Checking
□	Saving	Saving

_9. Compare that the values are shown like in the below screenshot, then press the "Next" button, then "Finish", then "Close"



Add Choice List Items

You can add new items to the choice list and optionally group the items.

- usr100 Account Type
  - Checking (Checking)
  - Saving (Saving)

_10. This would bring up the "Choice Lists" window again. Press "Refresh" and verify that your new Choice list also appears.

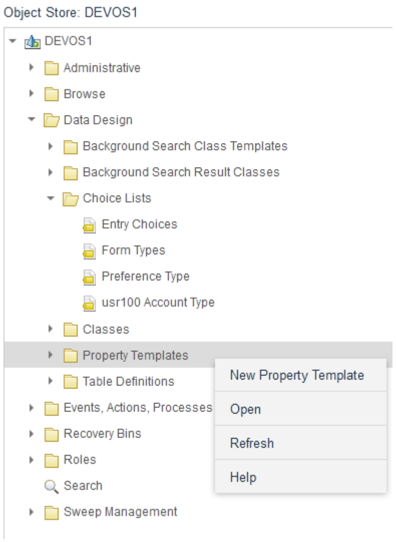
2.2.2 Defining the Property Templates

The following steps will need to be repeated a number of times to define all the property templates needed on the document classes later. Find the details in the below table

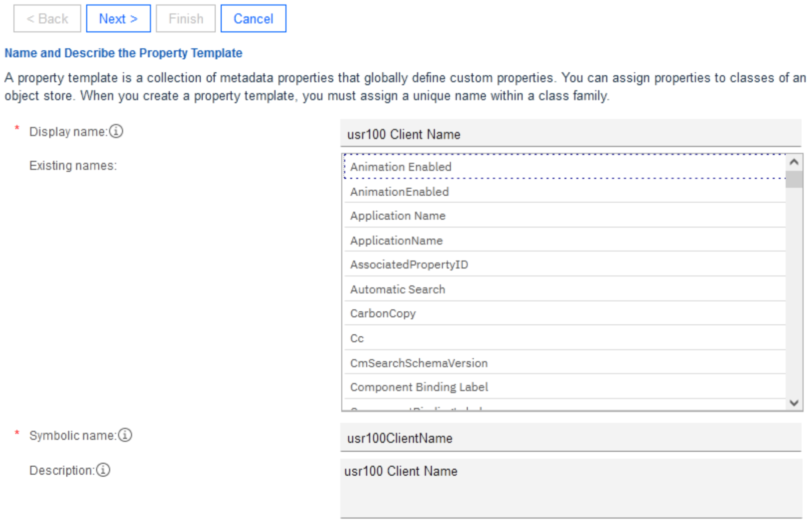
Display Name	Symbolic Name	Type	Choice List	Set other attributes	Value Required	Max String Length
usrxxx Client Name	usrxxxClientName	String		X		40
usrxxx Reference ID	usrxxxReferenceID	String		X	X	10
usrxxx First Name	usrxxxFirstName	String		X		40
usrxxx Last Name	usrxxxLastName	String		X		40
usrxxx Date Of Birth	usrxxxDateOfBirth	DateTime				
usrxxx Identification Number	usrxxxIdentificationNumber	String		X		32
usrxxx Account Number	usrxxxAccountNumber	String		X		40
usrxxx Account Type	usrxxxAccountType	String	X	X		10
usrxxx Routing Number	usrxxxRoutingNumber	String		X		40
usrxxx Bank Name	usrxxxBankName	String		X		40

_1. From the last section, you still have a browser open, navigated to ACCE, the Administration Console for Content Engine. You are logged on using the username and the password which you obtained before, and you have the Object store "DEVOS1" open.

_2. On the Navigation pane, open "Data Design", then right-click on "Property Templates". Click on "New Property Template".



_3. Provide the Display Name of the property, and the Symbolic Name of the property. As both need to be unique, prefix the names with your user id. For example provide Display Name "usrxxx Client Name", and Symbolic Id: "usrxxxClientName". The symbolic name must consist only of characters, numbers or the underscore, so remove any spaces, the Wizard will create such a name for you. Then click "Next".



Name and Describe the Property Template

A property template is a collection of metadata properties that globally define custom properties. You can assign properties to classes of an object store. When you create a property template, you must assign a unique name within a class family.

* Display name: usr100 Client Name

Existing names:

* Symbolic name: usr100ClientName

Description: usr100 Client Name

_4. On the next page, select the Data Type for the new property. The Client Name for example would be using the "String" type. Click "Next" to get to the next page.

> For numeric information, there are the Integer and Float types. However, Integer properties are numbers, they don't allow leading zeroes and have a maximum value. If these characteristics are not desirable, for example because the property would be storing an Id, then better choose the String type.

_5. On the next page, Choice Lists and Marking Sets can be assigned. If you are defining the Account Type property, click on "Assign choice list", and choose the previously defined Account type choice list. It should start with the username you are using. Click on "Next" to get to the next page.

> The marking sets are used to provide access rights to the object, e.g. a document, based on the values of its properties. For example, access to a "Reviewers" group could be provided based on the state property

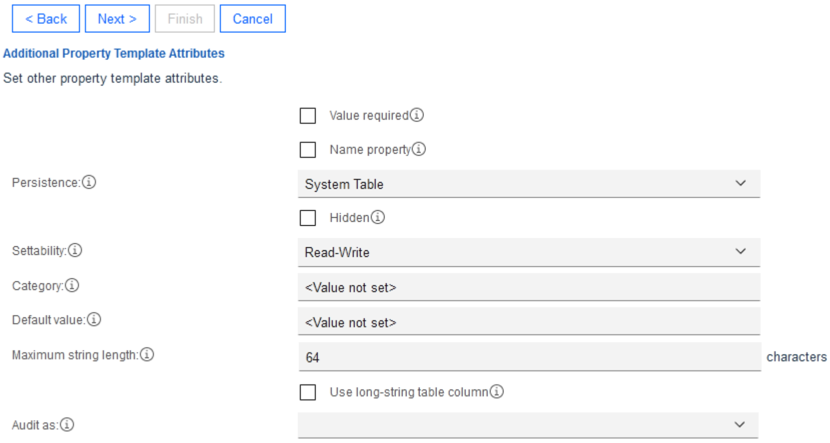
> of a document having the value "In Review". For details refer to https://www.ibm.com/docs/en/filenet-p8-platform/5.5.x?topic=markings-overview

_6. On the next page, it is possible to define an array of values, instead of only a single one. This is not needed for any of the properties defined here. The check box "set other attribute" needs to be enabled only, if it is needed to assign further properties of the new property template, such as changing the default length of string properties. When this is not needed, leave it unselected, then click "Next" and "Finish". Then proceed with defining the next property template.

> Multi-value properties are internally stored in a separate table, in the database. For example, all multi value String properties are stored in the table ListOfString. This can have adverse effects on the runtime of queries on these properties.

_7. If you are not defining the property templates for the "Date of Birth" enable the option "Set other attributes", and click "Next".

_8. On the next page, a couple additional properties can be defined. In our use case it is only needed to update the maximum string length, and the "Value required" for the Reference ID, then click "Next".



Additional Property Template Attributes

Set other property template attributes.

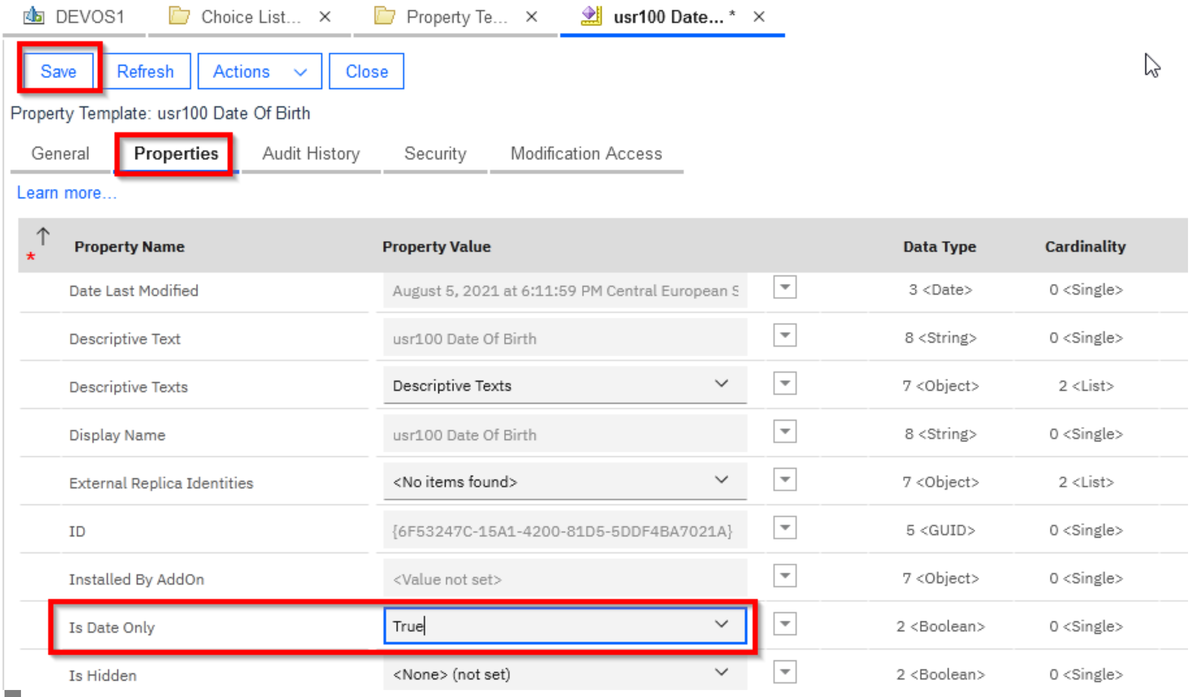
	□ Value required
	□ Name property
Persistence:	System Table
	□ Hidden
Settability:	Read-Write
Category:	<Value not set>
Default value:	<Value not set>
Maximum string length:	64 characters
	□ Use long-string table column
Audit as:	

> The default maximum size of a string property template is 80 characters. Reducing the maximum string length for a property, if possible, is a good practice to shorten the maximum database row size used for the storage of the metadata of the documents. The different database vendors support different maximum row sizes, for details see https://www.ibm.com/docs/en/filenet-p8-platform/5.5.x?topic=reference-database-engine-differences.
>
> On this page it is also possible to enable the usage of long string table columns. Long strings will be stored internally into a database LOB, allowing these string properties to grow to several megabytes length, based on limitations of the database. However, as long strings are not stored directly in the database row, it is not possible to include them in database indexes. Refer to https://www.ibm.com/docs/en/filenet-p8-platform/5.5.x?topic=ip-useslongcolumn-property-1 for details.

_9. No modifications needed on the next page, click "Next" and "Finish".

> On this page you can define additional conditions to grant modification access right on the property being defined. This covers the use case, when you have users who might modify some, but not all of the properties of a document.

_10. When you are defining the Date of Birth property, after clicking "Finish" click on "Open" to bring up the properties of the new Property Template. Select the "Properties" tab and scroll down to find the Property "Is Date Only". Set the value to "True", and click "Save".



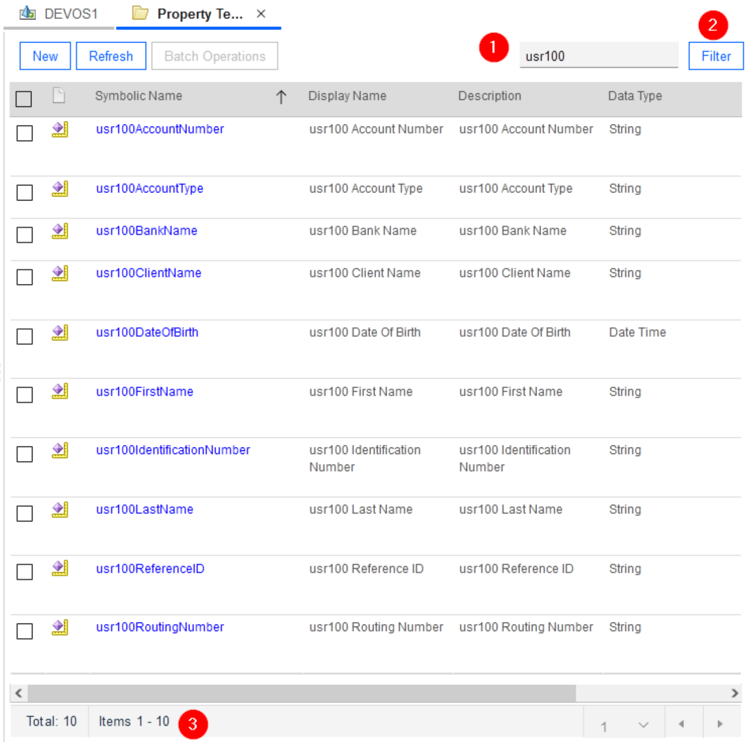
Property Template: usr100 Date Of Birth

Property Name	Property Value	Data Type	Cardinality
Date Last Modified	August 5, 2021 at 6:11:59 PM Central European S	3 <Date>	0 <Single>
Descriptive Text	usr100 Date Of Birth	8 <String>	0 <Single>
Descriptive Texts	Descriptive Texts	7 <Object>	2 <List>
Display Name	usr100 Date Of Birth	8 <String>	0 <Single>
External Replica Identities	<No items found>	7 <Object>	2 <List>
ID	{6F53247C-15A1-4200-81D5-5DDF4BA7021A}	5 <GUID>	0 <Single>
Installed By AddOn	<Value not set>	7 <Object>	0 <Single>
Is Date Only	True	2 <Boolean>	0 <Single>
Is Hidden	<None> (not set)	2 <Boolean>	0 <Single>

Selecting "Is Date Only" will enable more efficient handling of the property in Content Navigator. The time portion of the property value will be set to noon UTC, and the property will also be shown without a time value in Content Navigator.

2.2.3 Defining the Document Class Definitions

_1. Before defining the Document Classes, review completeness of the defined property templates. In ACCE click on "Property Templates" within "Data Design" to bring up a window with all the property templates on the right side. In upper right corner, type in your username to filter the property templates. Review the result count, it should be 10. If some are missing, go back to define the missing ones.



usr100

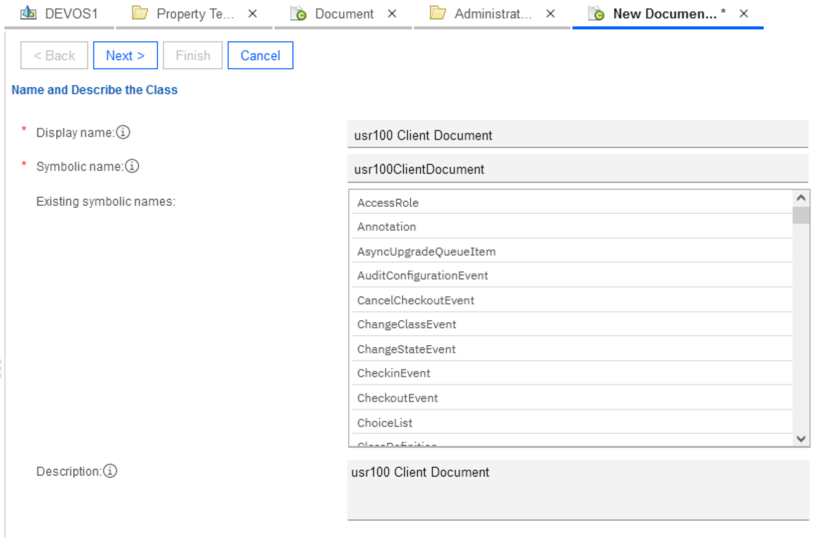
Symbolic Name	Display Name	Description	Data Type
usr100AccountNumber	usr100 Account Number	usr100 Account Number	String
usr100AccountType	usr100 Account Type	usr100 Account Type	String
usr100BankName	usr100 Bank Name	usr100 Bank Name	String
usr100ClientName	usr100 Client Name	usr100 Client Name	String
usr100DateOfBirth	usr100 Date Of Birth	usr100 Date Of Birth	Date Time
usr100FirstName	usr100 First Name	usr100 First Name	String
usr100IdentificationNumber	usr100 Identification Number	usr100 Identification Number	String
usr100LastName	usr100 Last Name	usr100 Last Name	String
usr100ReferenceID	usr100 Reference ID	usr100 Reference ID	String
usr100RoutingNumber	usr100 Routing Number	usr100 Routing Number	String

Total: 10 Items 1 - 10

_2. In the navigation pane on the left side in the "DEVOS1" Object Store, navigate to "Data Design", "Classes", and right-click on the "Document" class and select "New class".

> Doing this will create a new class as a subclass of the "Document" class. It will inherit all the properties defined for the Document class along with other definitions

_3. The first new class to define would be the "Client Document" class. Prefix the name again with your username, to get a unique name. The symbolic name will be generated accordingly. Click "Next" twice, then "Finish", and then "Open" to open the new document class to make further changes.



Name and Describe the Class

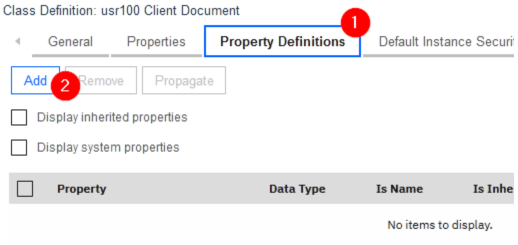
* Display name: usr100 Client Document

* Symbolic name: usr100ClientDocument

Existing symbolic names:
AccessRole
Annotation
AsyncUpgradeQueueItem
AuditConfigurationEvent
CancelCheckoutEvent
ChangeClassEvent
ChangeStateEvent
CheckinEvent
CheckoutEvent
ChoiceList

Description: usr100 Client Document

_4. With the document class open for editing, select the "Property definitions" tab, and click "Add" to add property definitions.

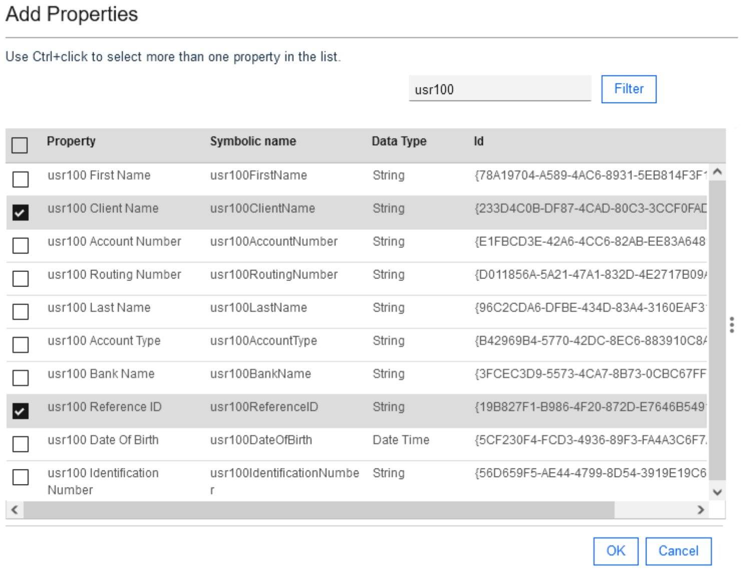


Class Definition: usr100 Client Document

□	Property	Data Type	Is Name	Is Inhe

No items to display.

_5. In the "Add properties" dialog, enter your username in the upper right corner, then click on "Filter". That will reduce the list to the properties you defined earlier. From the 10 property templates displayed, check the one for "Client Name", and the one for "Reference ID". Then click "OK"



Add Properties

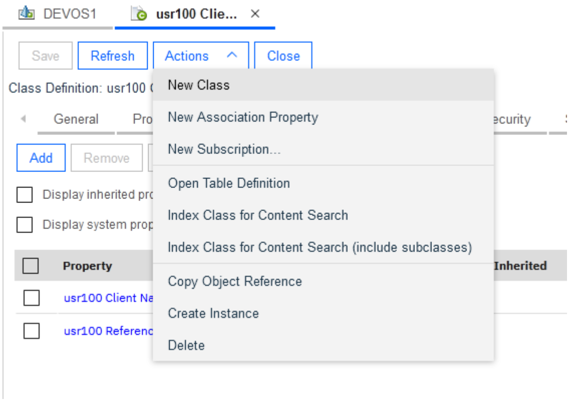
Use Ctrl+click to select more than one property in the list.

usr100

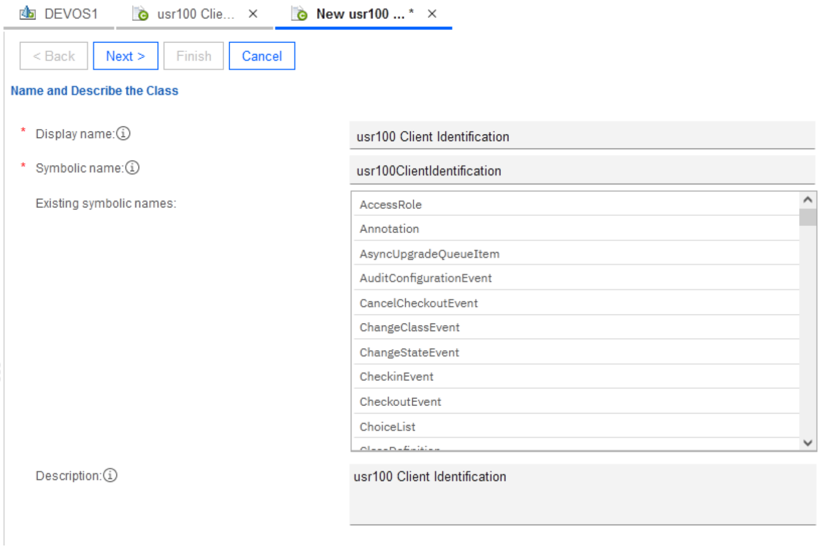
□	Property	Symbolic name	Data Type	Id
□	usr100 First Name	usr100FirstName	String	{78A19704-A589-4AC6-8931-5EB814F3F
☑	usr100 Client Name	usr100ClientName	String	{233D4C0B-DF87-4CAD-80C3-3CCF0FAD
□	usr100 Account Number	usr100AccountNumber	String	{E1FBCD3E-42A6-4CC6-82AB-EE83A648
□	usr100 Routing Number	usr100RoutingNumber	String	{D011856A-5A21-47A1-832D-4E2717B09
□	usr100 Last Name	usr100LastName	String	{96C2CDA6-DFBE-434D-83A4-3160EAF3
□	usr100 Account Type	usr100AccountType	String	{B42969B4-5770-42DC-8EC6-883910C8
□	usr100 Bank Name	usr100BankName	String	{3FCEC3D9-5573-4CA7-8B73-0CBC67FF
☑	usr100 Reference ID	usr100ReferenceID	String	{19B827F1-B986-4F20-872D-E7646B549
□	usr100 Date Of Birth	usr100DateOfBirth	Date Time	{5CF230F4-FCD3-4936-89F3-FA4A3C6F7
□	usr100 Identification Number	usr100IdentificationNumber	String	{56D659F5-AE44-4799-8D54-3919E19C6

_6. On the Client Document class, click on "Save".

_7. Click the "Actions" dropdown menu and invoke "New Class" to create a subclass of the "Client Document" class.



_8. Define the "Client Identification" class, as a subclass of the "Client Document" class. Prefix the name again with your username. Click on "Next", "Next", "Finish" and "Open".



Name and Describe the Class

* Display name: usr100 Client Identification

* Symbolic name: usr100ClientIdentification

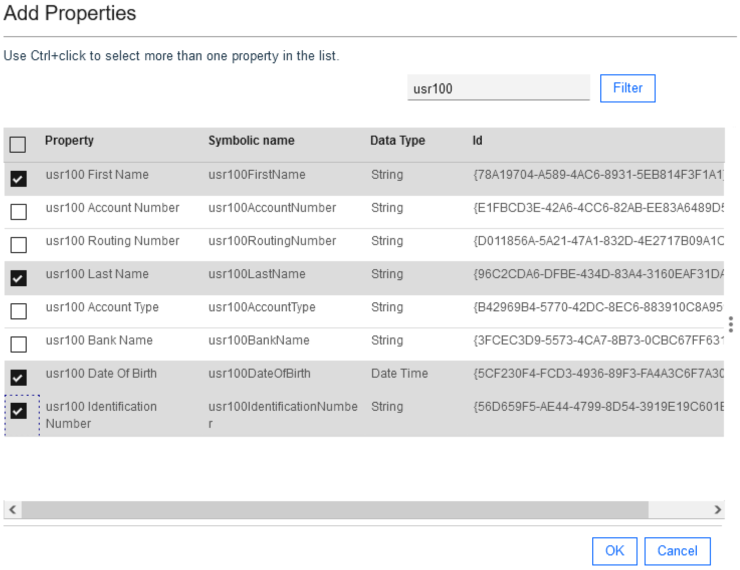
Existing symbolic names:
AccessRole
Annotation
AsyncUpgradeQueueItem
AuditConfigurationEvent
CancelCheckoutEvent
ChangeClassEvent
ChangeStateEvent
CheckinEvent
CheckoutEvent
ChoiceList

Description: usr100 Client Identification

_9. Again select to the "Property definitions" tab, and add properties. Enter your username in the upper right corner, and click "Filter". Only 8 properties are shown this time.

The other two got inherited already, and you can't add them a second time.

_10. The "Client Identification" class has the 4 displayed additional properties. Select them, then click "OK". Then click "Save" on the "Client Identification" class and close it.



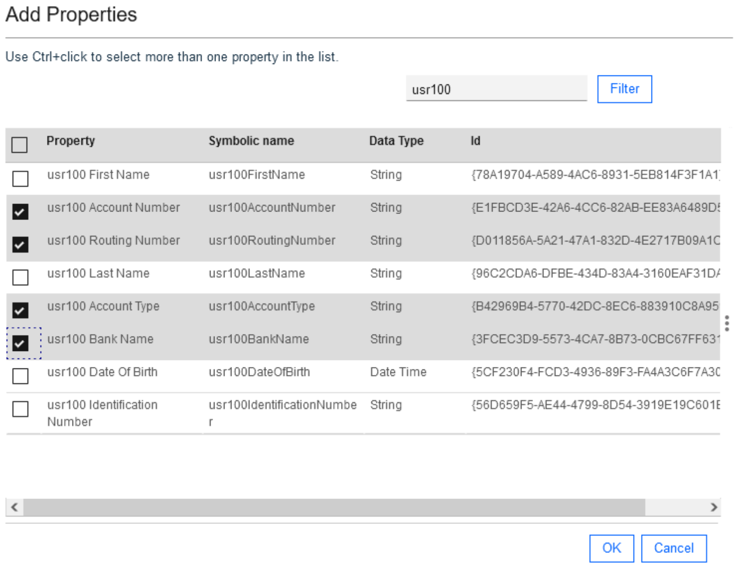
Add Properties

Use Ctrl+click to select more than one property in the list.

usr100 Filter

□	Property	Symbolic name	Data Type	Id
☑	usr100 First Name	usr100FirstName	String	{78A19704-A589-4AC6-8931-5EB814F3F1A1
□	usr100 Account Number	usr100AccountNumber	String	{E1FBCD3E-42A6-4CC6-82AB-EE83A6489D5
□	usr100 Routing Number	usr100RoutingNumber	String	{D011856A-5A21-47A1-832D-4E2717B09A1C
☑	usr100 Last Name	usr100LastName	String	{96C2CDA6-DFBE-434D-83A4-3160EAF31DA
□	usr100 Account Type	usr100AccountType	String	{B42969B4-5770-42DC-8EC6-883910C8A95
□	usr100 Bank Name	usr100BankName	String	{3FCEC3D9-5573-4CA7-8B73-0CBC67FF631
☑	usr100 Date Of Birth	usr100DateOfBirth	Date Time	{5CF230F4-FCD3-4936-89F3-FA4A3C6F7A30
☑	usr100 Identification Number	usr100IdentificationNumber	String	{56D659F5-AE44-4799-8D54-3919E19C601E

_11. Create the third and last class, named "Bank Information". Again prefix the display and the symbolic name with your username. Create the class **as a subclass of the class "Client document"**, and add the following property definitions to it:



Add Properties

Use Ctrl+click to select more than one property in the list.

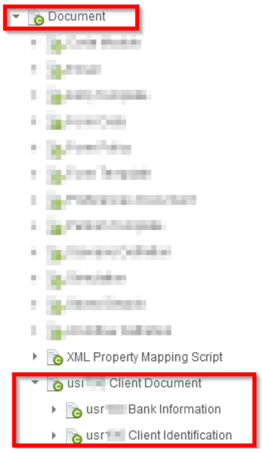
usr100 Filter

☐	Property	Symbolic name	Data Type	Id
☐	usr100 First Name	usr100FirstName	String	{78A19704-A589-4AC6-8931-5EB814F3F1A1
☑	usr100 Account Number	usr100AccountNumber	String	{E1FBCD3E-42A6-4CC6-82AB-EE83A6489D5
☑	usr100 Routing Number	usr100RoutingNumber	String	{D011856A-5A21-47A1-832D-4E2717B09A1C
☐	usr100 Last Name	usr100LastName	String	{96C2CDA6-DFBE-434D-83A4-3160EAF31DA
☑	usr100 Account Type	usr100AccountType	String	{B42969B4-5770-42DC-8EC6-883910C8A95
☑	usr100 Bank Name	usr100BankName	String	{3FCEC3D9-5573-4CA7-8B73-0CBC67FF631
☐	usr100 Date Of Birth	usr100DateOfBirth	Date Time	{5CF230F4-FCD3-4936-89F3-FA4A3C6F7A30
☐	usr100 Identification Number	usr100IdentificationNumbe r	String	{56D659F5-AE44-4799-8D54-3919E19C601E

> If after filtering with your username you see only 4 or 10 properties displayed, you have created a subclass of the wrong base class.

2.3 Verification Instructions

_1. Verify that the class hierarchy shows your three classes below the Document class, as outlined here:


Document
XML Property Mapping Script
usr[redacted] Client Document
usr[redacted] Bank Information
usr[redacted] Client Identification

_2. Verify that the property definitions tab shows the properties as shown in the class diagram at the beginning of the exercise.

3 Exercise: Security

3.1 Introduction

In an Enterprise Content Management (ECM) system, it is very important that only those parties can access the entities such as folders and documents, who are allowed to do so. For this reason, every folder, every document, every document class, and every other object might have different security settings assigned to them.

These security settings which are directly assigned to the object are called direct security, and are the security source with the highest priority. A good friend of the direct security is the so-called default instance security, which can be assigned to classes. These settings are COPIED to the objects created for that class, when a new instance of the class is being created. This means, that when you change the default instance security settings of a class, this will not affect the documents which have already been created for that class.

The default instance security has highest priority. This means, that if an individual, or a group in which the individual is member of, is allowed or denied a specific access to a specific object, this has absolute priority. Only when a specific access is neither granted nor revoked, then other mechanisms for obtaining security are considered. For examples, can an object inherit security settings from another object, e.g. a document from a folder where it is contained in. Furthermore, security settings might be set by a template which assigns preconfigured security settings when the state of a document changes, such as a document getting released.

Another important concept are the access levels. An access level comprises a set of basic access rights, for example does the access level "Add to folder" for a folder mean, that the individual can view all properties of the folder, can perform the FileInFolder operation, as well as the UnfileFromFolder, and can view the Folder permissions. Please note that the access levels might allow different access rights on different object types.

Then there is the concept of the owner of an object. The owner of an object can be specified together with the default instance security, and can be given special access rights, by defining access for the special name "#CREATOR-OWNER". The idea is, that a user who created an object should also be able to have special access to it, such as the right to delete it, which might not be granted to other users.

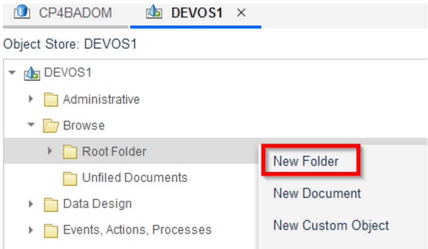
For a discussion of the security concepts of FileNet Content Manager, a recommended reading is the section about Authorization: https://www.ibm.com/docs/en/filenet-p8-platform/5.5.x?topic=security-authorization. The last reference on that page in the documentation discusses the fairly recently added concept of role-based security. With a role, different access to different object types can be abstracted as a role object, and assigned to the objects by reference. The idea is, that with later needed security updates, updates are not needed to millions of documents but just to the few role objects used for specifying the security settings of other objects like documents and folders.

3.2 Exercise Instructions

3.2.1 Creating a Folder

_1. From the last section, you still have a browser open, navigated to ACCE, the Administration Console for Content Engine. You are logged on using the username and the password which you obtained before, and you have the Object store "DEVOS1" open.

_2. In the navigation area on the left side, expand "Browse", and right-click on "Root Folder". In the context menu which appears, click on "New Folder".



_3. Give the new folder the name " Client Onboarding", prefixed with your username, as always. Then click on "Next"

> In this dialog you can also select a different folder class to use. Much like you created a subclass of the "Document" class, to define new document class types, in the last exercise, you can also create new folder types, having for example additional metadata. A good example is Case Manager, which introduces a new folder type to store cases.

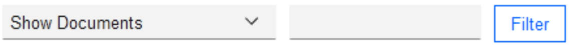
_4. Click "Next", then "Finish" then "Open".

> In this dialog, you can specify the retention for the folder, specifying the time span, in which it is not allowed to delete the folder. When this period expires, the folder will not be automatically deleted, though. Notice the different options provided, with the default retention being "None". Be sure to understand the difference between "Indefinite" and "Permanent", by hovering over the information icon and reading the explanations. The default retention (in this case for the folder) can be defined in the class, in this case this would be the Folder class.

_5. On the opened folder, choose the "Security" tab. Check that you have 4 Access Control Entries (ACEs) on the Security tab, one would be for your username listing "Full Control". Select the one for "cb4ausers", and click on "Remove" to delete it. Then click "Save".

> The Access Control Entry for your username was automatically generated from the default instance security settings for the "Folder" class, which you can check:
> - Find and open the Folder class in the navigation pane, its listed under Data Design → Classes.
> - Navigate to the "Default Instance Security" tab. Compare it to the original security of the new folder.
> - Scroll to the bottom of the Default Instance Security page, usually below the visible area to find that the owner of the new folder class will be set to the #CREATOR-OWNER as well.
>
> Close the Folder class again.

_6. Select the "Contents" tab. Make sure that in the upper right corner, "Show documents" is selected so that the contained documents of the folder are shown. Click on "Actions" above the row of tabs, and click on "New Document" in the drop down menu.

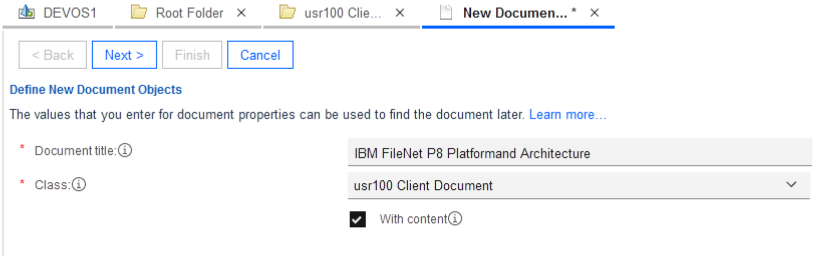


> Notice the other options to select. Custom Objects are similar to documents, but have no content.

3.2.2 Creating a Sample Document

_1. In a different browser window or a different tab, navigate to http://www.redbooks.ibm.com. Search and download the pdf version of the Redbook named "IBM FileNet P8 Platform and Architecture". Close that browser tab again and get back to ACCE.

_2. In the "New Document" window, put the title of the Redbook which you have just downloaded. For the document class, choose the "Client Document" class prefixed with your username. Then click "Next".



Define New Document Objects

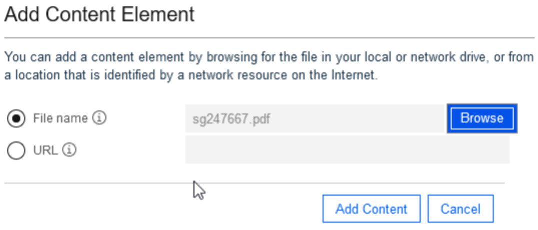
The values that you enter for document properties can be used to find the document later. Learn more...

* Document title: IBM FileNet P8 Platformand Architecture

* Class: usr100 Client Document

☑ With content

_3. On the next page, click "Add" to add a Content Element, and select the downloaded Redbook. Then click on "Add content", and "Next".


Add Content Element

You can add a content element by browsing for the file in your local or network drive, or from a location that is identified by a network resource on the Internet.

● File name: sg247667.pdf

○ URL

_4. On the next page notice the properties for the Reference ID and the Client Name. For now, fill in only the Client Name with the value "John Doe". This should give an error message later, as the Reference ID was specified as required. Click "Next". This should bring up a pop up asking to provide all required values. Click "Ok" and set the Reference ID to "1". Click "Next" again.

_5. No changes needed on "Document Content and Version" page. Click on "Next".

> Notice that you can select to create a minor version document here. For major version documents an approval and release process can be setup, which would then not be applied to minor version documents. You can later see the versions of a document in the "Versions" tab of the document properties.

_6. No changes needed on the "Specify Settings on Retaining Objects" page. Click on "Next".

_7. On the "Advanced Features" page again no changes are needed, click on "Next", then "Finish", then "Open".

> Notice the Storage policy set to "Default Database Storage Policy". The next exercise in detail looks at Storage topic.

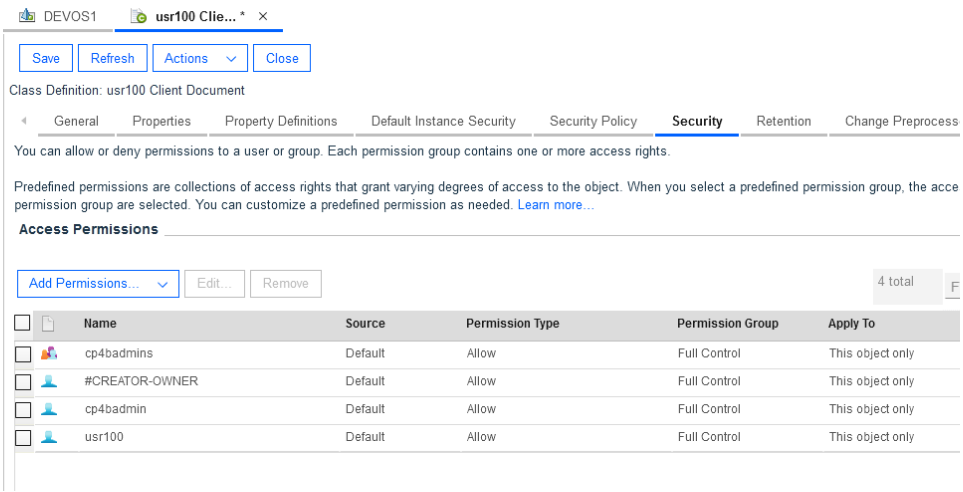
_8. Select the "Security" tab, and notice that again we have the group"cp4bausers". Leave it there for now, and close the document properties window.

3.2.3 Updating the Default Instance Security

_1. Navigate to the class "Client Document" which is prefixed with your username, and open it.

_2. Select the "Security" tab. Check that your username is contained with "Full Control" access level. Remove the group "cp4bausers". Click "Save"

> The entry for "#CREATOR-OWNER" in this table is responsible for the access rights on new subclasses for this class.



Class Definition: usr100 Client Document

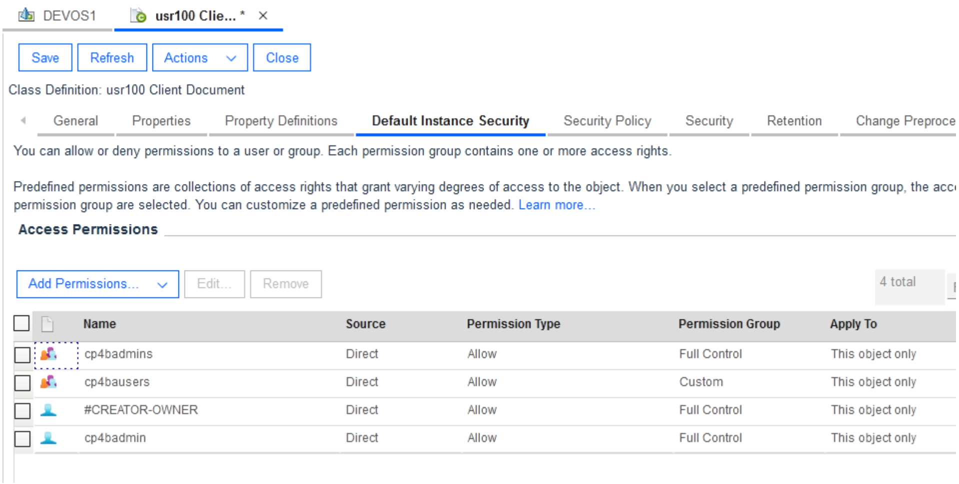
You can allow or deny permissions to a user or group. Each permission group contains one or more access rights.

Predefined permissions are collections of access rights that grant varying degrees of access to the object. When you select a predefined permission group, the acce permission group are selected. You can customize a predefined permission as needed. Learn more...

Access Permissions

□	Name	Source	Permission Type	Permission Group	Apply To
□	cp4badmins	Default	Allow	Full Control	This object only
□	#CREATOR-OWNER	Default	Allow	Full Control	This object only
□	cp4badmin	Default	Allow	Full Control	This object only
□	usr100	Default	Allow	Full Control	This object only

_3. Select the "Default Instance Security" tab. Remove the "cp4bausers". Click on "Save".



Class Definition: usr100 Client Document

You can allow or deny permissions to a user or group. Each permission group contains one or more access rights.

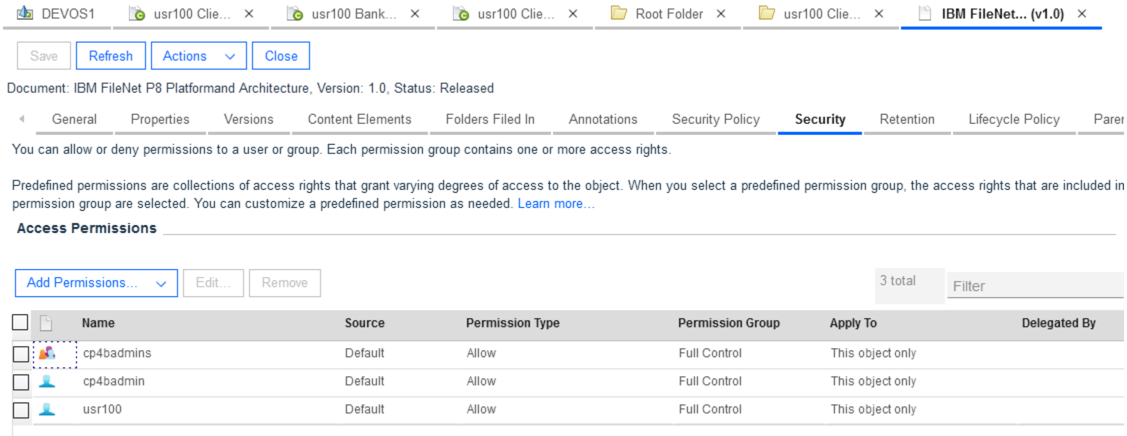
Predefined permissions are collections of access rights that grant varying degrees of access to the object. When you select a predefined permission group, the acc permission group are selected. You can customize a predefined permission as needed. Learn more...

Access Permissions

Name	Source	Permission Type	Permission Group	Apply To
cp4badmins	Direct	Allow	Full Control	This object only
cp4bausers	Direct	Allow	Custom	This object only
#CREATOR-OWNER	Direct	Allow	Full Control	This object only
cp4badmin	Direct	Allow	Full Control	This object only

_4. Repeat the modification on the two subclasses for "Client Identification" and "Bank Information" including your username. Remember saving the changes.

_5. Create another document in the "Client Onboarding" folder including your username, using the same document title "IBM FileNet P8 Platform and Architecture", using the same document class, and the same Redbook as the content. Set the client "Reference ID" property to "2", for the "Client Name" property no value is needed at this time. After the clicking on "Finish" click on "Open", and review the settings on the "Security" tab. Notice the "cp4bausers" group is missing this time.



Document: IBM FileNet P8 Platformand Architecture, Version: 1.0, Status: Released

You can allow or deny permissions to a user or group. Each permission group contains one or more access rights.

Predefined permissions are collections of access rights that grant varying degrees of access to the object. When you select a predefined permission group, the access rights that are included in permission group are selected. You can customize a predefined permission as needed. Learn more...

Access Permissions

Name	Source	Permission Type	Permission Group	Apply To	Delegated By
cp4badmins	Default	Allow	Full Control	This object only	
cp4badmin	Default	Allow	Full Control	This object only	
usr100	Default	Allow	Full Control	This object only	

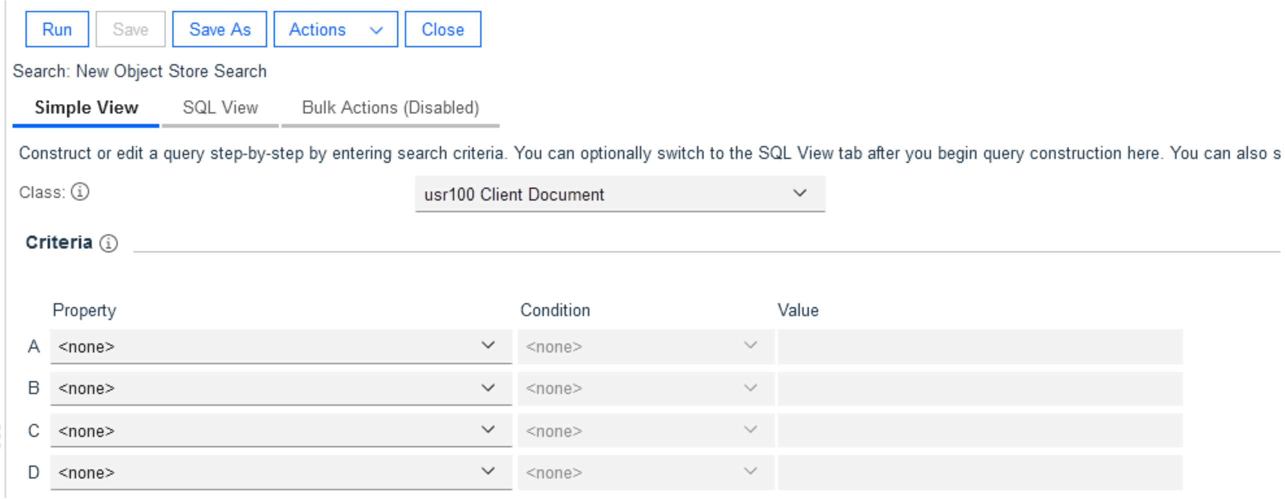
_6. In the window with the "Client Onboarding" folder including your username, click on "Refresh" and notice that the document was automatically renamed to have a unique name.

> The name which was renamed is not the document title, suggest to check it. To do so, open the document and go to the "Properties" tab. Search the Document title and find its original name there. The renaming occurred to the so-called "Containment name" which is the name given to the containment relationship between the folder and the document. This containment name needs to be unique.
> When lots of documents with the same name are stored in the same folders, finding unique containment names can produce a lot of overhead on the database side.

_7. Open the original document, the one which was not renamed, and review its security. The security settings still contain the "cp4bausers" group.

_8. In the navigation pane on the left, click on "Search". On the window which opens on the right side, click on "New Object Store Search".

_9. In the New Object Store Search dialog, set the class name in the first section of the dialog to the "Client Document" class prefixed with your username. Then click on "Run". Dismiss the warning dialog. The search shall return the two documents you created before, in a new "Search results" tab.



Run | Save | Save As | Actions | Close

Search: New Object Store Search

Simple View | SQL View | Bulk Actions (Disabled)

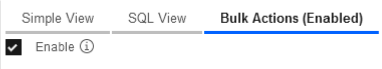
Construct or edit a query step-by-step by entering search criteria. You can optionally switch to the SQL View tab after you begin query construction here. You can also s

Class: usr100 Client Document

Criteria

	Property	Condition	Value
A	<none>	<none>	
B	<none>	<none>	
C	<none>	<none>	
D	<none>	<none>	

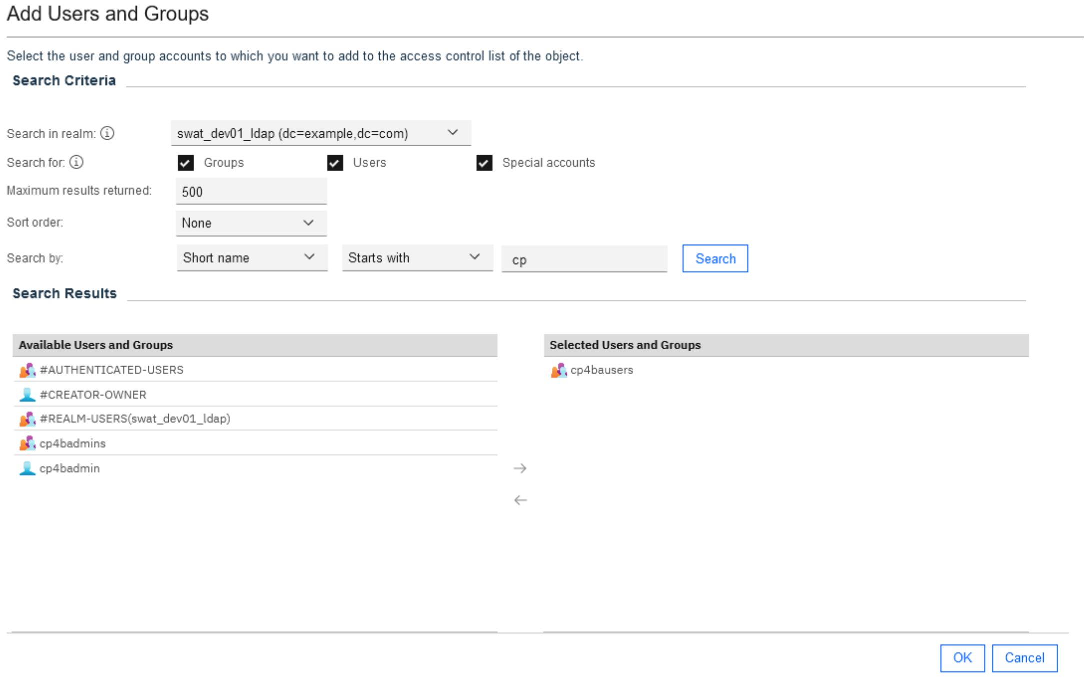
_10. Close the "Search Results" tab and select the "Bulk actions (disabled)" tab. In the topmost row, select to enable it. Notice the name of the tab changes to "Bulk actions (enabled)".



Simple View | SQL View | Bulk Actions (Enabled)

☑ Enable

_11. Scroll further down until you find the section named "Security", and enable it by selecting "Update security". Click on the "Add" button and select "Add User/Group permission...". Search for the group named "cp4bausers" by typing in a prefix (e.g. "cp") then click on "Search". In the list below, then find "cp4bausers", and move it to the right side. Then click "OK".



Add Users and Groups

Select the user and group accounts to which you want to add to the access control list of the object.

Search Criteria

Search in realm: swat_dev01_ldap (dc=example,dc=com)

Search for: ☑ Groups ☑ Users ☑ Special accounts

Maximum results returned: 500

Sort order: None

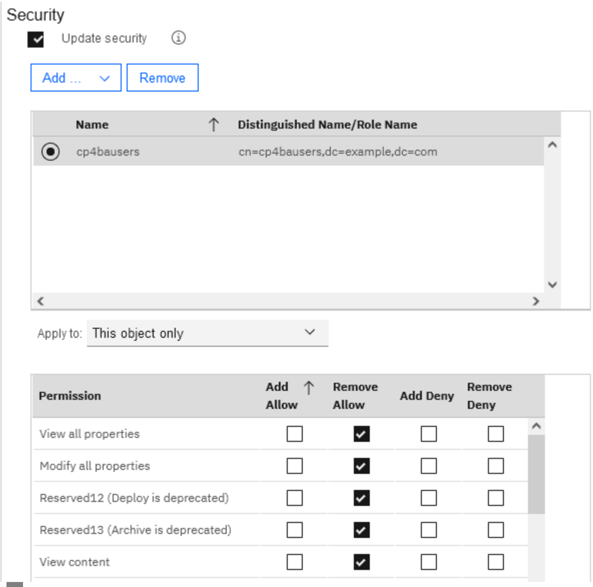
Search by: Short name | Starts with | cp | Search

Search Results

Available Users and Groups	Selected Users and Groups
#AUTHENTICATED-USERS	cp4bausers
#CREATOR-OWNER	
#REALM-USERS(swat_dev01_ldap)	
cp4badmins	
cp4badmin	

OK | Cancel

_12. Back on the "Bulk actions (Enabled)" tab, a dynamic section appeared below the username. On that table, tick all boxes on the column with the heading "Remove Allow".



Security

☑ Update security

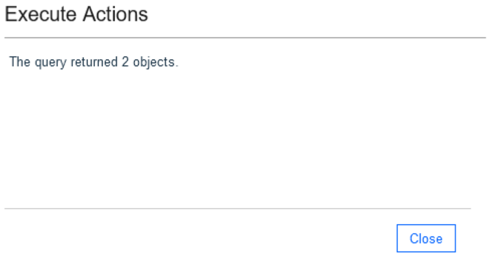
	Name	Distinguished Name/Role Name
●	cp4bausers	cn=cp4bausers,dc=example,dc=com

Apply to: This object only

Permission	Add Allow	Remove Allow	Add Deny	Remove Deny
View all properties	□	☑	□	□
Modify all properties	□	☑	□	□
Reserved12 (Deploy is deprecated)	□	☑	□	□
Reserved13 (Archive is deprecated)	□	☑	□	□
View content	□	☑	□	□

> This will remove the access rights of the cp4bausers group, which allows them to do any of the actions selected. This is actually very different from adding access rights of the "Deny" type.

_13. Click "Run" again to execute the search. Dismiss the warning again. This will automatically remove the cp4bausers group of every found document and list the results. Press "Close".

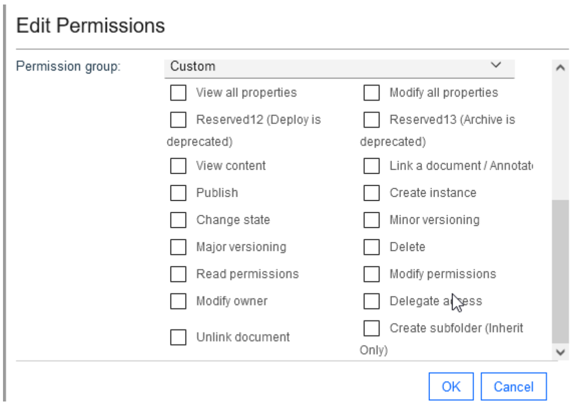


Execute Actions

The query returned 2 objects.

_14. Close the search, navigate to the folder "Client onboarding" which is prefixed with your name, and open the document again which we had created first (the one whose containment name was not renamed). If the document was open already, click "Refresh", then select the "Security" tab and verify that the entry for the cp4bausers group is still there.

_15. Click the checkbox in front of the "cp4bausers" group and click on "Edit". Verify that no permissions are granted.



Edit Permissions

Permission group: Custom

□ View all properties
□ Modify all properties
□ Reserved12 (Deploy is deprecated)
□ Reserved13 (Archive is deprecated)
□ View content
□ Link a document / Annotat
□ Publish
□ Create instance
□ Change state
□ Minor versioning
□ Major versioning
□ Delete
□ Read permissions
□ Modify permissions
□ Modify owner
□ Delegate access
□ Unlink document
□ Create subfolder (Inherit Only)

_16. Close all windows.

3.3 Verification Steps

At this stage, your classes, folders and documents would be invisible to any other user, and their folders, documents and classes invisible for you.

4 Exercise: Storage

4.1 Introduction

For storing the documents, FileNet Content Platform Engine offers a wide range of possibilities.

The **Database Storage Areas** store documents in database Binary Large Objects (BLOBs) in the Object store database. One of the advantages is, that the storage of the documents is in this way always in sync with the Object store database. The most important limitations are, that the database does not support arbitrary file sizes, and that storage in the database is usually more expensive than other storage technology. Find more details about the Database Storage Areas here: https://www.ibm.com/docs/en/filenet-p8-platform/5.5.x?topic=areas-database-storage.

The **File Storage Areas** store the documents in the filesystem. When talking about the container releases of a FileNet Content Platform Engine environment, that would mean in the persistent volume which is associated with the mounting point "/opt/ibm/asa" inside the container. On this mounting point, a subdirectory would be created and would be configured as the root directory of the File Storage Area. Further information about the File Storage Areas can be found here: https://www.ibm.com/docs/en/filenet-p8-platform/5.5.x?topic=areas-file-storage.

The **Fixed Storage Area** is the third of the traditional Storage Area technologies. It is very similar to the File Storage Area on the aspect that it uses a filesystem. The filesystem of the Fixed Storage Area is only used as a staging area, a temporary location where documents are stored until they could be uploaded to one of the Fixed Content Devices supported by FileNet Content Platform Engine. Optionally, the Fixed Storage Devices can be configured to support a retention period. That is the time which the documents need to be retained and cannot be deleted. Find information about Fixed Storage Areas here: https://www.ibm.com/docs/en/filenet-p8-platform/5.5.x?topic=types-fixed-storage-areas.

The **Advanced Storage Area** are a relatively new addition, they were introduced with the Version 5.2.1. With the Advanced Storage Areas, documents are stored on so-called Storage Devices, for example on File Storage Devices. Cloud Storage Devices and Hadoop are also supported. Important advantages of Advanced Storage Areas are replication (documents can be stored more than once, e.g. once on every site having a FileNet Content Platform Engine Server in a distributed environment), the support for cloud storage providers, and the direct upload of the documents (uploaded files are sent directly to the attached storage devices they are not locally stored on the FileNet Content Platform Engine Server).

If more than one storage device is included in the Advanced Storage area, and thus more than one replica of the files exist, then content can also be repaired, if needed. This is done by running a Consistency Check Sweep Job, which is available under Sweep Management / Job Sweeps. Find more information in the documentation: https://www.ibm.com/docs/en/filenet-p8-platform/5.5.x?topic=types-advanced-storage-areas.

All of the Storage Areas allow stored documents to be compressed and/or encrypted. The encryption keys for the documents can be stored on an external key management service, which might allow customers to store even confidential files on Cloud storage providers, as long as they control the key management service. Find more information on this topic in the documentation: https://www.ibm.com/docs/en/filenet-p8-platform/5.5.x?topic=tasks-v552-later-optional-preparing-external-key-management.

4.2 Exercise Instructions

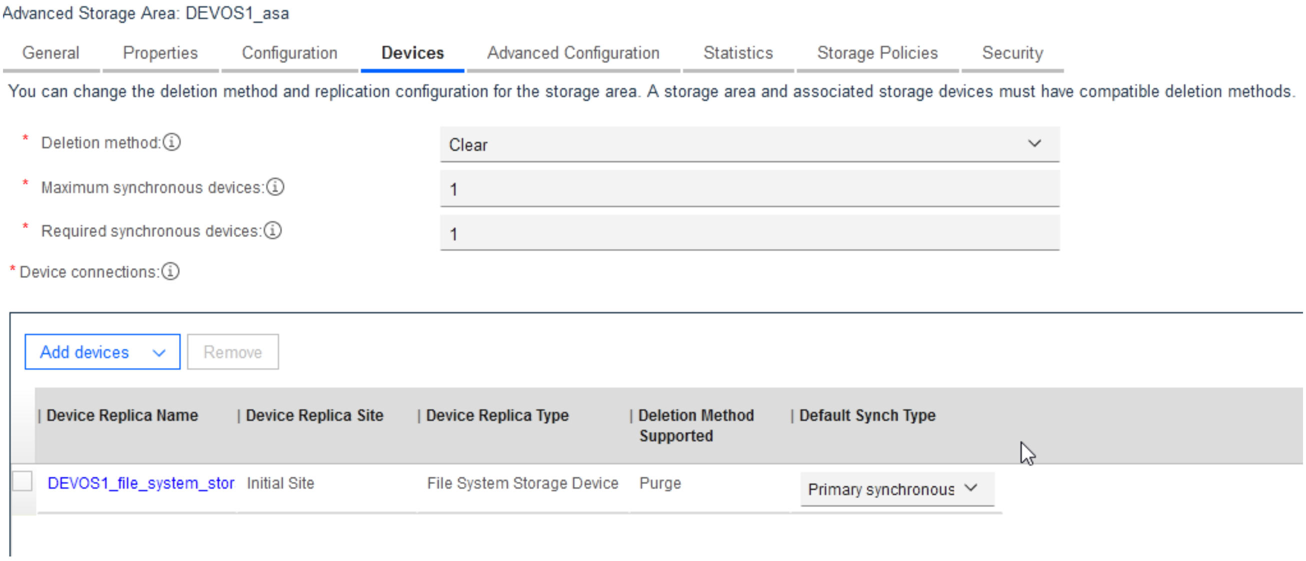
4.2.1 Review the Storage Areas

_1. From the last section, you still have a browser open, navigated to ACCE, the Administration Console for Content Engine. You are logged on using the username and the password which you obtained before, and you have the Object store "DEVOS1" open.

_2. In the navigation area on the left side, open "Administrative" → "Storage" → "Advanced Storage". Click on "Advanced Storage Areas".

_3. On the right side a list view opens with the Advanced storage areas of the DEVOS1 Object store. One exists with the name "DEVOS1_asa". Click on its link in the "Display Name" column to open its properties.

_4. Review the information on the “Configuration”, the “Devices” and the “Advanced Configuration” tabs to determine that the Advanced Storage Area supports compression, is using only one Storage Device, and is configured to not use any Content Integrity options.



Advanced Storage Area: DEVOS1_asa

You can change the deletion method and replication configuration for the storage area. A storage area and associated storage devices must have compatible deletion methods.

* Deletion method: Clear

* Maximum synchronous devices: 1

* Required synchronous devices: 1

* Device connections:

Device Replica Name	Device Replica Site	Device Replica Type	Deletion Method Supported	Default Synch Type
DEVOS1_file_system_stor	Initial Site	File System Storage Device	Purge	Primary synchronous

_5. Close the Advanced Storage Area.

_6. In the navigation area on the left side, open "Administrative" → "Storage". Click on " Storage Areas".

_7. On the right side opens a list view with the other storage areas defined in the system. Notice the Default Database Storage Area.

> The "Default Database Storage Area" is always created, when the Object Store is created.

_8. In the navigation area on the left side, open "Administrative" → "Storage". Click on "Storage Policies".

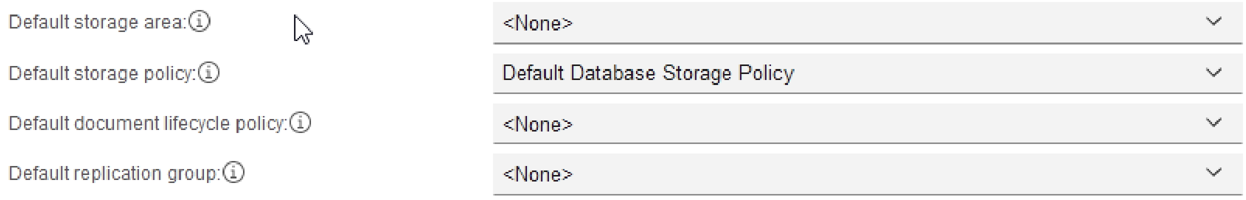
_9. On the right side a list view opens with the storage policies. Notice that we have two storage policies, one for each storage area.

> When a storage area is completely filled, the administrator needs to provision new storage and create a new storage area for it. Now, if the storage area objects are directly used in the documents and the document classes, a huge number of objects would need to be updated. For this reason, a level of indirection is introduced: In the documents and the document classes, the storage policy is used. The storage policy in turn stores the storage area(s) into which the documents should currently be stored.
>
> Storage policies also allow to distribute storage of documents to multiple storage areas to support a higher throughput for storing new documents.
>
> See further details in the documentation: https://www.ibm.com/docs/en/filenet-p8-platform/5.5.x?topic=area-storage-policies.

4.2.2 Change default storage policy for the "Client Document" document classes

_1. In the navigation area on the left side, open "Data Design" → "Classes" → "Document". Click on your "usrxxx Client Document" class to bring up its properties.

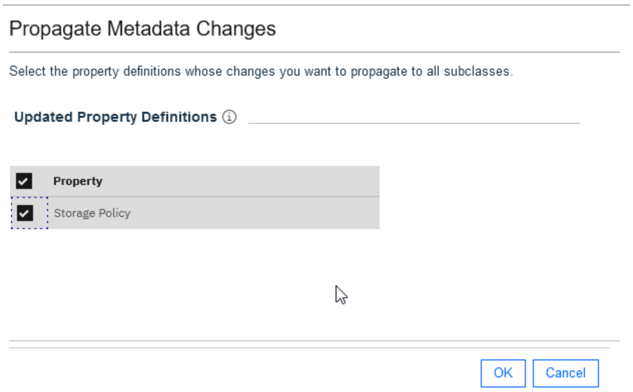
_2. On the properties, scroll down on the "General" tab to find the “Default Storage policy” section.



Default storage area:ⓘ	<None>
Default storage policy:ⓘ	Default Database Storage Policy
Default document lifecycle policy:ⓘ	<None>
Default replication group:ⓘ	<None>

_3. Set the "Default storage policy" for the class to the storage policy of the advanced storage area "Default Advanced Storage Area". Save the change.

_4. When the "Propagate Metadata Changes" window appears, enable the propagation of the "Storage policy" property, and click “OK”.



Propagate Metadata Changes

Select the property definitions whose changes you want to propagate to all subclasses.

Updated Property Definitions ⓘ

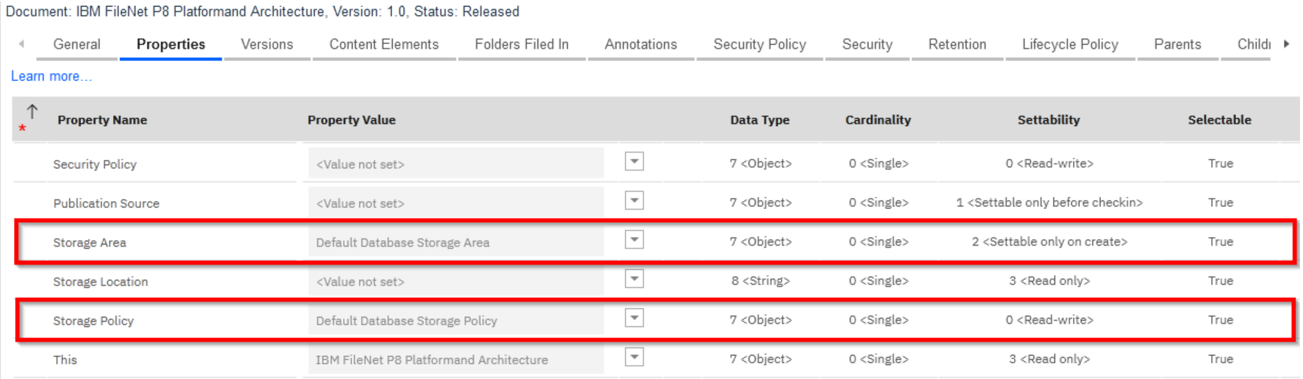
☑	Property
☑	Storage Policy

> New documents will now be stored in the Advanced Storage Area. The existing documents however are unaffected, they are still stored on the "Default Database Storage Area".

_5. In the navigation area on the left side, navigate to "Browse"-→ "Root Folder". Click on the "usrxxx Client Onboarding" folder to bring up the list of contained documents on the right side.

_6. Bring up the document properties of one of the documents, by clicking the link in the "Containment name" column.

_7. Select the "Properties" tab and scroll down until you find the properties "Storage Policy" and "Storage Area".



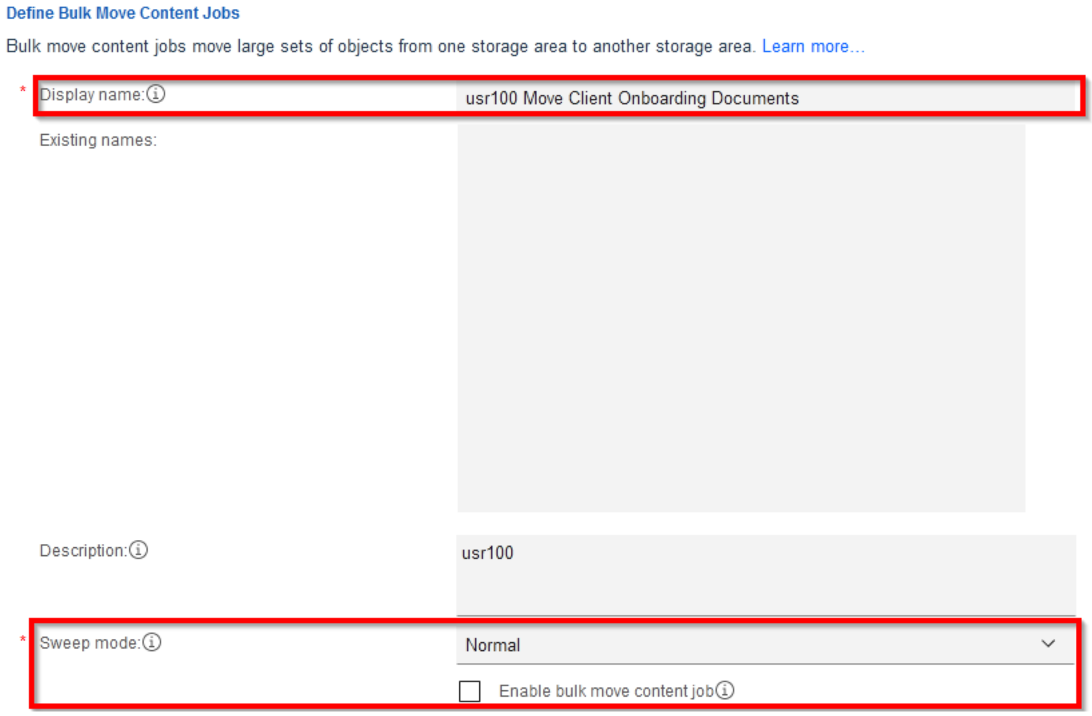
Document: IBM FileNet P8 Platformand Architecture, Version: 1.0, Status: Released

Property Name	Property Value	Data Type	Cardinality	Settability	Selectable
Security Policy	<Value not set>	7 <Object>	0 <Single>	0 <Read-write>	True
Publication Source	<Value not set>	7 <Object>	0 <Single>	1 <Settable only before checkin>	True
Storage Area	Default Database Storage Area	7 <Object>	0 <Single>	2 <Settable only on create>	True
Storage Location	<Value not set>	8 <String>	0 <Single>	3 <Read only>	True
Storage Policy	Default Database Storage Policy	7 <Object>	0 <Single>	0 <Read-write>	True
This	IBM FileNet P8 Platformand Architecture	7 <Object>	0 <Single>	3 <Read only>	True

4.2.3 Move documents between Storage Areas

_1. In the navigation area on the left side, navigate to "Sweep Management" → "Job Sweeps". Right-click on "Bulk Move Content Jobs" and select "New Bulk Move Content Job".

_2. Set the name to "Move Client Onboarding Documents". Prefix the name with your username. Set the Sweep mode to "Normal". Don't enable the bulk move job yet. Then click on "Next".



Define Bulk Move Content Jobs

Bulk move content jobs move large sets of objects from one storage area to another storage area. Learn more...

* Display name: usr100 Move Client Onboarding Documents

Existing names:

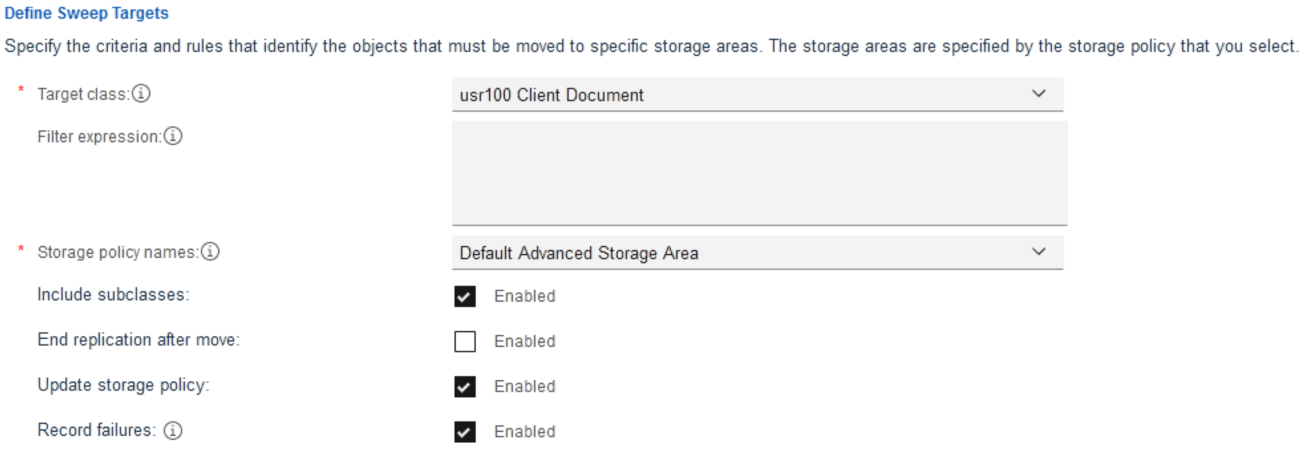
Description: usr100

* Sweep mode: Normal

□ Enable bulk move content job

_3. On the next page, select the "usrxxx Client Document" document class containing your username. Also select the storage policy of the Advanced Storage Area "Default Advanced Storage Area". Then select the three options "Include subclasses", "Update Storage policy", and "Record failures". Click on "Next", and dismiss the warning.

> The meaning of "Include subclasses" and "Record failures" is self-explanatory. The "Update Storage Policy" likewise: When it is enabled it will update the storage policy of the document. The storage policy of the documents controls among others, which storage area is selected for a new version of the document.



Define Sweep Targets

Specify the criteria and rules that identify the objects that must be moved to specific storage areas. The storage areas are specified by the storage policy that you select.

* Target class: usr100 Client Document

Filter expression:

* Storage policy names: Default Advanced Storage Area

Include subclasses: ☑ Enabled

End replication after move: □ Enabled

Update storage policy: ☑ Enabled

Record failures: ☑ Enabled

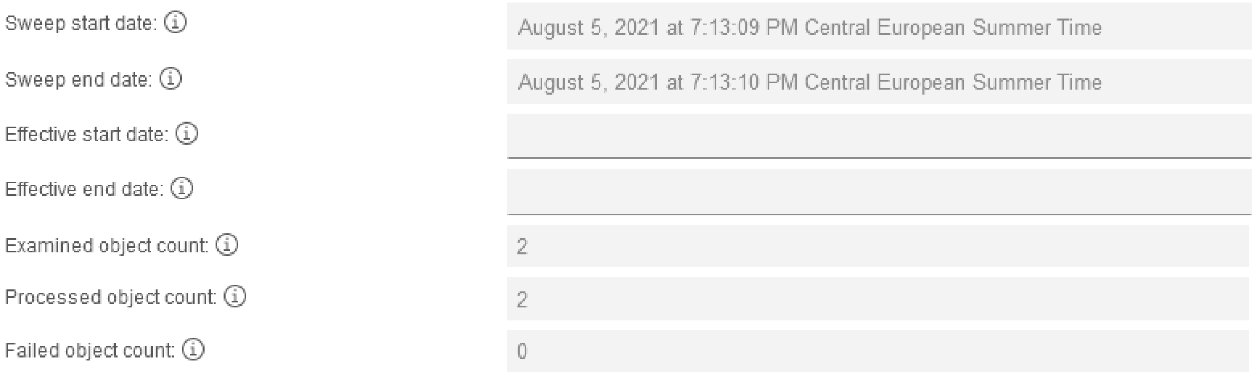
_4. No change is needed on the next page. Without assigning a schedule, the job will run as soon as it is enabled. Click "Next", then "Finish" and then "Open".

_5. The properties of the new Bulk move content job are shown. Enable the Bulk move content job on the top row of the properties dialog, then press "Save".

_6. Wait some time for the system to start the bulk move job. The sweep job has started when the property "Sweep start date" has a value. Keep clicking on "Refresh".

> The default time between runs of the Bulk Move Processing is 300 seconds, it can be configured for the FileNet Content Platform Engine Domain in the Sweep Subsystem tab. So in worst case, it can take 5 minutes before the sweep starts.

_7. The job has been completed when a time stamp appears for the "Sweep end date" property. The number of successfully moved documents is shown on the "Processed object count" row.



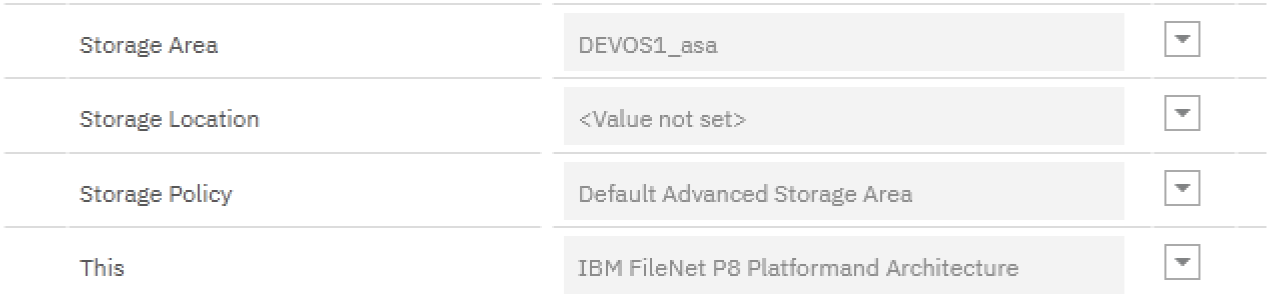
Sweep start date:	August 5, 2021 at 7:13:09 PM Central European Summer Time
Sweep end date:	August 5, 2021 at 7:13:10 PM Central European Summer Time
Effective start date:	
Effective end date:	
Examined object count:	2
Processed object count:	2
Failed object count:	0

> If the "Failed object count" is having a positive value, and the sweep job has been configured accordingly, the list of failed documents should be visible on the "Sweep results" tab. With no errors, the tab would remain empty, though.

4.3 Verification Steps

At this step, your Document Classes should be configured to store new documents in the advanced storage area.

Furthermore, also the already existing documents would be stored in the advanced storage area. You can check it by reviewing the Storage Area and Storage Policy properties of the documents. You might need to click on "Refresh" if properties window for the documents were still open.



Storage Area	DEVOS1_asa
Storage Location	<Value not set>
Storage Policy	Default Advanced Storage Area
This	IBM FileNet P8 Platformand Architecture

5 Exercise: Triggering actions

5.1 Introduction

When using FileNet Content Engine in a business application, it is often needed to trigger some actions, when for example a new document or folder is created, or a new version of a document has been uploaded. In FileNet Content Engine, *Subscriptions* can be created for this purpose. They define, that when a specific event on an object of a specific class is observed, a specified action would need to be performed. The subscription object is thereby independent of the action. The action can run in one of two modes, synchronous, or asynchronous, leading to synchronous subscriptions and asynchronous subscriptions.

Synchronous subscriptions are invoked as part of the processing of the update to the object causing it. For example, a synchronous subscription, configured to be invoked with a document creation, will be processed as part of the document creation. This means, it will always be processed together with the document creation, and if it is failing, the creation of the document will fail and be aborted with an error message. An *asynchronous* subscription on the other hand is processed after and independently of the action which triggered it. If it is failing, it will be retried, by default 8 times, then it is marked as "poisoned" and must be handled by an administrator.

The actions processed by the subscriptions could in the easiest case be code modules written in JavaScript or Java. These code modules are stored in the FileNet Object store and are then executed by the FileNet Content Engine. Other alternatives would be to start a FileNet Process Engine workflow, or invoke a webhook.

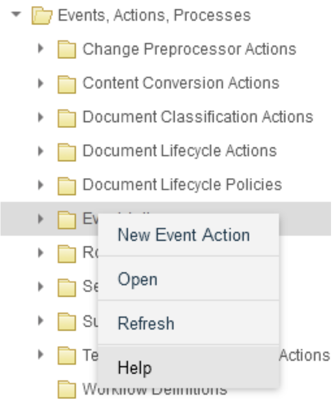
In our example here, any document uploaded using one of the Customer Information document classes, which were defined in the previous exercises, shall be filed automatically in all the case folders, which have the matching Reference id. For this purpose, some case folders have been defined on the Object store with the reference ids "TEST1" and "TEST2". The case folders are instances of the subclass named "SWAT Jam Case Folder".

5.2 Exercise Instructions

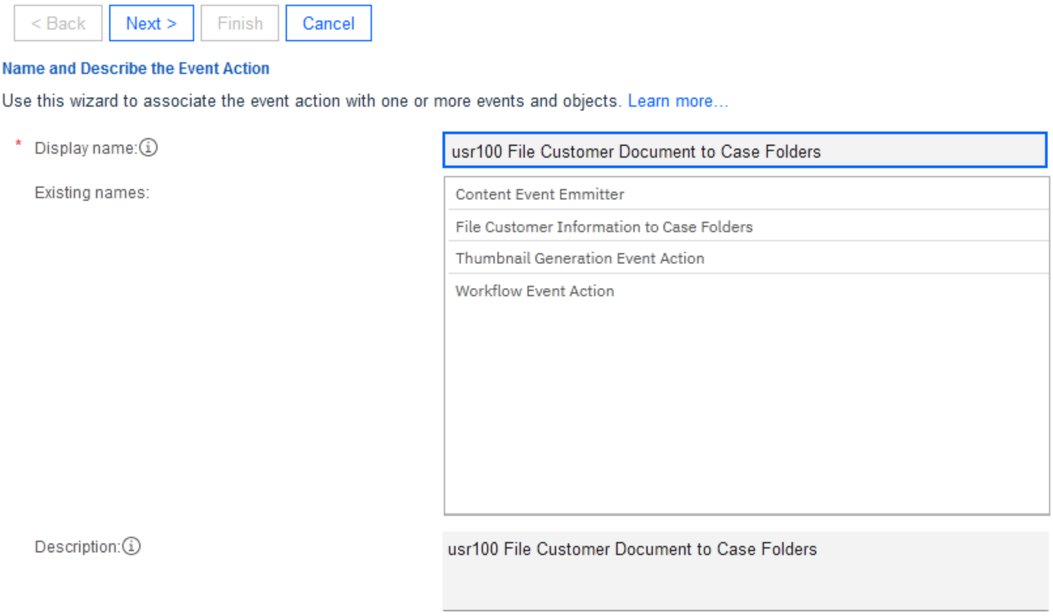
5.2.1 Creating the Event Action Script

_1. From the last section, you still have a browser open, navigated to ACCE, the Administration Console for Content Engine. You are logged on using the username and the password which you obtained before, and you have the Object store "DEVOS1" open.

_2. In the navigation area on the left side, open "Event, Actions, Processes", by clicking on the triangle. Then right-click on "Event Actions" and select "New Event Action".


Events, Actions, Processes
Change Preprocessor Actions
Content Conversion Actions
Document Classification Actions
Document Lifecycle Actions
Document Lifecycle Policies
New Event Action
Open
Refresh
Help
Actions

_3. Provide the name "usrxxx File Customer Document to Case Folders". Use your username as a prefix so that you can find the right subscription later. Press "Next".



Name and Describe the Event Action

Use this wizard to associate the event action with one or more events and objects. Learn more...

* Display name: usr100 File Customer Document to Case Folders

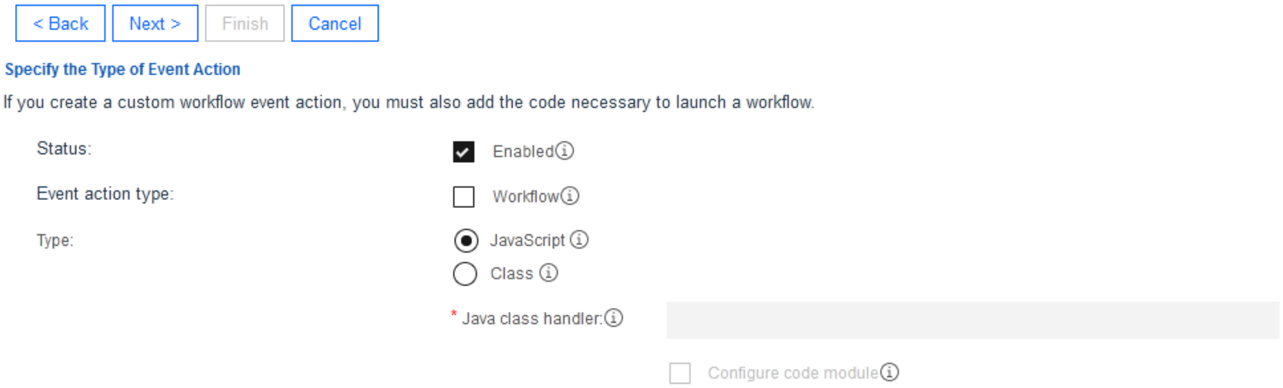
Existing names:

- Content Event Emmitter
- File Customer Information to Case Folders
- Thumbnail Generation Event Action
- Workflow Event Action

Description: usr100 File Customer Document to Case Folders

_4. On the next page, set the event action type to "Event Action", and leave the Status set to "Enabled", then press "Next".

_5. On the next page set the Type field to "JavaScript", press "Next".



Specify the Type of Event Action

If you create a custom workflow event action, you must also add the code necessary to launch a workflow.

Status: ☑ Enabled

Event action type: □ Workflow

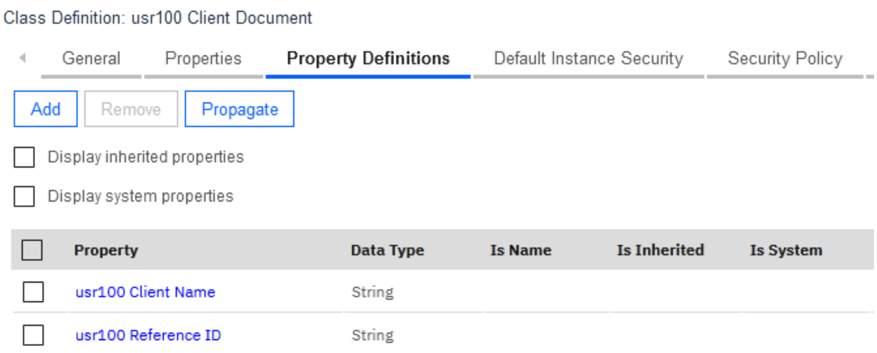
Type: ● JavaScript ○ Class

* Java class handler:

□ Configure code module

_6. Download the FileClientDocumentEventAction.js script from the lab materials page. Open the file in a text editor.

_7. In ACCE, leave the Wizard for the New Event action open. In the navigation area, open "Data Design" → "Classes" → "Document". Open the "usrxxx Client Document" class with your username at the beginning. Go to the Property Definitions tab.



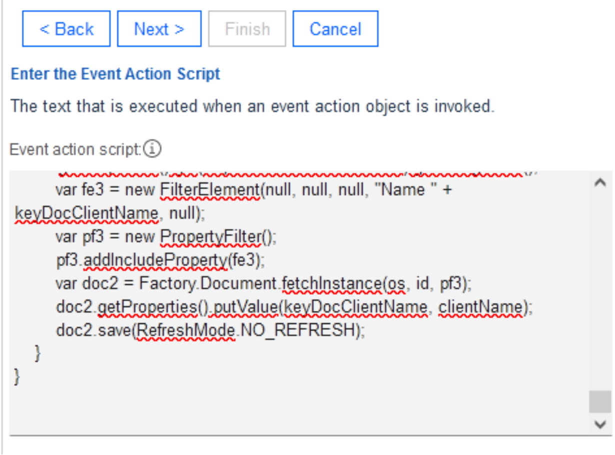
□ Display inherited properties

□ Display system properties

□	Property	Data Type	Is Name	Is Inherited	Is System
□	usr100 Client Name	String			
□	usr100 Reference ID	String			

_8. Click on the two property definitions, one after the other. From the dialog which opens, copy the value shown for the symbolic name, and paste it into the script at the right places. The symbolic name for the Client name goes into the variable named keyDocClientName, and the symbolic name for the Reference ID goes into the variable named keyDocReferenceIdentifierName. No further updates are needed.

_9. In ACCE get back to the definition of the Event Action. Completely remove everything from the "Event action script" text box. Then select the complete content of the file FileClientDocumentEventAction.js in the text editor. Copy and paste the content into the "Event action script" text box. Then click "Next", then "Finish", and then "Close".



Enter the Event Action Script

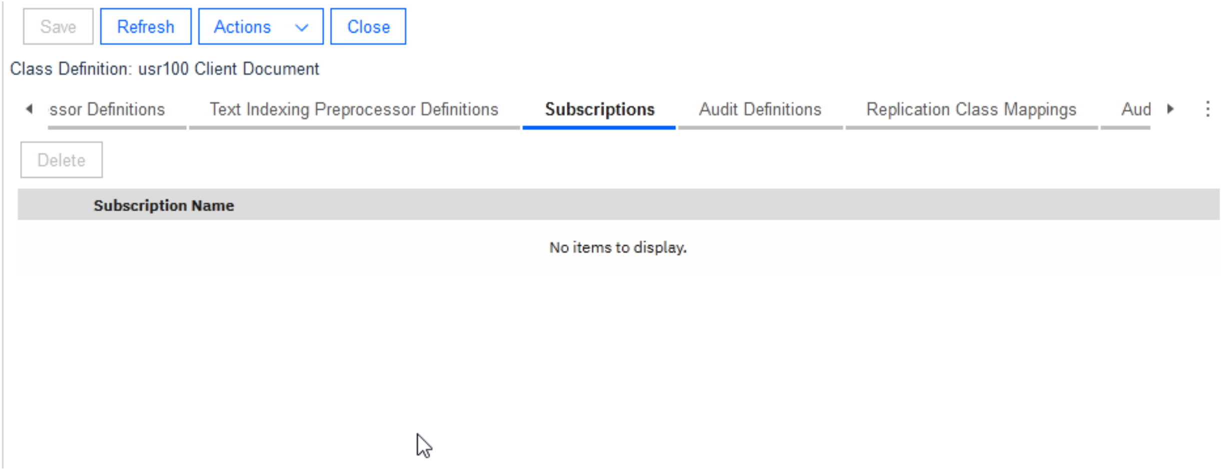
The text that is executed when an event action object is invoked.

Event action script:

```
      var fe3 = new FilterElement(null, null, null, "Name " +
keyDocClientName, null);
      var pf3 = new PropertyFilter();
      pf3.addIncludeProperty(fe3);
      var doc2 = Factory.Document.fetchInstance(os, id, pf3);
      doc2.getProperties().putValue(keyDocClientName, clientName);
      doc2.save(RefreshMode.NO_REFRESH);
   }
}
```

5.2.2 Creating the Subscription

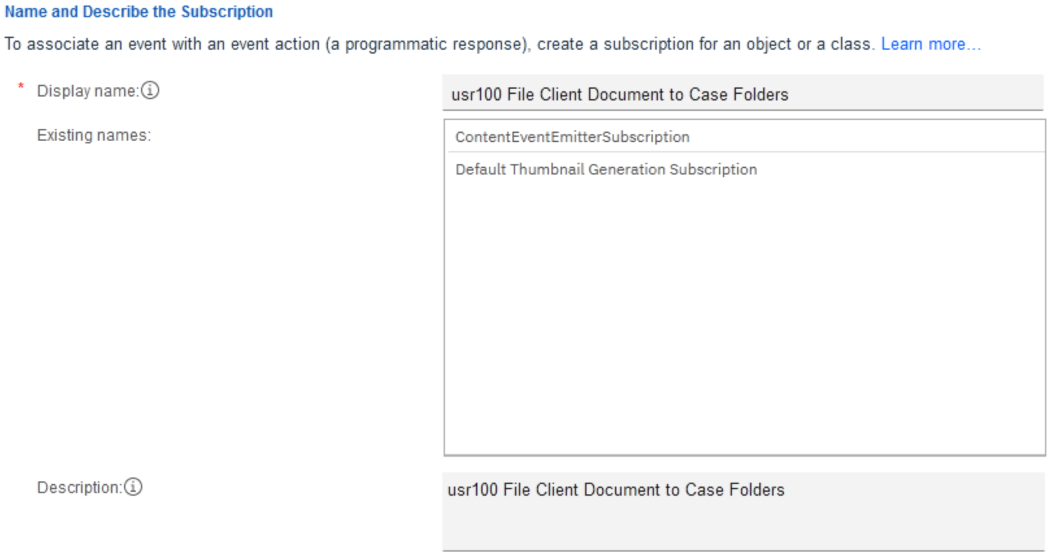
_1. In ACCE, in the object store "DEVOS1", open "Data Design", "Classes", and "Document". Then click on the "usrxxx Client Document" class which is prefixed with your username to bring up its properties. With the open properties window for this class, select the "Subscriptions" tab. You might have to scroll to the right to find it.



Class Definition: usr100 Client Document

No items to display.

_2. From the "Actions" pulldown menu, select "New Subscription". Provide the name "usrxxx File Client Document to Case Folders" as the subscription name. It is not necessary to use the same name as the event action. Press "Next".



Name and Describe the Subscription

To associate an event with an event action (a programmatic response), create a subscription for an object or a class. Learn more...

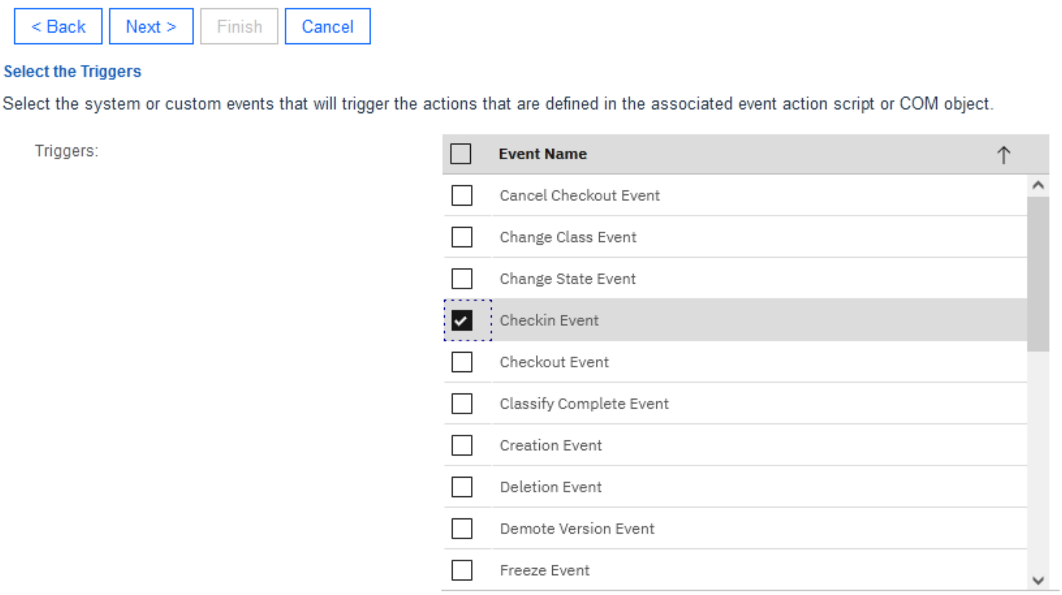
* Display name: usr100 File Client Document to Case Folders

Existing names:
ContentEventEmitterSubscription
Default Thumbnail Generation Subscription

Description: usr100 File Client Document to Case Folders

_3. No changes on the next page of the wizard. Press "Next".

_4. On the "Select the triggers" page, select the "Checkin Event". Press "Next".



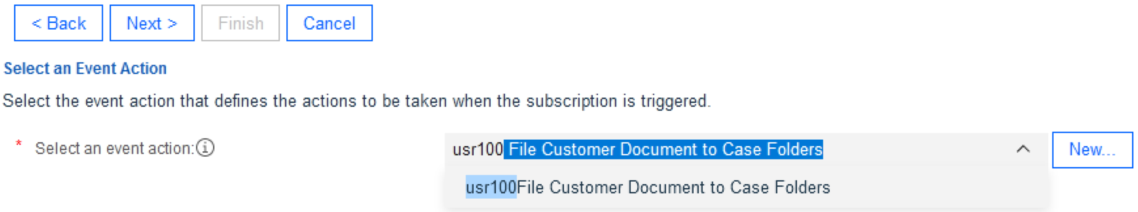
Select the Triggers

Select the system or custom events that will trigger the actions that are defined in the associated event action script or COM object.

Triggers:

☐	Event Name
☐	Cancel Checkout Event
☐	Change Class Event
☐	Change State Event
☑	Checkin Event
☐	Checkout Event
☐	Classify Complete Event
☐	Creation Event
☐	Deletion Event
☐	Demote Version Event
☐	Freeze Event

_5. On the next page, select the Event Action created before. The best method might be to start typing the username in the text box, to bring up the right name. Then click "Next"



Select an Event Action

Select the event action that defines the actions to be taken when the subscription is triggered.

* Select an event action: usr100File Customer Document to Case Folders

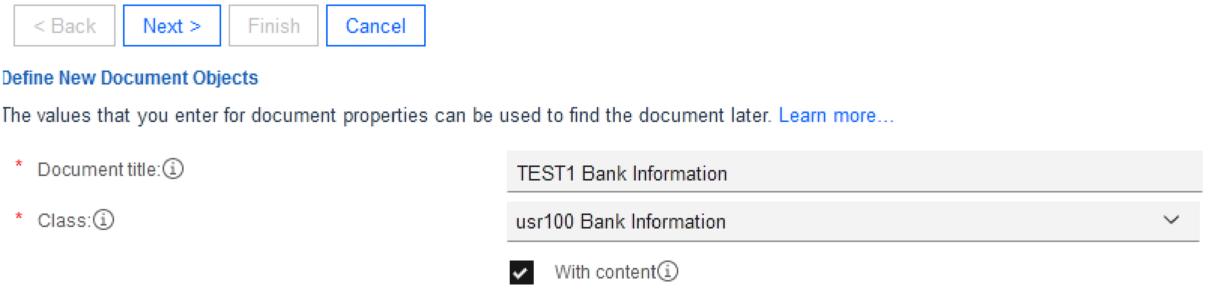
usr100File Customer Document to Case Folders

_6. On the next page, select to include the subclasses in the subscription. Through this, you can create documents for the two subclasses "Client Identification" or "Bank Information" and still get the document filed in the case folder. Leave the "run synchronously" disabled, this subscription should be an asynchronous subscription. Then press "Next", then "Finish", then "Close"

5.3 Verification Steps

_1. In ACCE, in the navigation pane open "Browse", "Root Folder" and click on the "usrxxx Client Onboarding" folder with your username to bring up its content on the right side. Click on the "Actions" menu and select "New document".

_2. Use the class "usrxxx Bank Information" which is prefixed with your username. Select the name "TEST1 Bank Information".



Define New Document Objects

The values that you enter for document properties can be used to find the document later. Learn more...

* Document title: TEST1 Bank Information

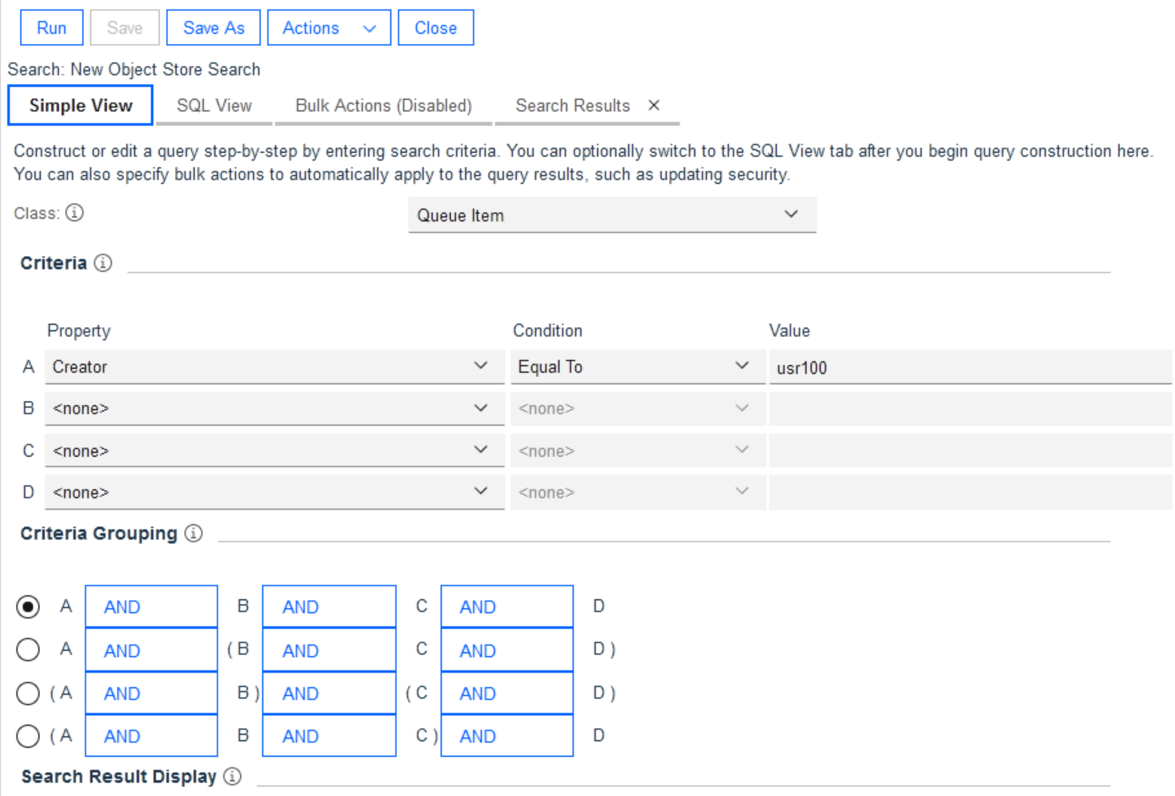
* Class: usr100 Bank Information

☑ With content

_3. On the next page, upload some content element to the new document, then click "Next".

_4. On the "Object properties" page set the property "usrxxx Reference ID" which is prefixed with your username to the value “TEST1”. You can leave the rest of the properties unchanged. No further changes are needed in the Add document wizard, so press "Next" until it does not show up anymore, then "Finish", then "Open".

_5. For troubleshooting, click on "Search" in the Navigation area and select "New Object Store Search". Provide the class name "Queue Item" for the search. Filter the results by selecting property "Creator" with condition "Equal to". Provide your username as the value. then click on "Run" to run the search.



Search: New Object Store Search

Construct or edit a query step-by-step by entering search criteria. You can optionally switch to the SQL View tab after you begin query construction here. You can also specify bulk actions to automatically apply to the query results, such as updating security.

Class: Queue Item

Criteria

	Property	Condition	Value
A	Creator	Equal To	usr100
B	<none>	<none>	
C	<none>	<none>	
D	<none>	<none>	

Criteria Grouping

●	A	AND	B	AND	C	AND	D
○	A	AND	(B	AND	C	AND	D)
○	(A	AND	B)	AND	(C	AND	D)
○	(A	AND	B	AND	C)	AND	D

Search Result Display

_6. If a result is shown, the subscription was not, or not yet successfully executed. Review the property values. One if the first ones is the "Retry Count". It starts with 7 and is decreased with every invocation. Another field is "Next retry date". If the script does not run successful, the "Retry Count" will decrease with every invocation, until finally it reaches zero, and the "Next retry date" is emptied. When it reaches this state, the entry is not tried anymore, is marked "Poisoned". An administrator can try setting the retry count to a positive value, and set the next retry date to a date in the future to retry the Event Subscription. If that is not done, the administrator is supposed to delete the unsuccessful queueitem entry.

_7. If the subscription is not successfully executed, it would be necessary to review the logfiles of the FileNet Content Platform Engine to determine clues why the execution was not successful. If you find errors in your script, you can delete the subscription. The dialog allows you to delete the event action together with the subscription. Then you need to recreate both anew.

_8. If the subscription is successful, you might not be able to run a query for it before it is removed by the system. In this case you should see the document being filed in the "Browse/Root Folder/Case Folders/Sample Test Folder TEST1" (if you have chosen Reference ID TEST1), or " Browse/Root Folder/Case Folders/Sample Test Folder TEST2" for the value TEST2. In the properties, the client name property of the document should have been filled with the client name from the case folder. If the document isn't filed on one of the case folders, then maybe the value for the Reference ID in the document did not match any of the case folders.

6 Exercise: Content Based Retrieval

6.1 Introduction

A central task of FileNet Content Platform Engine is to find entities - for example documents and folders. In order to assist with this, metadata of the entities being stored are separately stored in a database. This makes it possible to use the metadata to find those documents and folders at a later time, e.g. using the name of a customer, or some ID. Now, the main concern of "Content Based Retrieval" (CBR) is the ability to find documents by search expressions based on their content, not, or not only using the metadata of the documents stored in the properties. To support Content based Retrieval, FileNet uses a separate component, the Content Search Services component.

In this exercise it will be reviewed how the Content Search Services component is configured, and how queries for the content of a document can be made in the FileNet Content Platform Engine. For more details on this topic, refer to the FileNet Content Platform Engine documentation on Content Based Retrieval: https://www.ibm.com/docs/en/filenet-p8-platform/5.5.x?topic=documents-finding-objects-content-based-retrieval

6.2 Exercise Instructions

_1. Open a browser and navigate to ACCE, the Administration Console for Content Engine. Login using the username and the password which you obtained before. Do **not** open the DEVOS1 Object store, at least not yet.

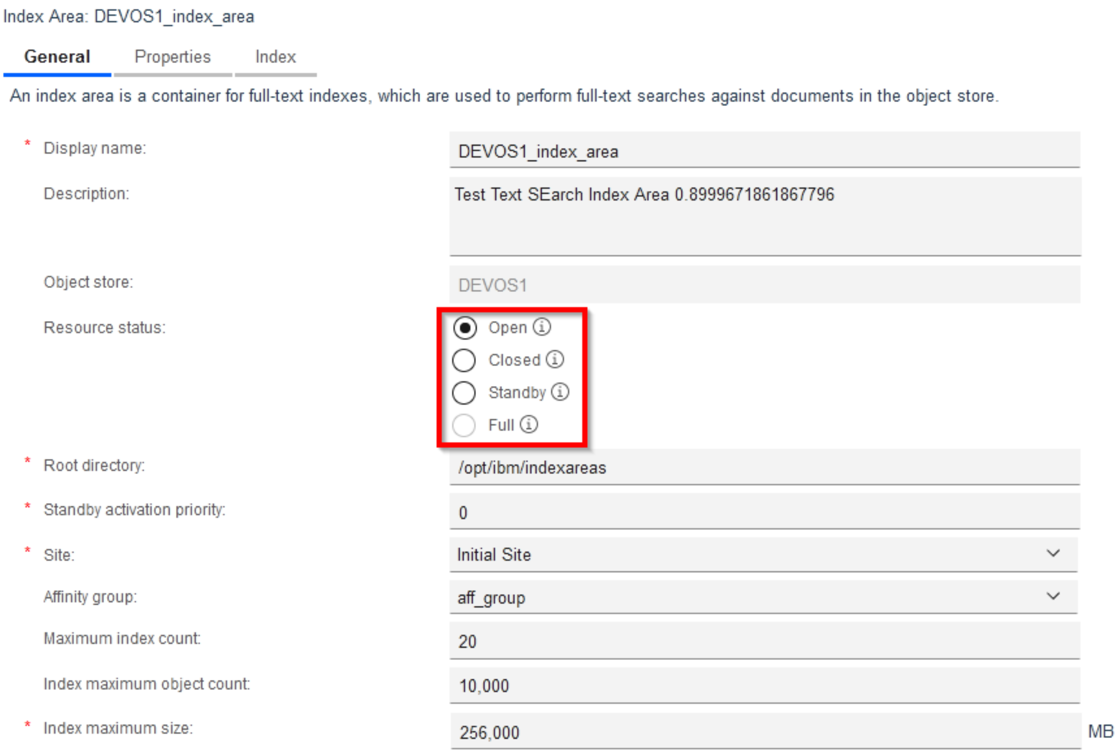
_2. In the navigation area, open "Global Configuration", and "Administration". Then click on "Text Search Servers".

_3. Click on the content search services servers defined here, to review their properties. Notice that communication is using SSL, and that both servers are using the same hostname icp4adeploy-css-svc-1.

> This is the name of a kubernetes service, which can be used inside the Openshift cluster for communication between the components. The name is not valid outside of the Openshift cluster.

_4. Now open the Object Store DEVOS1. On the DEVOS1 Object Store properties select the "Text Search" tab. Notice here that the "IBM Content search services" are enabled, and that the indexing language is configured to be English (en).

_5. In order to perform Content Based Retrieval, it is also needed to define one or more Index areas. However, you cannot see the index areas as access is only granted to Object store administrators. Therefore, find the screenshots below:



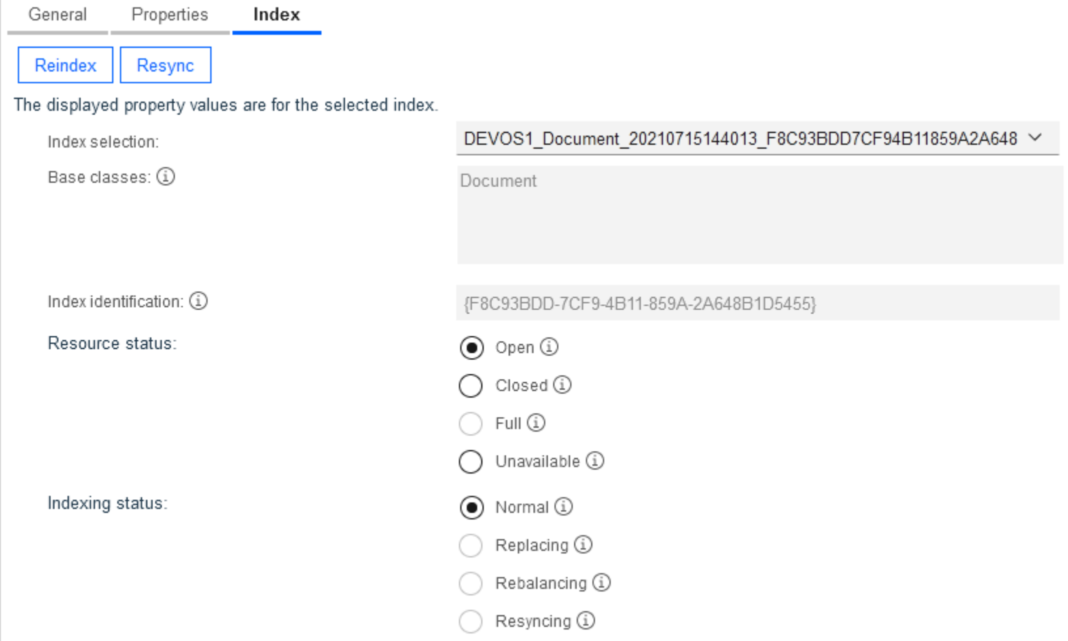
Index Area: DEVOS1_index_area

General | Properties | Index

An index area is a container for full-text indexes, which are used to perform full-text searches against documents in the object store.

* Display name:	DEVOS1_index_area
Description:	Test Text SEarch Index Area 0.8999671861867796
Object store:	DEVOS1
Resource status:	● Open ⓘ ○ Closed ⓘ ○ Standby ⓘ ○ Full ⓘ
* Root directory:	/opt/ibm/indexareas
* Standby activation priority:	0
* Site:	Initial Site
Affinity group:	aff_group
Maximum index count:	20
Index maximum object count:	10,000
* Index maximum size:	256,000 MB

On the "General" tab, the status of the whole index area can be set. If the index area diskspace is reached, the index area can for example be set to the status "Full", and a new Index area can be defined, which will then be used for indexing new documents.



General | Properties | **Index**

Reindex | Resync

The displayed property values are for the selected index.

Index selection:	DEVOS1_Document_20210715144013_F8C93BDD7CF94B11859A2A648
Base classes: ⓘ	Document
Index identification: ⓘ	{F8C93BDD-7CF9-4B11-859A-2A648B1D5455}
Resource status:	● Open ⓘ ○ Closed ⓘ ○ Full ⓘ ○ Unavailable ⓘ
Indexing status:	● Normal ⓘ ○ Replacing ⓘ ○ Rebalancing ⓘ ○ Resyncing ⓘ

On the "Index" tab, you can review the settings of the individual indexes (by selecting the index in the "Index selection" entry box). Notice that only a single index for the "Document" class exists. The lower portion of the dialog (not included in the screenshot) shows the size of the index and the number of documents stored in the index. Furthermore the resource status can be set and the indexing status of the index is shown

_6. In the Navigation area, now open "Data Design", "Classes", "Document" and click on the "usrxxx Client Document" class with your username as a prefix. The properties of the class are shown on the right side. Review the content of the "General" tab. Scroll down to find the checkbox for "CBR enabled". Notice it is enabled.

> CBR stands for "Content Based Retrieval". It is enabled as it was inherited from the value set for the "Document" class, for which it was enabled too.

_7. If the "CBR enabled" would not have been set already, and you would have needed to enable it newly, then you would also need to open the "Actions" menu and invoke the action "Index for Content Search (include subclasses)". Upon successful completion (required administrative access on the Object store) a dialog informs that an Index job has been created.

_8. The administrator can further go to "Administrative", and click on "Index Jobs Manager. Through the "Actions" menu you can Pause, Resume and Cancel Index Jobs.

6.3 Verification Steps

_1. Now let's run a CBR query, to verify the indexing actually works. In the navigation area, click on "Search". The Saved searches dialog appears on the right side. Click on the button with "New Object Store Search".

_2. In the "New Object Store Search" dialog, select the "SQL View" tab and remove all text entered here. Replace it by below text, then click "Run" to try running the search:

```
SELECT d.This
FROM Document d
INNER JOIN ContentSearch c ON d.This = c.QueriedObject
WHERE CONTAINS(d.*, 'CBR and CSS')
```

_3. The IBM FileNet P8 Platform and Architecture document uploaded earlier should be found. If you have used a different document, try to find some words from that document instead.

Congratulations you have successfully completed the Setting up FileNet Content Platform Engine for Automation Projects on Cloud Pak for Business Automation