Application Deployment guide

Contents

[1 Introduction 2](#_Toc533868521)

[1.1 Purpose 2](#_Toc533868522)

[2 Components 2](#_Toc533868523)

[2.1 EDS 2](#_Toc533868524)

[2.1.1 EDS war creation in DMS 2](#_Toc533868525)

[2.1.2 Database Tables Required : 3](#_Toc533868526)

[2.2 FDM Artifacts 4](#_Toc533868527)

[2.2.1 DMS Class and WorkFlow Subscription Details 4](#_Toc533868528)

[2.2.2 CE Component 4](#_Toc533868529)

[2.2.3 Entry Template for DMS class 4](#_Toc533868530)

[2.2.4 Search Template for DMS 4](#_Toc533868531)

[2.3 PCC-Configuration 5](#_Toc533868532)

[2.3.1 ComponentName : 6](#_Toc533868533)

[2.3.2 Update Server details in below property file 6](#_Toc533868534)

[2.3.3 In-basket Details. 7](#_Toc533868535)

[2.3.4 Roles 7](#_Toc533868536)

[2.3.5 In-baskets 8](#_Toc533868537)

[2.4 Security Proxy: 10](#_Toc533868538)

# Introduction

## Purpose

The purpose of this document is to provide instructions for DMS build deployment into FileNet environment.

# Components

## EDS

### EDS war creation in DMS

As EDS War already deployed please add below .class files in path and property file.

1. Modified “ObjectTypes.json” file included below code

{"symbolicName": "Edelwesis"},

{"symbolicName": "DMS"},

{"symbolicName": "EdelweissDMS.Workflow.Dispatch"},

{"symbolicName": "EdelweissDMS.Workflow.Archival"}

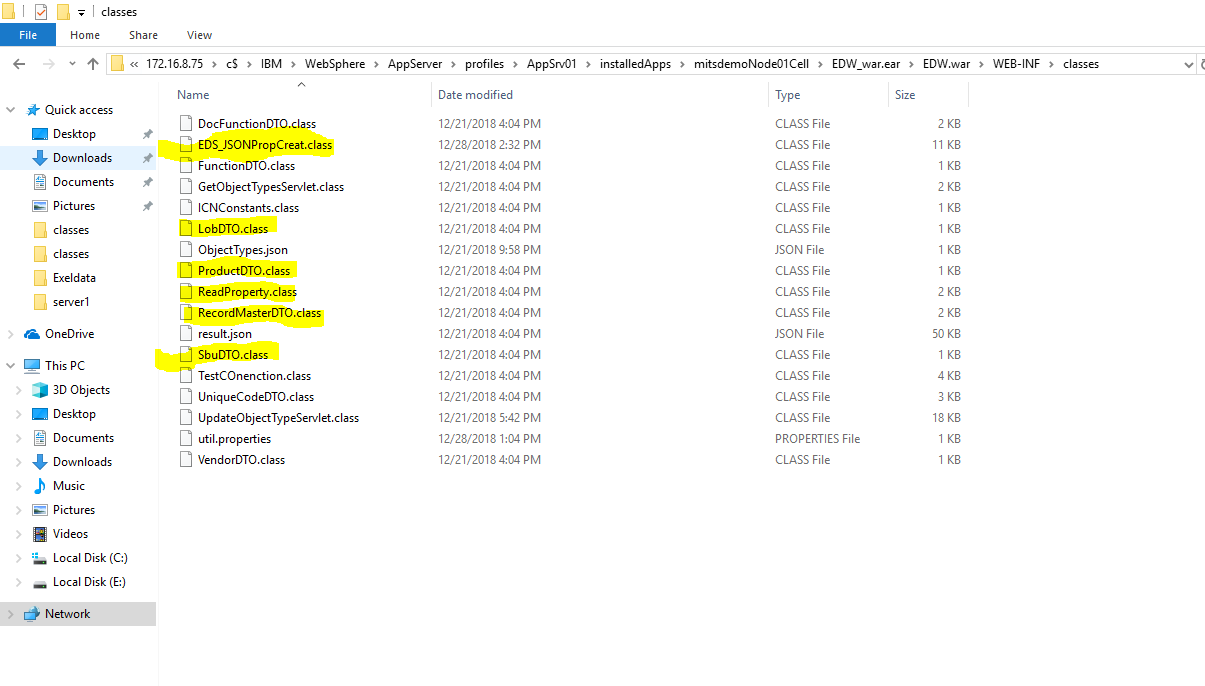
2.Modified Util.properties file with included below code.

DMS\_CLSSES = DMS, EdelweissDMS.Workflow.Dispatch,EdelweissDMS.Workflow.Archival

WorkFlowMandartoryClasses = EdelweissDMS.Workflow.Dispatch,EdelweissDMS.Workflow.Archival

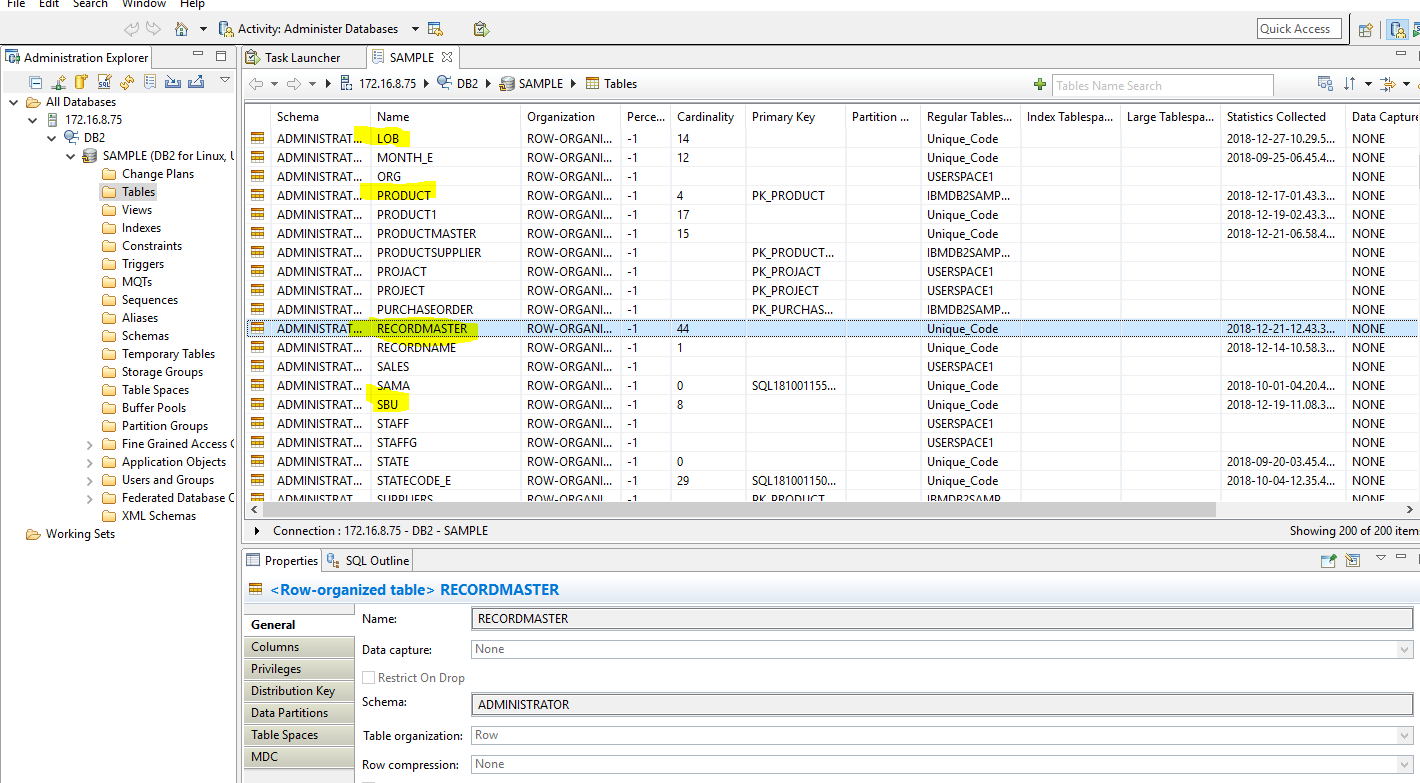
3. Included below class names in EDS.

1. EDS\_JSONPropCreat
2. SbuDTO
3. LobDTO
4. ProductDTO
5. RecordMasterDTO
6. ReadProperty



### Database Tables Required :

1. SBU
2. LOB
3. PRODUCT
4. RECORDMASTER
5. ENTITY



## FDM Artifacts

### DMS Class and WorkFlow Subscription Details

1.Created Document class name with “DMS” in FNOS and Workflow name with ‘EdelweissDMS’ with properties

2. Created workflow subscription with “create event” action.

### CE Component

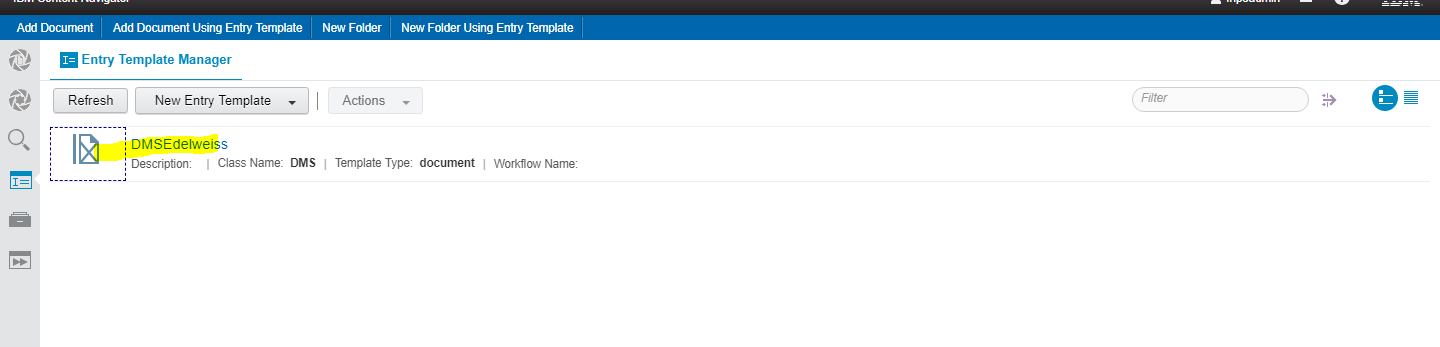
1. component to update dispatch and archival date to workflow properties.(UpdateArchivalDateinContentEngine).

2. Component for add content to existing document if Product and Client ID when same(Updateddocumentproperties).

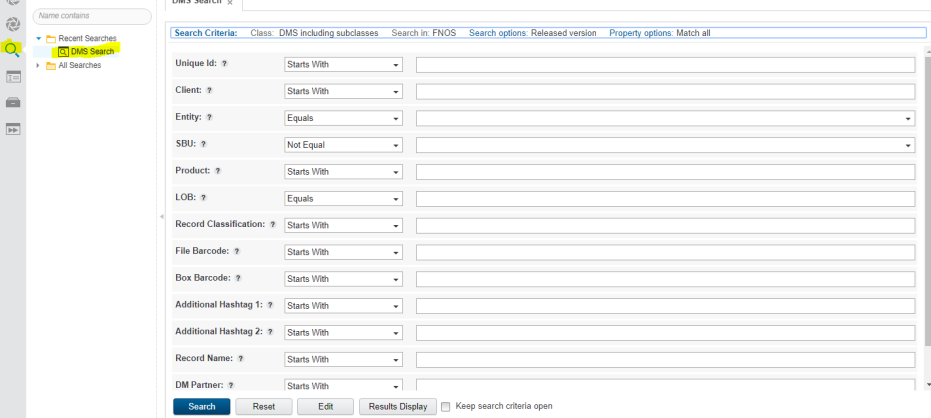
3. To maintain versions(UpdateWorkflowComponent).

4. Security to add document and folder.

### Entry Template for DMS class

****

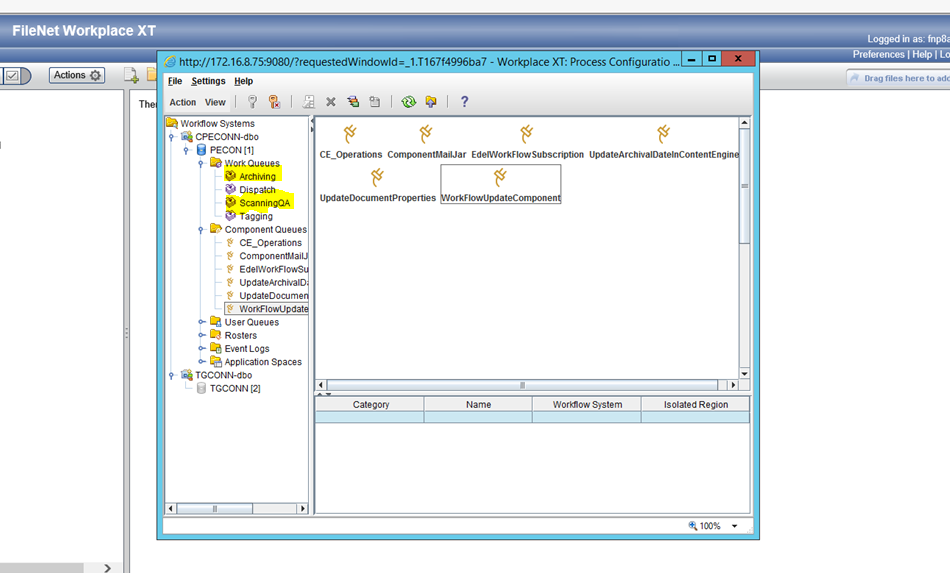
### Search Template for DMS

****

## PCC-Configuration

In “work Queue” create below two queues.

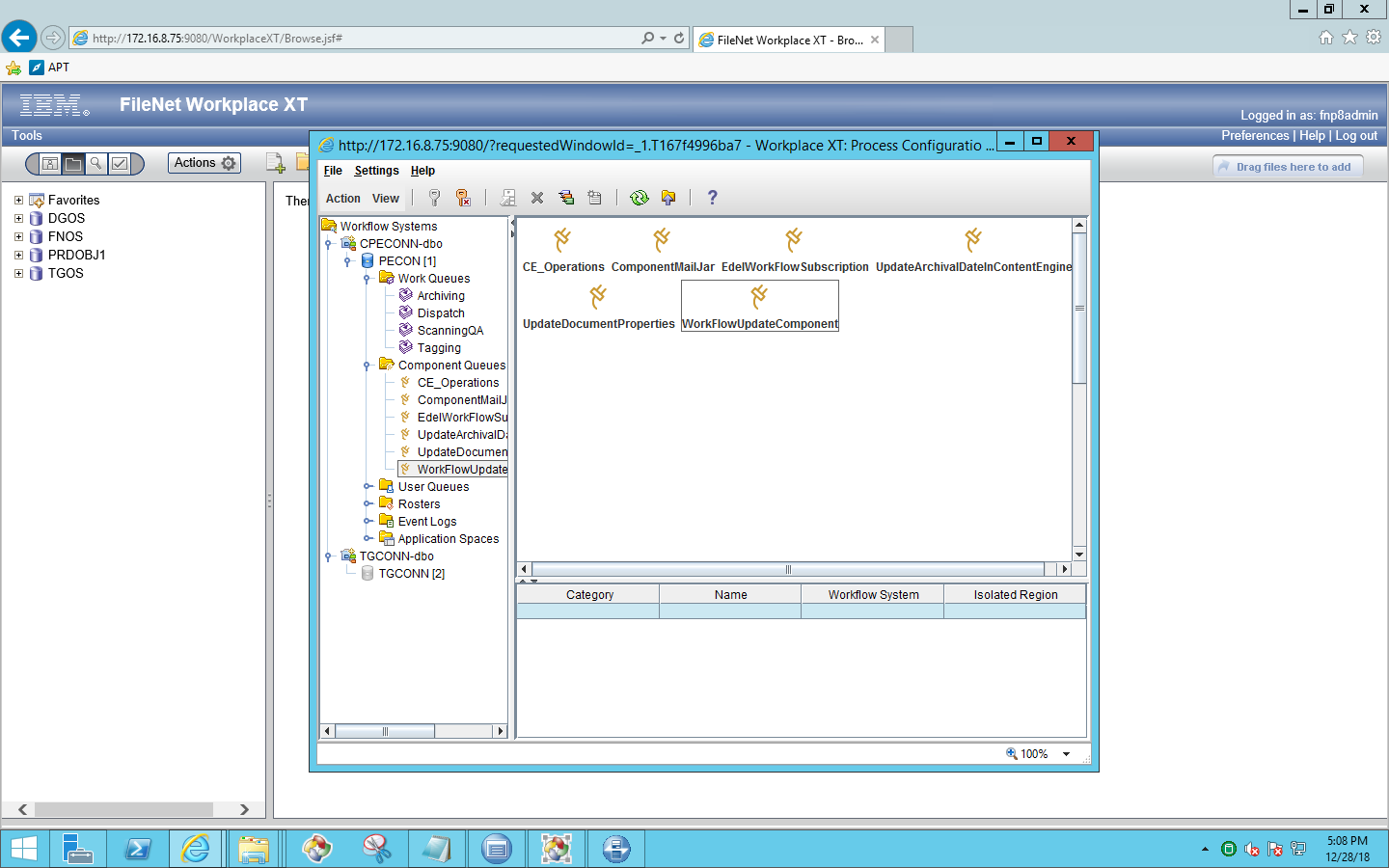
1. Archiving
2. ScanningQA.



### ComponentName :

Create below component :-

* Workflowupdatecomponent.

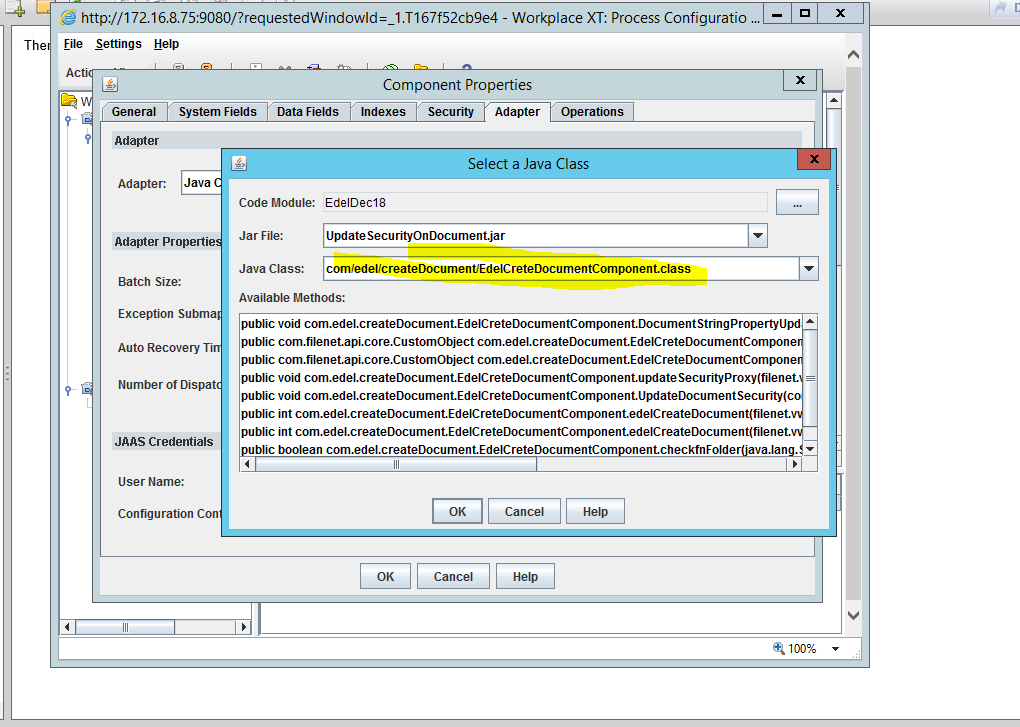


### Update Server details in below property file

Property File Path: C:\Edel\CreateDocument.properties

Expose below Component Methods

* Document string property update
* Edelcreatedocumentproperty.
* Update security proxy
* Checkinfolder
* CreateFolder



### In-basket Details.

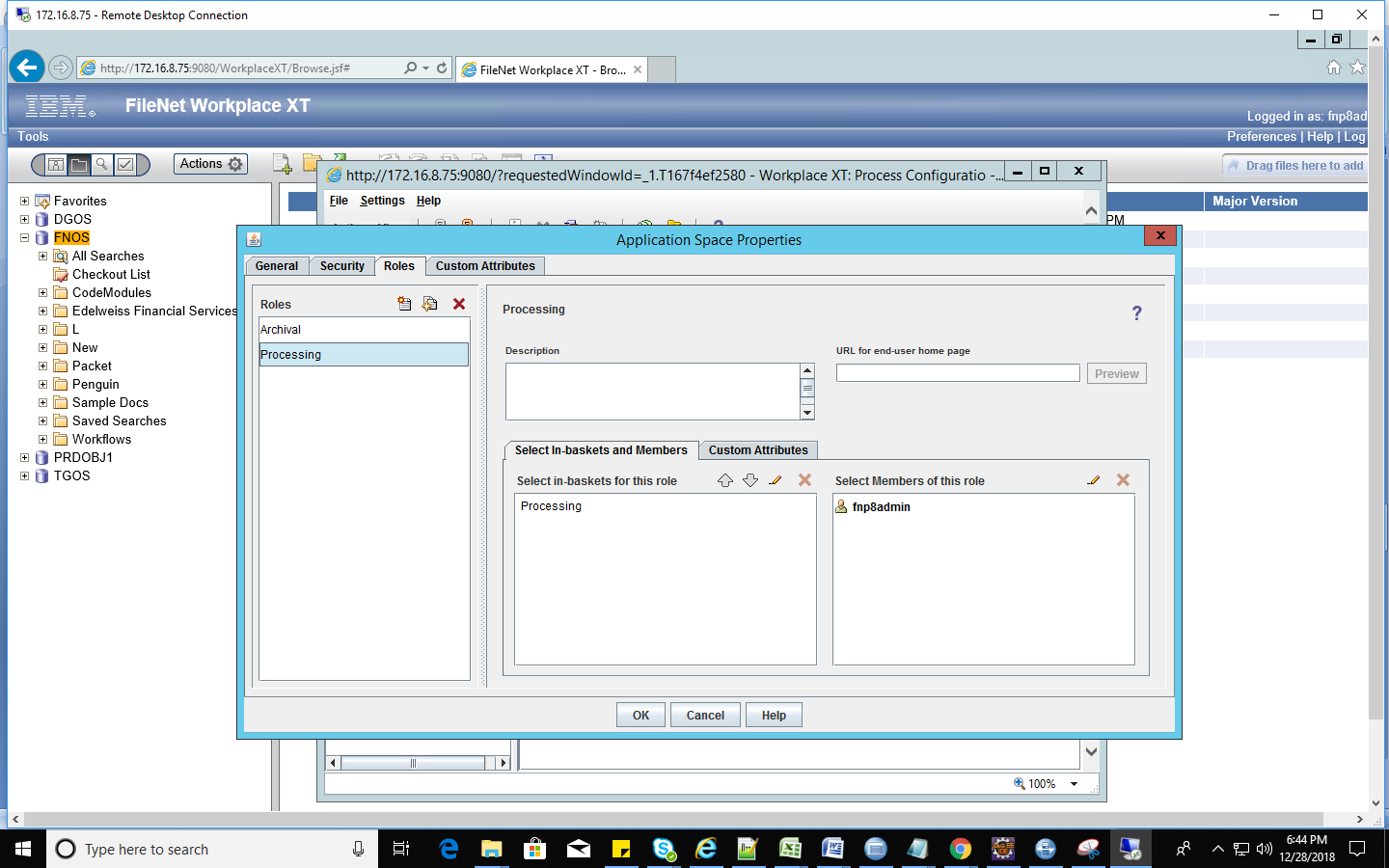
1. Created In-basket’s for different roles.
   * 1. Processing
     2. Archival
2. Processing step used In-basket ‘Processing’ and “Scanning QA” Queue used.
3. Dispatch step used In-basket ’Archival’ and “Archiving” Queue used.
4. Archive step used In-basket ’Archival’ and “Archiving” Queue used.

**Note**: Same In-basket and Same Queue used for above two –steps.

### Roles

Below are the roles

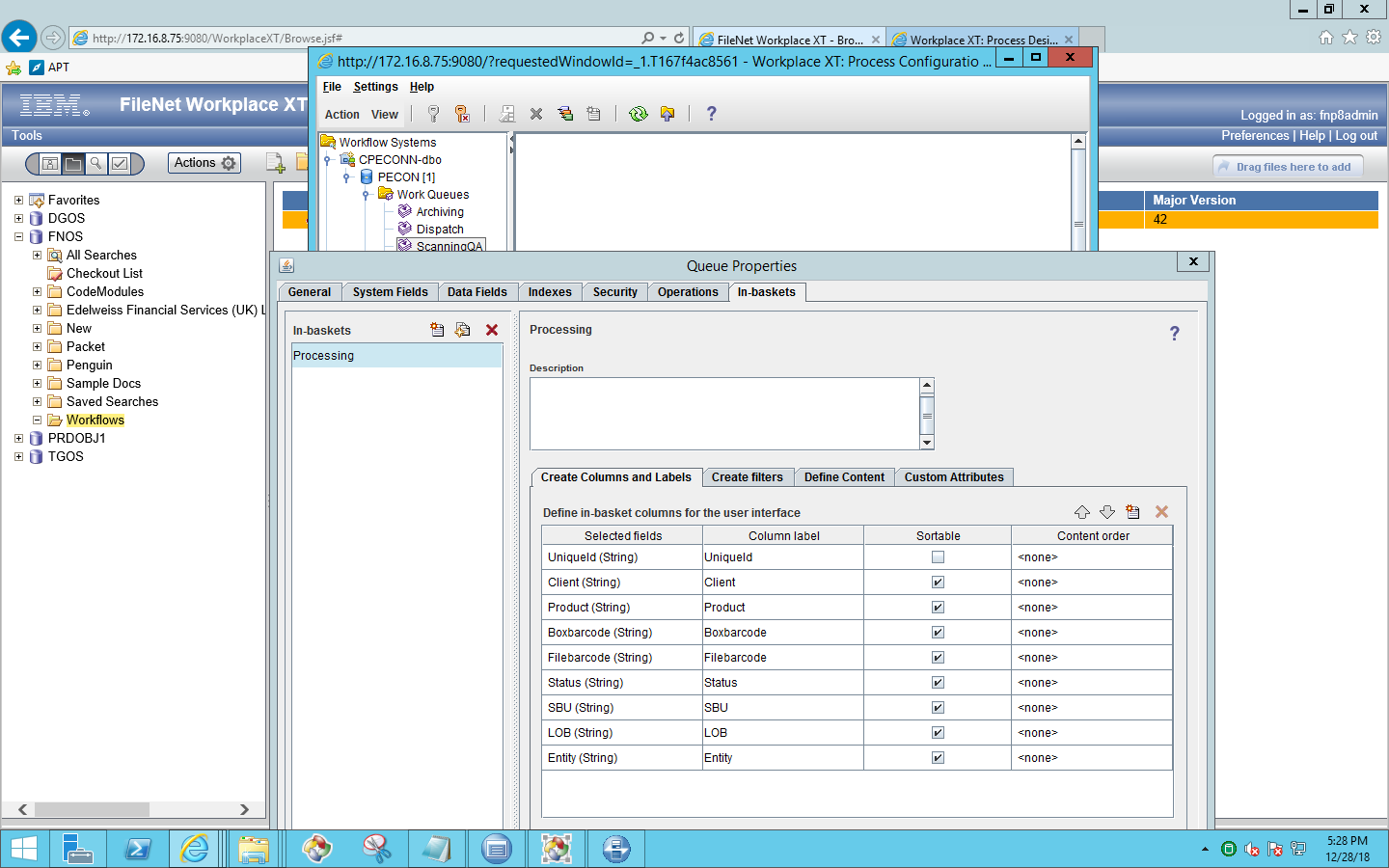
* Archival
* Processing

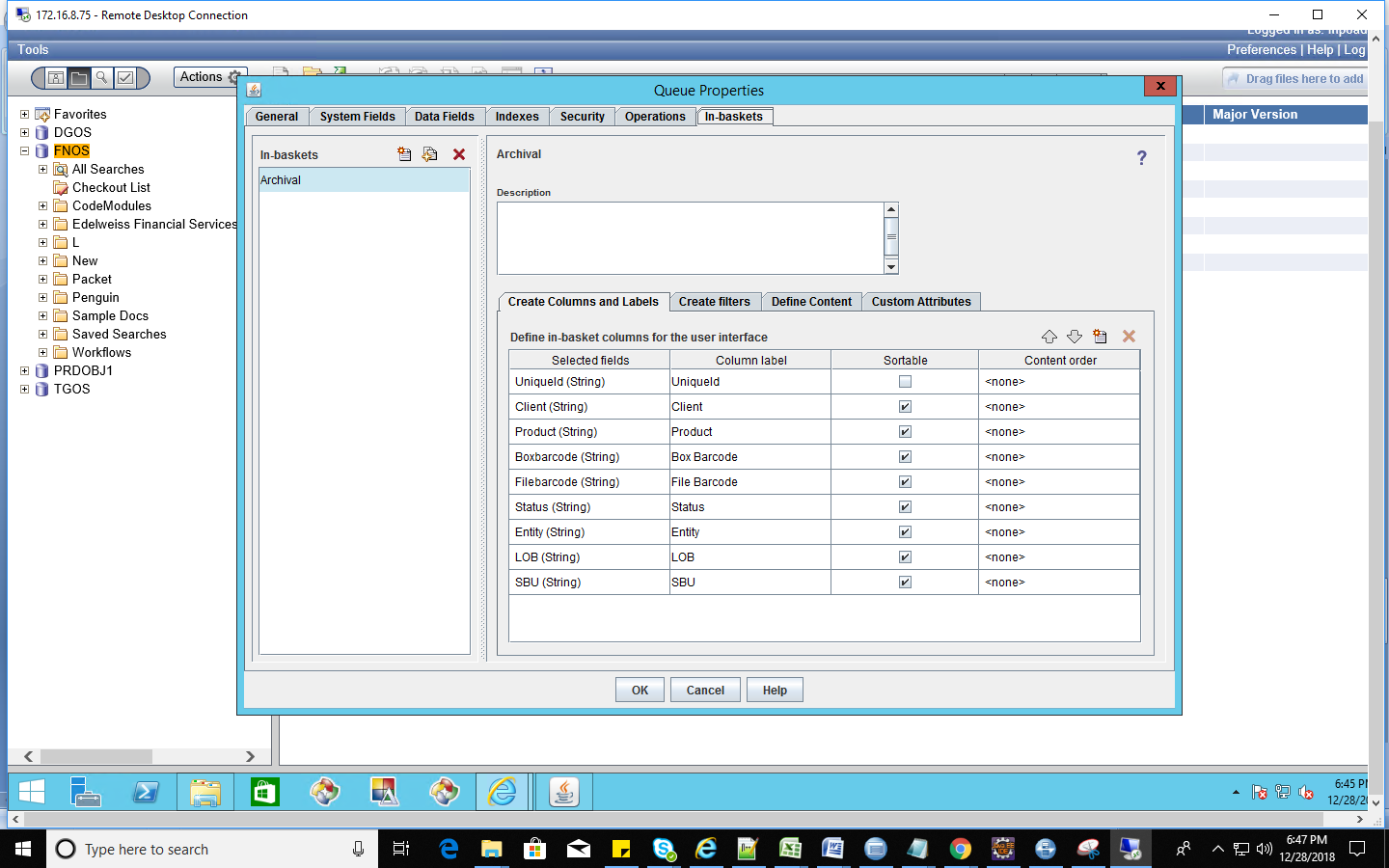


### In-baskets

Below are the In-baskets for roles.

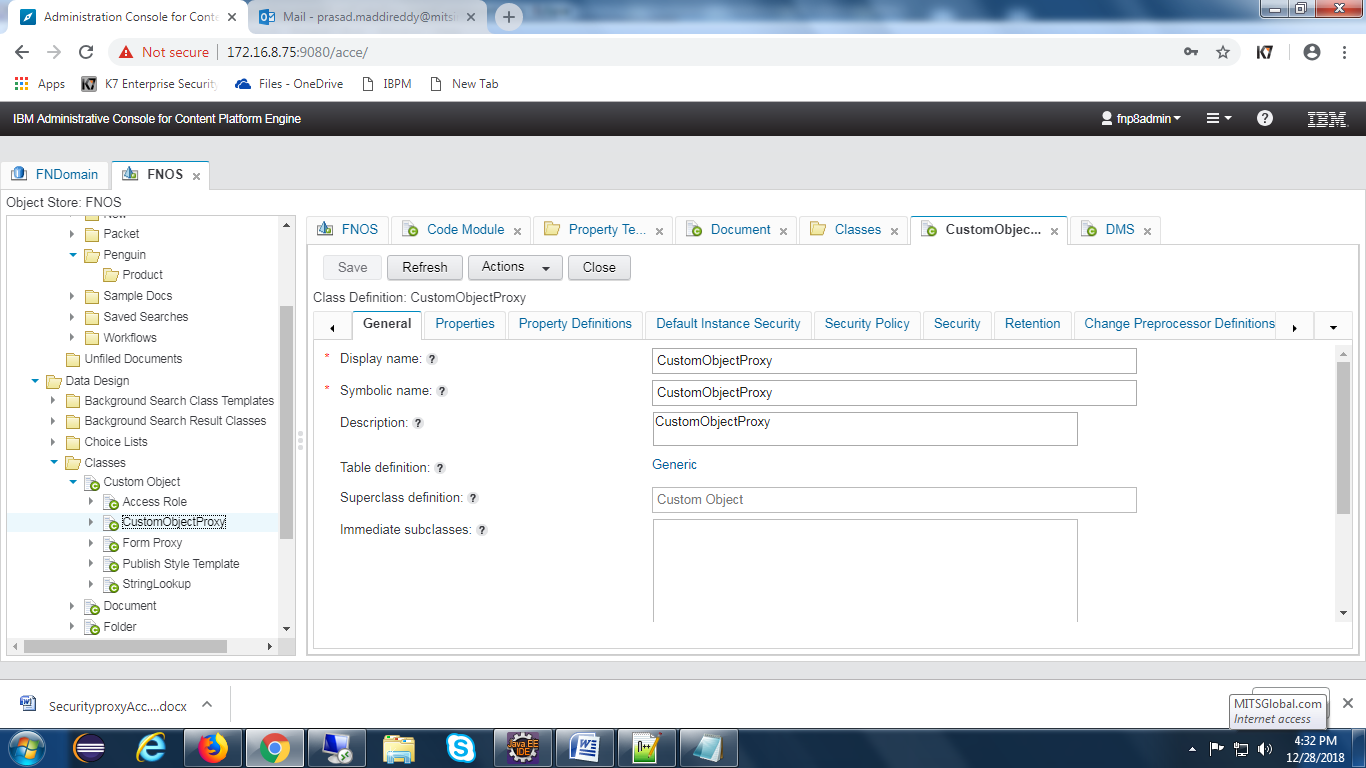
* Processing
* Archival



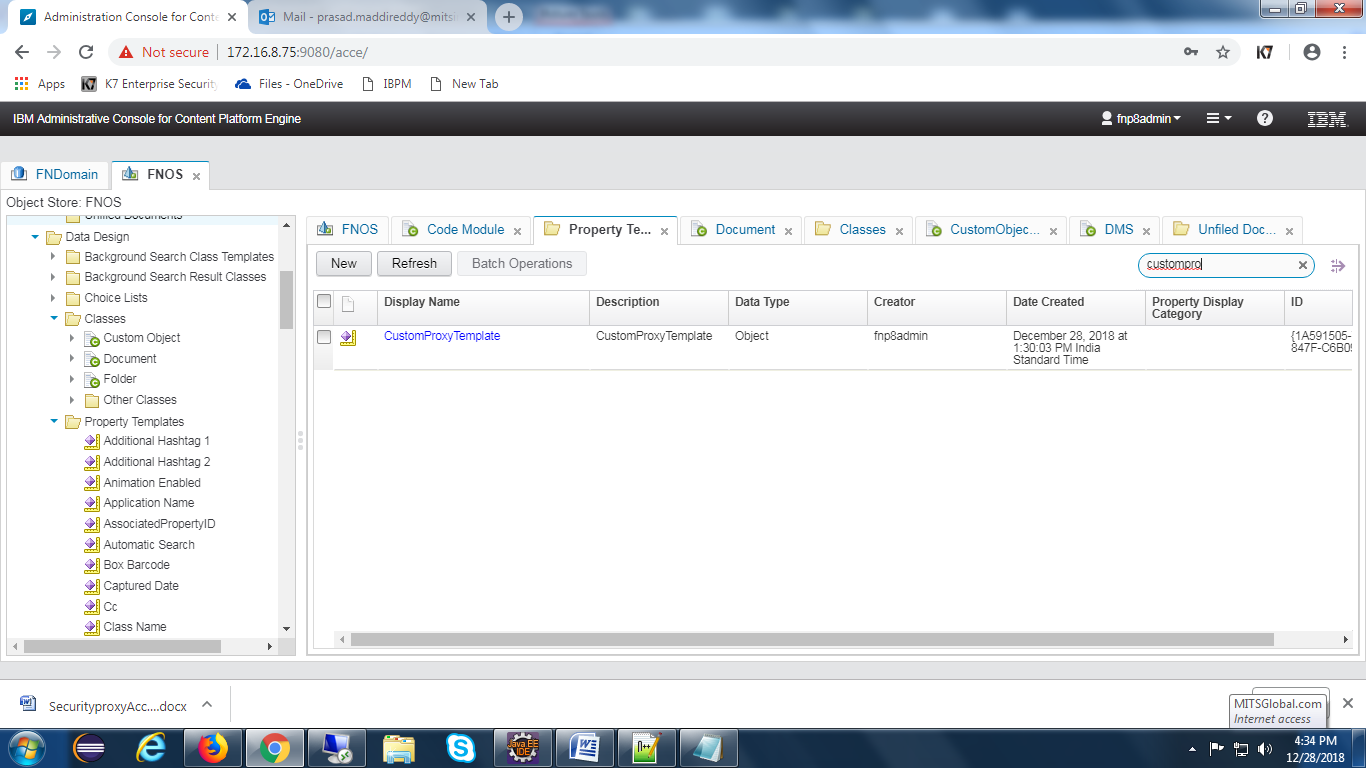


## Security Proxy:

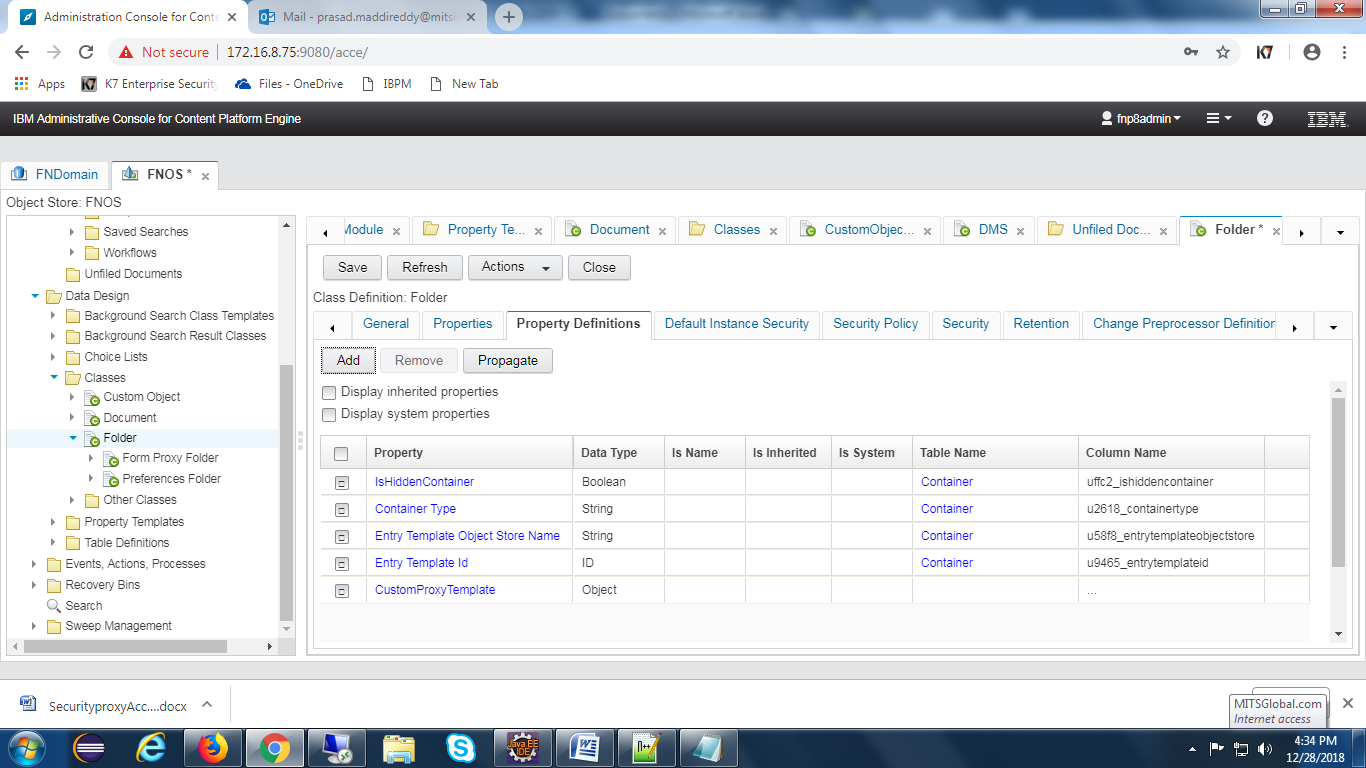
Create Proxy Class in Customobject



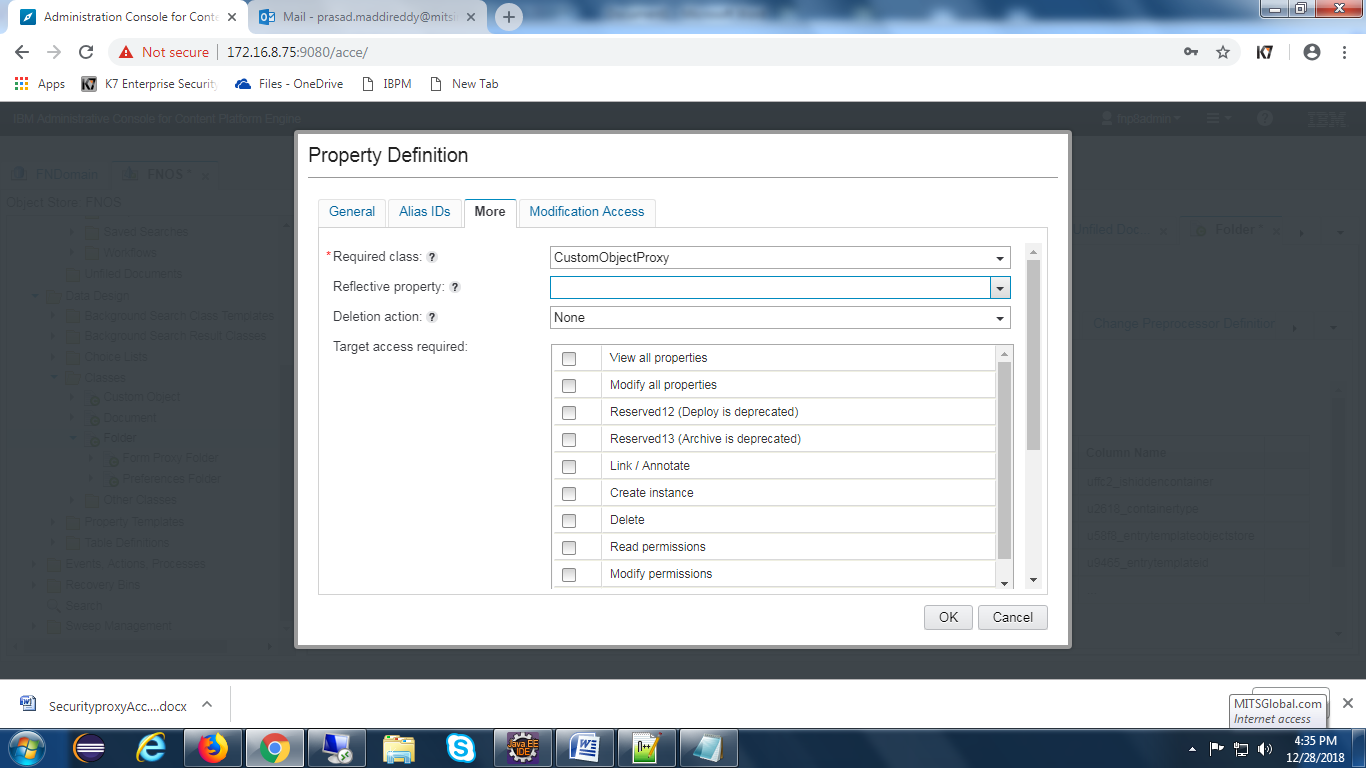
Create proxy property

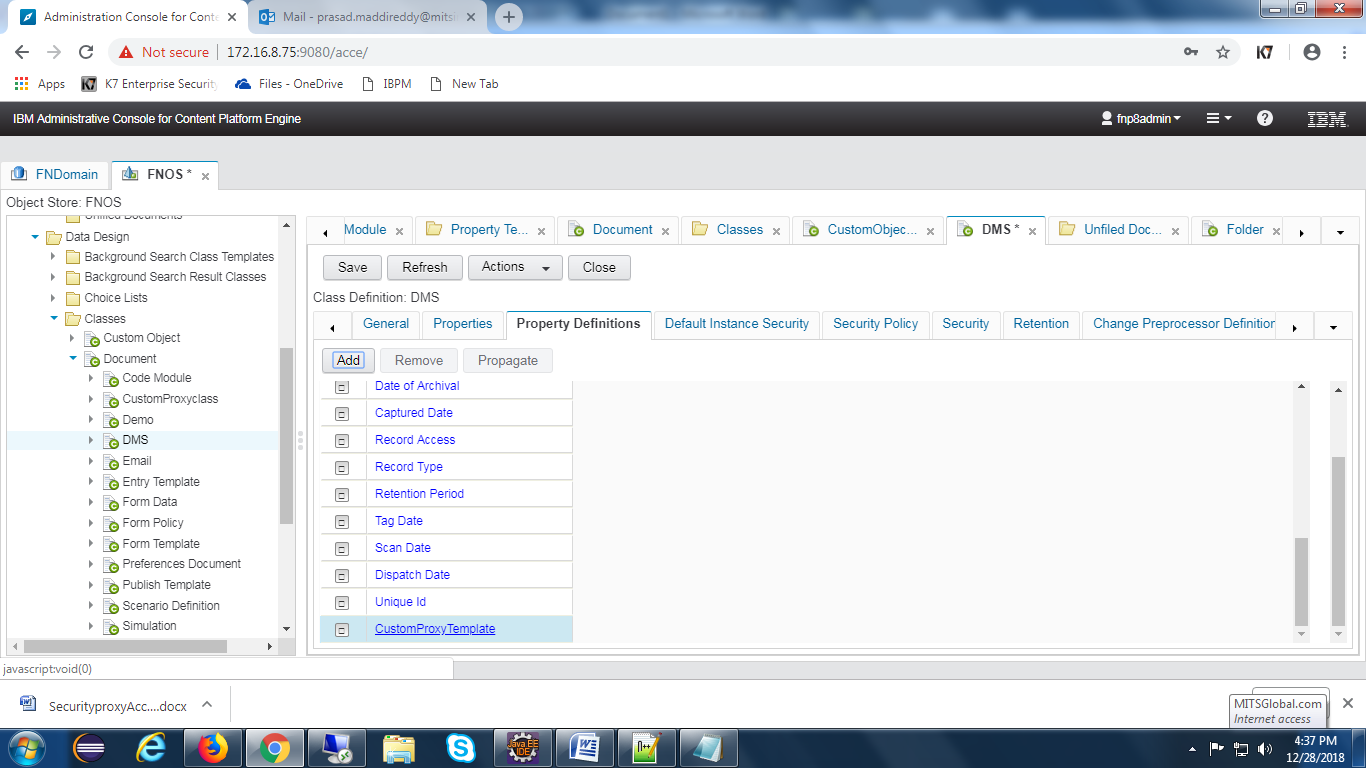


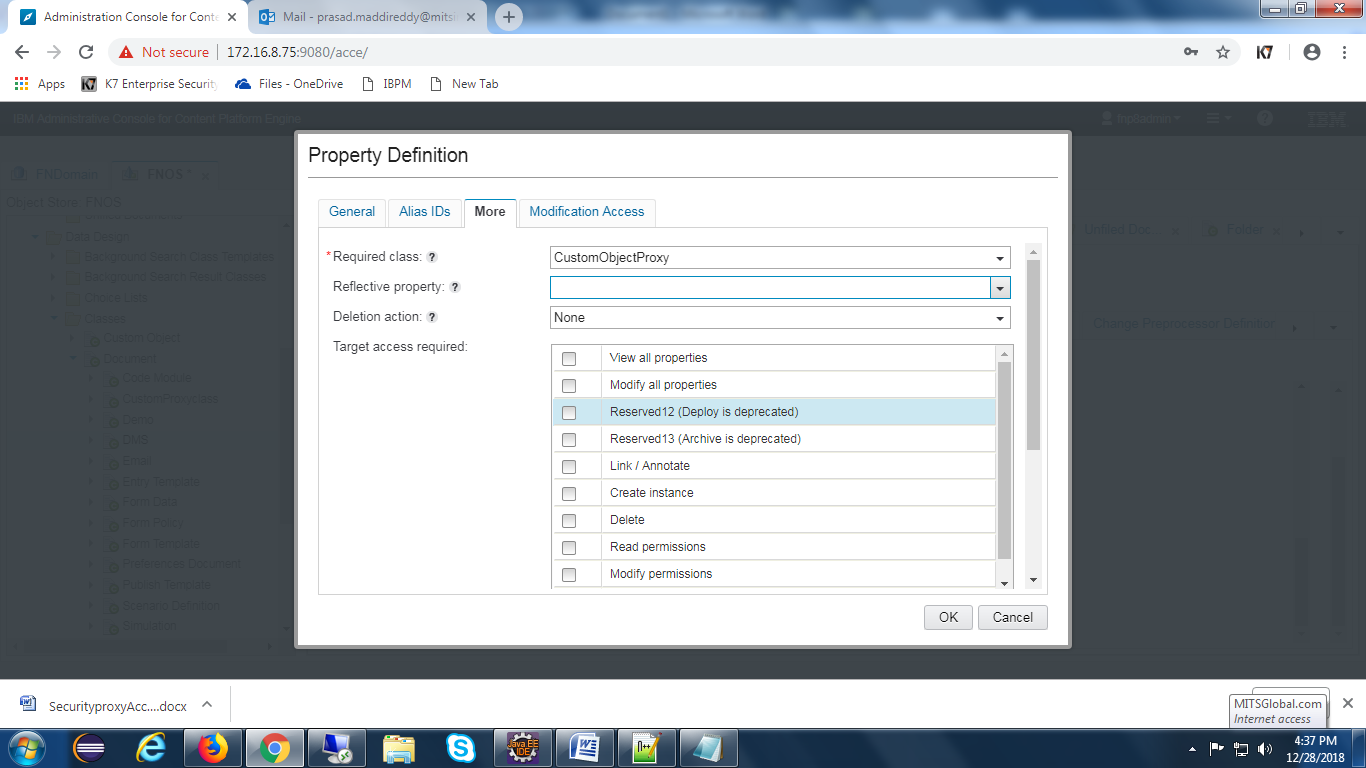
Add property to folder class



Set class type as CustomObjectProxy to the property







Create two proxies for document and folders

