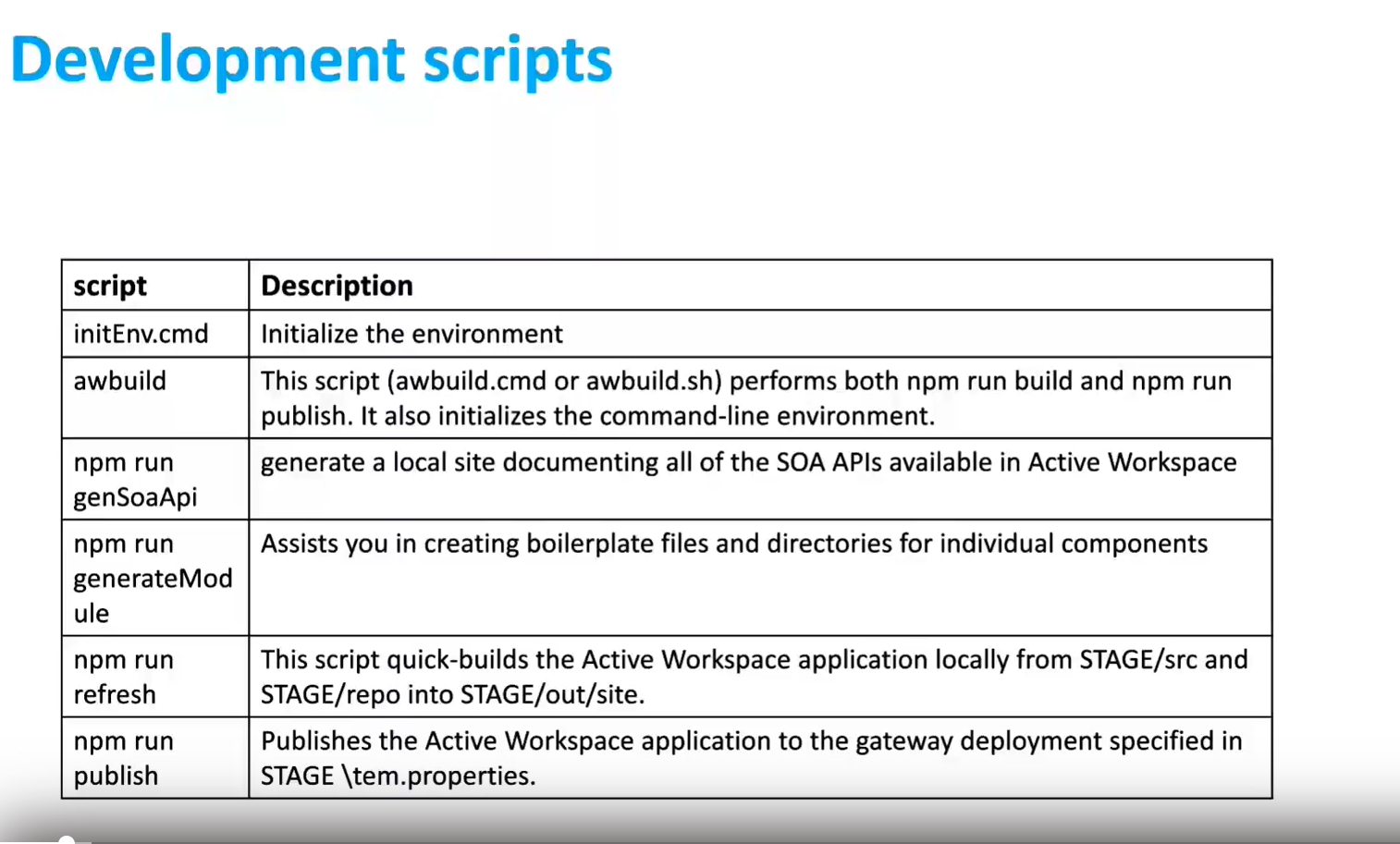
**Active Workspace Architecture**



**AWC MAIN COMMAND TO RUN AFTER MAKING CHANGES**



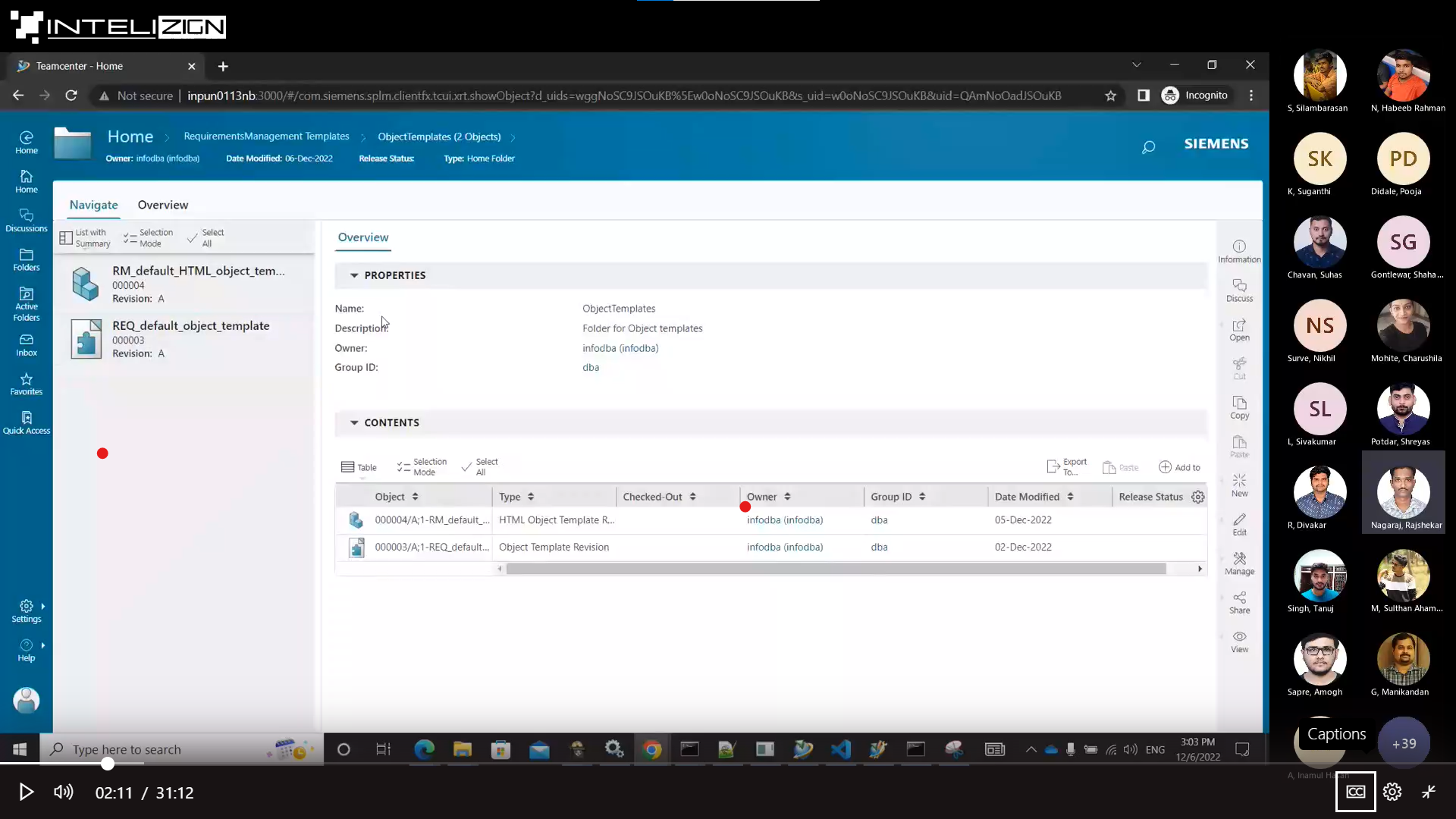
------------------------------------------------------------------------------------------------------------------------------------

DATE:05-04-23

**TO PIN AN OBJECT TO HOME PAGE**

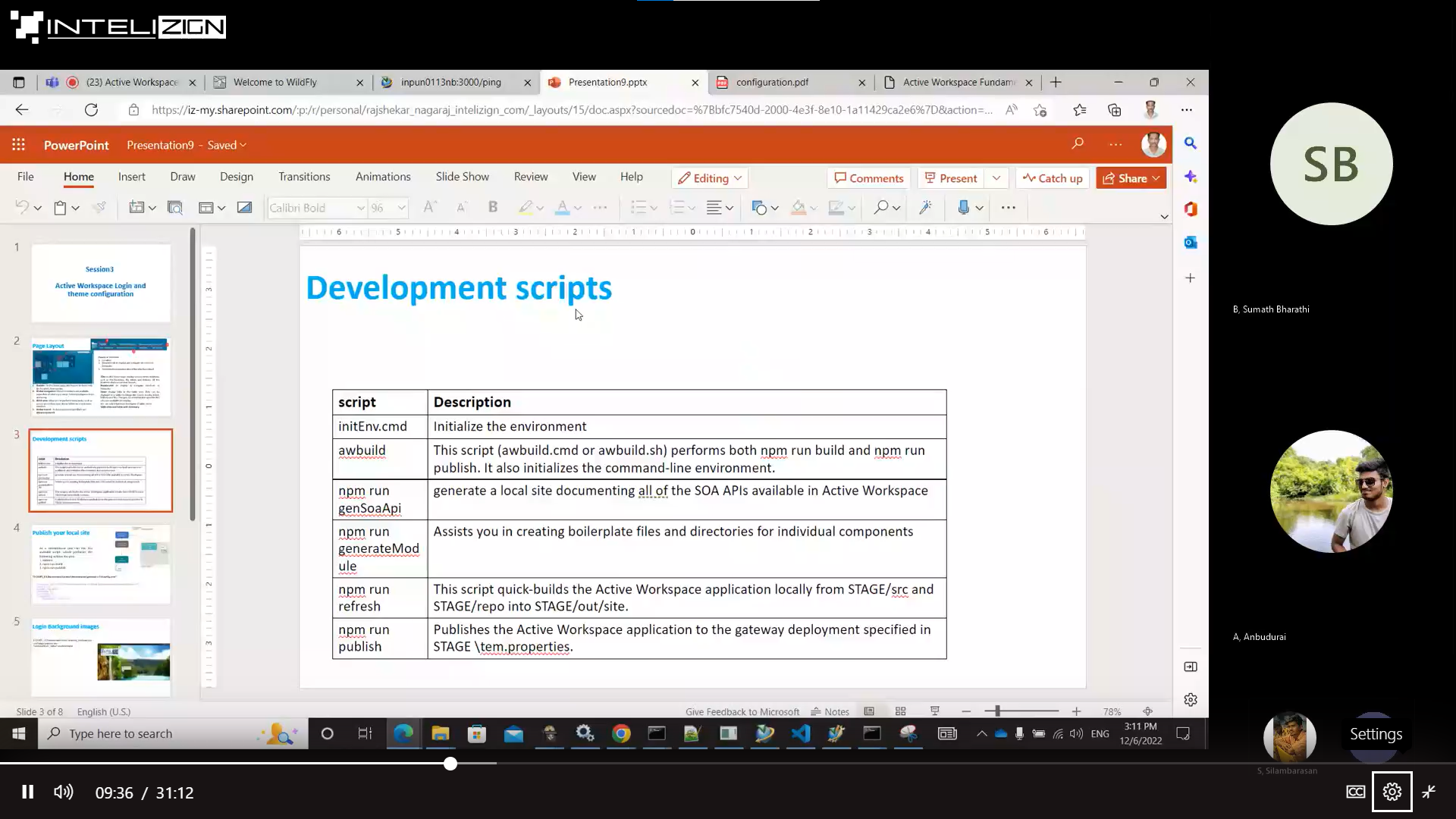
* Open the object in component view
* Go to manage
* At right side I.e., primary toolbar**->** manage->Select Pin to home

-------------------------------------------------------------------------------------------------------------------------------------

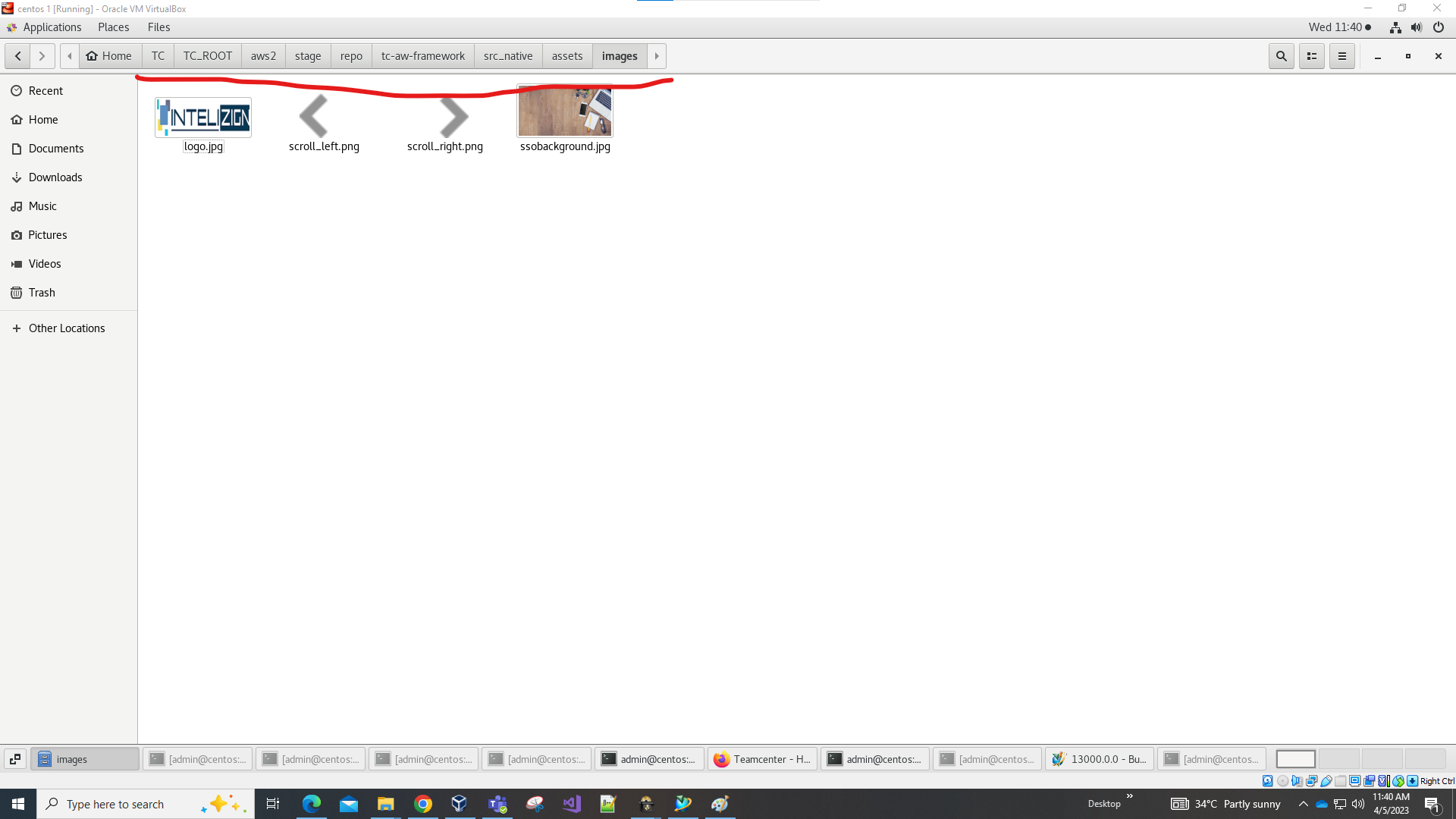


1. Global navigator
2. Header
3. Primary toolbar
4. Bread crumb

**DEVELOPMENT SCRIPTS**



**CHANGING BACKGROUND IMAGE AND SETTING LOGO**



* Copy the logo and background pics with the path specified in above image.
* Logo pixels should be (width=360, Height=152) and it should be in **jpg** format.
* Background image pixels should be (width=1920, Height=1080) and it should be in **jpg** format.

},

IN THE GIVEN PATH : /home/admin/TC/TC\_ROOT/aws2/stage

**MAKE CHANGES TO THE BELOW CODE**

"solutionDef": {

"solutionId": "TcAW",

"solutionName": "MY Active Workspace",

"solutionVersion": "5.2",

"browserTitle": "Teamcenter AWC",

"brandName": "TEAMCENTER",

"companyName": "Intelizign Lifecycle Services Pvt Ltd",

"copyrightText": "Copyright © 2023 Intelizign",

"solutionWar": "awc",

"workspaces": [

**RUN THE BELOW COMMAND**

(I used VS code in linux, I directly opened the stage folder and in the below console I run the command )

[admin@centos stage]$ ./awbuild.sh

Then you can reload your AWC, there you can see the changes.

**CHANGING THEME IN AWC**

IN THE GIVEN PATH : /home/admin/TC/TC\_ROOT/aws2/stage

**Run below command**

[admin@centos stage]$ npm run generateModule

Enter type to generate: theme

Theme name: LightGreen

Color: #808000

[admin@centos stage]$ ./awbuild.sh

**CREATING COMMAND**

First run the command in

[admin@centos stage]$ npm run generateModule

Enter type to generate: module

Module name: AWCTraining

[admin@centos stage]$ npm run generateModule

Enter type to generate: command

Command name: newcommand

Icon name: cmdUpArrow

Placement name: aw\_rightWall

Placements will be 3 types:

* aw\_globalToolbar
* aw\_rightWall
* aw\_contextMenu2

[admin@centos stage]$ ./awbuild.sh

**Refresh AWC Window OR Login Again.**

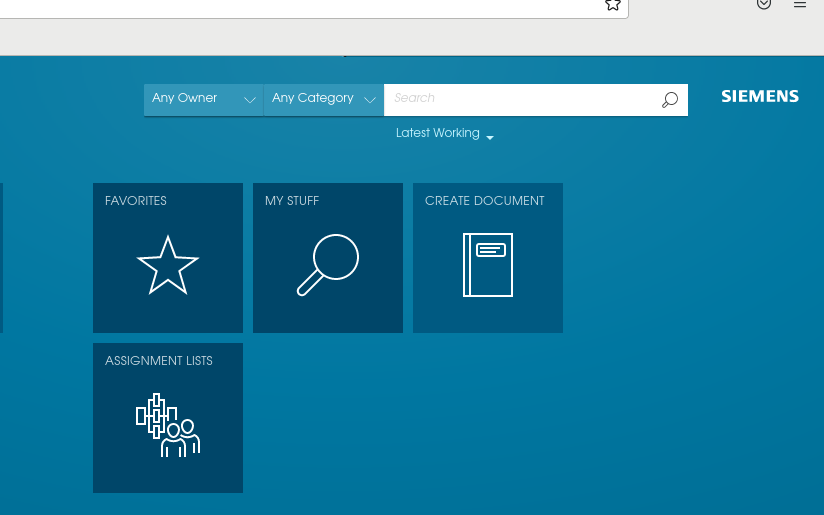
Check the Icon of your new command in the placement you mentioned.

**CREATING TILE IN HOME FOLDER**

* First in General search , Search for **tile Template** type**.**
* Copy “Awp0HomeFolderCreateTemplate”
* Click on newStuff (where you want to create )
* Go to file –> New -> Other -> Tile ->Next
* In tile template paste the copied template.Give the necessary filelds “Action parameters” in Specific format.
* Click finish
* Copy that tile.
* After that goto General search , Search for **tile** **collection** type**.**
* Select the workspace you want and paste it.
* Now you can see that tile in AWC.

**HIDING/UNHIDING ADVANCED SEARCH OPTION**

* To hide the Advanced Search option in AWC we must set the preference “**AW\_Advanced\_Search\_Visibility**” to **false**.



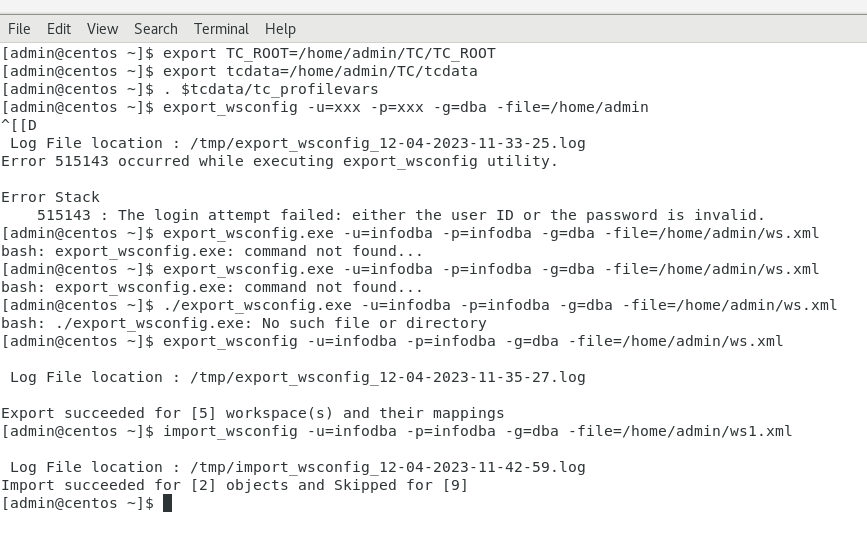
* To unhide the Advanced Search option in AWC we must set the preference “**AW\_Advanced\_Search\_Visibility**” to **true**.

**CREATING CUSTOM WORKSPACE**

* GOTO **/home/admin/TC/TC\_ROOT/aws2/stage**
* **Run the command “npm run generateModule”**
* **Enter type to generate: workspace**
* **ID: MyWorkspace**
* **Workspace name: NewWorkspace**
* **Type: Inclusive**
* **Default page: showHome**

**After that run the utility in below image**

* **Created one xml file(export)**
* **Make changes to the xml file which is exported I.e., add your workspace.**



* GOTO **/home/admin/TC/TC\_ROOT/aws2/stage**
* Run “[admin@centos stage]$ ./awbuild.sh”
* Check the changes in AWC by refreshing. Whether custom workspace is added or not.

**OVERLAY ICON**

* **First create a .json file in D:\SOFT\TEAMCENTER\TC\_ROOT\aws2\stage\src\AWCTraining(under your module)**.
* Then convert your image into .svg.
* After that paste that image here path= **D:\SOFT\TEAMCENTER\TC\_ROOT\aws2\stage\src\image**
* Then open .json file you created and paste the below code

[{

"type": {

"names": [

"A2ObjectRevision"

],

"prop": {

"names": [

"object\_desc"

],

"iconFileName": "A4ObjectOut",

"conditions": {

"object\_desc": {

"$ne": " "

}

}

}

}

}]

* Change name, icon file name, condition.
* Then run **awbuild.cmd**
* **TASK:Change the icon for any of the LOV property as well, icons should change for different LOV values**

**Write the code in .json file**

[

{

"type": {

"names": [

"A2ObjectRevision"

],

"prop": {

"names": [

"a2Color"

],

"iconFileName": "Yellow",

"conditions": {

"a2Color": {

"$eq": "Yellow"

}

}

}

}

},

{

"type": {

"names": [

"A2ObjectRevision"

],

"prop": {

"names": [

"a2Color"

],

"iconFileName": "Red",

"conditions": {

"a2Color": {

"$eq": "Red"

}

}

}

}

},

{

"type": {

"names": [

"A2ObjectRevision"

],

"prop": {

"names": [

"a2Color"

],

"iconFileName": "Blue",

"conditions": {

"a2Color": {

"$eq": "Blue"

}

}

}

}

},

**4.Populate different logo for this 2 conditions if object\_name is "Test" and object\_desc is "Testing"( object\_desc should take highest priority)**

[{

"type": {

"names": ["A2FruitRevision"],

"prop": {

"names": ["object\_name"],

"iconFileName": "Red",

"conditions": {

"object\_name": { "$eq": "Test" }

}

}

},

"priority": 50

},

{

"type": {

"names": ["A2FruitRevision"],

"prop": {

"names": ["object\_desc"],

"iconFileName": "Blue",

"conditions": {

"object\_desc": {"$eq": "Testing"}

}

}

},

"priority": 65

}]

**3.Change the logo if object\_name is "Test" and object\_desc is "Testing"**

[

{

"type": {

"names": [

"A2FruitRevision"

],

"prop": {

"names": [

"object\_name",

"object\_desc"

],

"iconFileName": "Red",

"conditions": {

"object\_name": {

"$eq": "Test"

},

"object\_desc": {

"$eq": "Testing"

}

}

}

}

}

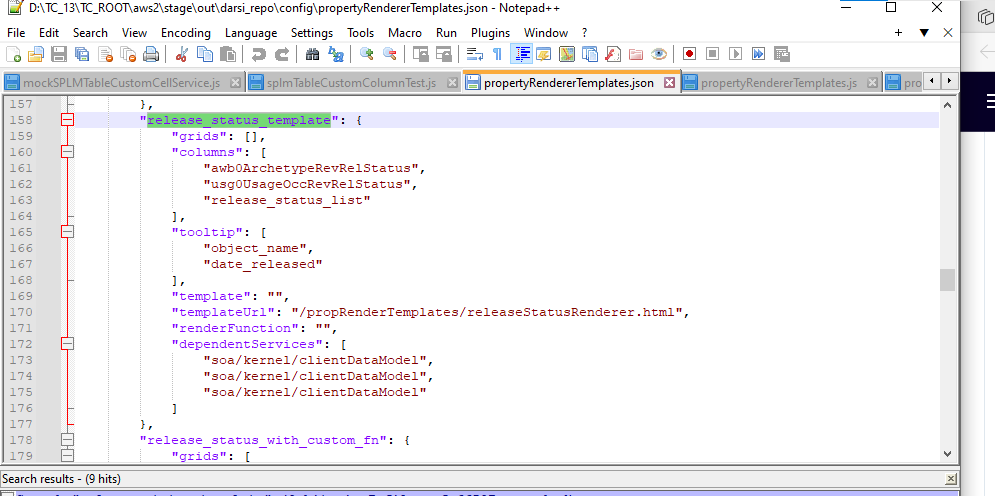
]

**CREATING CUSTOM SOA SERVICE**

* **npm run convertTemplates D:\TC\_13\TC\_DATA\model**
* In this path D:\TC\_13\TC\_ROOT\aws2\stage\out\soa\json
* Copy a2hello\_template to D:\TC\_13\TC\_ROOT\aws2\stage\src\soa\json this path

**DISPLAYING STATUS NAME INSTEAD OF ICON**

* Remove the code as shown in the figure

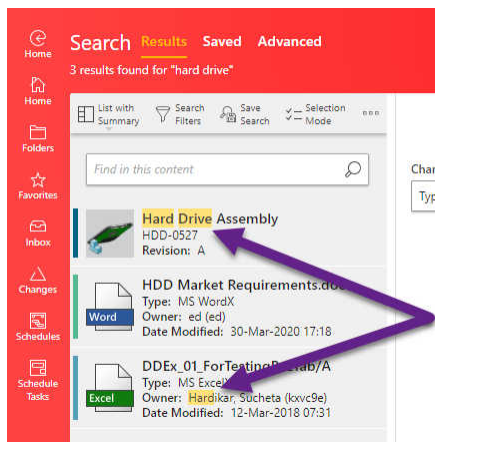


* From below paths



**CHANGING HIGHLIGHT COLOR WHILE SEARCHING**

* **Initially the highlighted color is like below**

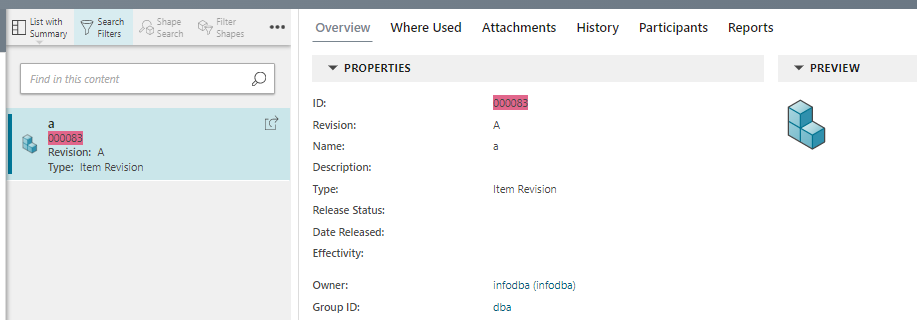


* **First open .scss file of your custom theme**
* **Add the below highlighted code to .scss file**

@import 'ui-theme-settings'; $aw\_highlighted\_background\_color:$AW\_SiemensAccentRed5;

@import 'ui-theme';

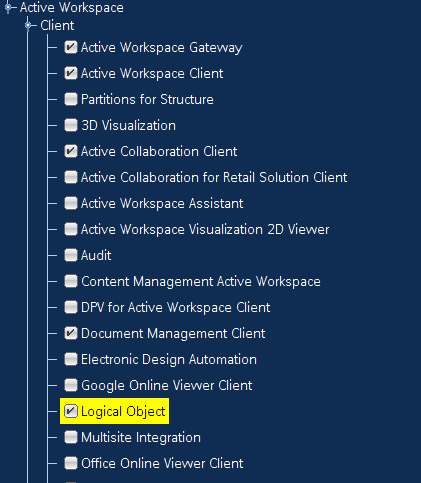
* Run awbuild.cmd
* The Result is

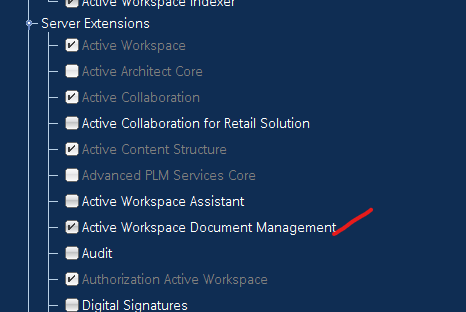


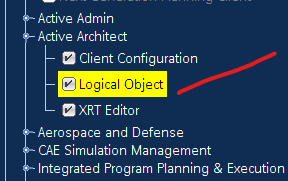
h

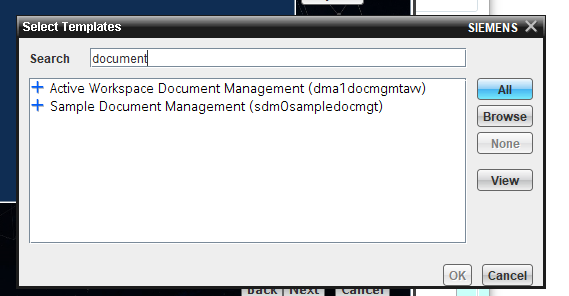
**LOGICAL OBJECTS:**

* A logical object is a runtime object designed to consolidate properties. If you have related business objects in Teamcenter and they contain useful properties, you might want to see them all in a single place.

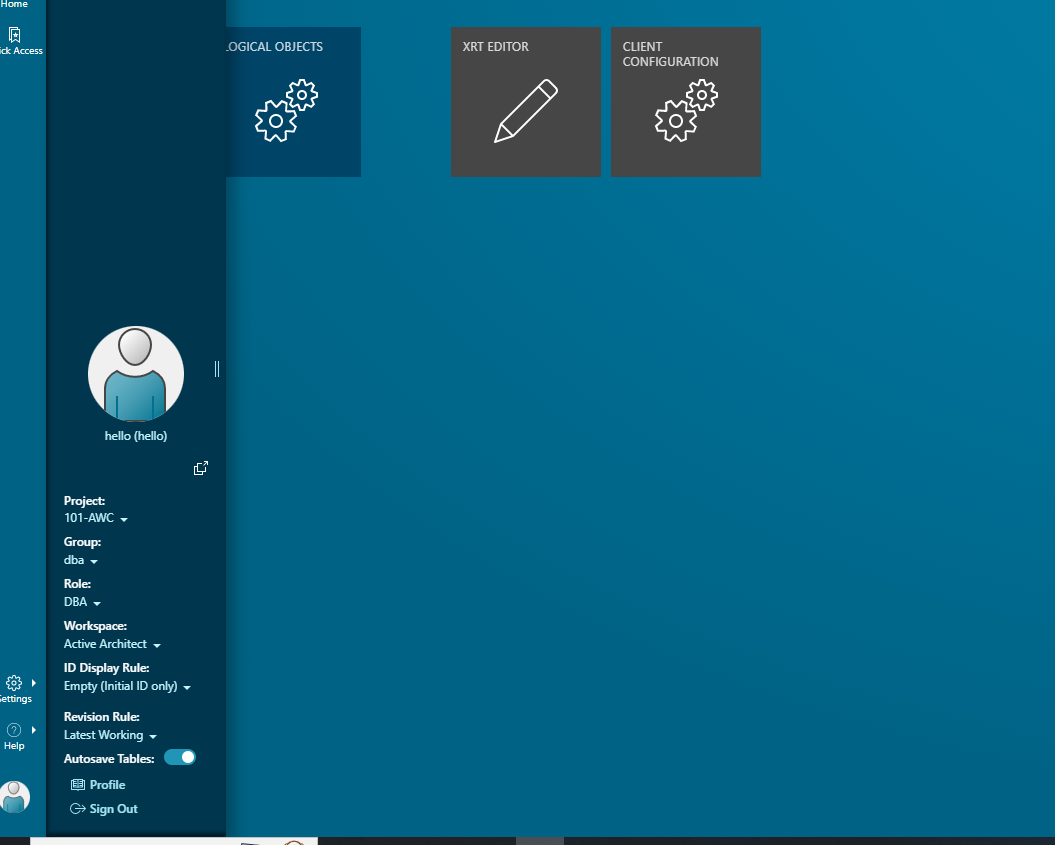


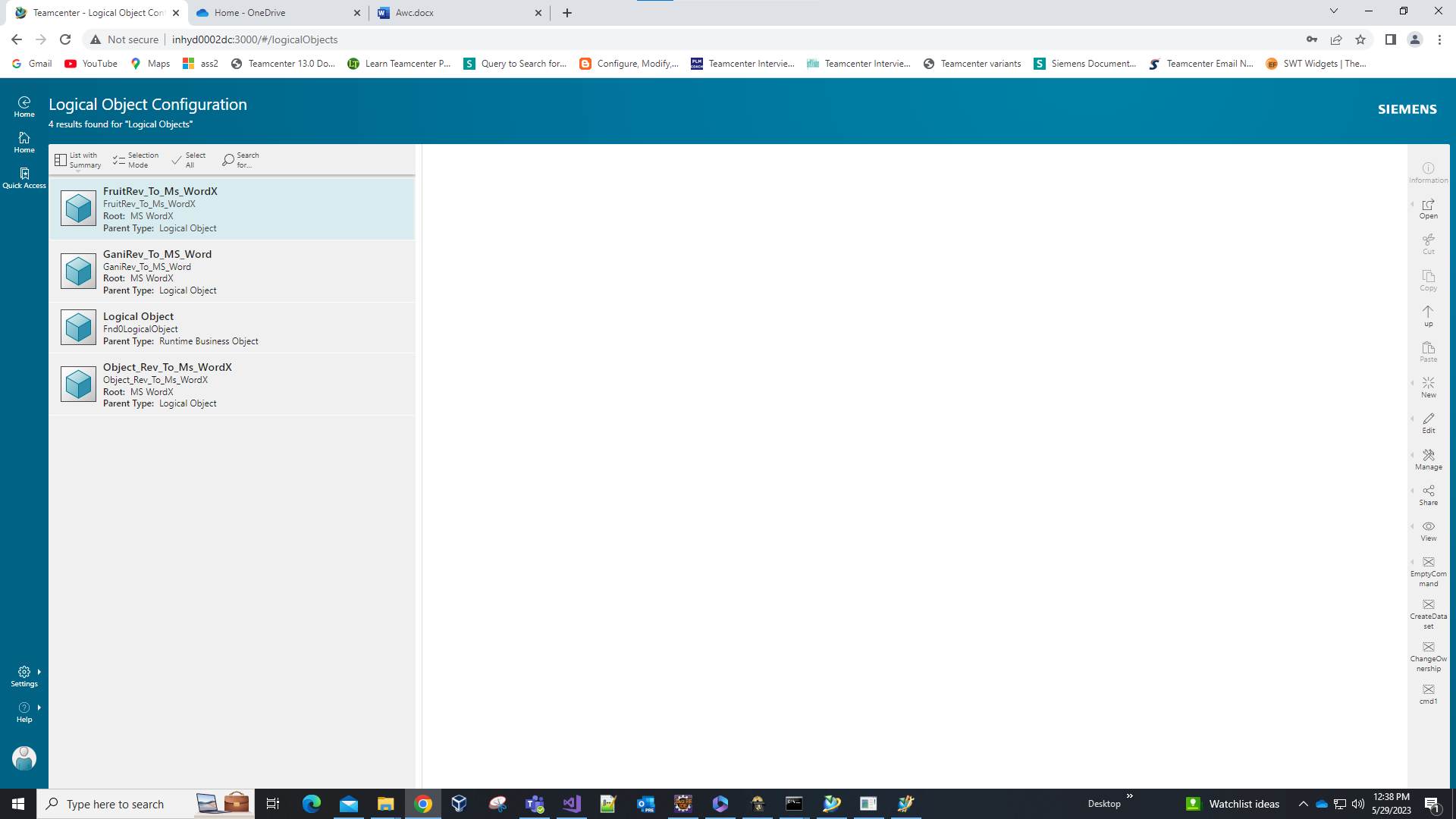


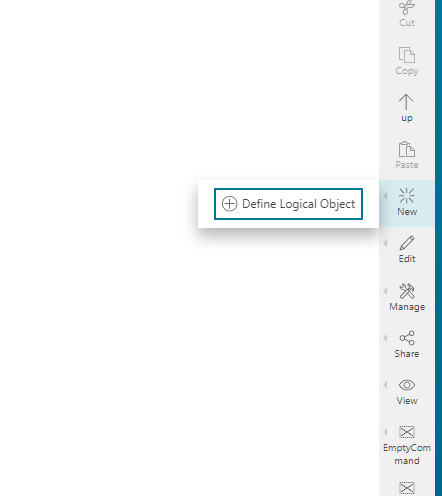


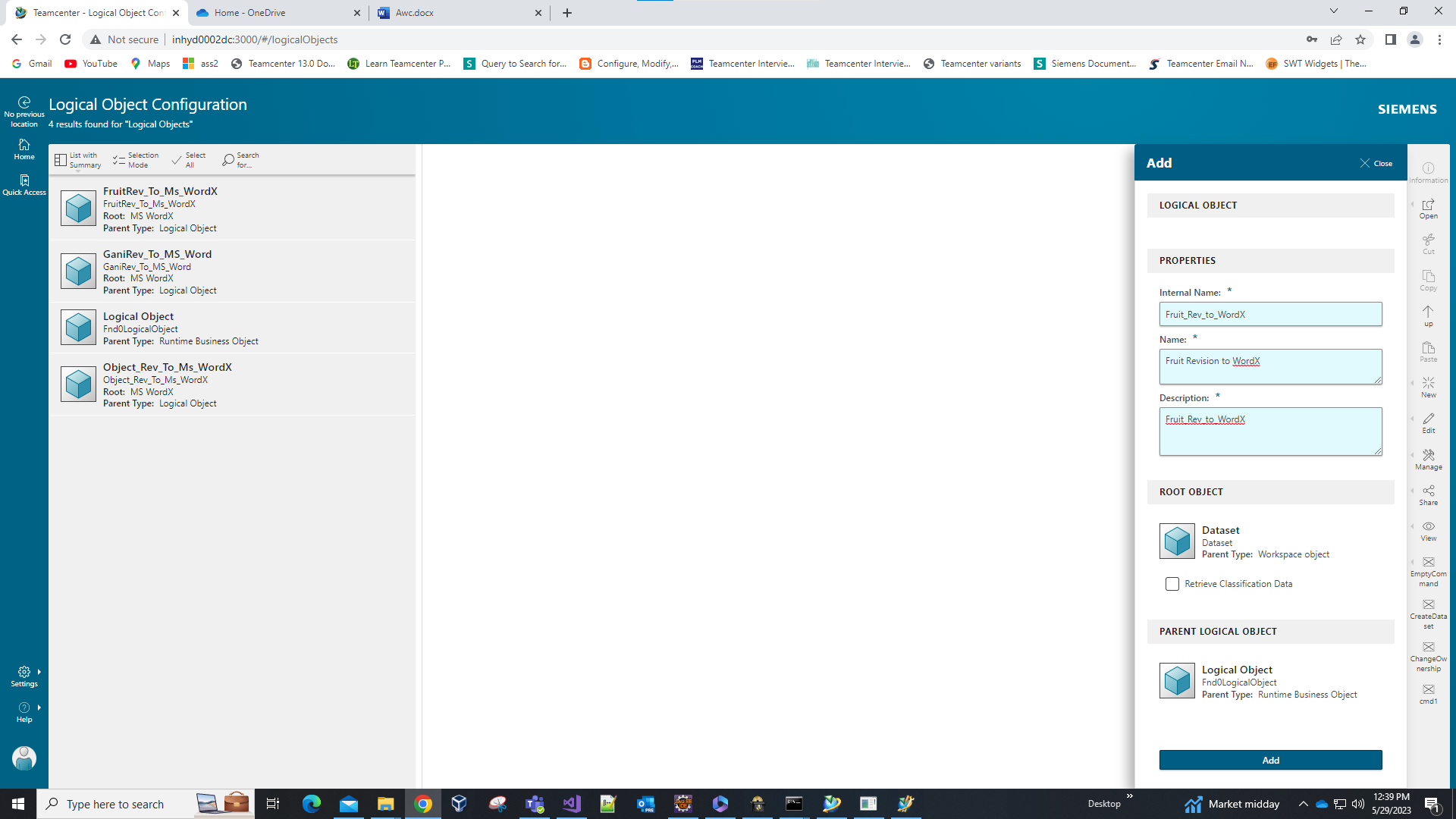


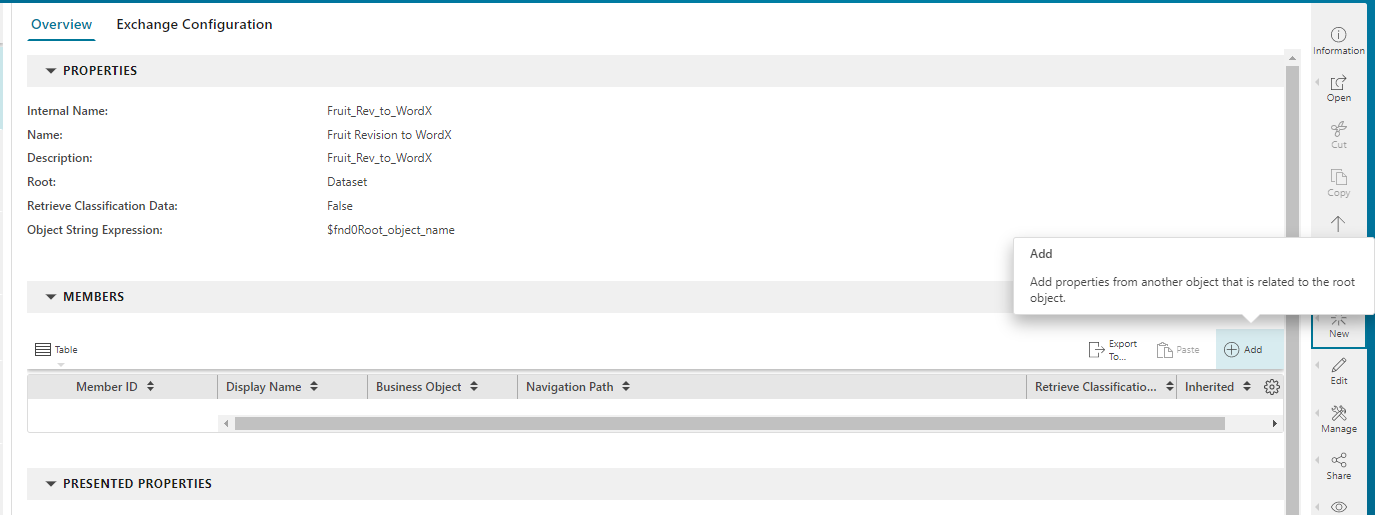
1. Login to AWC with admin privilages
2. Goto “**Active Architect”** Workspace.
3. Click on Logical objects tile
4. Select new,create new logical object



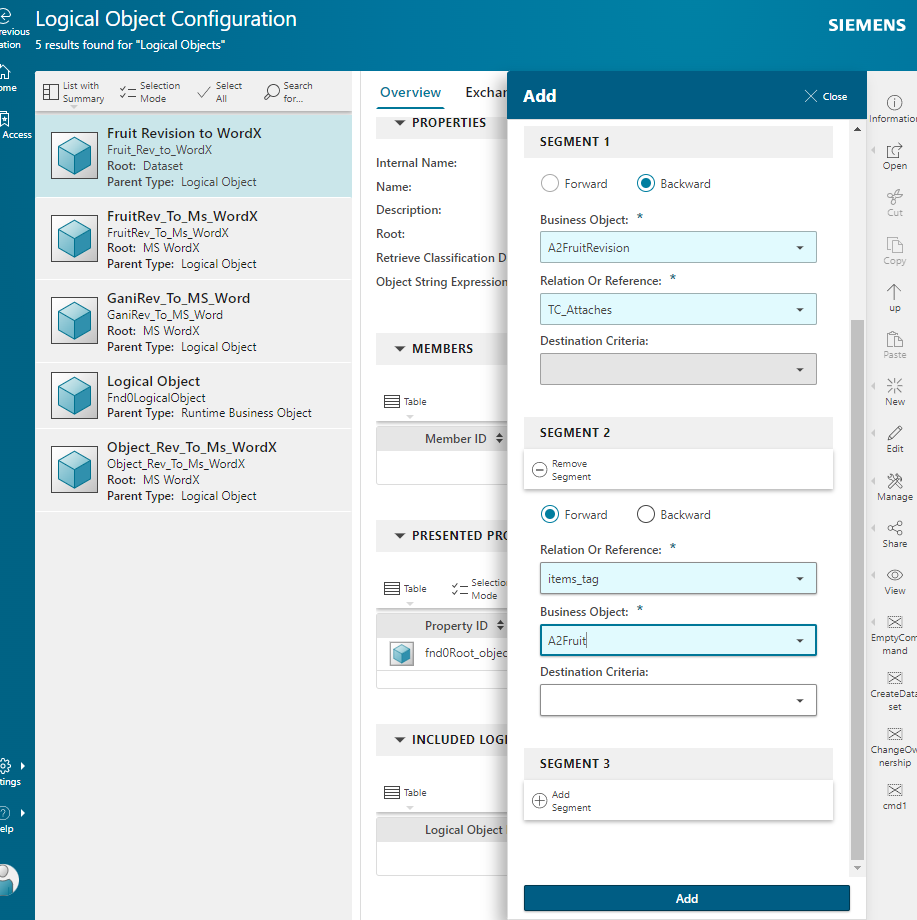




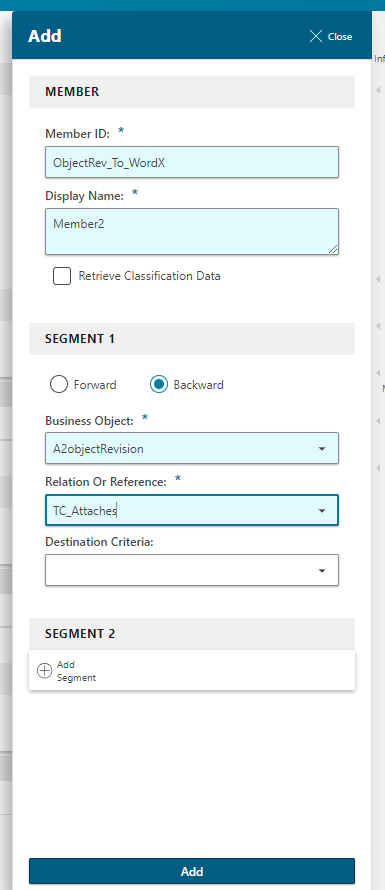


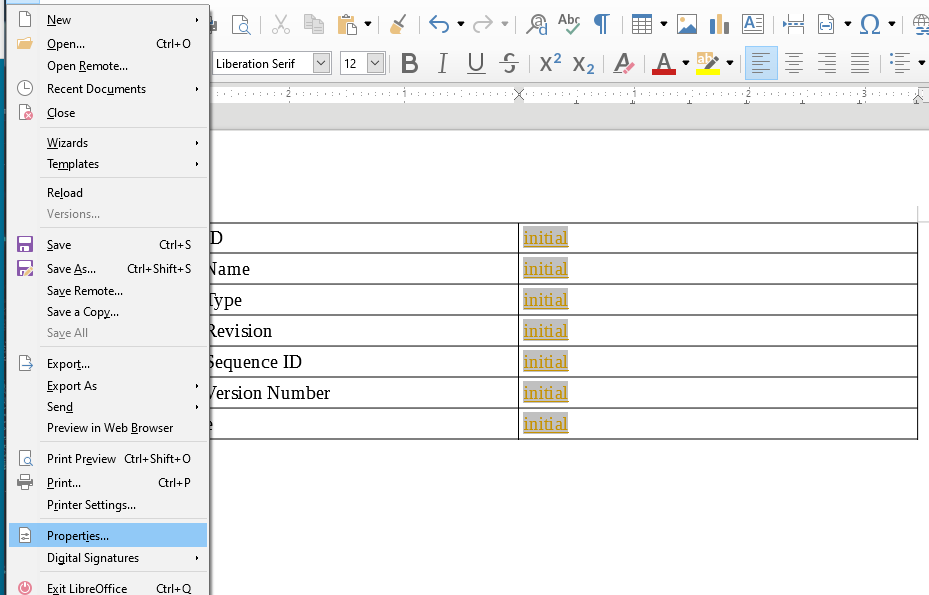


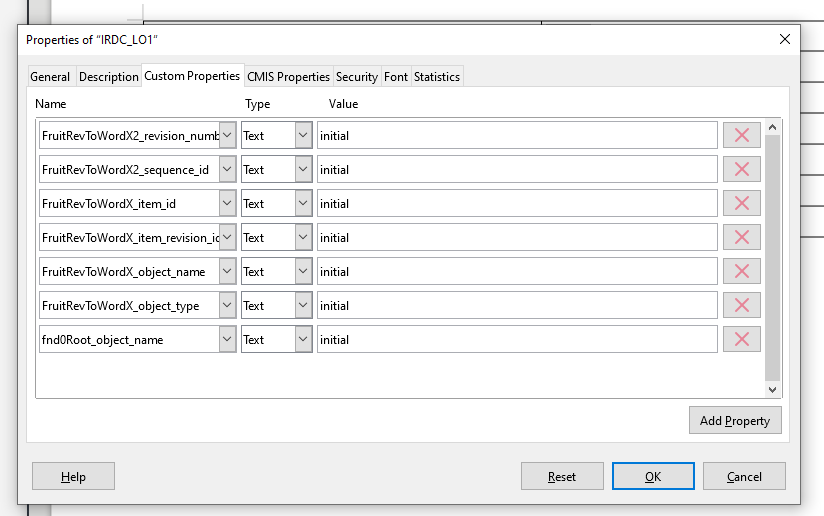
For item properties



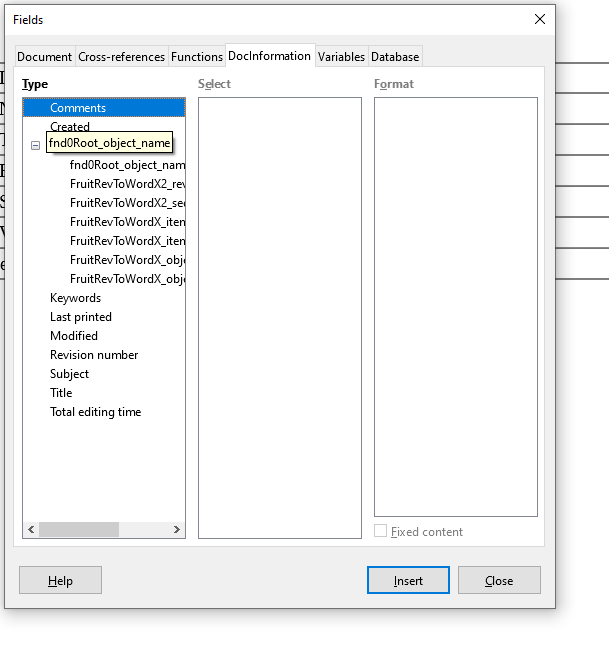
For item revision properties

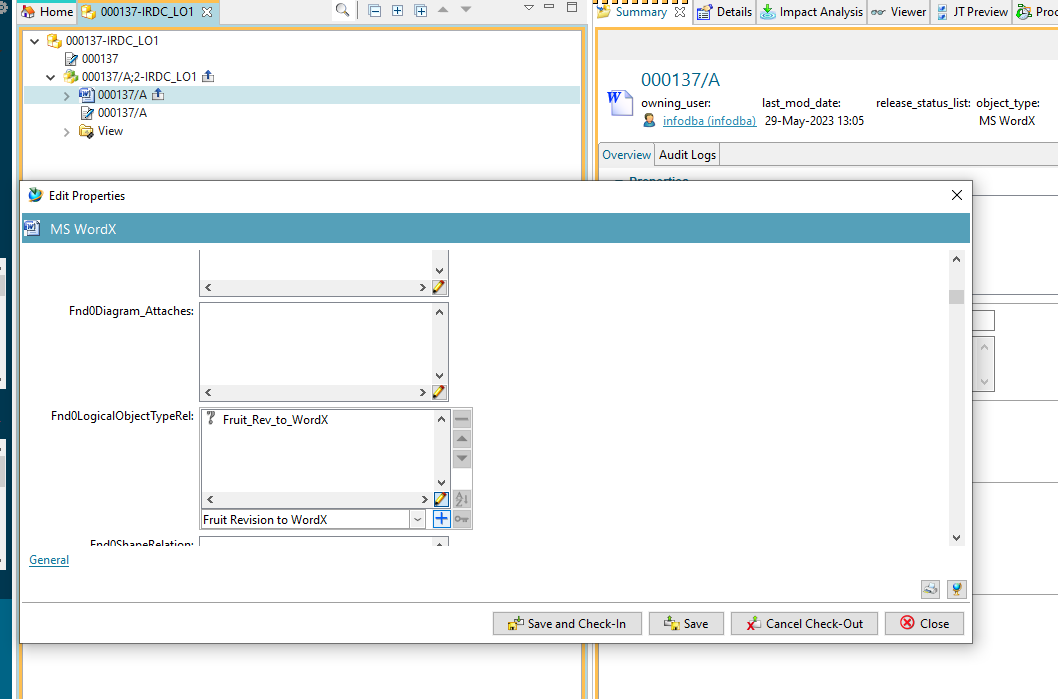


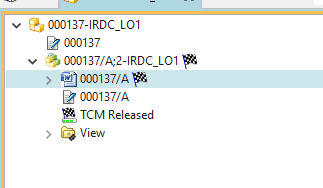


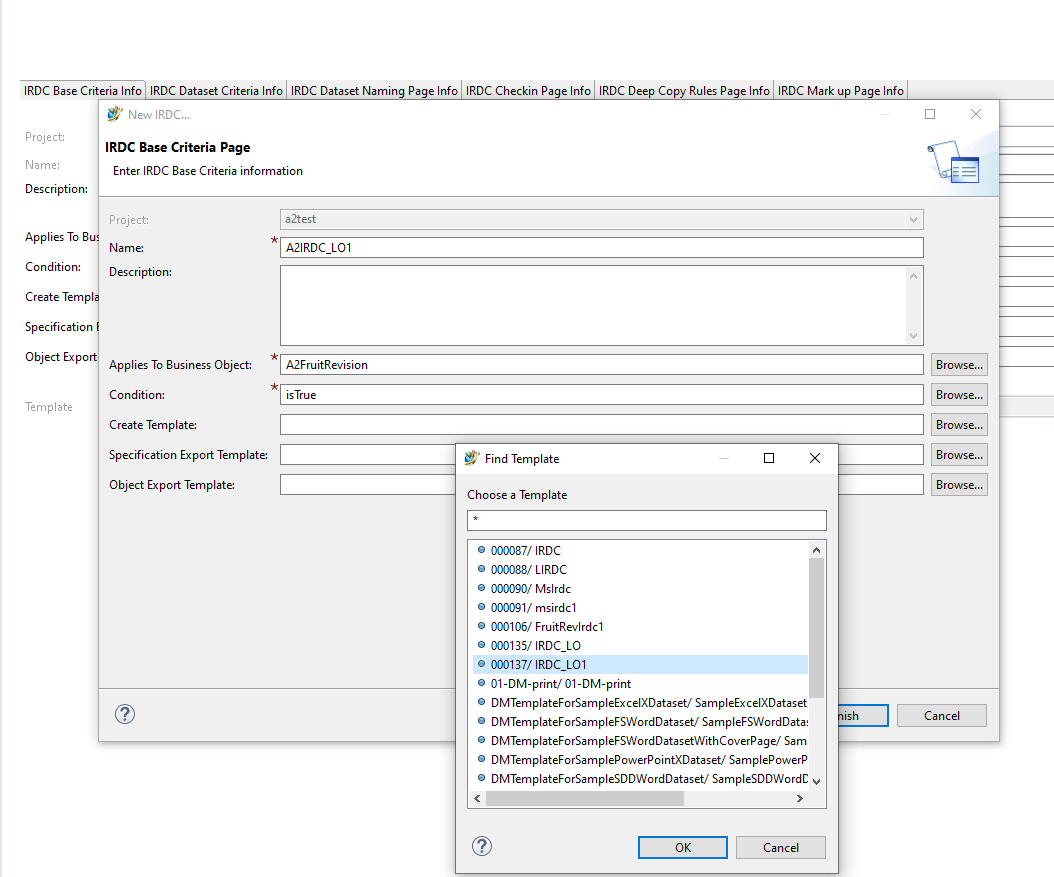


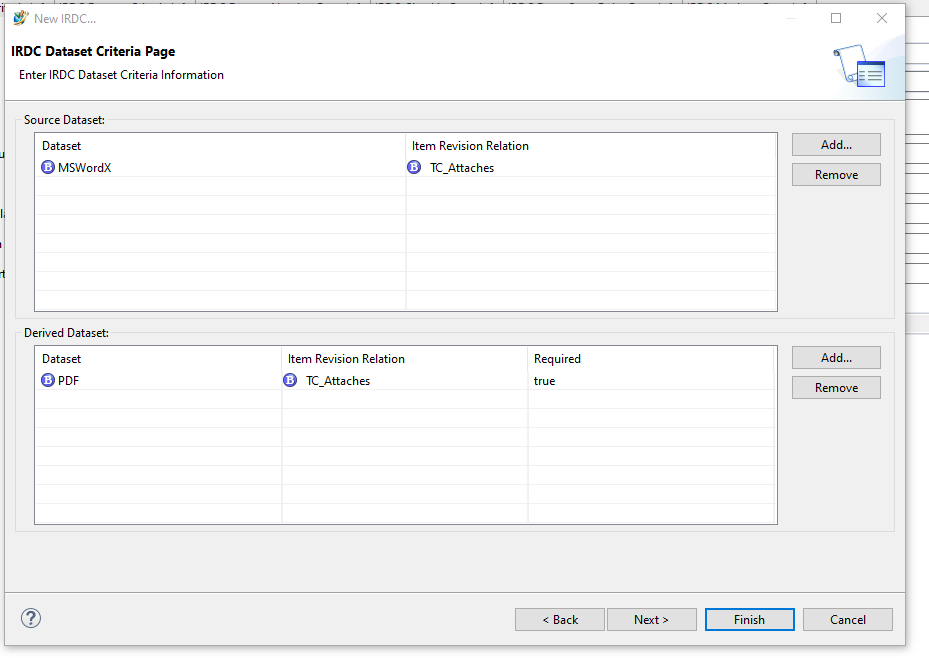


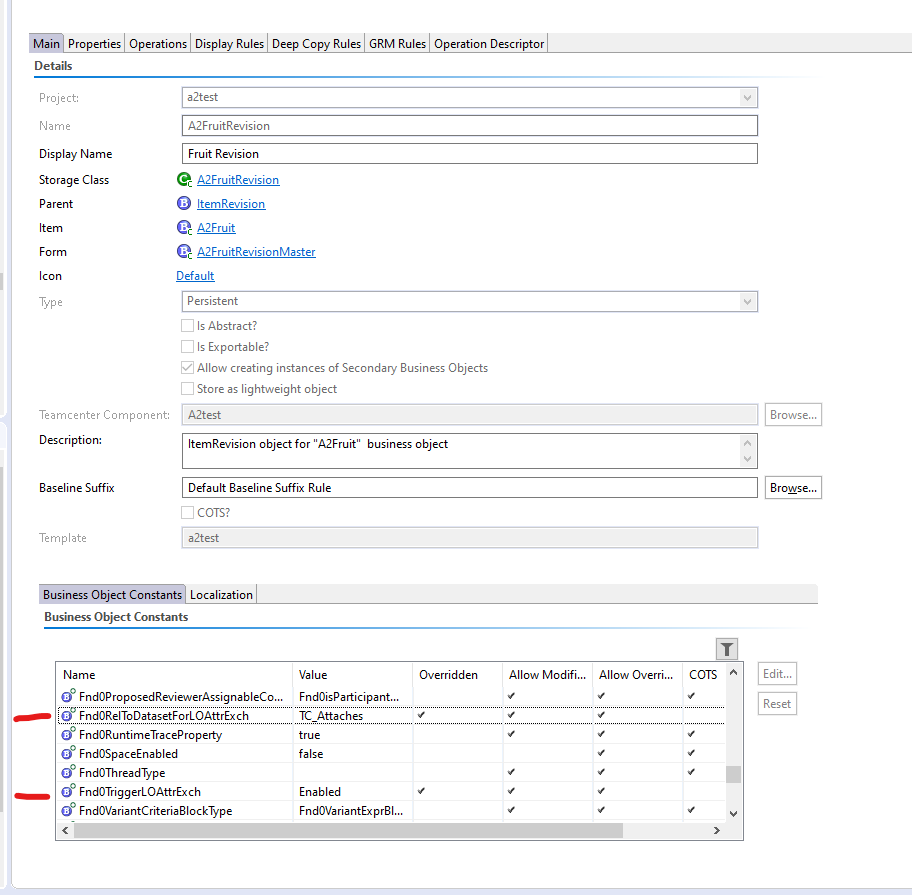




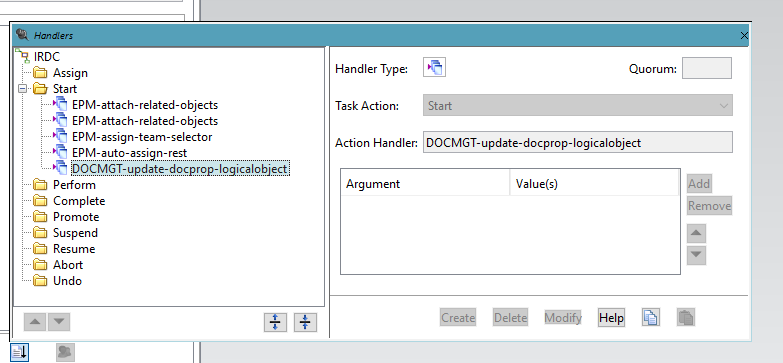








To update values into doc after making changes we need to run the workflow by putting the Highlighted Handler in the below image



[GTAC: Siemens PLM Software](https://solutions.industrysoftware.automation.siemens.com/view.php?sort=desc&q=IRDC&file_type=html&i=002-8513177&k=9&o=0)

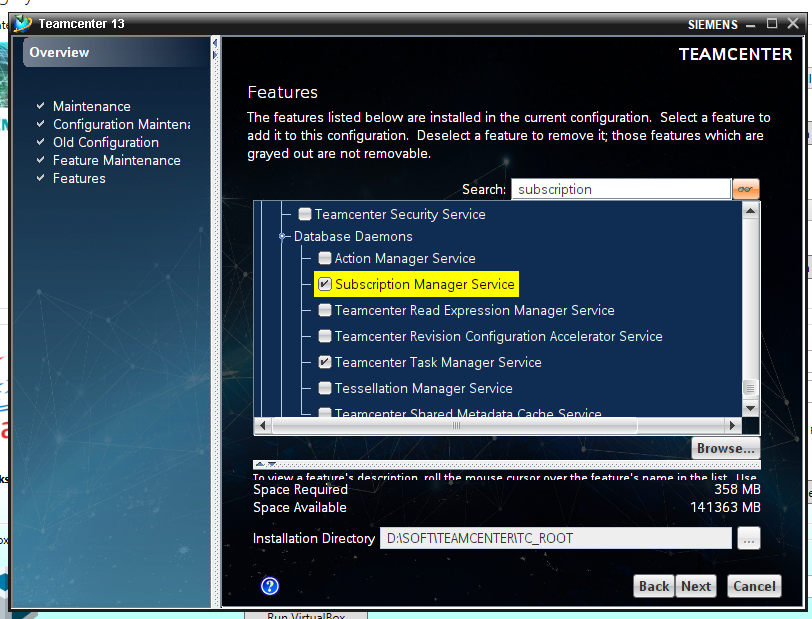
https://solutions.industrysoftware.automation.siemens.com/view.php?sort=desc&q=IRDC&file\_type=html&i=002-8513177&k=9&o=0

The above link is to create IRDC for generating derived datatype.

Here we have to create custom dispatcher service.

**SUBSCRIPTION MANAGER**

* **Run TEM**
  + Select Active Workspace → Client→ Subscription
  + Select Active Workspace → Server Extensions → Subscription
  + Add below feature also



* After deployment once run “**awbuild.cmd”.**
* **Set environment variable** 
  + Variable name = TC\_MESSAGING\_MUX\_URL
  + Variable path = [http://INHYD0002DC:8087](http://INHYD0002DC:8087/)

**-> 8087 IS MUX PORT**

* On Microsoft Windows machines, add the following line to the **tc\_profilevars.bat** file:
  + set TC\_MESSAGING\_MUX\_URL=http://blserver1:8087
* Open preferences “**SCM\_notification\_mode”** and edit value to **3.**

**-->>** In Teamcenter add the module **subscription adminstration**

* Add object type to get notification and properties

**-->** Check the service is running: **Teamcenter Subscription Manager Service**

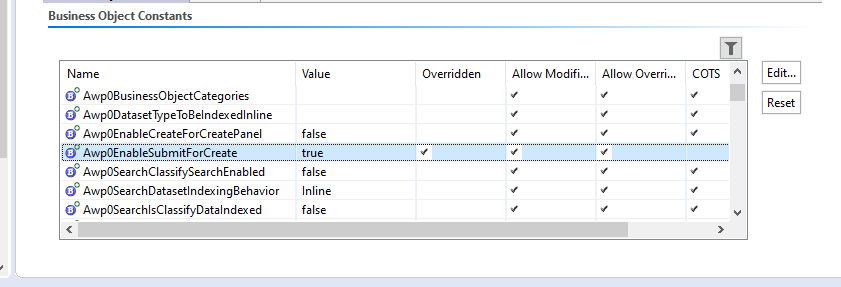
Now go to AWC and

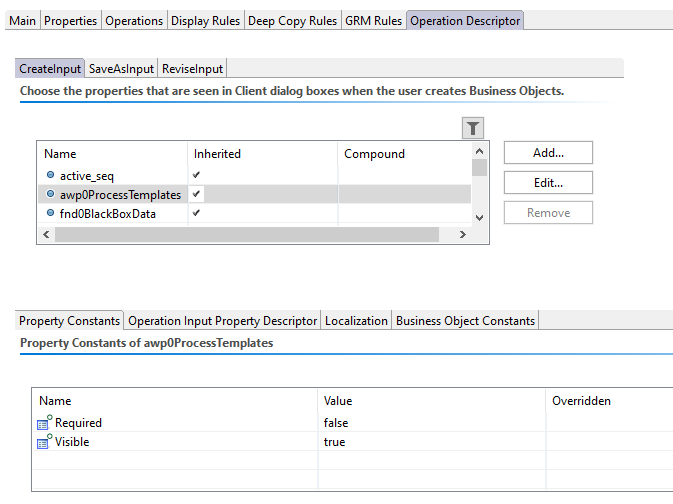
* Select subscription tile
* Click on **share** command -> click on **follow type** and add the subscription events

**Enabling the option to submit the normal item to the workflow process while creation in Active Workspace**

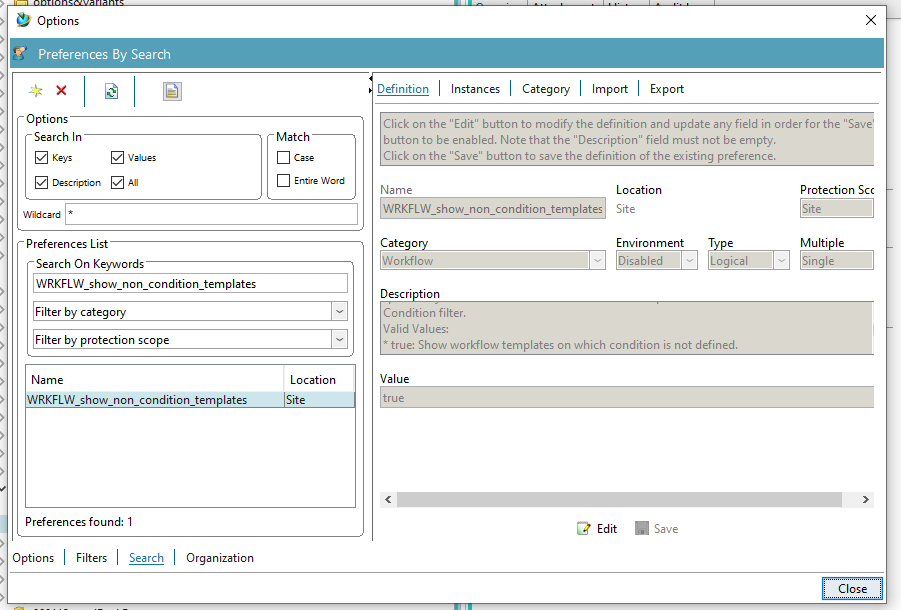
**Configuration in BMIDE**

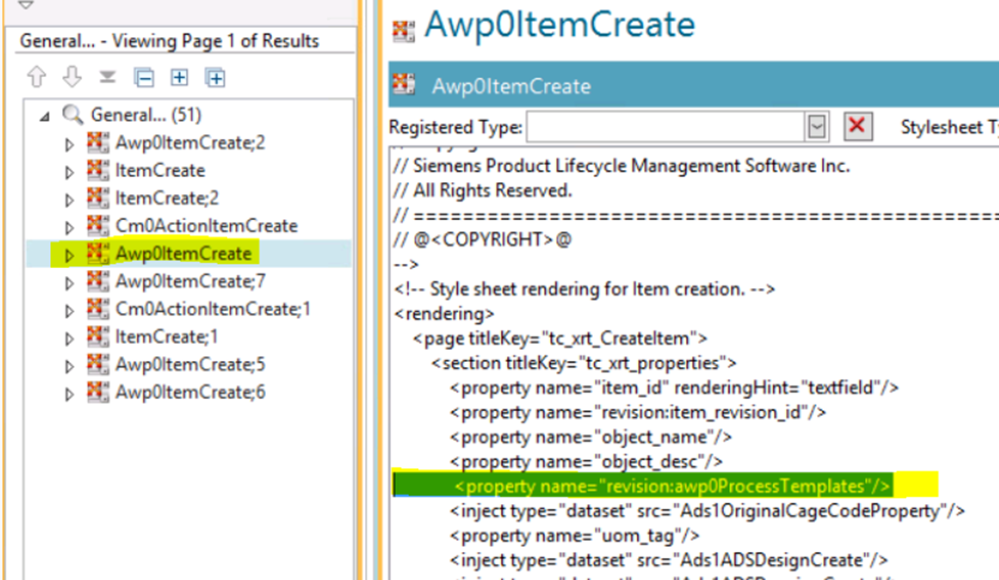
1. Open your business Object, let's say A4CustomItem.
2. In Main Tab, go to Business Objects Constant and set value for Awp0EnableSubmitForCreate to true.
3. Similar for Custom Item Revision



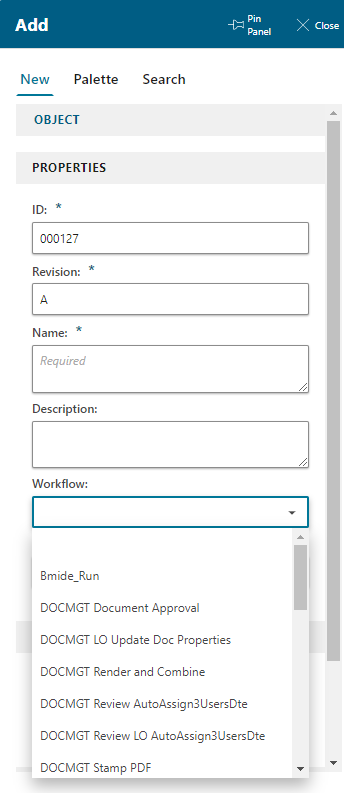
1. On the Custom Item Revision Make sure in Operation Descriptor, awp0ProcessTemplates property should be added in CreateInput with visible = true.
2. Save and Deployed the Data model.

**Configuration in Teamcenter**

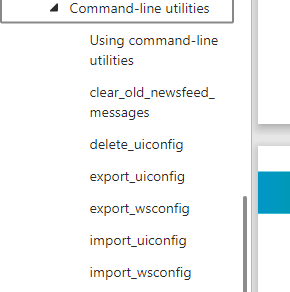
1. Set the preference value to true
2. Paste the line in awp0ItemCreate stylesheet .The line will display the property in AWC while you create item



The result will be like below



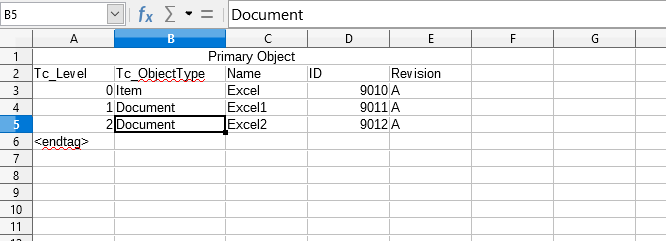
**Utilities we used in AWC**

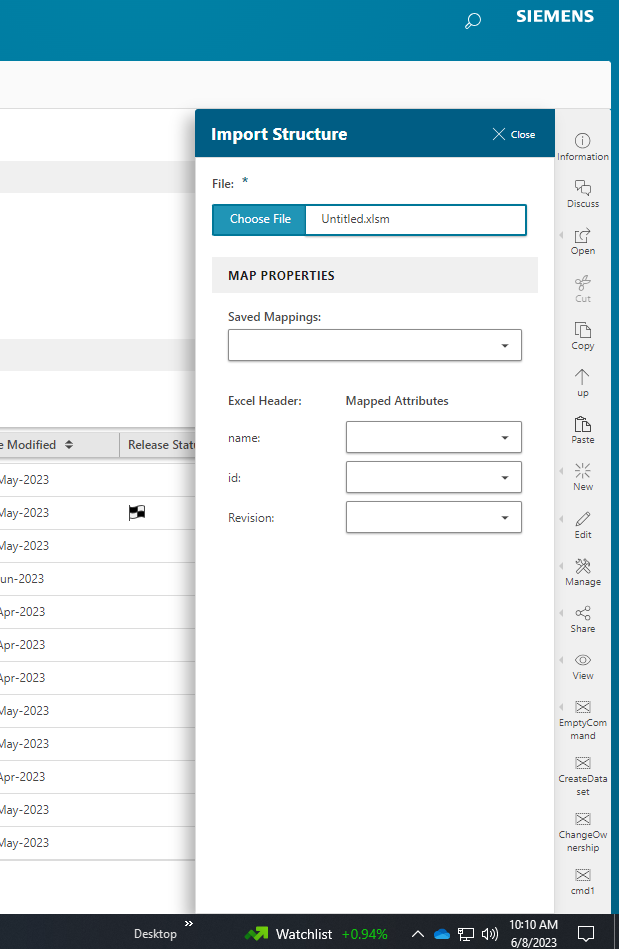


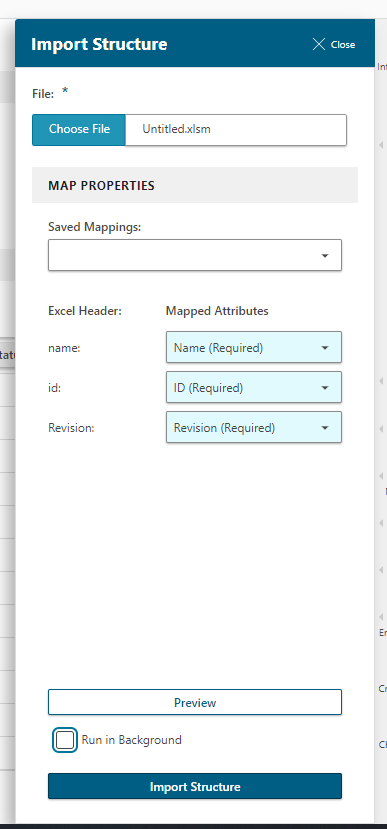


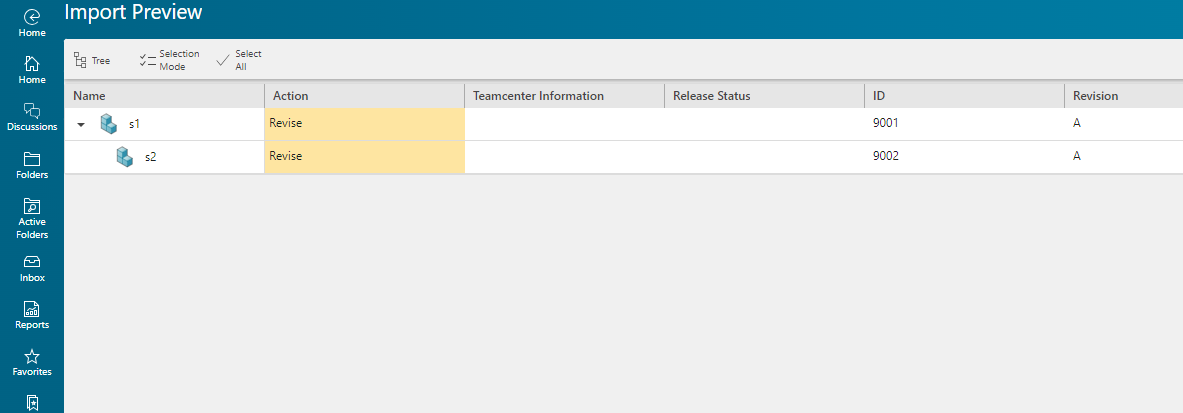
**IMPORTING STRUCTURE INTO AWC FROM EXCEL FILE**

* Excel file should be of this type



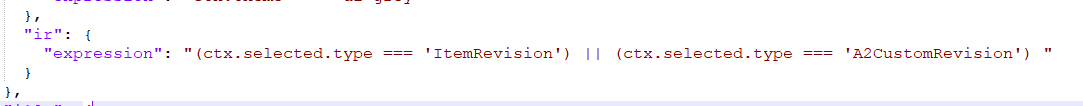






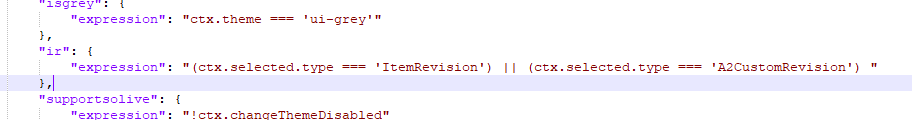
DISPLAYING COMMANDS ONLY WHEN PARTICULAR TYPE IS SELECTED

CONDITION:



CHANGING DEFAULT THEME

* **TO DISPLAY THE COMMAND ONLY WHEN SPECIFIC TYPE IS CLICKED**



**Closing a panel After an action is successful**

* Keep the code inside the action

"events": {

"success": [

{

"name": "complete",

"condition": "!data.isDestroyed()",

"eventData": {

"source": "toolAndInfoPanel"

}

}

]

},



* And in onEvent[], place the code,

"onEvent": [

{

"eventId": "updateProperty.contentUnloaded",

"action": "UpdateProperty"

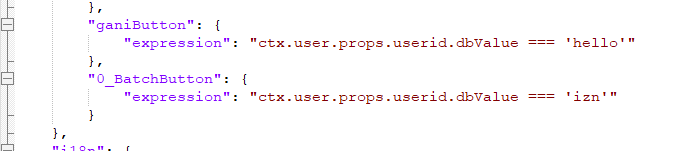
}

],



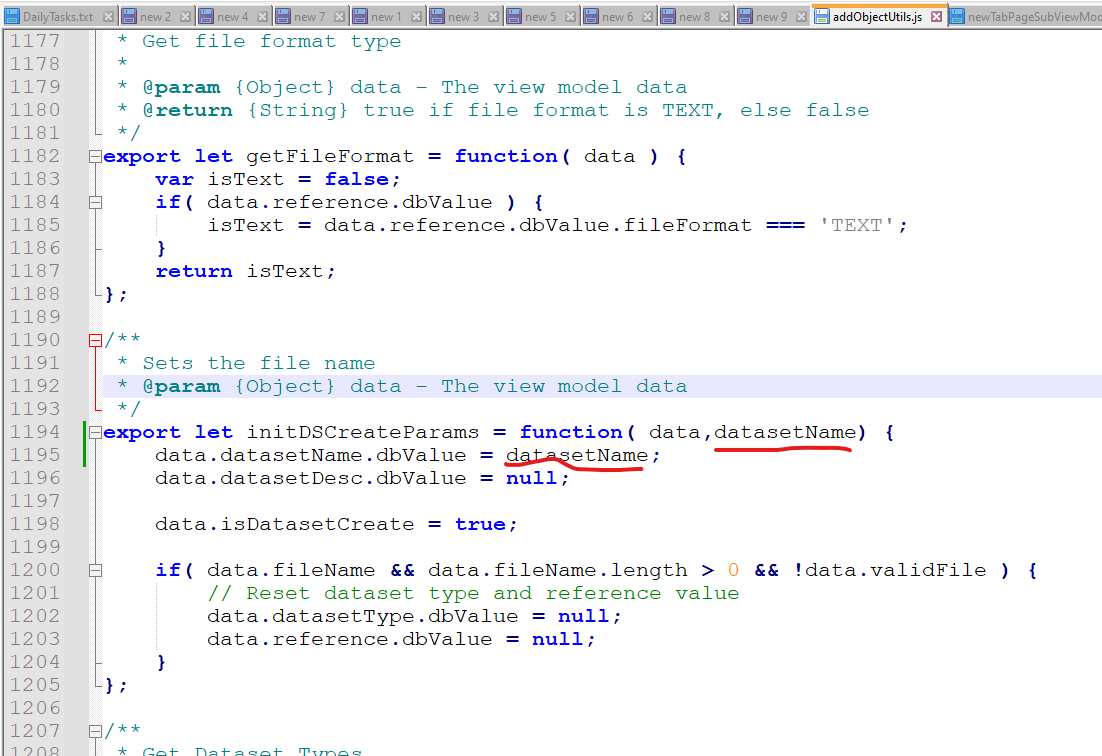
**DISPLAYING COMMANDS FOR SPECIFIC USER**

* Write the conditions like below image

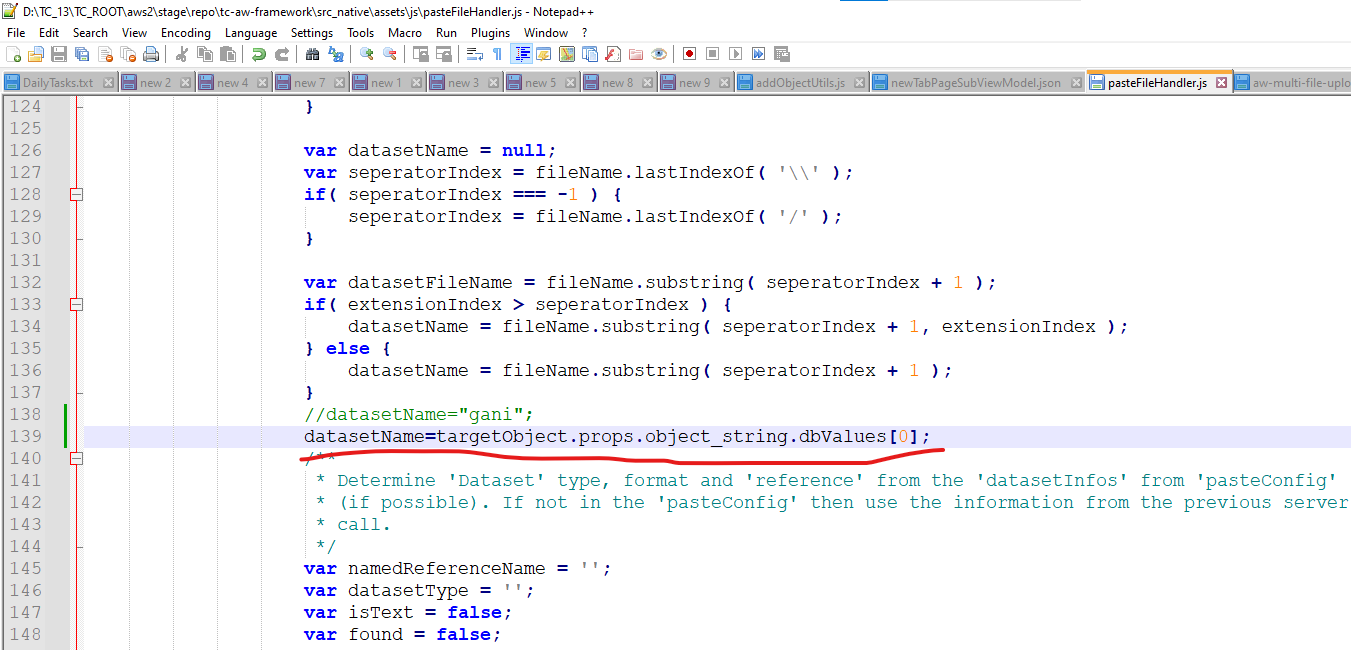


**DISPLAYING DATASET NAME AS REVISION NAME WHILE ADDING IT AS ATTACHMENT**





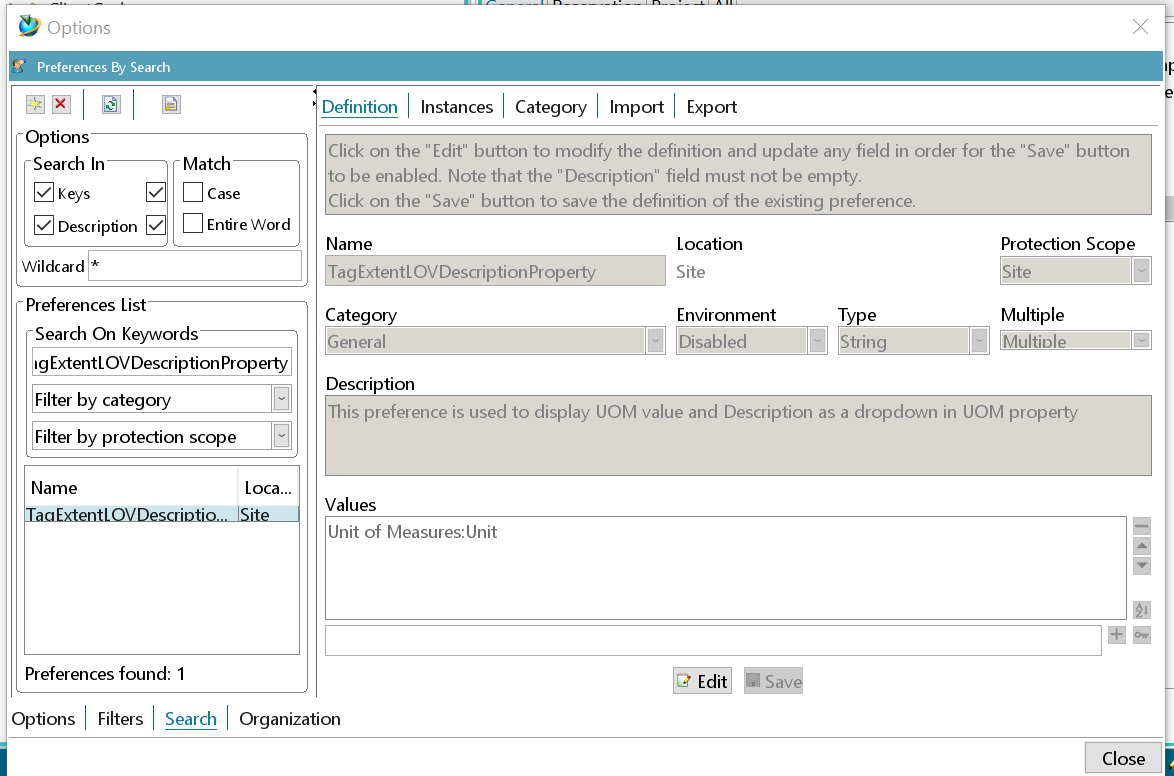
For Drag and drop service



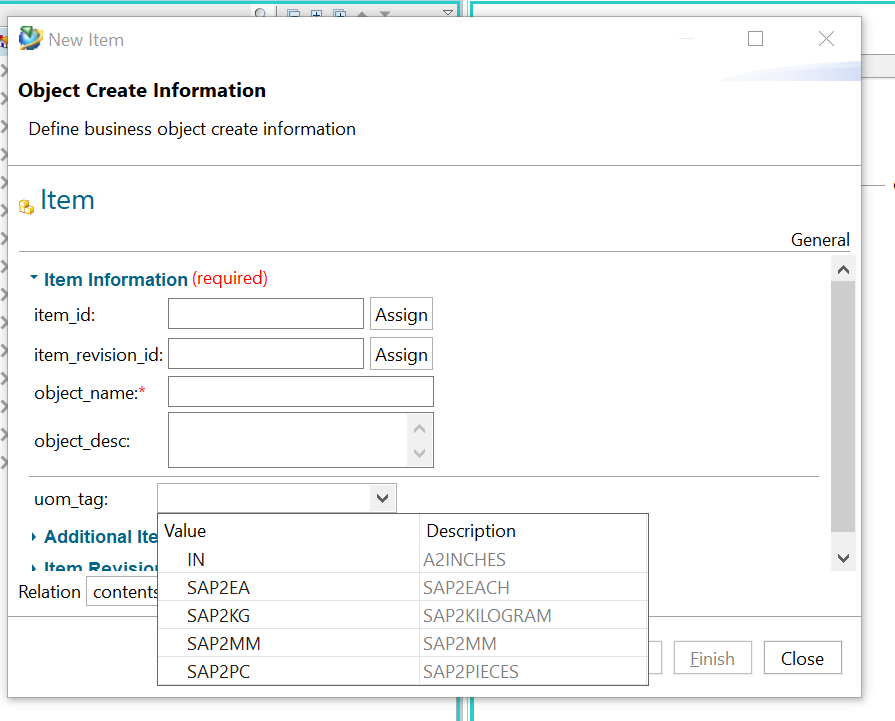
datasetName=targetObject.props.object\_string.dbValues[0];

DISPLAY UOM DESCRIPTION AND VALUE AS DROP DOWN

1.Create a preference with name :TagExtentLOVDescriptionProperty

2.Change the preference to Multiple and add the following value to the preference.  
Value in preference : Unit of Measures:Unit  
Save the preference.

3.Afte this the UOM description will appear as dropdown as appeared in below



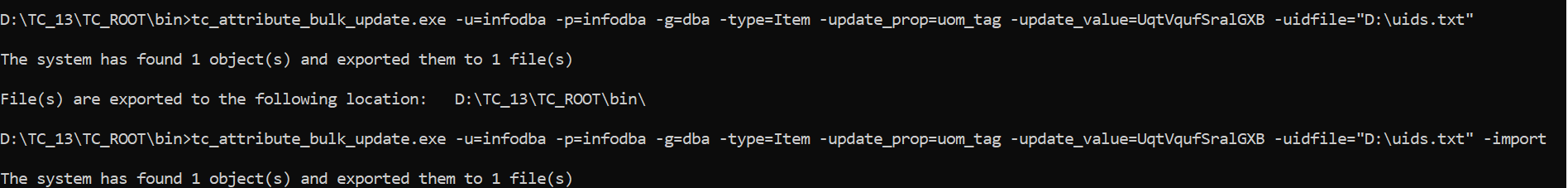
**Updating UOM property for an item that is used in structure**

In Below utility: the updated value is uid of symbol

uidfile is the uid of an item that is to be updated

>tc\_attribute\_bulk\_update.exe -u=infodba -p=infodba -g=dba -type=Item -update\_prop=uom\_tag -update\_value=UqtVqufSralGXB -uidfile="[D:\uids.txt](file:///D:/uids.txt)"

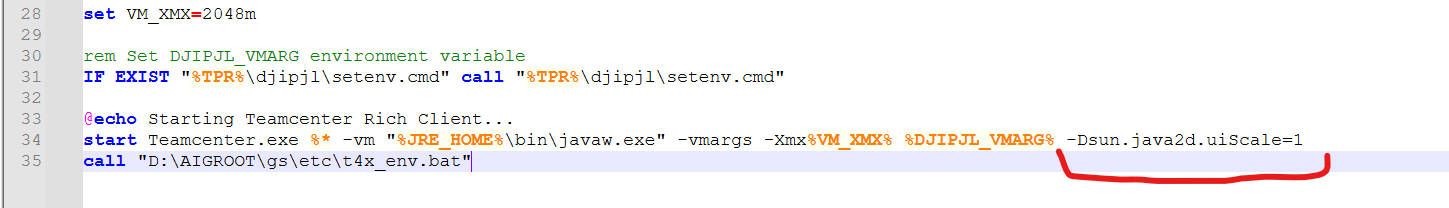
>tc\_attribute\_bulk\_update.exe -u=infodba -p=infodba -g=dba -type=Item -update\_prop=uom\_tag -update\_value=UqtVqufSralGXB -uidfile="D:\uids.txt" -import



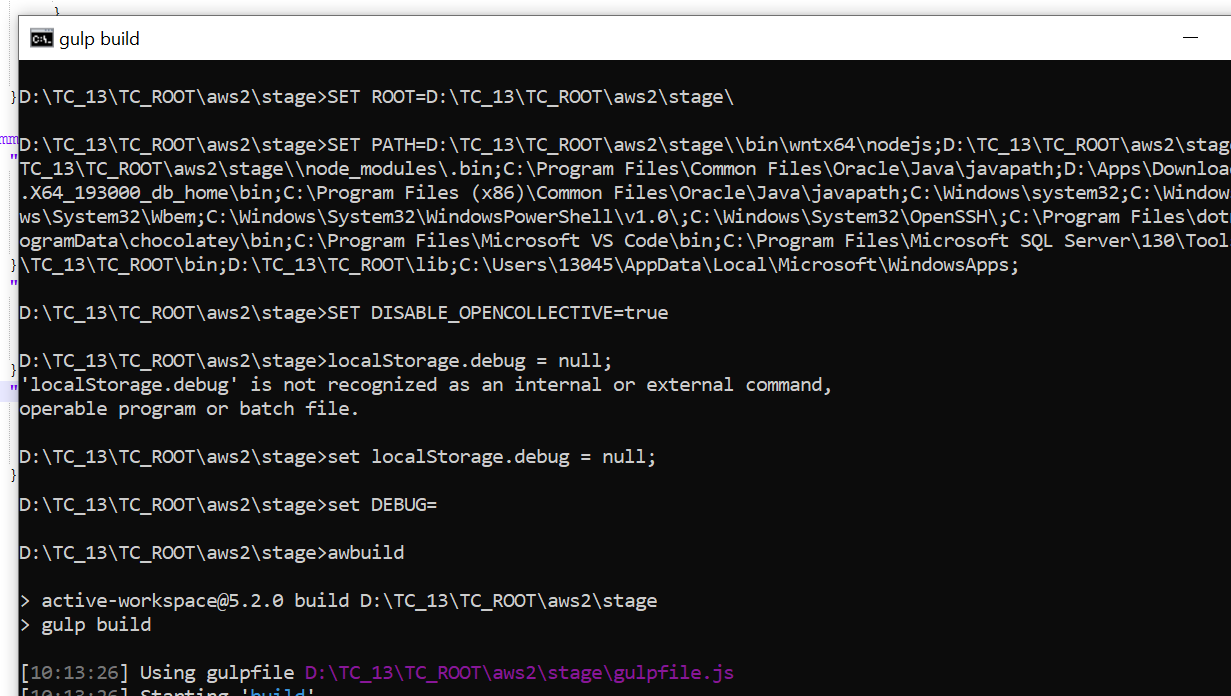
**IF YOU DONT WANT TO CHANGE THE FONT AS PER YOU SYSTEM**

If you don't want to change TC font as per Windows display setting then add **-Dsun.java2d.uiScale=1** in portal.bat , this option is ignoring OS scale

Add the highlighted line to **portal.bat**



**To disable the debuging environment while building AWCBUILD**



ICON SIZES

