

LAB 1: Sequence Maintenance

LAB 1: Add CreateSerial Business Component

LAB 1: Modify CreateProductSerial Operation

LAB 1: Add a User Formula Function

LAB 1: Connect the Operation to the Button

LAB 1: Test Run TRNXX_STP Screen

LAB 1: Test Mode Panel

End of LAB 1

LAB 2: Create a Screen to Assign Products to a Warehouse Location

LAB 2: Create a Screen to Assign Products to a Warehouse Location

LAB 2: Create Warehouse Locations List

LAB 2: Create TRNXX_STP-010 Screen

LAB 2: Copy and Link a View

LAB 2: Change View Actions

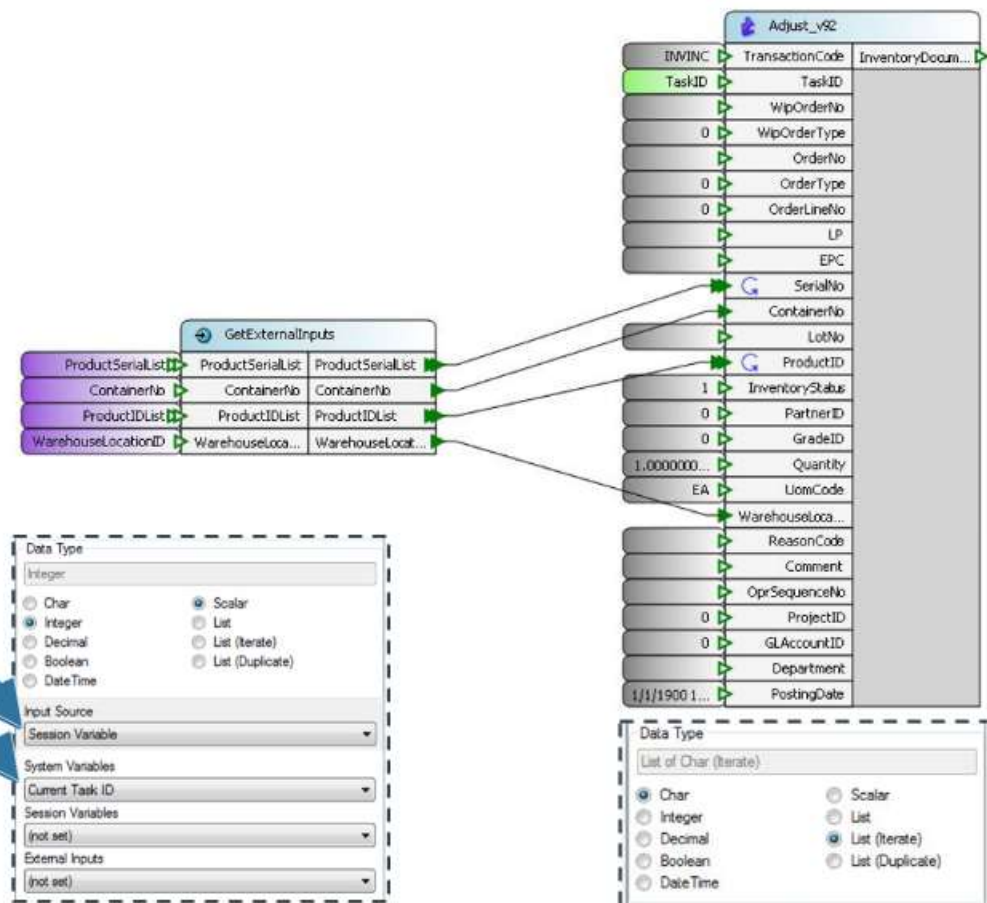
LAB 2: Create TRNXX_STP.CreateContainer Operation

LAB 2: CreateContainer Business Component

LAB 2: Create AdjustProductSerials Operation

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



Search...

LAB 1: Modify CreateProduction Operation

LAB 1: Add a User Formula Function

LAB 1: Connect the Operation to the Button

LAB 1: Test Run TRNXX_STP Screen

LAB 1: Test Mode Panel

End of LAB 1

LAB 2: Create a Screen to Assign Products to a Warehouse Location

LAB 2: Create a Screen to Assign Products to a Warehouse Location

LAB 2: Create Warehouse Locations List

LAB 2: Create TRNXX_STP-010 Screen

LAB 2: Copy and Link a View

LAB 2: Change View Actions

LAB 2: Create TRNXX_STP.CreateContainer Operation

LAB 2: CreateContainer Business Component

LAB 2: Create AdjustProductSerials Operation

LAB 2: Adjust_v92 Business Component

LAB 2: Configure TRNXX_STP-010 Screen

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.

Adjust_v92

Details

Category: **AprisoInventoryMovement**
 Assembly: **FlexNet.BusinessFacade.Inventory.dll**
 Class: **FlexNet.BusinessFacade.Inventory.InventoryAdjustController**
 Status: **Active**

Description

The purpose of the adjust_v92 component is to generate non balanced inventory movements. It is used to receive inventory against a PO or a Production order, ship products, issue components against production order or maintenance order, process adjustments of inventory. Transaction code holds will be validated before the adjust is performed as per the "GetHoldDetail" method found in the "FlexNet.BusinessFacade.Common.HoldDetailProvider" class. If this method returns that any object defined to be check by that transaction code is on hold, and error will be returned to the caller.

Parameter Information

I/O	Name	Data Type	Description	Required
+	TransactionCode	Char	Transaction Code defined in Transaction_ used to determine the inventory change	Yes

TaskID

Transaction

Transactions

Options

Transaction Code	Description	Host Transaction	Reverse Transacti	Allow Auto-revers	Host Message Typ	Inventory Change
INVINC	INVINC	MB01		-		1-Increment

LAB 1: Modify CreateProductSerial
Operation

LAB 1: Add a User Formula
Function

LAB 1: Connect the Operation to
the Button

LAB 1: Test Run TRNXX_STP Screen

LAB 1: Test Mode Panel

End of LAB 1

LAB 2: Create a Screen to Assign
Products to a Warehouse Location

LAB 2: Create a Screen to Assign
Products to a Warehouse Location

LAB 2: Create Warehouse Locations
List

LAB 2: Create TRNXX_STP-010
Screen

LAB 2: Copy and Link a View

LAB 2: Change View Actions

LAB 2: Create
TRNXX.STP.CreateContainer
Operation

LAB 2: CreateContainer Business
Component

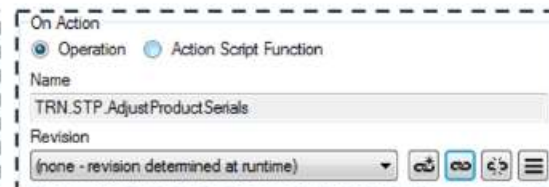
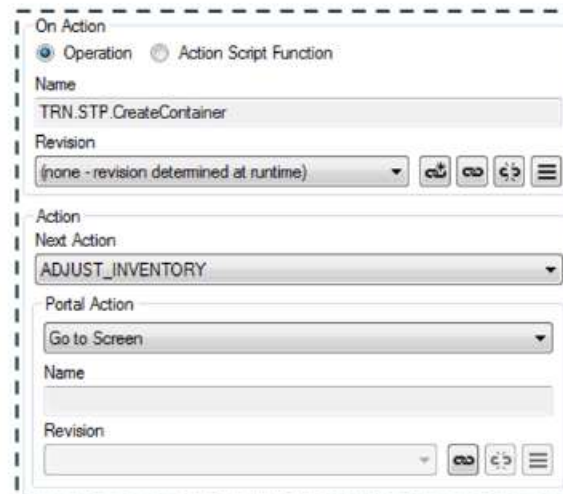
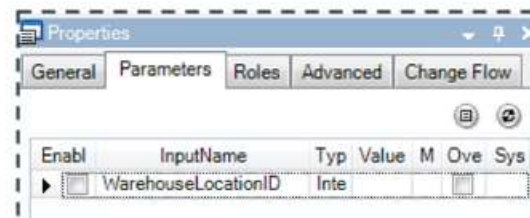
LAB 2: Create AdjustProductSerials
Operation

LAB 2: Adjust_v92 Business
Component

LAB 2: Configure TRNXX_STP-010
Screen

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 1: Add a User, Formula Function

LAB 1: Connect the Operation to the Button

LAB 1: Test Run TRNXX_STP Screen

LAB 1: Test Mode Panel

End of LAB 1

LAB 2: Create a Screen to Assign Products to a Warehouse Location

LAB 2: Create a Screen to Assign Products to a Warehouse Location

LAB 2: Create Warehouse Locations List

LAB 2: Create TRNXX_STP-010 Screen

LAB 2: Copy and Link a View

LAB 2: Change View Actions

LAB 2: Create TRNXX_STP.CreateContainer Operation

LAB 2: CreateContainer Business Component

LAB 2: Create AdjustProductSerials Operation

LAB 2: Adjust_v92 Business Component

LAB 2: Configure TRNXX_STP-010 Screen

LAB 2: Provide the Navigation Between Screens

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

Properties

General Parameters Roles Advanced

Name
NEXT

☒ Show Title

Title

Code:

Translation: Choose Warehouse

☒ Defined

Type

☐ Button (Primary) ☐ Calculated ☐ Tab

☒ Button (Secondary) ☐ Numeric

Image

Group Tag

On Action

☒ Operation ☐ Action Script Function

Name

Revision

Action

Next Action

Portal Action

Go to Screen

Name

TRN_STP-010

Revision

(none - revision determined at runtime)





LAB 1: Test Run TRNXX_STP Screen

LAB 1: Test Mode Panel

End of LAB 1

LAB 2: Create a Screen to Assign
Products to a Warehouse LocationLAB 2: Create a Screen to Assign
Products to a Warehouse LocationLAB 2: Create Warehouse Locations
ListLAB 2: Create TRNXX_STP-010
Screen

LAB 2: Copy and Link a View

LAB 2: Change View Actions

LAB 2: Create
TRNXX_STP>CreateContainer
OperationLAB 2: CreateContainer Business
ComponentLAB 2: Create AdjustProductSerials
OperationLAB 2: Adjust_v92 Business
ComponentLAB 2: Configure TRNXX_STP-010
ScreenLAB 2: Provide the Navigation
Between Screens

End of LAB 2

LAB 3: Show Final Result Screen

LAB 3: Show Final Result Screen

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

Overview:

ProductDescription	Serial
INNER SWASHPLATE RINGS	11070
LICENSE PLATE	S11071

Container: L10

WarehouseLocation: C1P1-->PRD-->PRDFLR04

End





End of LAB 1

LAB 2: Create a Screen to Assign Products to a Warehouse Location

LAB 2: Create a Screen to Assign Products to a Warehouse Location

LAB 2: Create Warehouse Locations List

LAB 2: Create TRNXX_STP-010 Screen

LAB 2: Copy and Link a View

LAB 2: Change View Actions

LAB 2: Create TRNXX_STP.CreateContainer Operation

LAB 2: CreateContainer Business Component

LAB 2: Create AdjustProductSerials Operation

LAB 2: Adjust_v92 Business Component

LAB 2: Configure TRNXX_STP-010 Screen

LAB 2: Provide the Navigation Between Screens

End of LAB 2

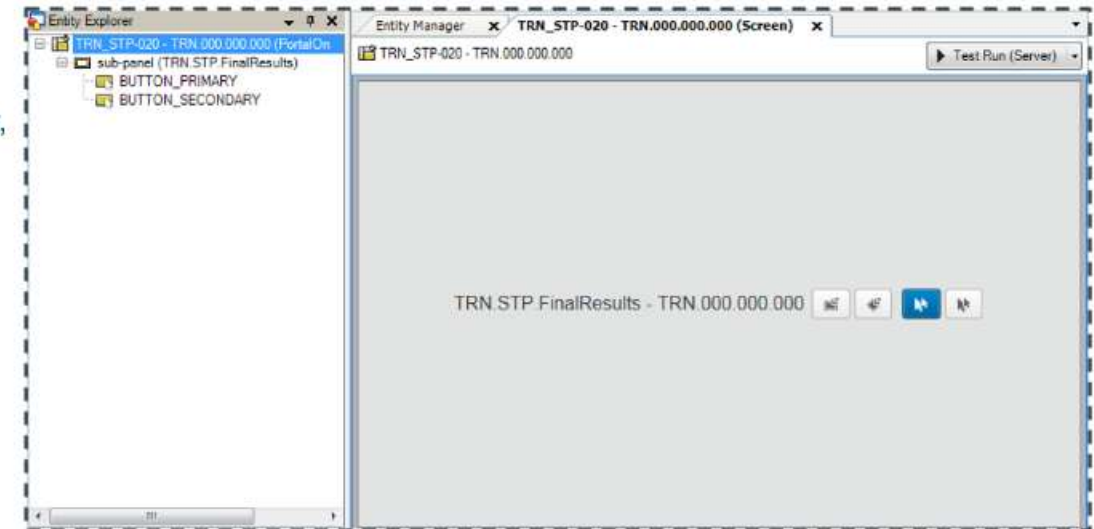
LAB 3: Show Final Result Screen

LAB 3: Show Final Result Screen

LAB 3: Create TRNXX_STP-020 Screen

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 2: Create a Screen to Assign Products to a Warehouse Location

LAB 2: Create a Screen to Assign Products to a Warehouse Location

LAB 2: Create Warehouse Locations List

LAB 2: Create TRNXX_STP-010 Screen

LAB 2: Copy and Link a View

LAB 2: Change View Actions

LAB 2: Create TRNXX_STP.CreateContainer Operation

LAB 2: CreateContainer Business Component

LAB 2: Create AdjustProductSerials Operation

LAB 2: Adjust_v92 Business Component

LAB 2: Configure TRNXX_STP-010 Screen

LAB 2: Provide the Navigation Between Screens

End of LAB 2

LAB 3: Show Final Result Screen

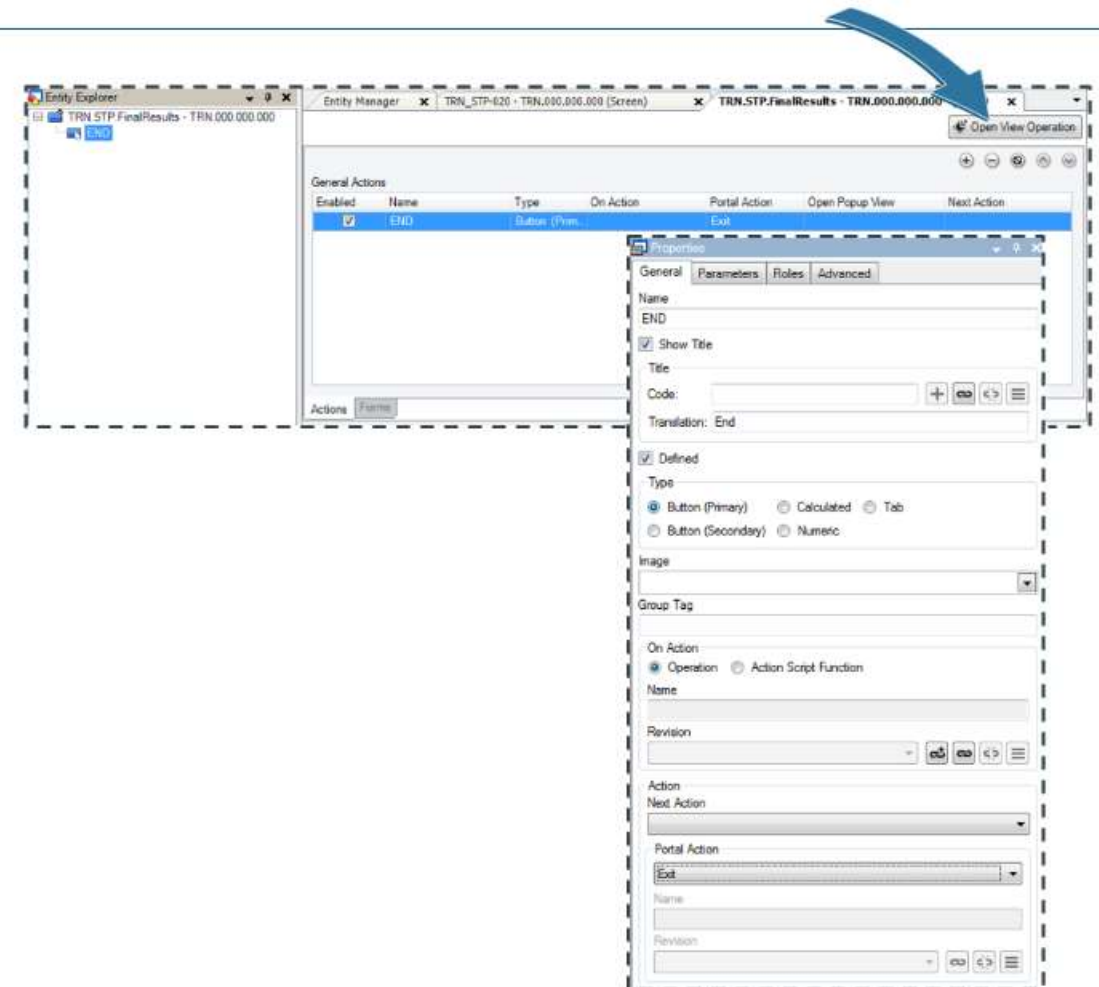
LAB 3: Show Final Result Screen

LAB 3: Create TRNXX_STP-020 Screen

LAB 3: Change View Actions

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



Component

LAB 2: Configure TRNXX_STP-010
ScreenLAB 2: Provide the Navigation
Between Screens

End of LAB 2

LAB 3: Show Final Result Screen

LAB 3: Show Final Result Screen

LAB 3: Create TRNXX_STP-020
Screen

LAB 3: Change View Actions

LAB 3: Add New Function

LAB 3: Add SQL Query Functions

LAB 3: Add Connections Between
FunctionsLAB 3: Make Changes in HTML
Layout EditorLAB 3: Provide the Navigation
Between Screens

LAB 3: Test Run TRNXX_STP

End of LAB 3

LAB 4 (Voluntary): Show Which
Serial Are CreatedLAB 4 (Voluntary): Show Which
Serial Are Created

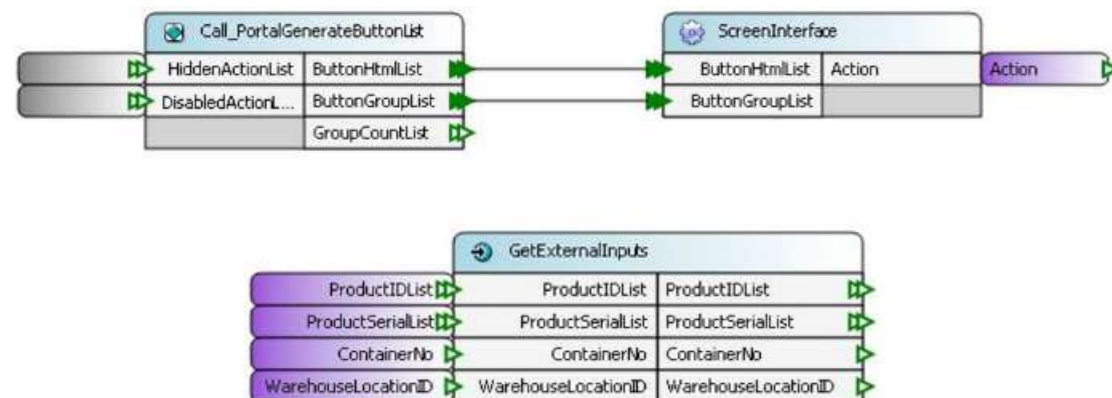
End of LAB 4 (Voluntary):

Summary

End of Course

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)



Component

LAB 2: Configure TRNXX_STP-010 Screen

LAB 2: Provide the Navigation Between Screens

End of LAB 2

LAB 3: Show Final Result Screen

LAB 3: Show Final Result Screen

LAB 3: Create TRNXX_STP-020 Screen

LAB 3: Change View Actions

LAB 3: Add New Function

LAB 3: Add SQL Query Functions

LAB 3: Add Connections Between Functions

LAB 3: Make Changes in HTML Layout Editor

LAB 3: Provide the Navigation Between Screens

LAB 3: Test Run TRNXX_STP

End of LAB 3

LAB 4 (Voluntary): Show Which Serial Are Created

LAB 4 (Voluntary): Show Which Serial Are Created

End of LAB 4 (Voluntary):

Summary

End of Course

LAB 3: Add SQL Query Functions

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

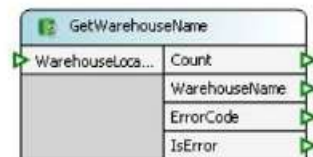
SCRIPT FILE: Desktop/Training Materials/Level

- ▶ 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



GetProductNo Properties

Name: GetProductNo

Description:

Type: SQL Query

Log Events: ☐ Cache Results: ☐

Database: FlexNet

SQL Query Editor:

```
SELECT
  TT.Medium as ProductDescList
FROM
  PRODUCT P
JOIN TEXT_TRANSLATION TT
ON P.TextID = TT.TextID
and TT.LanguageId = @LanguageId
WHERE
  Id = @ProductIdList
```

Report error if:

☐ No rows are returned

☐ Multiple rows are returned

☐ Scalar output

GetWarehouseName Properties

Name: GetWarehouseName

Description:

Type: SQL Query

Log Events: ☐ Cache Results: ☐

Database: FlexNet

SQL Query Editor:

```
SELECT
  Facility + '--->' +
  Warehouse + '--->' + Location
AS WarehouseName
FROM
  WAREHOUSE_LOCATION
WHERE
  Id = @WarehouseLocationId
```

Report error if:

☒ No rows are returned

☒ Multiple rows are returned

☒ Scalar output

Component

LAB 2: Configure TRNXX_STP-010 Screen

LAB 2: Provide the Navigation Between Screens

End of LAB 2

LAB 3: Show Final Result Screen

LAB 3: Show Final Result Screen

LAB 3: Create TRNXX_STP-020 Screen

LAB 3: Change View Actions

LAB 3: Add New Function

LAB 3: Add SQL Query Functions

LAB 3: Add Connections Between Functions

LAB 3: Make Changes in HTML Layout Editor

LAB 3: Provide the Navigation Between Screens

LAB 3: Test Run TRNXX_STP

End of LAB 3

LAB 4 (Voluntary): Show Which Serial Are Created

LAB 4 (Voluntary): Show Which Serial Are Created

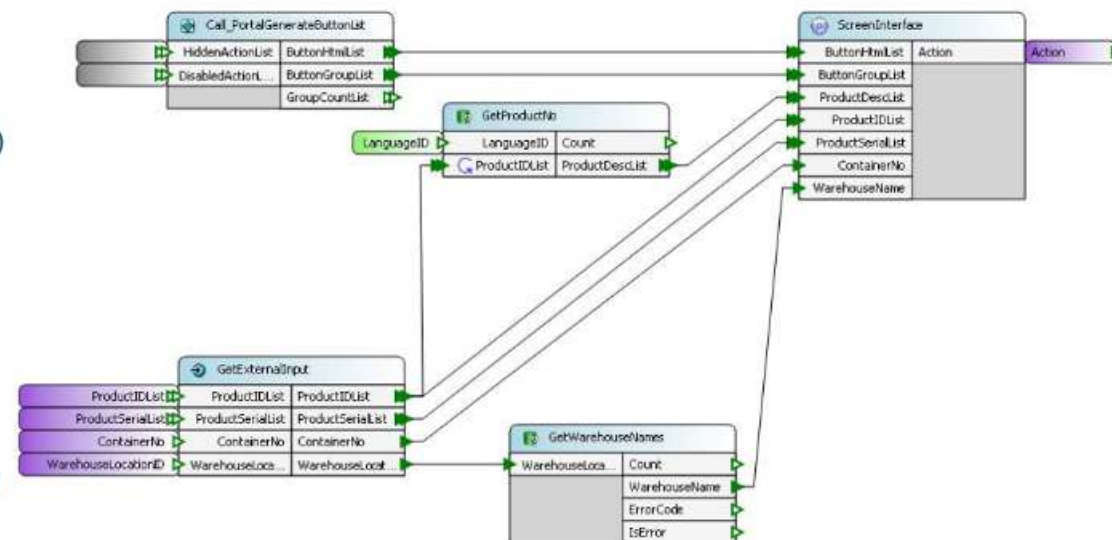
End of LAB 4 (Voluntary):

Summary

End of Course

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.

Component

LAB 2: Configure TRNXX_STP-010
ScreenLAB 2: Provide the Navigation
Between Screens

End of LAB 2

LAB 3: Show Final Result Screen

LAB 3: Show Final Result Screen

LAB 3: Create TRNXX_STP-020
Screen

LAB 3: Change View Actions

LAB 3: Add New Function

LAB 3: Add SQL Query Functions

LAB 3: Add Connections Between
FunctionsLAB 3: Make Changes in HTML
Layout EditorLAB 3: Provide the Navigation
Between Screens

LAB 3: Test Run TRNXX_STP

End of LAB 3

LAB 4 (Voluntary): Show Which
Serial Are CreatedLAB 4 (Voluntary): Show Which
Serial Are Created

End of LAB 4 (Voluntary):

Summary

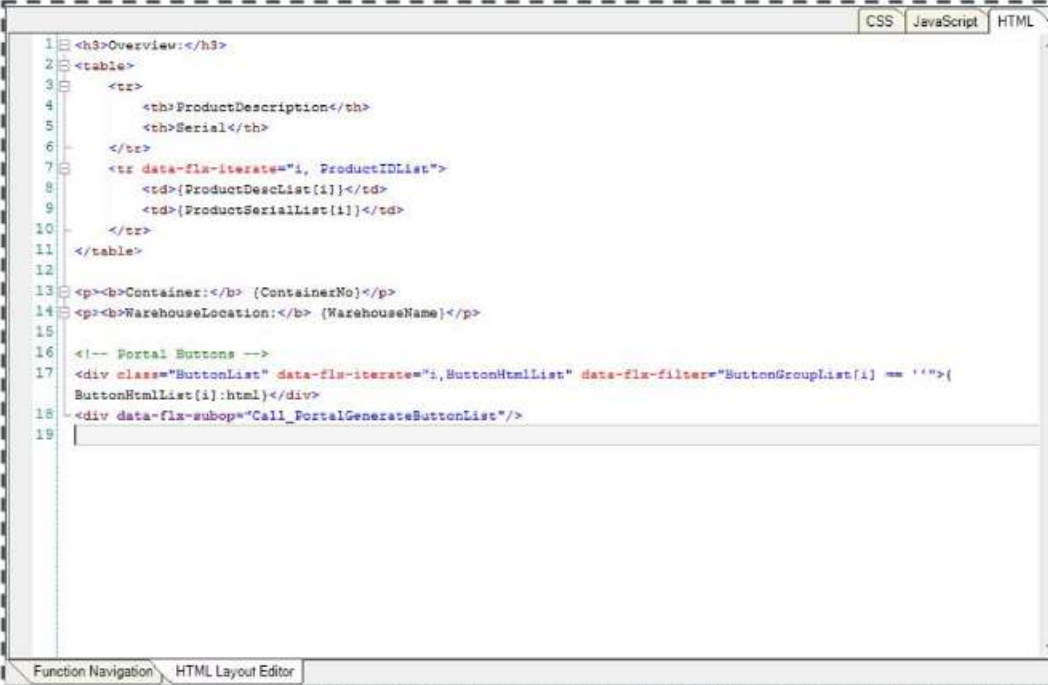
End of Course

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



```
1 <h3>Overview:</h3>
2 <table>
3   <tr>
4     <th>ProductDescription</th>
5     <th>Serial</th>
6   </tr>
7   <tr data-flx-iterate="i, ProductIDList">
8     <td>{ProductDescList[i]}</td>
9     <td>{ProductSerialList[i]}</td>
10  </tr>
11 </table>
12
13 <p><b>Container:</b> {ContainerNo}</p>
14 <p><b>WarehouseLocation:</b> {WarehouseName}</p>
15
16 <!-- Portal Buttons -->
17 <div class="ButtonList" data-flx-iterate="i,ButtonHtmlList" data-flx-filter="ButtonGroupList[i] == ''">
18   {ButtonHtmlList[i]:html}</div>
19 <div data-flx-subop="Call_PortalGenerateButtonList"/>
```


Component

LAB 2: Configure TRNXX_STP-010
ScreenLAB 2: Provide the Navigation
Between Screens

End of LAB 2

LAB 3: Show Final Result Screen

LAB 3: Show Final Result Screen

LAB 3: Create TRNXX_STP-020
Screen

LAB 3: Change View Actions

LAB 3: Add New Function

LAB 3: Add SQL Query Functions

LAB 3: Add Connections Between
FunctionsLAB 3: Make Changes in HTML
Layout EditorLAB 3: Provide the Navigation
Between Screens

LAB 3: Test Run TRNXX_STP

End of LAB 3

LAB 4 (Voluntary): Show Which
Serial Are CreatedLAB 4 (Voluntary): Show Which
Serial Are Created

End of LAB 4 (Voluntary):

Summary

End of Course

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

The screenshot shows the 'Properties' window for the 'ADJUST_INVENTORY' action. The 'General' tab is selected. The 'Name' field is 'ADJUST_INVENTORY'. The 'Show Title' checkbox is unchecked. The 'Title' field is empty. The 'Code' field is empty, and the 'Translation' field is 'Button (secondary)'. The 'Defined' checkbox is checked. The 'Type' section has 'Calculated' selected. The 'Image' field is empty. The 'Group Tag' field is empty. The 'On Action' section has 'Operation' selected. The 'Name' field is 'TRN.STP.AdjustProductSerials'. The 'Revision' field is '(none - revision determined at runtime)'. The 'Action' section has 'Next Action' set to a dropdown menu. The 'Portal Action' section has 'Go to Screen' selected. The 'Name' field is 'TRN_STP-020'. The 'Revision' field is '(none - revision determined at runtime)'. A blue arrow points from the 'Go to Screen' dropdown to the 'Name' field.



Component

LAB 2: Configure TRNXX_STP-010
ScreenLAB 2: Provide the Navigation
Between Screens

End of LAB 2

LAB 3: Show Final Result Screen

LAB 3: Show Final Result Screen

LAB 3: Create TRNXX_STP-020
Screen

LAB 3: Change View Actions

LAB 3: Add New Function

LAB 3: Add SQL Query Functions

LAB 3: Add Connections Between
FunctionsLAB 3: Make Changes in HTML
Layout EditorLAB 3: Provide the Navigation
Between Screens

LAB 3: Test Run TRNXX_STP

End of LAB 3

LAB 4 (Voluntary): Show Which
Serial Are CreatedLAB 4 (Voluntary): Show Which
Serial Are Created

End of LAB 4 (Voluntary):

Summary

End of Course

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

Overview:

ProductDescription	Serial
INNER SWASHPLATE RINGS	S11070
LICENSE PLATE	S11071

Container: L10

WarehouseLocation: C1P1-->PRD-->PRDFLR04

End





Component

LAB 2: Configure TRNXX_STP-010
ScreenLAB 2: Provide the Navigation
Between Screens

End of LAB 2

LAB 3: Show Final Result Screen

LAB 3: Show Final Result Screen

LAB 3: Create TRNXX_STP-020
Screen

LAB 3: Change View Actions

LAB 3: Add New Function

LAB 3: Add SQL Query Functions

LAB 3: Add Connections Between
FunctionsLAB 3: Make Changes in HTML
Layout EditorLAB 3: Provide the Navigation
Between Screens

LAB 3: Test Run TRNXX_STP

End of LAB 3

LAB 4 (Voluntary): Show Which
Serial Are CreatedLAB 4 (Voluntary): Show Which
Serial Are Created

End of LAB 4 (Voluntary);

Summary

End of Course

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

Product:

Added Serials:
ProductDescriptionSerial



Search... 

Component

LAB 2: Configure TRNXX_STP-010
Screen

LAB 2: Provide the Navigation
Between Screens

End of LAB 2

LAB 3: Show Final Result Screen

LAB 3: Show Final Result Screen

LAB 3: Create TRNXX_STP-020
Screen

LAB 3: Change View Actions

LAB 3: Add New Function

LAB 3: Add SQL Query Functions

LAB 3: Add Connections Between
Functions

LAB 3: Make Changes in HTML
Layout Editor

LAB 3: Provide the Navigation
Between Screens

LAB 3: Test Run TRNXX_STP

End of LAB 3

LAB 4 (Voluntary): Show Which
Serial Are Created

LAB 4 (Voluntary): Show Which
Serial Are Created

End of LAB 4 (Voluntary):

Summary

End of Course

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.

