OUTLINE

Search...

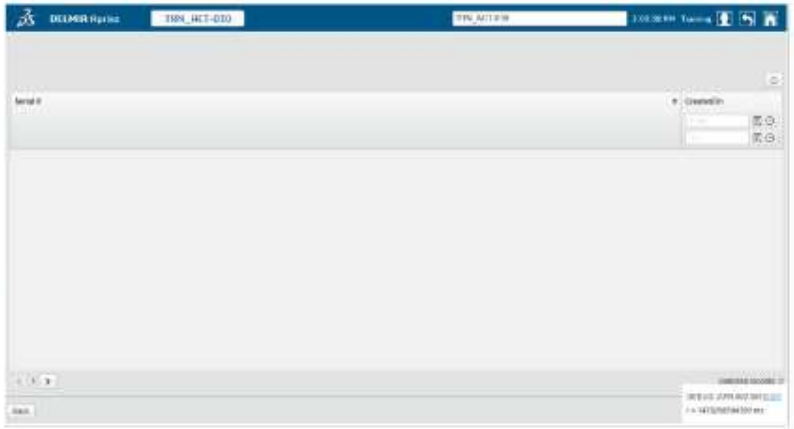


# LAB 4: Calculated Action - Screens

# LAB 4: Calculated Action - Screens

Task:

- Create two screens with grids:
  - TRNXX_ACT-020 to display serials for a selected product
  - TRNXX_ACT-030 to display lots for the selected product



What you will learn:

- How to perform basic configuration of a screen grid using the Grid 1.0 Business Control

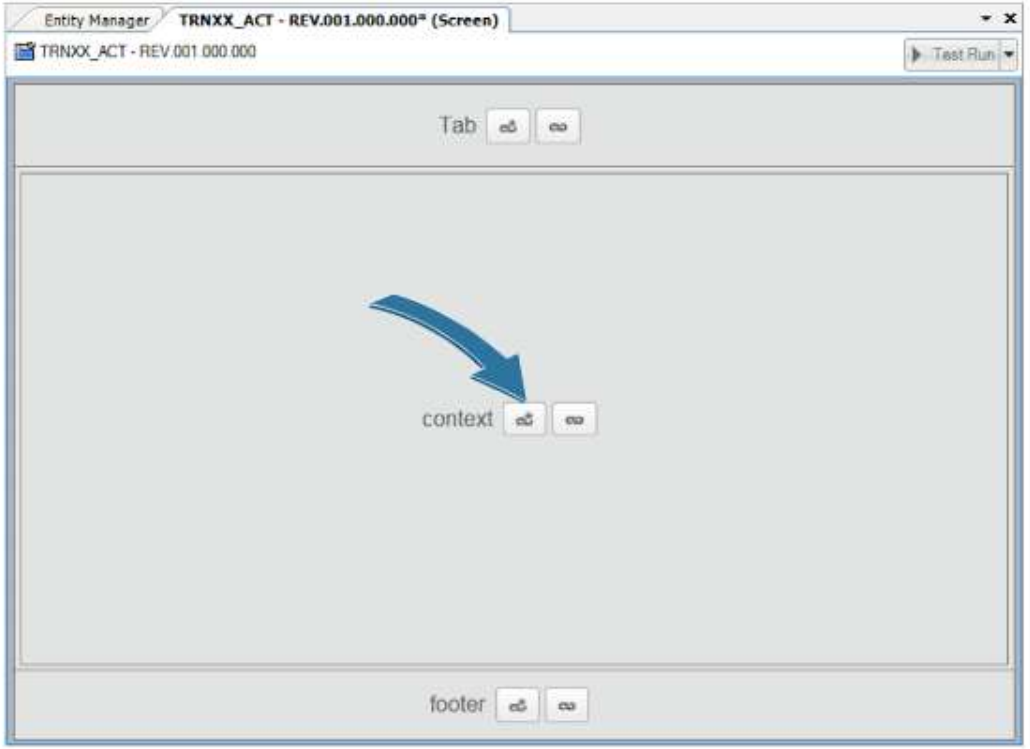
Requirements:

- In case of any technical problems, please contact DELMIA.Apriso.training@3ds.com

1 hour

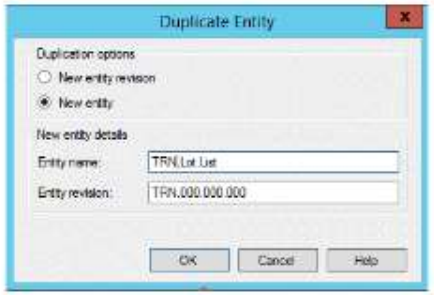
# LAB 4: Create New Screen TRNXX_ACT-020

- Create a new Screen
  - Name **TRNXX_ACT-020**
  - Revision: TRN.000.000.000
  - Layout: **Portal1PanelAndTabAndFooter**
  - Header: **PortalDefaultHeader**
- In the context panel click on the **Copy and Link** button



## LAB 4: Copy and Link PortalGrid View

- In the popup, select the View named **PortalGrid**
- Give the View a name **TRNXX.Lot.List**
- Use Revision: TRN.000.000.000
- Click on the **Open Operation** button



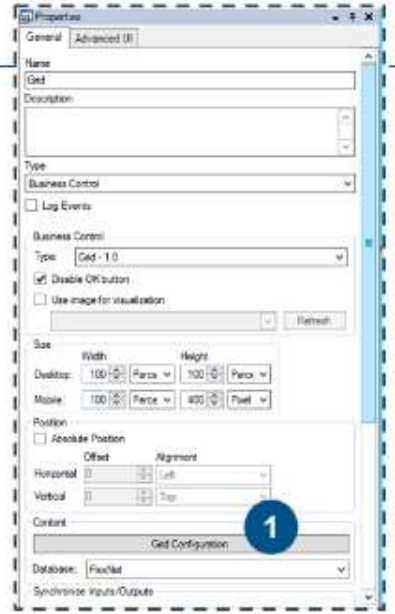
# LAB 4: Configure Grid, part 1

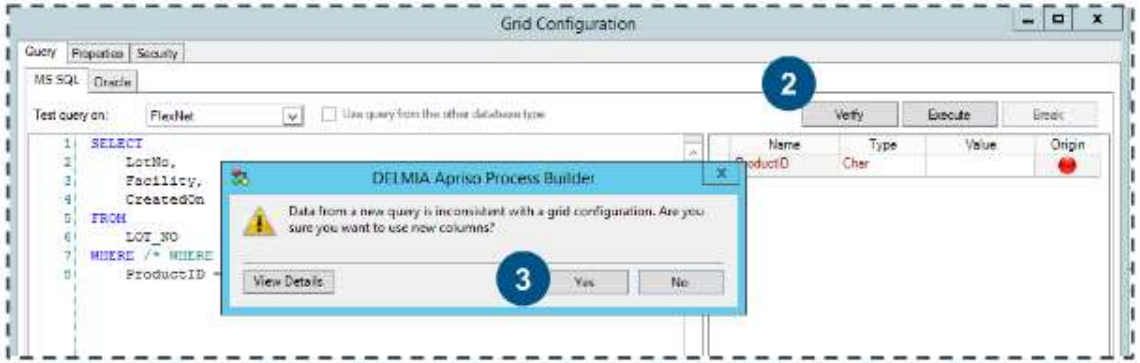
Click on the Grid function header and go to Grid function properties

1. Open the Grid Configuration and the query from scripts:

**SCRIPT FILE: Desktop/Training Materials/Level 1**

2. Click on „Verify" button
3. Click „Yes" to confirm you want to use new columns

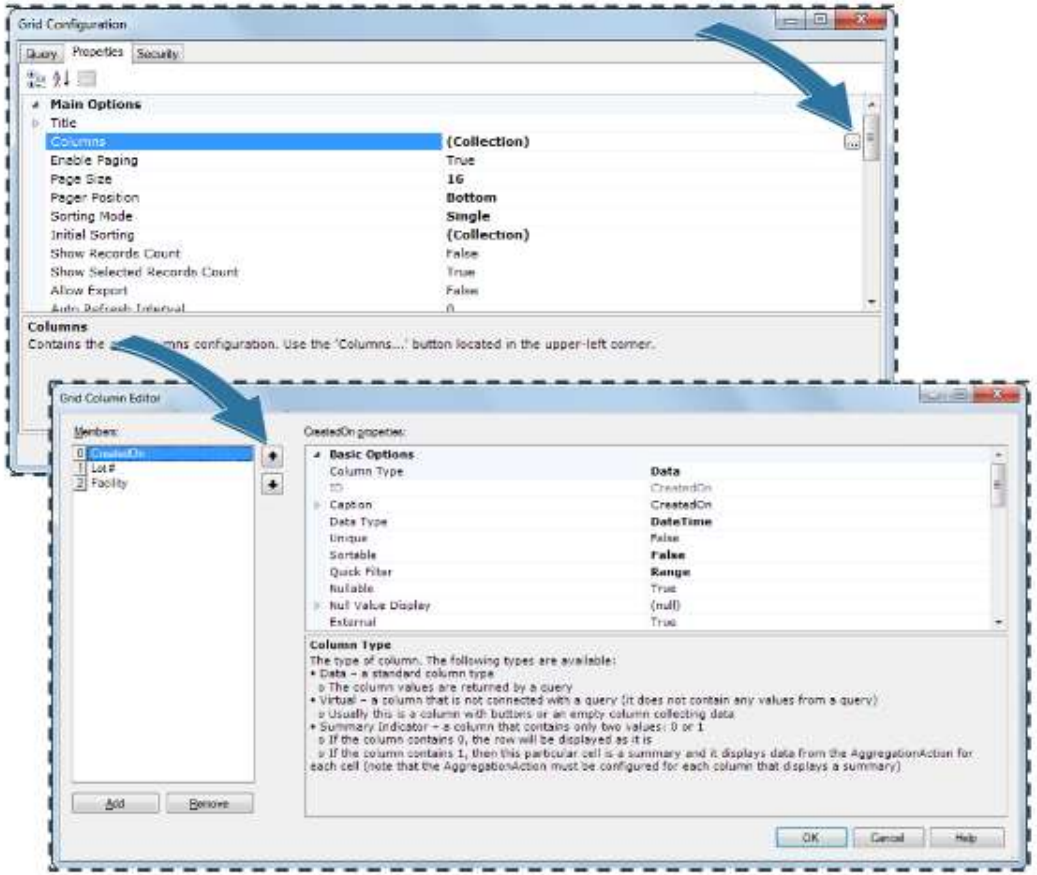


# LAB 4: Configure Grid, part 2

- Stay in the Grid Configuration, and go to the Properties tab
- Click on the Columns. This will make a button appear on the right side of the Columns record. Click on the button
- Use the Down button to move the CreatedOn column to the bottom of the list

You just rearranged the order of the columns in the grid.

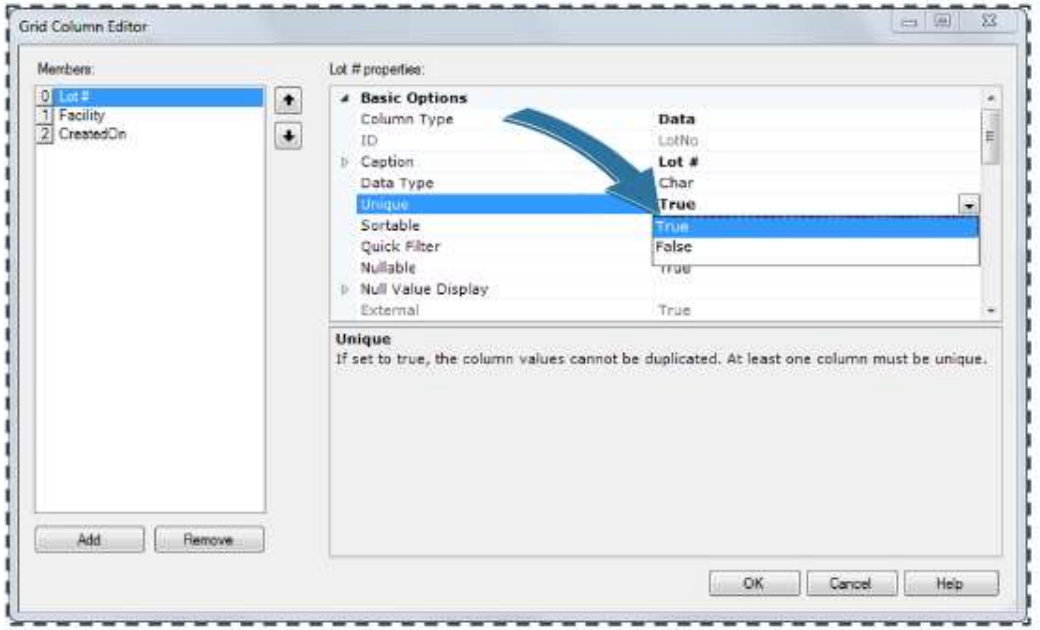


# LAB 4: Configure Grid, part 3

- Click on the **Lot #** in the Members section on the left and set the **Unique** property value to: **True**

You have just configured a grid which will show all lots which contain the product selected on the screen TRNXX_ACT (provided the product is lot-tracked).

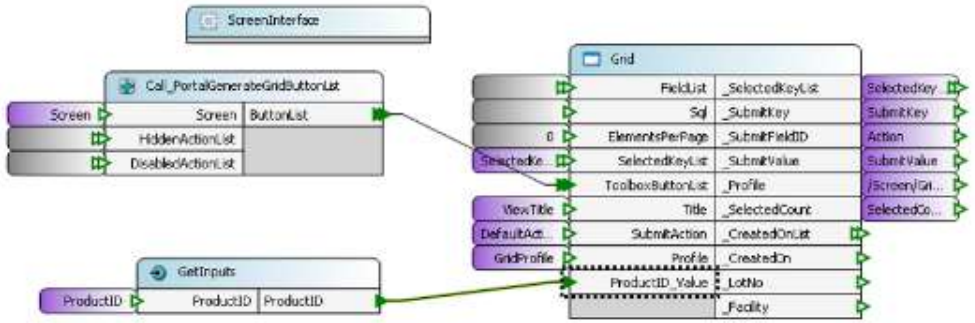
- Close Grid Column Editor
- Close Grid Configuration

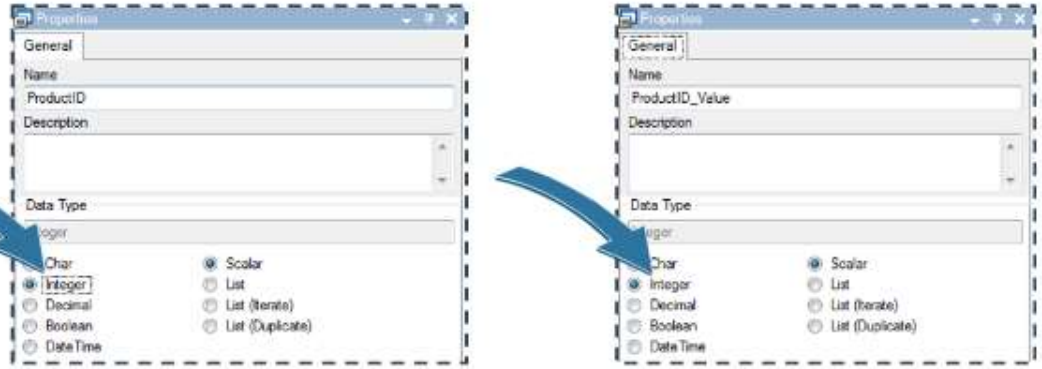


# LAB 4: Add Mapping for Product_ID

Provide a mapping for the ProductID_Value:

- Add an Input to Output function, and call it **GetInputs**
- Add a pair, and call it **ProductID**, data type should be **Integer** and **Scalar**, input should be **External**
- Map the ProductID output from the GetInputs function into the Grid function's ProductID_Value input
- Make sure the ProductID_Value is **Integer** and **Scalar**
- Save the Operation, and change its status to Prototype
- Change the status for the View, too



## LAB 4: Copy and Link New View in the TRNXX_ACT-020 Footer, part 1

- Go back to the TRNXX_ACT-020 Screen
- Click on the **Copy and Link** button in the footer panel

The Grid View does not have any buttons, and you will want to navigate between your Screens. Therefore the next step will be to add a View with a button to this Screen.

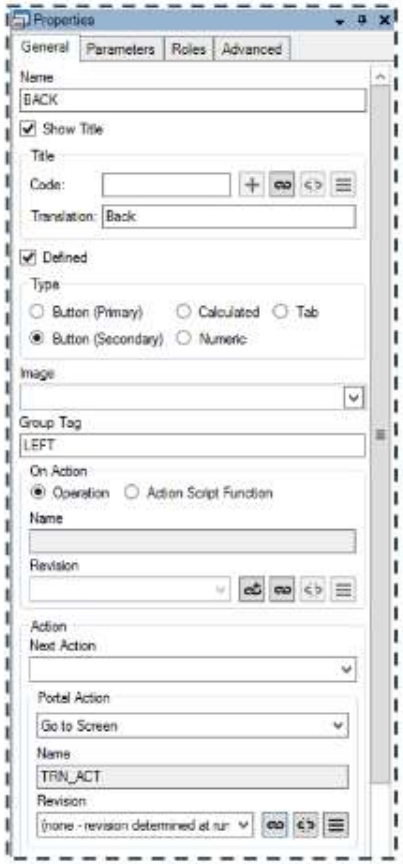
## LAB 4: Copy and Link New View in the TRNXX_ACT-020 Footer, part 2

- In the popup, select the **Portal2ButtonGroups** View
- Name the new View **TRNXX.ACT.Button**
- Use Revision: TRN.000.000.000
- Click on the **Open View** Button

# LAB 4: Configure Buttons in TRNXX_ACT-020

Configure the button that leads back to TRNXX_ACT Screen:

- In the **TRNXX.ACT.Button** View, **delete** the **BUTTON_RIGHT** Action
- Edit BUTTON_LEFT properties:
  - Name **BACK**
  - Translation **Back**
  - Type: Button (Secondary)
  - Link to Screen TRNXX_ACT
- **Save** the View and Screen TRNXX_ACT-020, and make sure both are in Prototype status

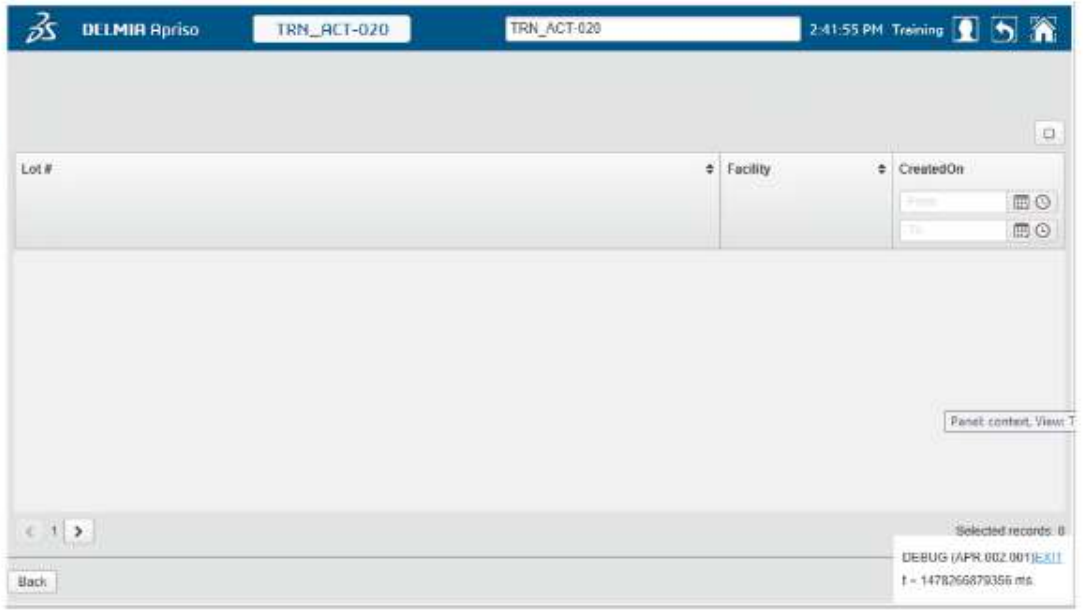


## LAB 4: Test Run TRNXX_ACT-020

You have just configured a grid which will show all lots associated with the product selected on the previous screen (provided the product is lot-tracked).

The Screen to display serials will be different from the ones displaying lots only by the grid definition.

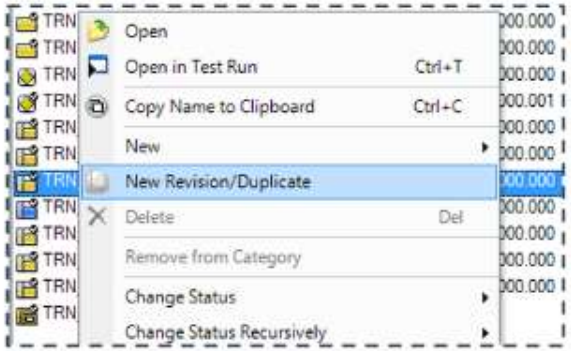
Therefore it makes sense not to create a new Screen, but to duplicate the TRNXX_ACT-020. This will be your next activity.

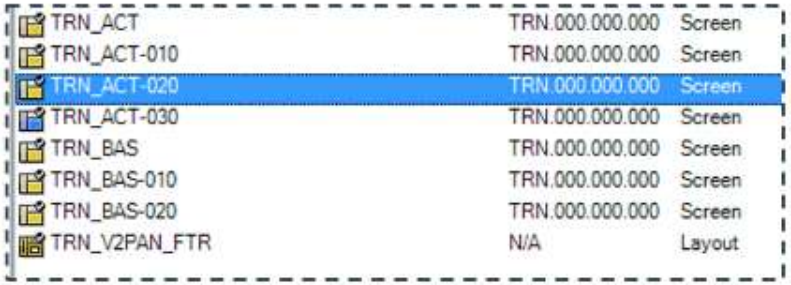


# LAB 4: Duplicate to New Screen TRNXX_ACT-030

The Screen to display Lot information is ready. Now we will create the Screen to show Serials, if the product is serial-tracked.

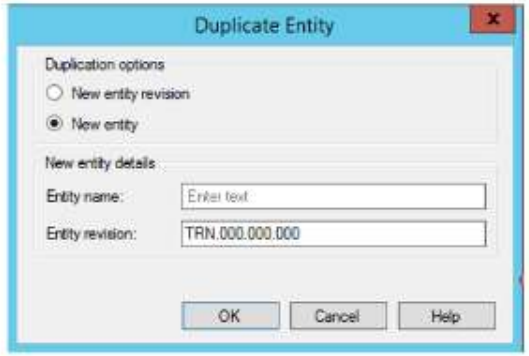
- **Duplicate** the TRNXX_ACT-020 Screen
- Name the new Screen **TRNXX_ACT-030**
- **Go to** the new Screen



# LAB 4: Copy and Link View to TRNXX.Lot.List

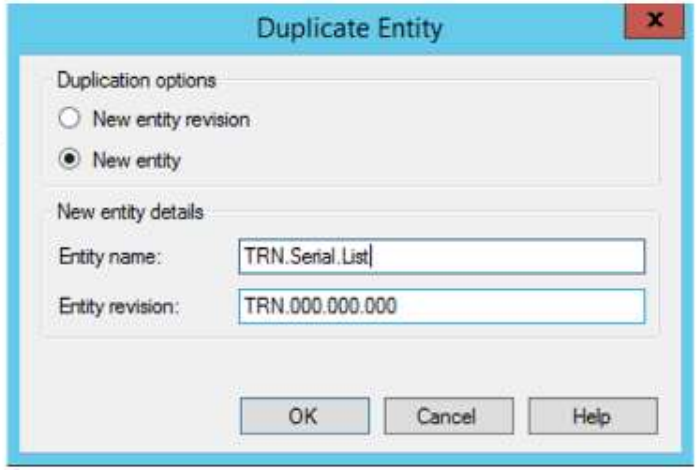
- Make sure the context panel is selected in the Entity Explorer
- Go to this panel's properties and click on the **Copy & Link** button at the View property
- Find the **TRNXX.Lot.List** View

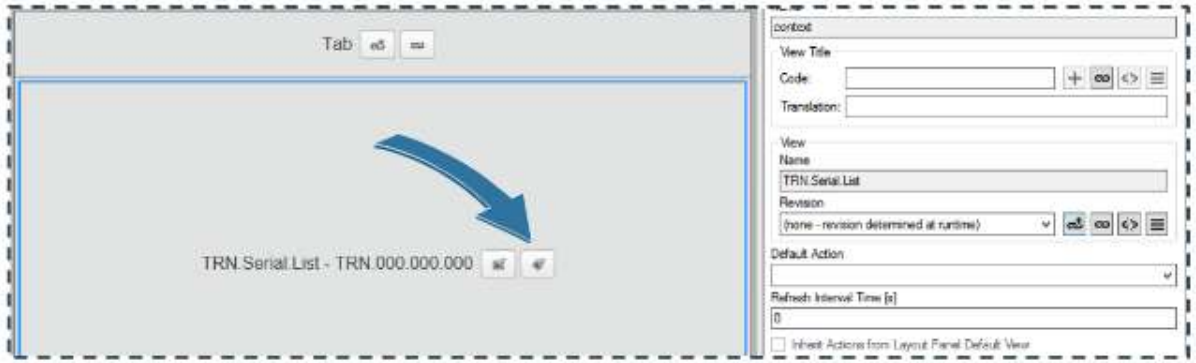




78

# LAB 4: Duplicate to New View TRNXX.Serial.List

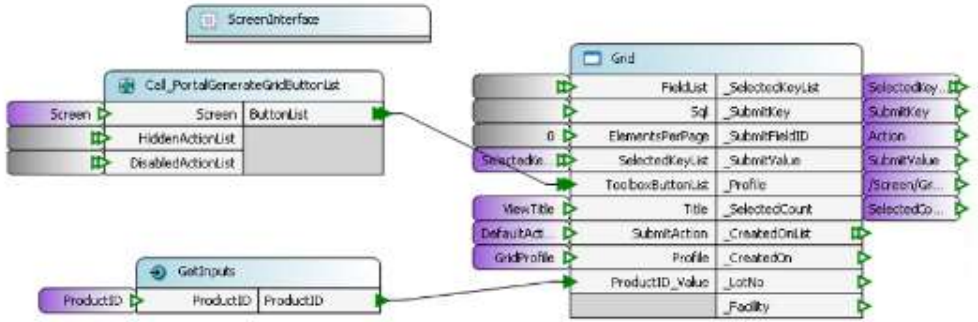
- Duplicate it to a new entity
- Give the new View a name **TRNXX.Serial.List**
- When the TRNXX_ACT-030 is refreshed with the new View, click on the **Open Operation** Button for this View

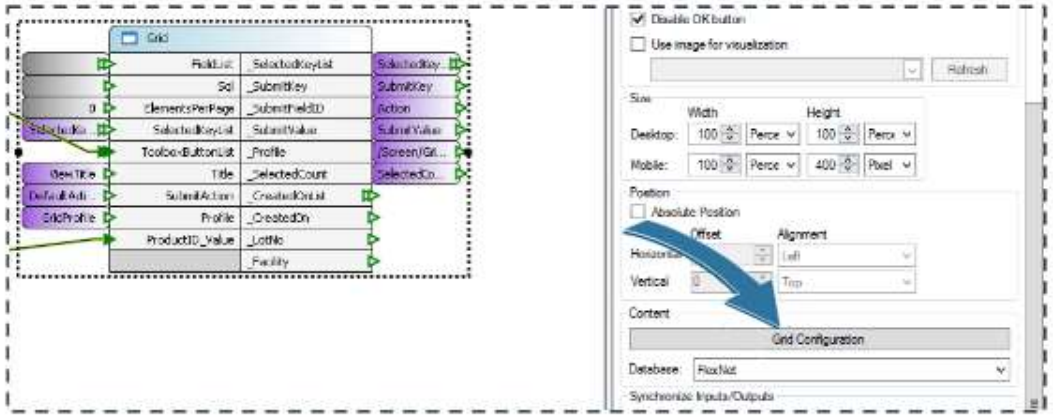


# LAB 4: Configure Grid, part 1

- Inside the Operation, click on the **Grid Function**
- In the Grid properties, click on the **Grid Configuration**



# LAB 4: Configure Grid, part 2

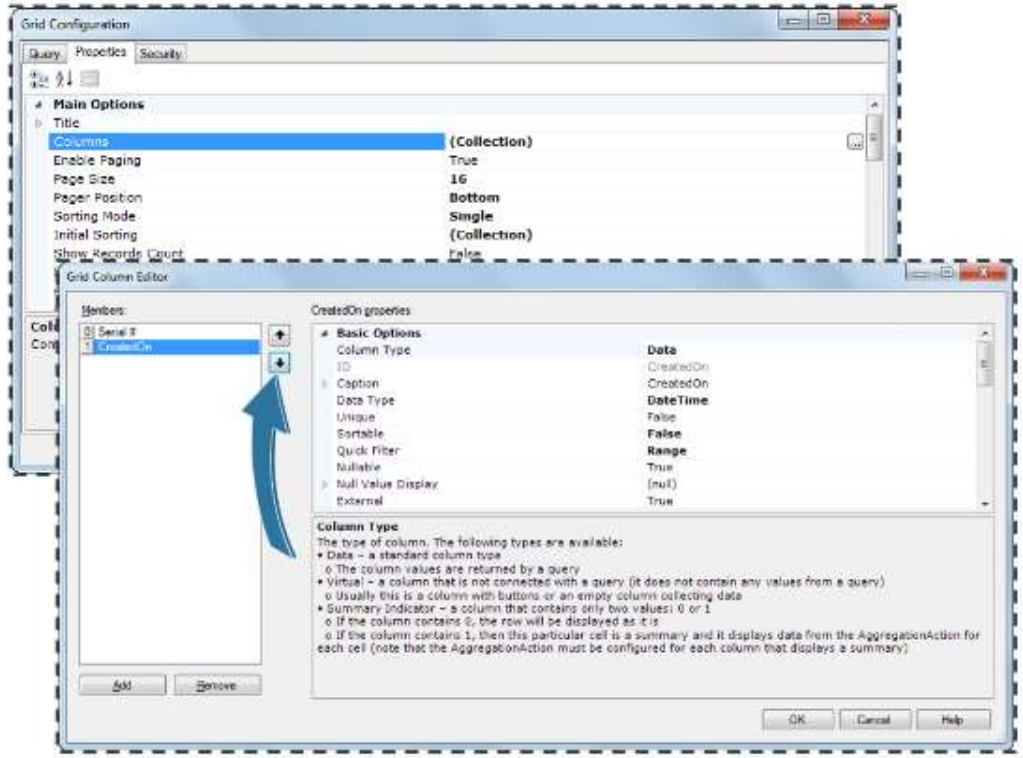
- Paste the query from scripts:

**SCRIPT FILE: Desktop/Training Materials/Level 1**

- When you verify the query (use the **Verify** button), confirm you want to use new columns

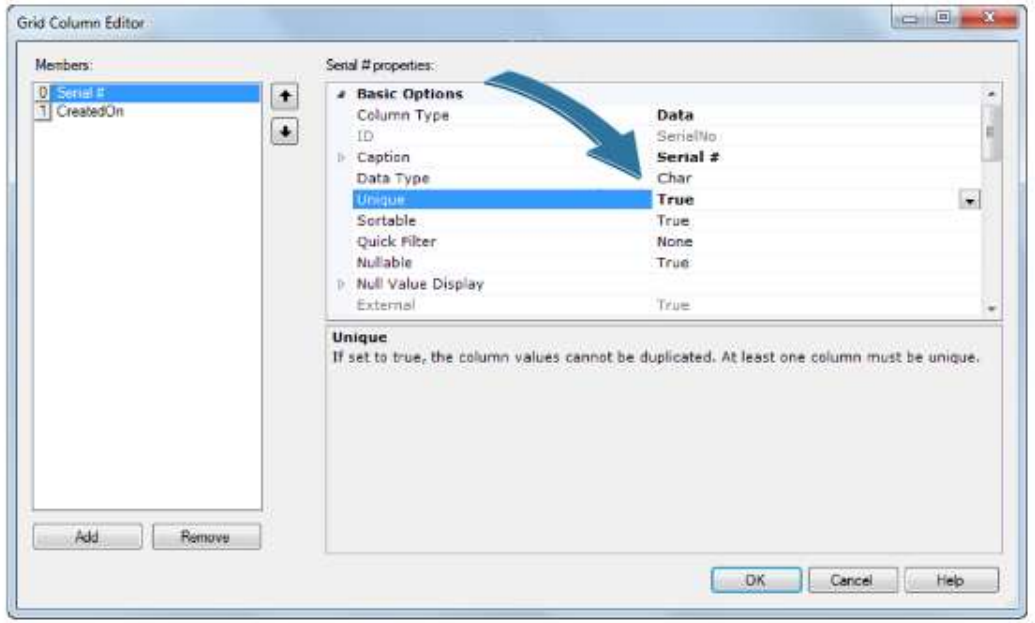
# LAB 4: Configure Grid, part 3

- Stay in the Grid Configuration, and go to the Properties tab
- Click on the Columns. This will make a button appear on the right side of the Columns record. Click on the button
- Use the Down button to move the CreatedOn column to the bottom of the list



## LAB 4: Configure Grid, part 4

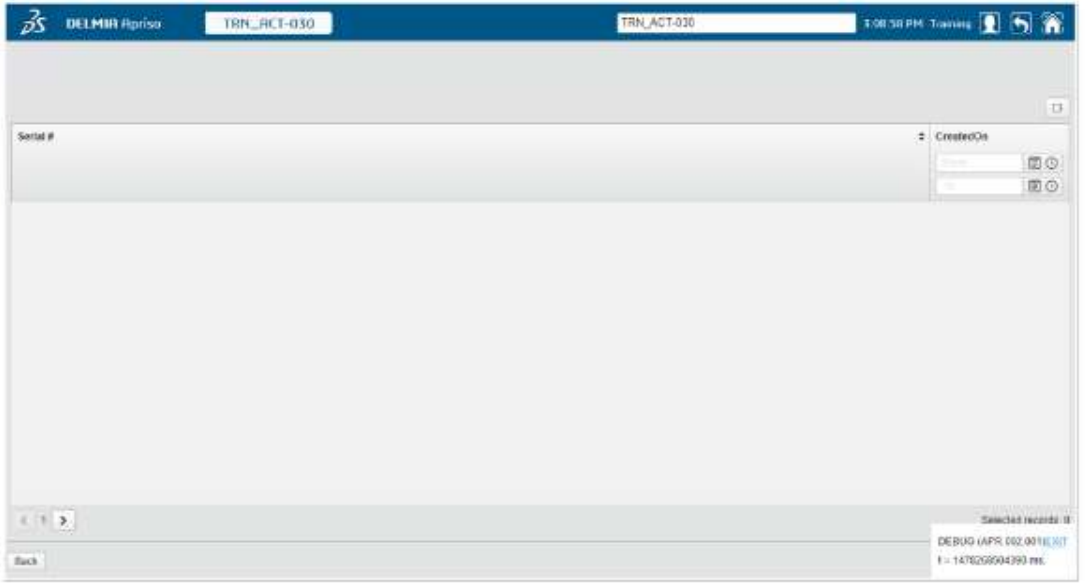
- Click on the Serial # in the Members section on the left and set the Unique property value to: True
- Go back to the Operation level, change its status to **Prototype**
- Change the status for the View and the Screen, too



# LAB 4: Test Run TRNXX_ACT-030

You have just configured a grid which will show all serial numbers associated with the product selected on the previous screen (provided the product is serial-tracked).

The next activity will be to tell the system which screen should be shown when a ProductNo is entered on the start screen.



OUTLINE

Search...



# LAB 5: Calculated Action - Business Logic

# LAB 5: Calculated Action - Business Logic

Task:

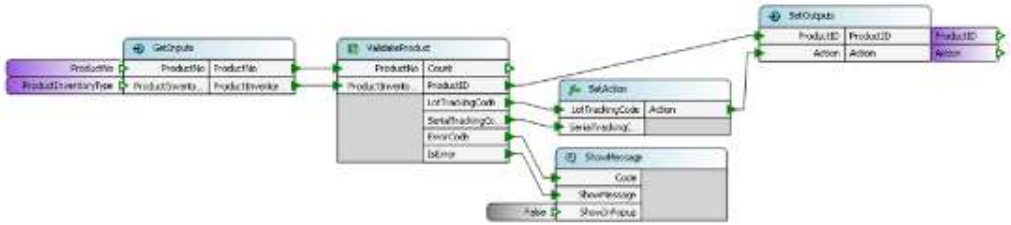
- Build a Standard Operation with a business logic to determine next screen
- Modify the TRNXX_ACT Screen to incorporate the updated Operation, and include changes to the Screen flow

What you will learn:

- How to expand your screen flow. When you enter a ProductNo on the main Screen, the system will use business logic to tell if the product is serial or lot tracked. Depending on this information, a screen with lots, or a screen with serials will be displayed.
- How to configure the business logic, add necessary screens, and modify the screen flow.

Requirements:

- In case of any technical problems, please contact DELMIA.Apriso.training@3ds.com



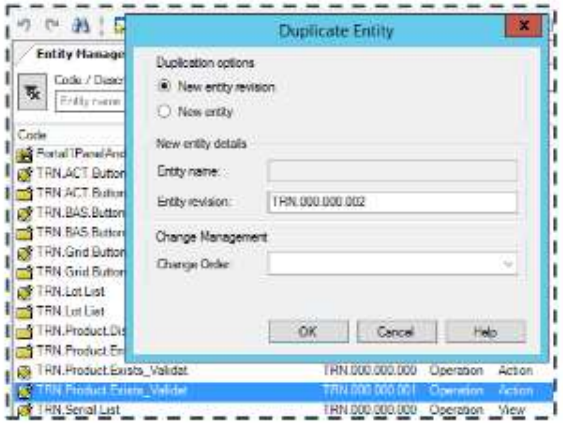
1 hour

# LAB 5: Duplicate Operation to TRNXX.Product.Exists_Validate

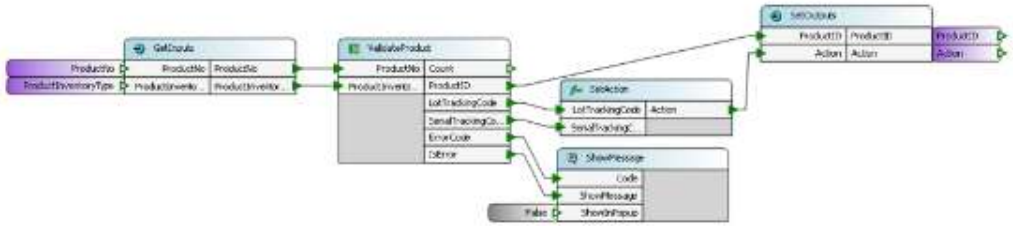
- Duplicate the Operation **TRNXX.Product.Exists_Validate**, Revision TRN.000.000.001, to another revision and make it a **default revision**. Go to the Operation

You will use your product validation Operation to determine which Screen to show after the product information is entered.

For this purpose, you need to expand the validation to check whether the product is lot or serial tracked.




If the Operation did not display check the **Has changes** option in the Entity Manager filter bar.



# LAB 5: Provide Script for ValidateProduct Function

- Change the SQL query in the ValidateProduct function to the one from scripts:

**SCRIPT FILE: Desktop/Training Materials/Level 1**

- Go to Outputs tab and mark the tracking codes as **Function Outputs**

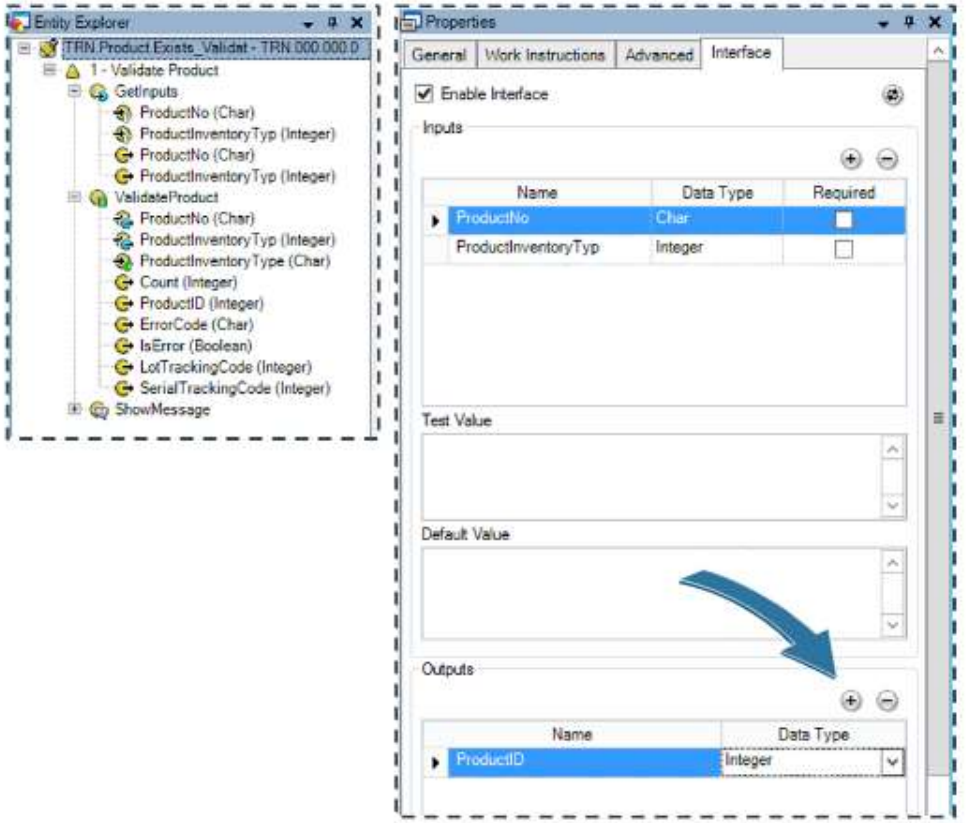
The tracking codes for lot and serial will be used in the screen routing validation. ProductID will be sent to the respective screens to determine the records to show.

# LAB 5: Change ProductID Output

Since the ProductID will be sent to other Screens, it needs to be marked as an external output from this Operation:

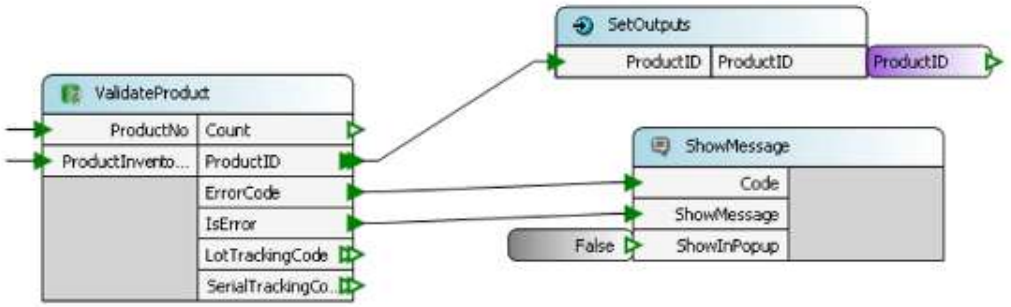
- In the **Entity Explorer**, make sure the top node (the Operation) is marked
- Go to **Operation Properties** and select the **Interface** Tab
- Add **ProductID** output as **Integer**

The name of the output has to be identical to the ProductID input which you used in queries in the grid Screens.



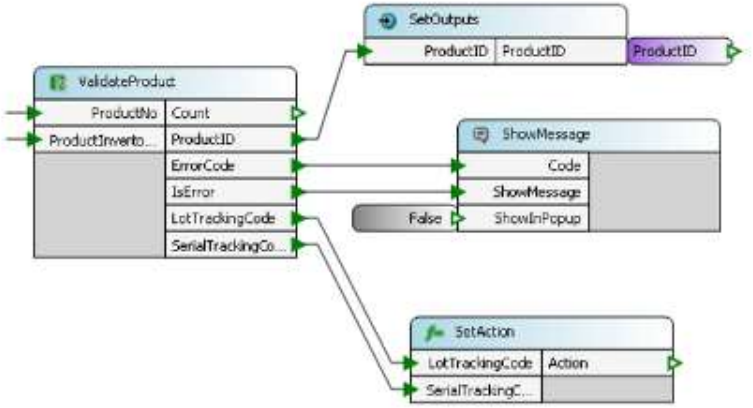
# LAB 5: Configure SetOutputs Function for ProductID

- Add a **SetOutputs** function (Input and Output type)
- Add a Pair there
- Name the Pair **ProductID**
- Link the ProductID output from the ValidateProduct function into the ProductID input of the SetOutputs function
- In the SetOutputs function, make the ProductID an external output (Add External Routing)



# LAB 5: Add SetAction Function

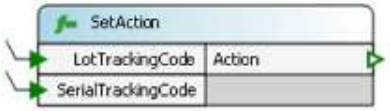
- Add a User Formula function, name it **SetAction**
- Create an Action output in the SetAction function. Make sure its type is Character
- Drag and drop both tracking codes outputs from the ValidateProduct function into the SetAction function
- Make sure both inputs are of **Integer** type, and **Scalar**

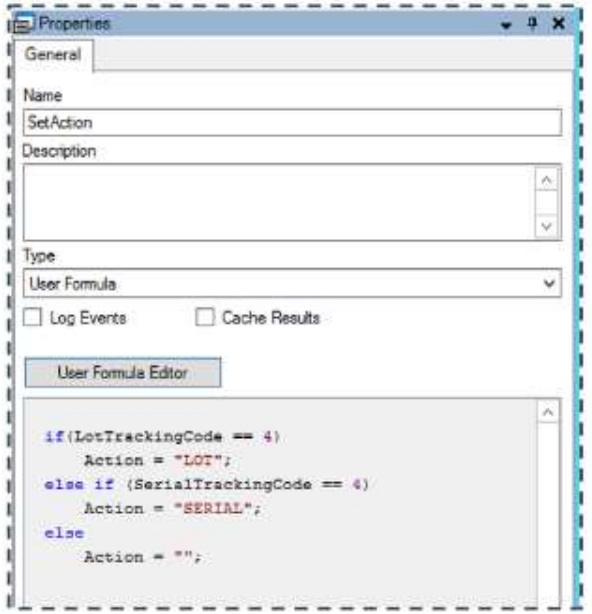




# LAB 5: Provide Script for SetAction

- In the **SetAction** function properties, click on the **User Formula Editor**
- Add the following C# formula from scripts:




**SCRIPT FILE: Desktop/Training Materials/Level 1**

The Action values LOT or SERIAL will be passed to the View, and each will have a respective View Action. If LOT is returned, the LOT View Action will be executed, and you will see the appropriate screen. Similar logic will apply to serial tracked product.

# LAB 5: Validate the SetAction Script Function

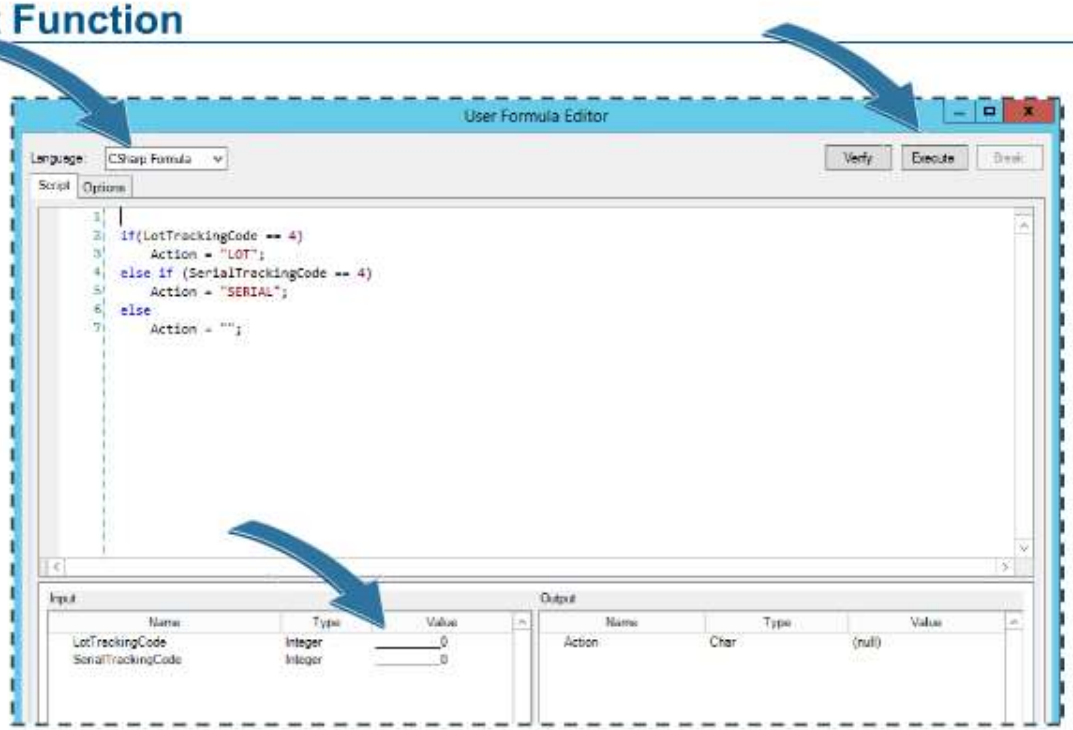
- When you paste the script, make sure the Language stays selected as CSharp Formula

Use the Verify button to check the script.

To test the script, type **4** in the input value field for LotTrackingCode and use the Execute button next to the Verify button.

To test the script for SerialTrackingCode, type **4** in the correct input value field, and use the Execute button.

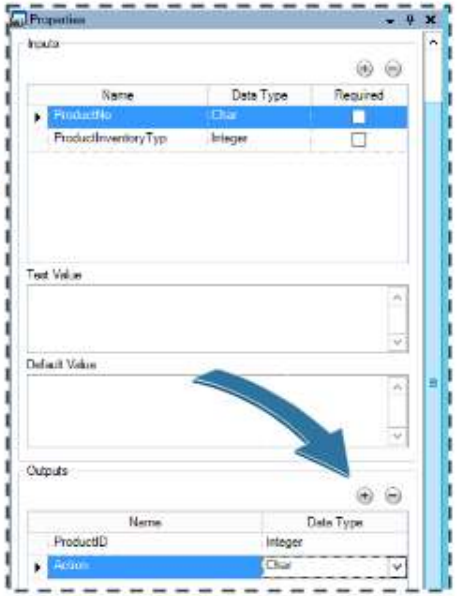
When happy with results, click OK.

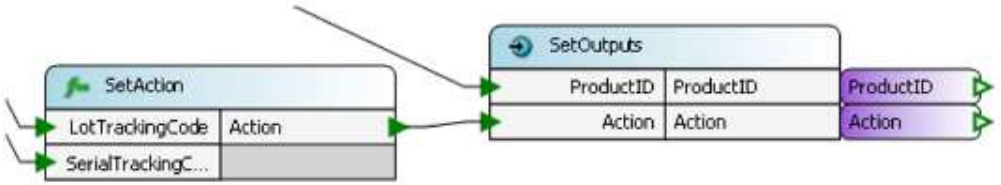


# LAB 5: Configure Operation Outputs

Now, let's add outputs to the Operation:

- Add Action external output in the **Operation Properties, Interface** tab. Make sure its type is **Character**
- Add a Pair in the SetOutputs function, and name it **Action (Char)**
- Connect the Action output from SetAction function to the Action input in the SetOutputs function
- Define the Action output from the SetOutput function as external output. Make sure the name of this external output is Action
- **Save** the Operation and change status to **Prototype**

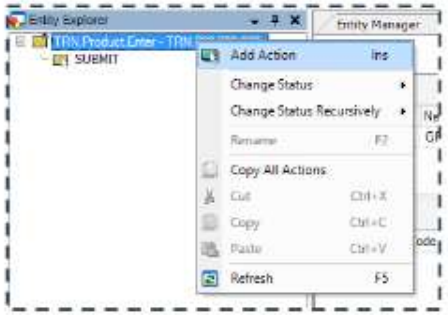


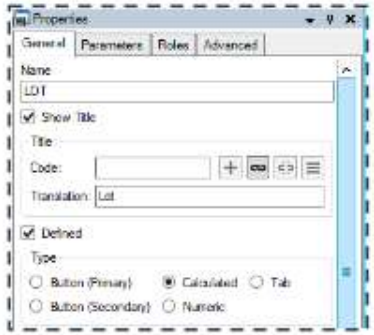



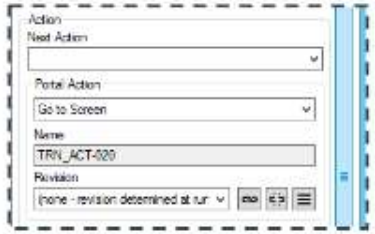
# LAB 5: Configure Actions Lot and Serial

Actions LOT or SERIAL returned by the Operation need to be configured in the View:

- Go to the **TRNXX.Product.Enter** View
- Make sure the View is selected in the Entity Explorer. Right-click on the View
- Select **Add Action**
- Rename the new Action to **LOT**
- Make it **Calculated**
- Set the LOT Action to trigger the **TRNXX_ACT-020** Screen



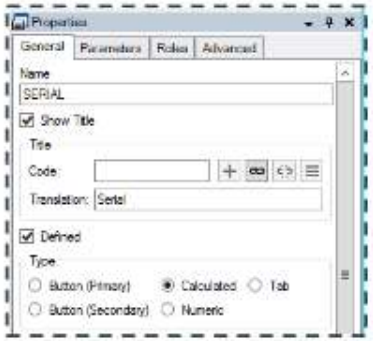



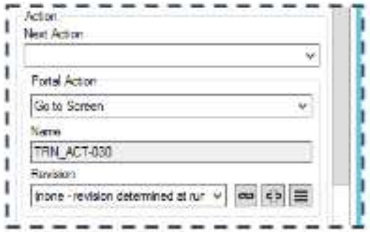



# LAB 5: Add Second Serial Calculated Action

- Use similar steps to add another Action **SERIAL**
- Make it **Calculated**
- Set the SERIAL Action to trigger the **TRNXX_ACT-030** Screen

When the Operation you just configured previously returns the Action value LOT, the View will use the LOT Action to direct the user to the Screen with information about lots. When **SERIAL** will be returned by the Operation, the SERIAL Action will be executed, and the user will be taken to the Screen with serials.

# LAB 5: Configure OnAction Operation in TRNXX.Product.Exists_Validate

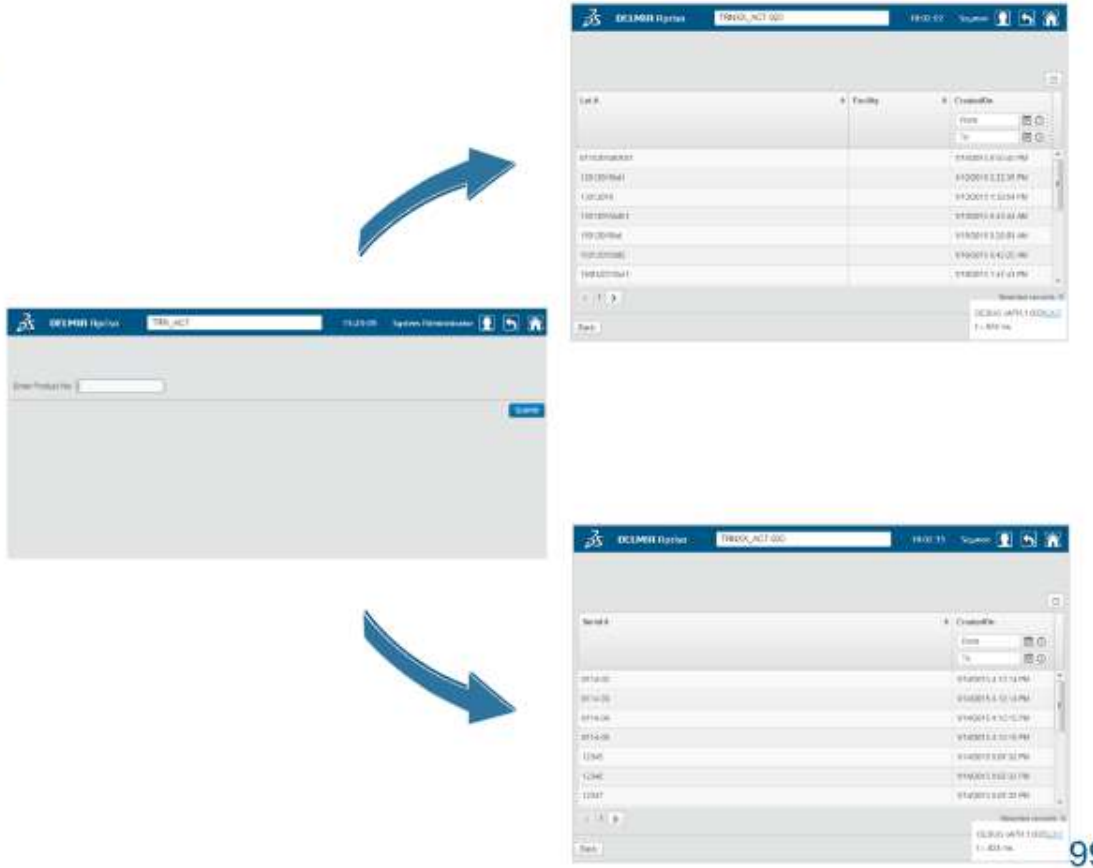
- In the SUBMIT button properties, make sure the OnAction Operation property is set to the TRNXX.Product.Exists_Validate
- **Unlink** the TRNXX_ACT-010 Screen in the Portal Action Property
- **Save** the View

With the LOT and SERIAL Actions, the purpose of the SUBMIT button is only to submit the Screen, and trigger the Operation to determine whether the serials or lots screen should be displayed.


# LAB 5: Test Run TRNXX_ACT

- Execute screen TRNXX_ACT
- Use the **B-BABY-001** product, to go to the Screen with all lots which have this product
- Use the **A-PIST-001** product, to see all serials with this product
- For a non tracked product (**G-OILS-001**), the Screen will be only refreshed

In this lab, you have seen how you can control the Screen flow using a Standard Operation with an Action external output, and View Actions of the Calculated type.



# Chapter 5: Action Chaining

**In this chapter you will learn how to call further Actions from previous Actions. This feature of the SFM will allow you to further expand the business logic you may need to link to your Screen Flows.**

## Here are the chapters to be covered:

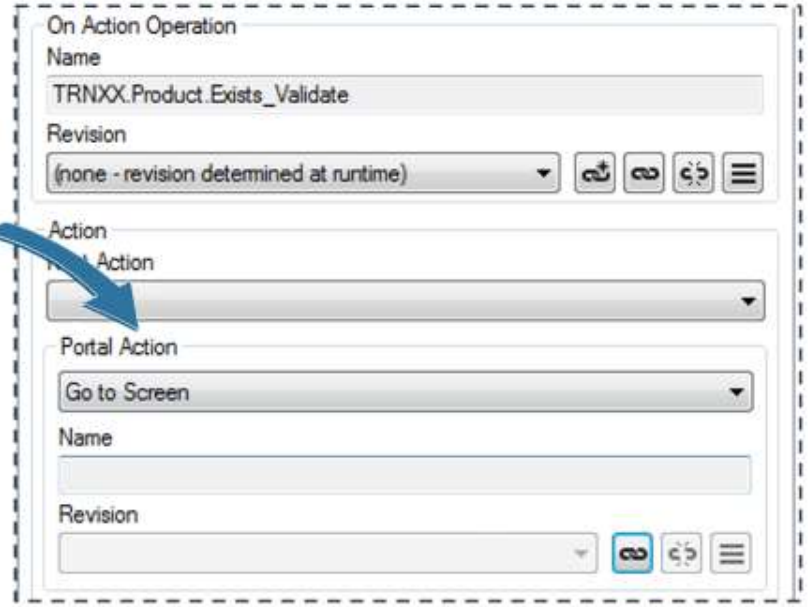
1. *Actions and Action Types*
2. *OnAction Property*
3. *Adding Parameters to OnAction Operation*
4. *Calculated Actions*
5. **Action Chaining**

# Next Action Property, part 1

Of the Action related properties, you have already used Portal Action and On Action.

Next Action is used to determine the screen flow behavior if the OnAction Operation returns no value in the Action output.

You have just seen an example in your lab, where a non-tracked product is not processed, the Screen is just refreshed, and no useful information is provided. To handle this better, the Next Action property can be used.

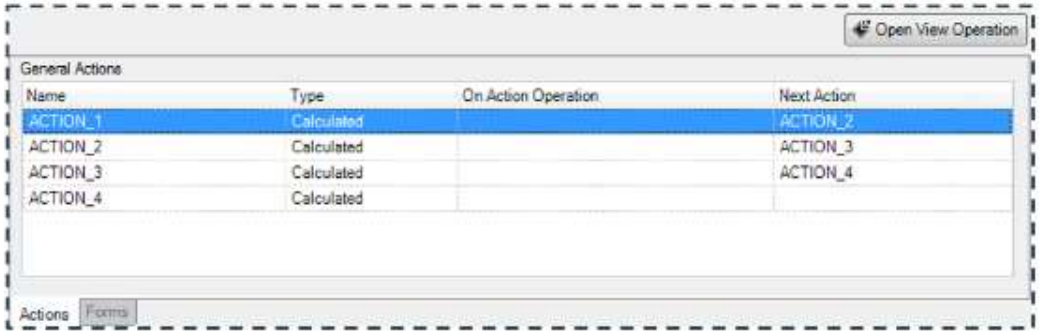


## Next Action Property, part 2

Next Action can also be used to chain Actions on a View, if you have multiple Actions and you want them to be executed in a particular order.

In the example shown here, 4 Actions will be executed one after another.

Obviously, in a realistic situation, all Actions need to have some business logic attached, e.g. with linked OnAction Operation.

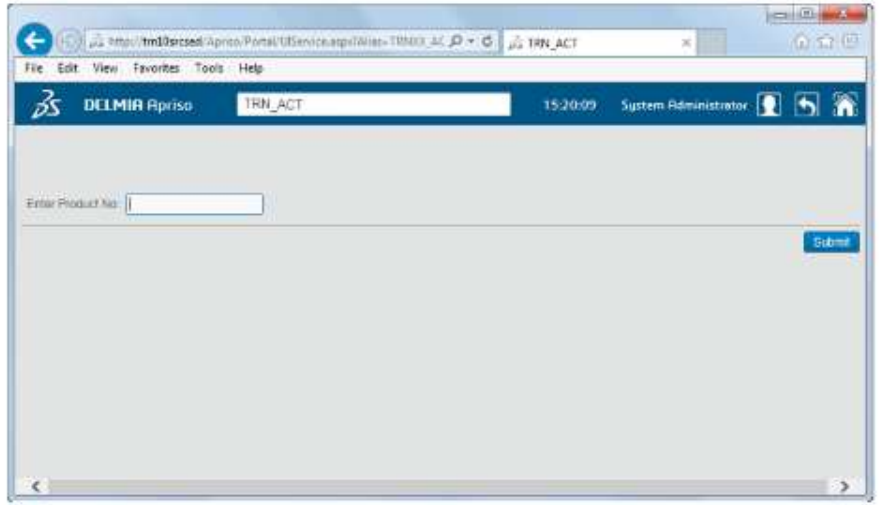


# LAB 6: NextAction in Use

# LAB 6: NextAction in Use

Task:

- Add a new Action to TRNXX.Product.Enter to route to TRNXX_ACT-010 Screen
- Mark the new Action as a Next Action to the SUBMIT Button
- Update the TRNXX_ACT-010 Screen to display a more meaningful text
- Publishing the TRNXX_ACT Screen as a FlexPart
- Test run your Screen




What you will learn:

- How to add another Action to the TRNXX.Product.Enter View, which will be executed if the product is neither lot nor serial tracked. This new Action will be configured as a Next Action on the SUBMIT button. It will be executed if the On Action on this button returns no value.

Requirements:

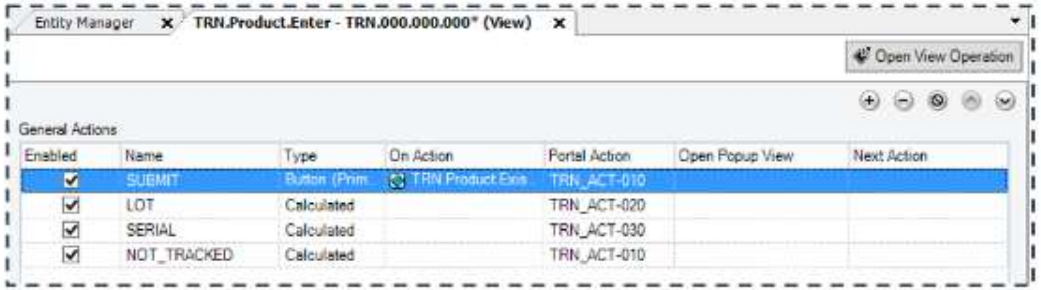
- In case of any technical problems, please contact DELMIA.Apriso.training@3ds.com

20 min

# LAB 6: Configure Actions Tab in TRNXX.Product.Enter View

- Go to the Screen **TRNXX_ACT**, and open the TRNXX.Product.Enter View
- Go to the Actions tab
- Add a new Action, **NOT_TRACKED**. Make it a **Calculated** Action, and use the Portal Action property to direct it to the **TRNXX_ACT-010** Screen

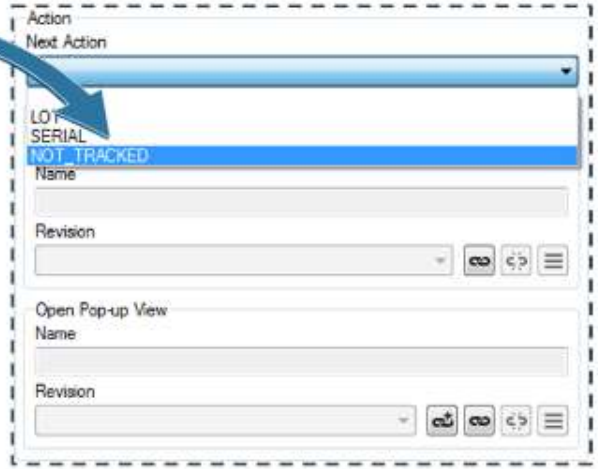
You may remember that the TRNXX_ACT-010 Screen displays the product information you entered on the TRNXX_ACT Screen.

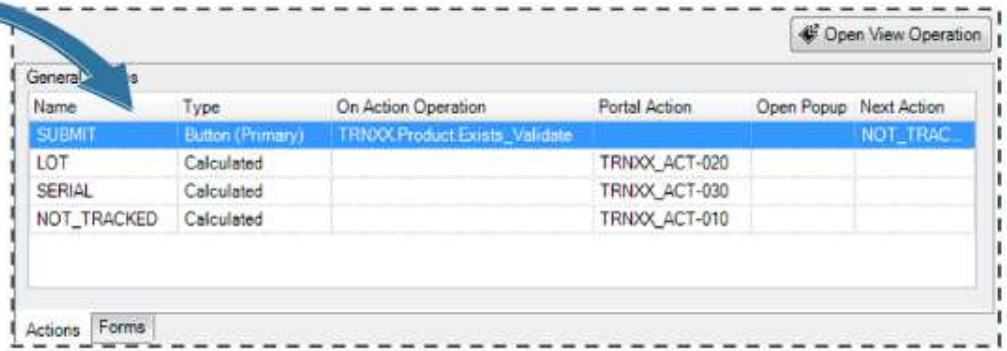


# LAB 6: Change Next Action

- Now go to the properties of the SUBMIT button, and select the **NOT_TRACKED** as the Next Action
- **Save** the View and change to **Prototype** status

With this configuration, when a product is not tracked, the user will see the TRNXX_ACT-010 Screen.



# LAB 6: Test Run TRNXX_ACT-010

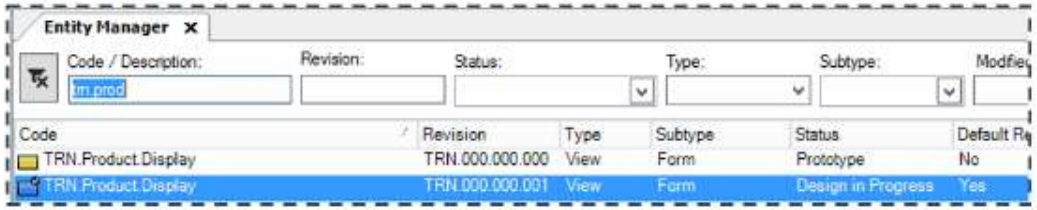
The TRNXX_ACT-010 Screen displays just the ProductNo.

With the current screen flow configuration, the reason to show this screen has changed. It is displayed if the product is not tracked.

You may want to change the TRNXX_ACT-010 Screen to adjust to this new context.

# LAB 6: Duplicate TRNXX.Product.Display to New Revision

- In **Entity Manager**, find the **TRNXX.Product.Display** View and duplicate to a new Revision
- Make the new Revision default
- **Go to** the View
- Go to the **Forms** tab, click on the **GROUP1**, and click on the ProductNo Control

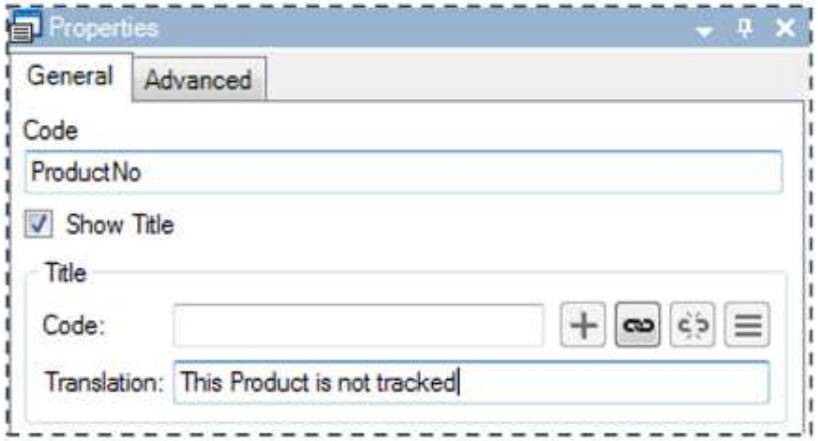



## LAB 6: Change ProductNo Details

- Go to the ProductNo properties
- Change the Translation to: **This Product is not tracked**

This will make the View still display the entered ProductNo, but with information that the product is not tracked.

- **Save** the View, and change to Prototype



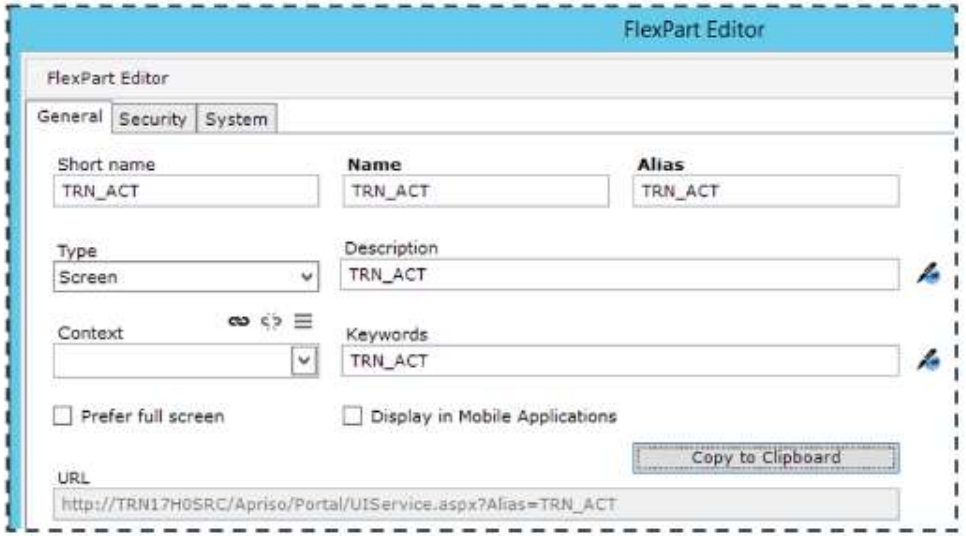
# LAB 6: Publish TRNXX_ACT as FlexPart

Publish the TRNXX_ACT Screen as a FlexPart.

- In TRNXX_ACT Properties, check the **Base Screen** box
- In **Entity Manager**, find the **TRNXX_ACT** Screen, right-click on it and select **Publish as FlexPart**

# LAB 6: Add Portal Admin Role

- In the popup, provide the name as **TRNXX_ACT**
- **Save** the FlexPart
- Copy the URL to clipboard
- Set security to **Portal Administrator** in the Security tab




# LAB 6: Test Run TRNXX_ACT in Web Browser

Test the complete solution, starting it from the URL in a browser.

- Lot-tracked product: **B-BABY-001**
- Serial-tracked product: **A-PIST-001**
- Not tracked product: **G-OILS-001**

The first 2 products will work like in the previous lab, and when you use the **G-OILS-001**, you will be taken to a Screen which shows the message „This Product is not tracked: G-OILS-001".