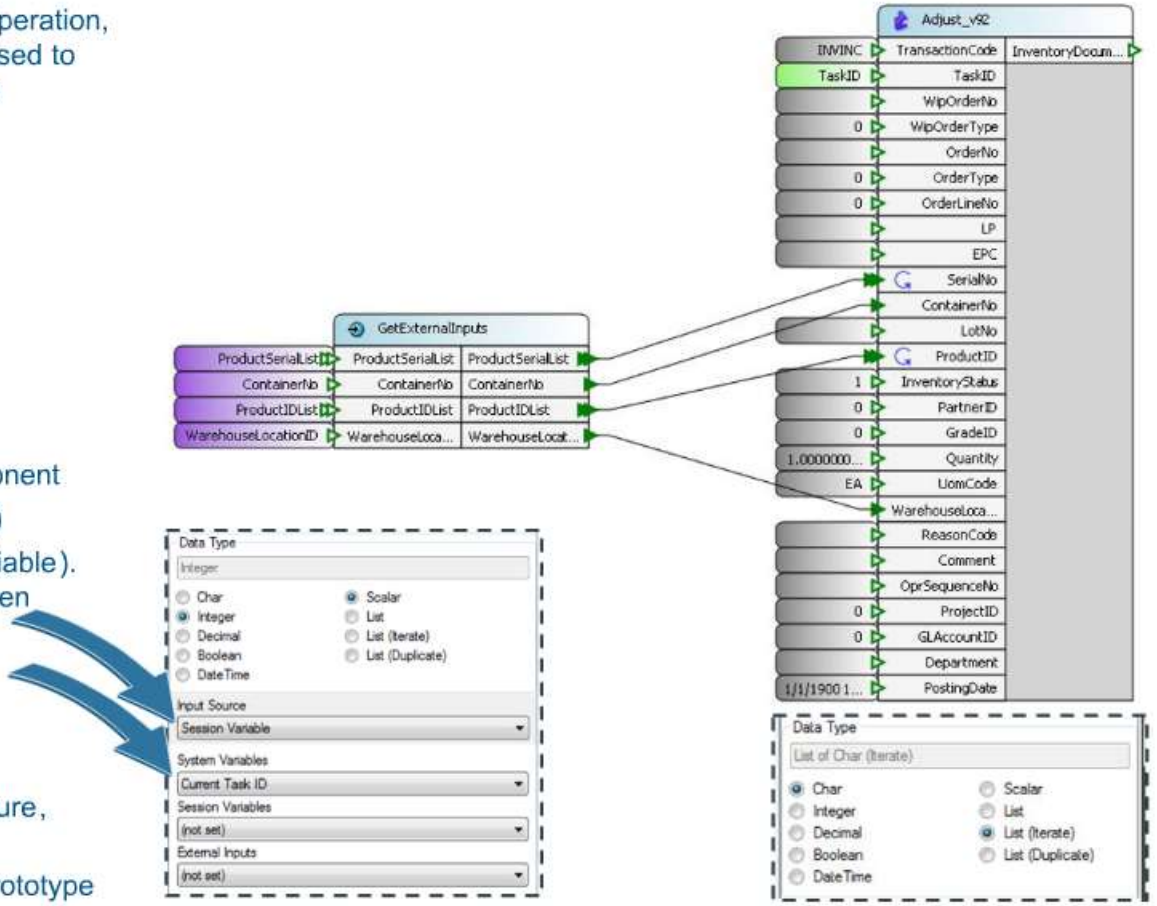
LAB 2: Create AdjustProductSerials Operation

- Create TRNXX.STP.AdjustProductSerials Operation, make sure the Subtype is Action. It will be used to move the Container with the Products to the selected warehouse location
- Create a GetExternalInputs function:
  - Add 4 pairs of inputs/outputs
  - ProductSerialList (List of Char)
  - ContainerNo (Char)
  - ProductIDList (List of Integer)
  - WarehouseLocationID (Integer)
- Add a Business Component function:
  - Link the Adjust_v92 Business Component
  - TransactionCode: INVINC (Constant)
  - TaskID: Current Task ID (System Variable). HINT: Select Session Variable and then select System Variable
  - InventoryStatus: 1 (Constant)
  - Quantity: 1 (Constant)
  - UomCode: EA (Constant)
- Connect both functions as shown in the picture, SerialNo and ProductID are List (Iterate)
- Save the Operation and change status to Prototype



LAB 2: Adjust_v92 Business Component

This Business Component is used to generate non-balanced inventory movements, i.e. inventory increase or decrease.

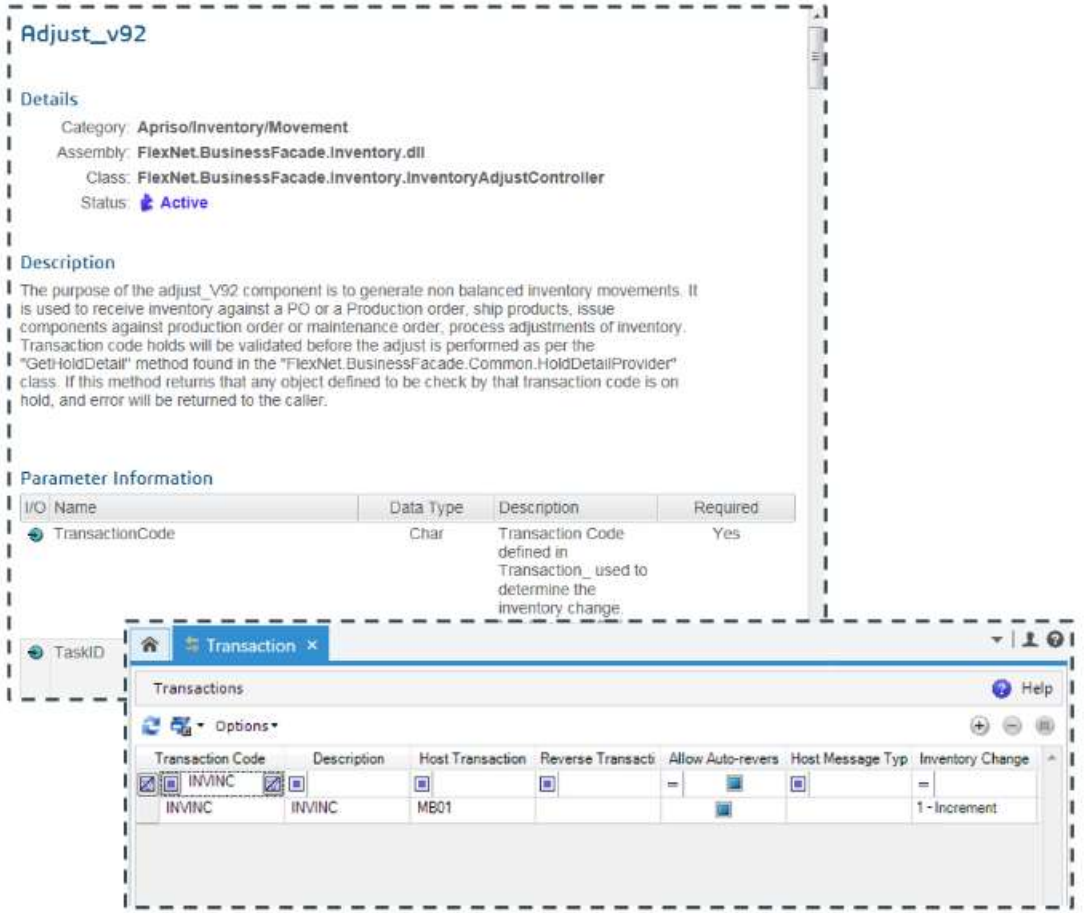
It is used to receive inventory against a Production Order, ship products, issue components against a Production or Maintenance Order, and process inventory adjustments.

You are using it here, because the 2 products with their serial numbers have just been produced and need to be added to inventory.

The TransactionCode input to this BC is very important, because it tells the system what kind of inventory adjustment to do (Increment, Decrement, or Move).

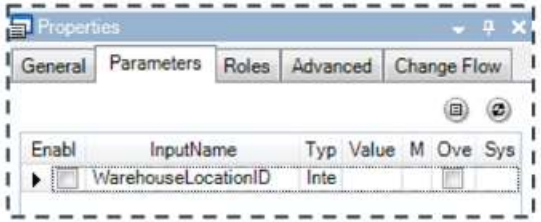
The transactions are maintained through the Transactions Screen in the DELMIA Apriso portal.

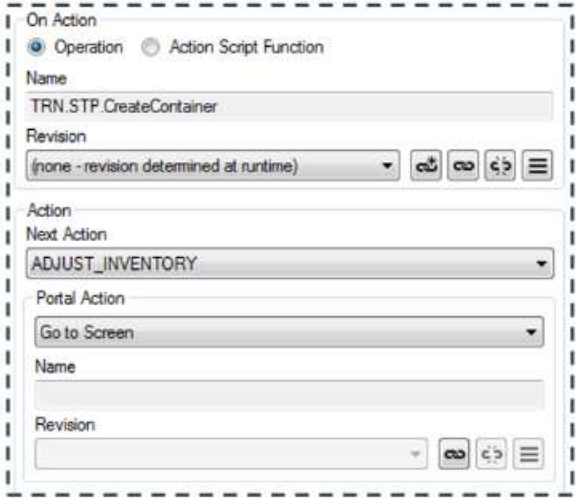
You can learn more about the Transaction properties in the Database Documentation, when you look there for the TRANSACTION_ table.



LAB 2: Configure TRNXX_STP-010 Screen

- In Screen TRNXX_STP-010, go to Properties, Parameters tab, and refresh parameters so that you can see the WarehouseLocationID input
- Now provide the navigation to the two operations we just created
- In the TRNXX_SelectWarehouse view, go to the Properties for the MOVE button
- For the On Action Operation, link TRNXX.STP.CreateContainer
- Under Action, set Next Action to ADJUST_INVENTORY
- Now go to the ADJUST_INVENTORY properties
- For the On Action Operation, link TRNXX.STP.AdjustProductSerials
- Save the Screen, the View, and the Operations, and make sure they are all in Prototype status.

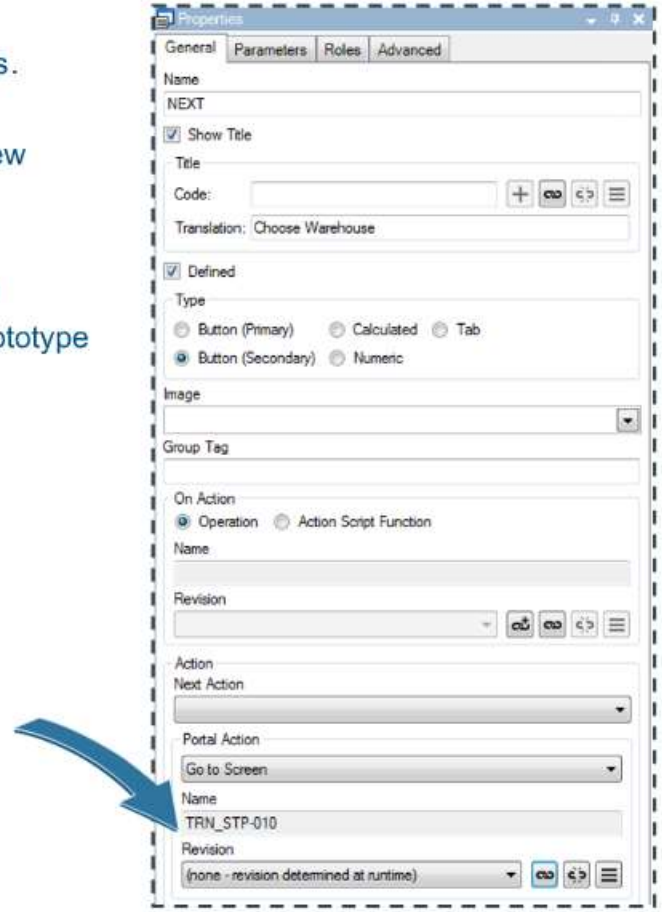




LAB 2: Provide the Navigation Between Screens

Provide the navigation between the TRNXX_STP and TRNXX_STP-010 Screens.

- Go to the TRNXX.STP.ProductSerial View
- Select the NEXT Action
- Set the Portal Action to Go to Screen
- Then Link the TRNXX_STP-010 Screen
- Save the View and make sure it is in Prototype



LAB 3: Show Final Result Screen

Task:

- Build an overview Screen which will confirm the results of the complete flow

What you will learn:

- How to build a Screen that consolidates results from other Screens

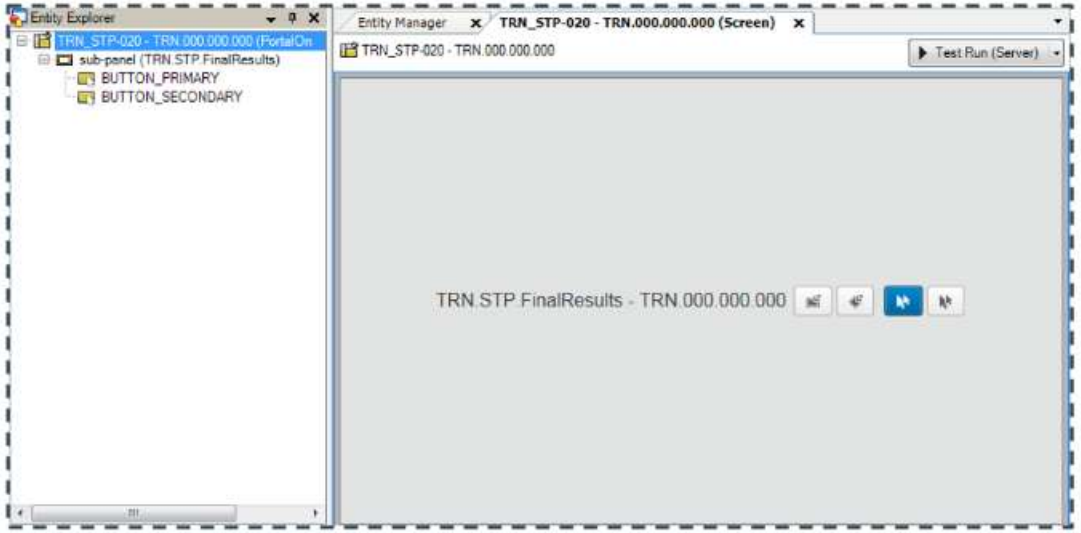
Training environment:

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

15 min

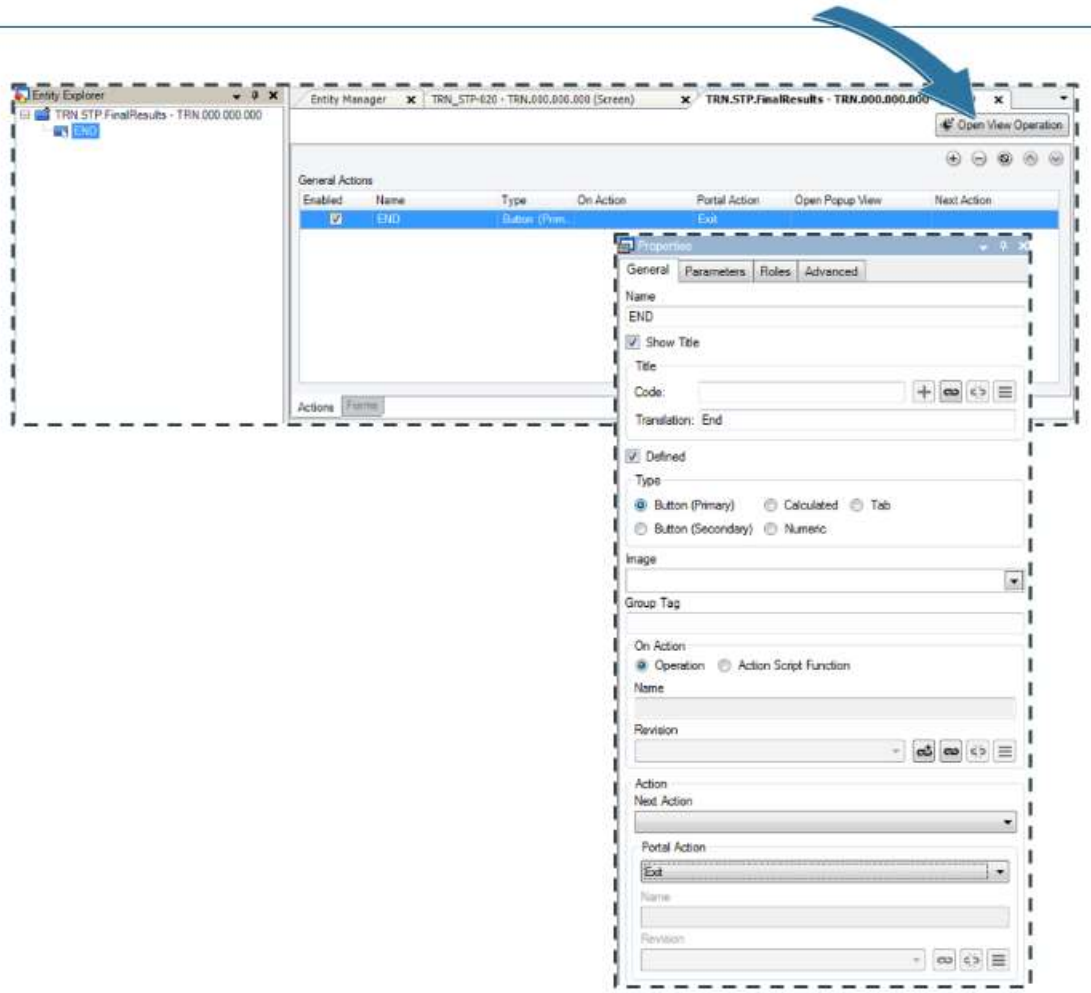
LAB 3: Create TRNXX_STP-020 Screen

- Create screen TRNXX_STP-020
- Choose PortalOneWindow layout
- In the sub-panel, copy and link PortalSimple View, rename as TRNXX.STP.FinalResults
- Save the Screen and change status to Prototype



LAB 3: Change View Actions

- Open the TRNXX.STP.FinalResults View
- Remove BUTTON_SECONDARY Action
- Edit the BUTTON_PRIMARY Action
  - Name: End
  - Show Title: Checked
  - Translation: End
  - Action / Portal Action: Exit
- Save View and change status to Prototype
- Open View Operation



LAB 3: Add New Function

- The Operation will have 2 default functions, Call_PortalGenerateButtonList and ScreenInterface
- Create a GetExternalInputs function
- Add 4 Inputs/Outputs
  - ProductIDList (List of Integer)
  - ProductSerialList (List of Char)
  - ContainerNo (Char)
  - WarehouseLocationID (Integer)

LAB 3: Add SQL Query Functions

- Add a SQL Query function, name it GetProductNo
- Enter the Query from scripts:

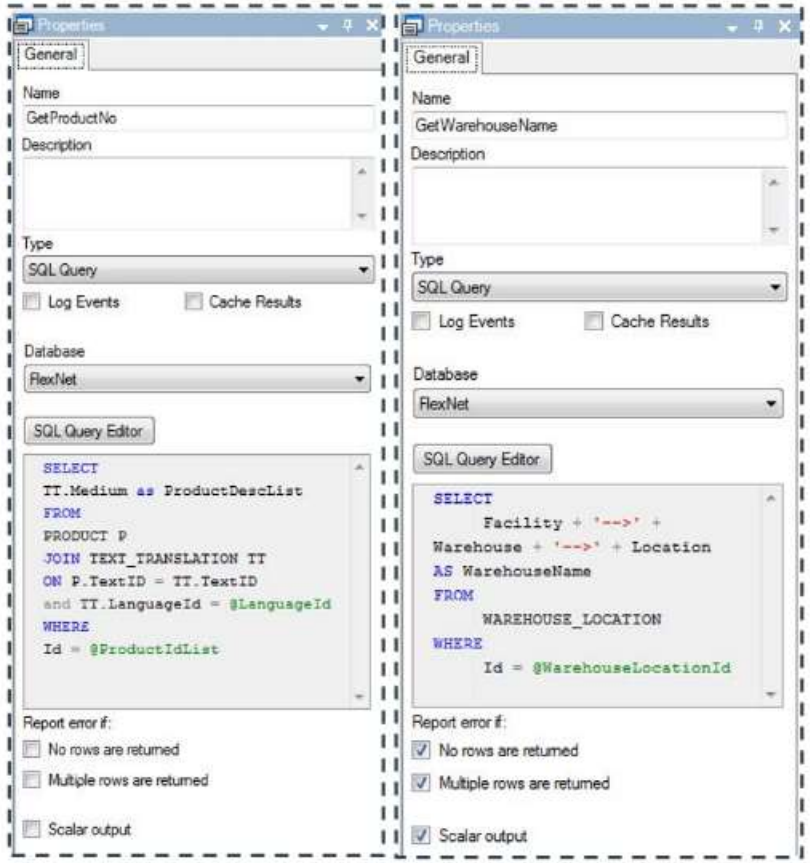
SCRIPT FILE: Desktop/Training Materials/Level 1

- Link the LanguageID input to the System Variable LanguageID
- Make sure to iterate the ProductIdList input

- Add another SQL Query function, name it GetWarehouseName
- Enter the SQL Query from scripts:

SCRIPT FILE: Desktop/Training Materials/Level 1

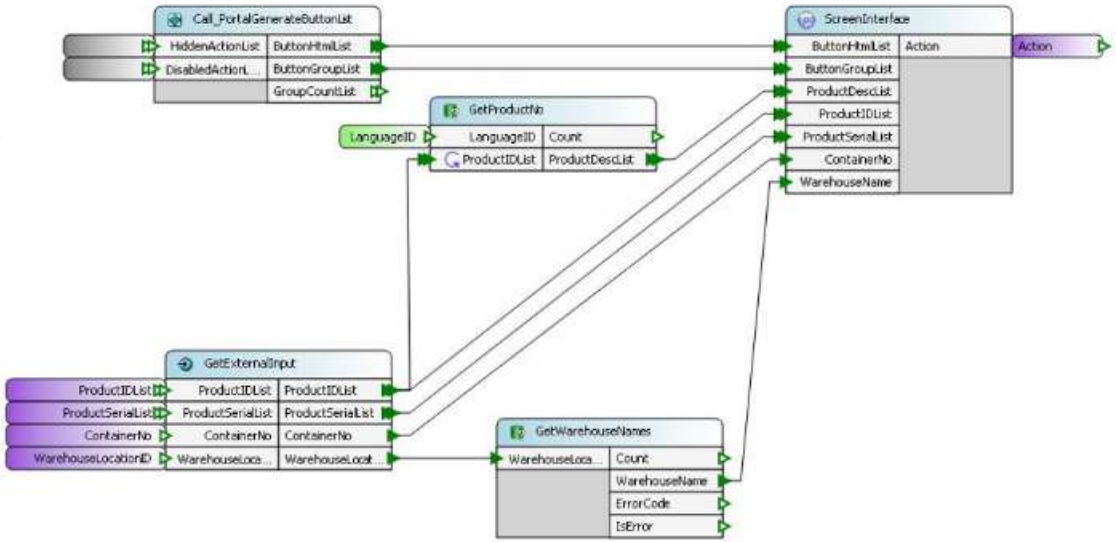
- In Properties, set for Scalar Output



LAB 3: Add Connections Between Functions

- Drag the following outputs to the input side of the Screen Interface function
  - ProductDescList (From GetProductNo)
  - ProductIDList (From GetExternalInputs)
  - ProductSerialList (From GetExternalInputs)
  - ContainerNo (From GetExternalInputs)
  - WarehouseName (From GetWarehouseName)
- Connect the ProductIDList output from GetExternalOutputs to GetProductNo
- Connect the WarehouseLocationID output from GetExternalOutputs to GetWarehouseName
- Add External Inputs to GetExternalInputs function

Your workspace should now look similar to this on the picture.



LAB 3: Make Changes in HTML Layout Editor

- Switch to the HTML Layout Editor and paste the query from scripts (it should match the picture):

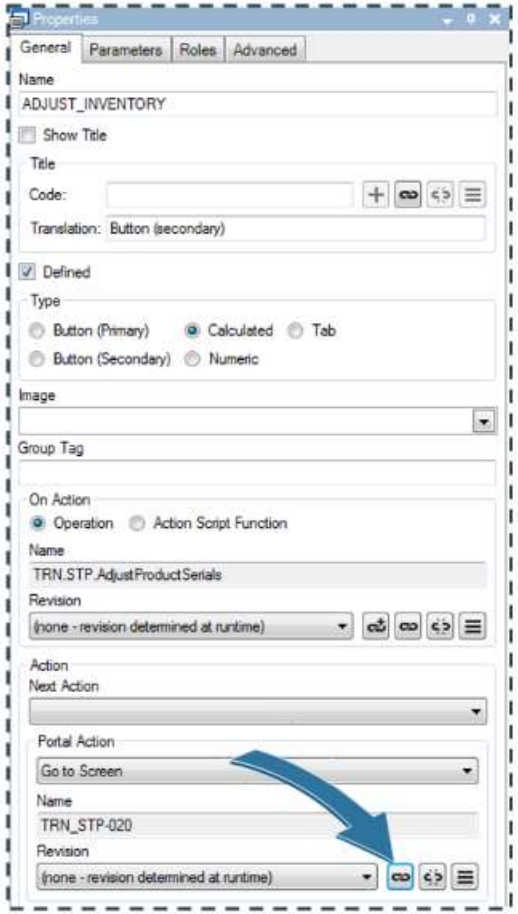
SCRIPT FILE: Desktop/Training Materials/Level 1

- Save the View Operation and set to Prototype

LAB 3: Provide the Navigation Between Screens

Provide the navigation between the TRNXX_STP-010 and TRNXX_STP-020 Screens.

- Go to the TRNXX.STP.SelectWarehouse View
- Select the ADJUST_INVENTORY Action
- Make sure that Portal Action is set to Go to Screen
- Then link the Screen TRNXX_STP-020
- Save the View and make sure it is in Prototype



LAB 3: Test Run TRNXX_STP

- Test run the TRNXX_STP Screen
- Choose one Product and click Add Serial
  - The screen refreshes, but stays the same
- Choose Warehouse Location, click the warehouse you would like to use, and click Move Container
- View the Overview
- Click the End button - it will close the test run

LAB 4 (Voluntary): Show Which Serial Are Created

If you like, you may try what you learned in the preceeding labs to add one more element in this screen flow.

- In the TRNXX_STP Screen, you are using the Add Serial button to create serials, but you are not seeing what is happening
- Use the skills you learned in LAB 3 to display serials on the TRNXX_STP Screen as you create them

Summary

In this training module you have created a simplified serial creation and warehouse movements flow.

You have used Business Components, which made your work quick and easy as compared to using programming to create the serial numbers, container, and the warehouse movements.

The DELMIA Apriso system has over 400 Business Components, and they are at the core of each business logic in every customer project.

Remember the Business Component Documentation is available in Process Builder. It will help you use the BCs in the best way possible.