Contents

[1. FileNet P8 Architecture 3](#_Toc167117557)

[1.1 What we will cover in this training 4](#_Toc167117558)

[2. Content Platform Engine 4](#_Toc167117559)

[2.1 Walkthrough of Administration Console for Content Platform Engine 4](#_Toc167117560)

[2.1.1 Domain 4](#_Toc167117561)

[2.1.2 Object Store 4](#_Toc167117562)

[2.1.3 Exploring Classes 4](#_Toc167117563)

[2.2 Security 4](#_Toc167117564)

[2.2.1 ACE/ACL (Access Control Entry/Access Control List) 4](#_Toc167117565)

[2.2.3 Role – based access control (RBAC) 5](#_Toc167117566)

[2.2.4 Security Inheritance (Security Proxy) 5](#_Toc167117567)

[2.2.5 Marking Sets 5](#_Toc167117568)

[2.2.6 Security Policies 5](#_Toc167117569)

[2.3 Document Class 5](#_Toc167117570)

[2.3.1 Creation of Property Templates 5](#_Toc167117571)

[2.3.2 Creation of Document Class 5](#_Toc167117572)

[2.3.3 Adding Property Definitions to the Document Class 5](#_Toc167117573)

[2.3.4 Checking the Security 5](#_Toc167117574)

[2.3.5 Creating a Document Instance 5](#_Toc167117575)

[2.3.5 Versioning of Documents 5](#_Toc167117576)

[2.3.6 Implementing Security Proxy 5](#_Toc167117577)

[2.3.7 Implementing Marking Sets 5](#_Toc167117578)

[2.3.8 Implementing RBAC 8](#_Toc167117579)

[2.3.9 Implementing Security Policy 8](#_Toc167117580)

[2.4 Events and Subscriptions 8](#_Toc167117581)

[2.4.1. Introduction 8](#_Toc167117582)

[2.4.2 Creating the Code Module 8](#_Toc167117583)

[2.4.3 Creating the Event Action 8](#_Toc167117584)

[2.4.4 Creating the Subscription 8](#_Toc167117585)

[2.4.5 Testing the Event Action 8](#_Toc167117586)

[2.5 Search Objects in Content Engine 8](#_Toc167117587)

[2.5.1 How to do Object Store Search 8](#_Toc167117588)

[2.5.2 How to do Bulk actions 8](#_Toc167117589)

[2.6 Audit History 8](#_Toc167117590)

[2.6.1 Introduction 8](#_Toc167117591)

[2.6.2 How to Configure and view Audit History in FileNet 8](#_Toc167117592)

[2.7 Content Engine Java API 9](#_Toc167117593)

[2.7.1 Setting up Eclipse 9](#_Toc167117594)

[2.7.2 Creating a FileNet Connection 9](#_Toc167117595)

[2.7.3 Working with Documents 9](#_Toc167117596)

[3. IBM Content Navigator 9](#_Toc167117597)

[3.1 Introduction to Navigator 9](#_Toc167117598)

[3.2 Walkthrough to Navigator Admin desktop 9](#_Toc167117599)

[3.3. Development in Navigator 9](#_Toc167117600)

[3.3.1 Creation of Object Store repositories 9](#_Toc167117601)

[3.3.2 Creation of Desktops 9](#_Toc167117602)

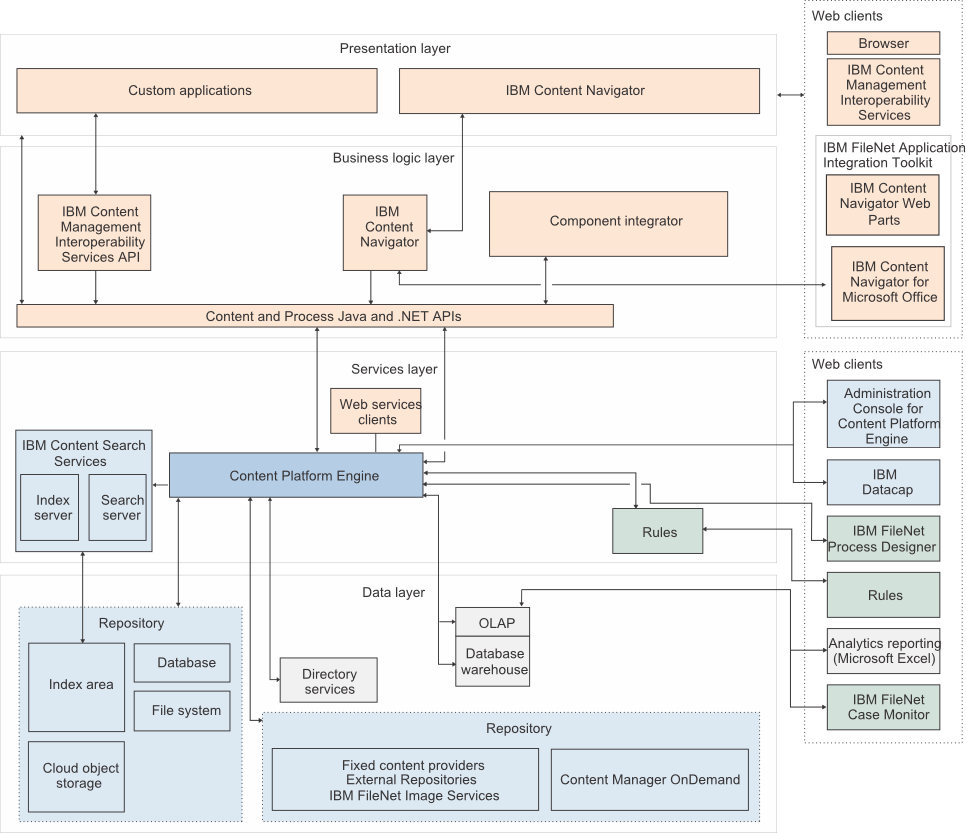
[3.4 Navigator Customizations 9](#_Toc167117603)

[4. IBM Case Manager 9](#_Toc167117604)

IBM FileNet Training

# FileNet P8 Architecture

The FileNet P8 family of products includes back-end services, development tools, and applications that address enterprise content and process management requirements



## What we will cover in this training

The below components we will cover as part of this training

* + 1. **Content Platform Engine (CPE)**
    2. **IBM Navigator**
    3. **IBM Case Manager**

# Content Platform Engine

## 2.1 Walkthrough of Administration Console for Content Platform Engine

### 2.1.1 Domain

### 2.1.2 Object Store

### 2.1.3 Exploring Classes

## 2.2 Security

The Content Engine resides within a Java Platform, Enterprise Edition (Java EE) application server, and uses the Java Authentication and Authorization (JAAS) standard as the basis for authentication. The JAAS programming model is a standard Java framework that manages authentication and authorization.

### 2.2.1 ACE/ACL (Access Control Entry/Access Control List)

**ACE Source:** Every ACE has a source, either Default,Direct, Inherited or Template.

* + 1. **Default:** Permissions are placed on an object by the Default instance security ACL of its class, as well as permissions placed on a sub classs by its parent class. Default ACEs are directly editable. If you do, its source type becomes Direct.
    2. **Direct:** Permissions are added directly to an object. Direct ACEs are directly editable
    3. **Inherited:** Permissions are placed on the object by a security parent or by setting up a relationship with an object-value property whose security proxy type has been set to Inherited. Inherited ACEs are not directly editable.

Example: ***Security Proxy Concept***

* + 1. **Template:** Permissions are placed on the object by a security policy. Template ACEs are not directly editable and do not appear on classes. Rather, a document, folder, or custom object class can have a default security policy which will pass template ACEs to the instances of the class, if all the conditions for the template apply.

Example**:** ***Security Policy Concept***

### 2.2.3 Role – based access control (RBAC)

You can create roles in your object store as another way to grant access to objects. Role-based access depends on three facets of authorization

* What access does the role grant?
* Who, meaning which users or groups, are the members of this role?
* Which objects can be accessed by members of this role

Using roles can simply security in some cases because you can update the role parameters, such a members or what kind of access the role has, without having to update all the objects to which members of the role have access.

### 2.2.4 Security Inheritance (Security Proxy)

### 2.2.5 Marking Sets

### 2.2.6 Security Policies

## 2.3 Document Class

### 2.3.1 Creation of Property Templates

### 2.3.2 Creation of Document Class

### 2.3.3 Adding Property Definitions to the Document Class

### 2.3.4 Checking the Security

### 2.3.5 Creating a Document Instance

### 2.3.5 Versioning of Documents

### 2.3.6 Implementing Security Proxy

### 2.3.7 Implementing Marking Sets

1: WHO Guidelines:

-> Document Title (default property)

-> Country Name -> (India, USA)

-> Severity

india users group

usa users group

-> Default Instance Security :

-> Marking set : can be used with the help of property template only

-> Security Proxy: can be used with the help of property template only

Marking security consists of the Add marking, Remove marking, and Use Marked Objects.

**Add marking and Remove marking**

A user with Add rights to a marking can set the property value associated with the marking, if it has not been set. Only those markings to which the user has Add rights will show in the list of marking values ​​available to be set in a property. A user with Remove rights to a marking can remove the marking value.

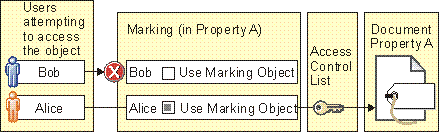
For example:

1. A Document has a property associated with a marking set. No value has been specified for the property.
2. The marking set has markings Red, Blue, and Green.
3. Alice has Use rights to Red; Use and Add rights to Blue and Use, Add and Remove rights to Green.

When Alice views the Document properties, she can set the property value to Blue or Green but not Red. If the property was set to Green, she could alter it to be Blue. If the property was set to Blue, Alice would be unable to alter the property's value.

**Use Marked Objects**

Use right determines whether the presence of the marking on an object constrains access to that object. If the user has Use right to the marking, access to the object will not be constrained.



In this example, Alice has the Use Marked Objects access right which lets her bypass the marking. Her access to the object will be evaluated by the object's ACL. Bob does not have Use Marked Objects and therefore will neither see nor have access to the object, regardless of any permissions the object's ACL might grant him.

Markings and marking sets are Content Platform Engine objects, each with a class description:

* Markings are objects that combine metadata behavior with access control behavior in a way that allows an object's access control to change by changing a property value.
* Marking sets have containers for markings. Marking sets are associated with a Property Template which can then be used to add a property to one or more classes.

Implementation:

Lets assume we have country property on a Covid Guidelines document class, and assume

We have 2 countries

* + 1. India ( intuser1 is the user belongs to india country )
    2. USA ( lucy is the user belongs to USA )

We have one admin user p8 admin:

Scenario: Now whatever Covid Guide lines document created for which country depending on the those country specific users only should be able to see that document.

Lets Achieve this using Marking set

1. Create a Marking Set named “Country Marking”
2. Give two marking india and usa
3. Add the appropriate users to the specific markings and save the marking
4. Create a property template named “Country “ and map the marking set
5. Assign the property template to Covid Guidelines document class
6. Create an instance and test the Marking set security

### 2.3.8 Implementing RBAC

### 2.3.9 Implementing Security Policy

## 2.4 Events and Subscriptions

### 2.4.1. Introduction

### 2.4.2 Creating the Code Module

### 2.4.3 Creating the Event Action

### 2.4.4 Creating the Subscription

### 2.4.5 Testing the Event Action

## 2.5 Search Objects in Content Engine

### 2.5.1 How to do Object Store Search

### 2.5.2 How to do Bulk actions

## 2.6 Audit History

### 2.6.1 Introduction

Audit History is used to store the event information of the Document, Folders and objects in FileNet P8. With Audit History, one can view historical changes to documents in a FileNet Content Manager repository.

* In IBM ACCE, Audit history is for all types of items like documents, folders and objects.
* In IBM Content Navigator, Audit history is for document only ( can see the document related audit history )

### 2.6.2 How to Configure and view Audit History in FileNet

**In ACCE:**

1. First, need to make the property “Enable Audit History to Yes on the Object Store

2. Go to the respective document class, where you want to enable the audit history tab and then create an Audit definition by choosing what are all the events you want to audit and save it.

3. Then create a document and trigger the respective events which you have configured and then go that document and go the Audit History tab and see whether it is updated or not

**In IBM Content Navigator:**

1. Go to the respective object store repository and enable the “Document History” setting
2. Do the same in Desktop level as well
3. Then go to respective document in the navigator and open properties and see the history tab

## 2.7 Content Engine Java API

### 2.7.1 Setting up Eclipse

### 2.7.2 Creating a FileNet Connection

### 2.7.3 Working with Documents

# 3. IBM Content Navigator

## 3.1 Introduction to Navigator

## 3.2 Walkthrough to Navigator Admin desktop

## 3.3. Development in Navigator

### 3.3.1 Creation of Object Store repositories

### 3.3.2 Creation of Desktops

## 3.4 Navigator Customizations

# 4. IBM Case Manager