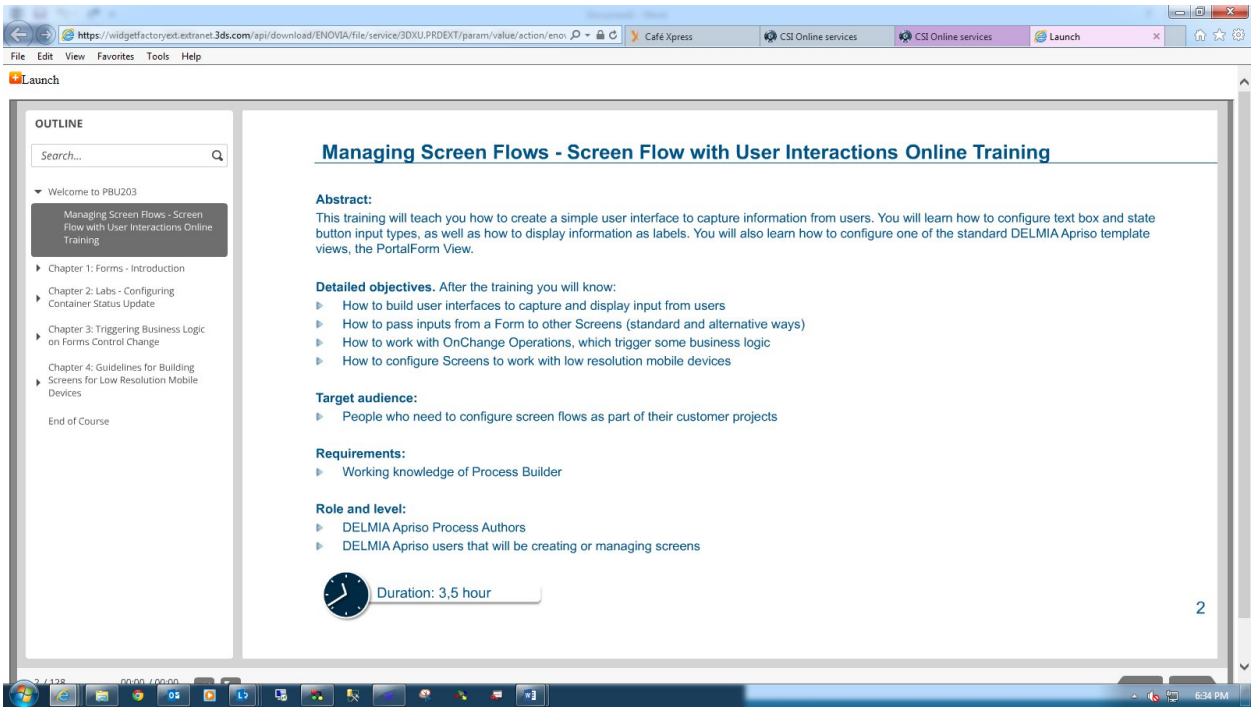
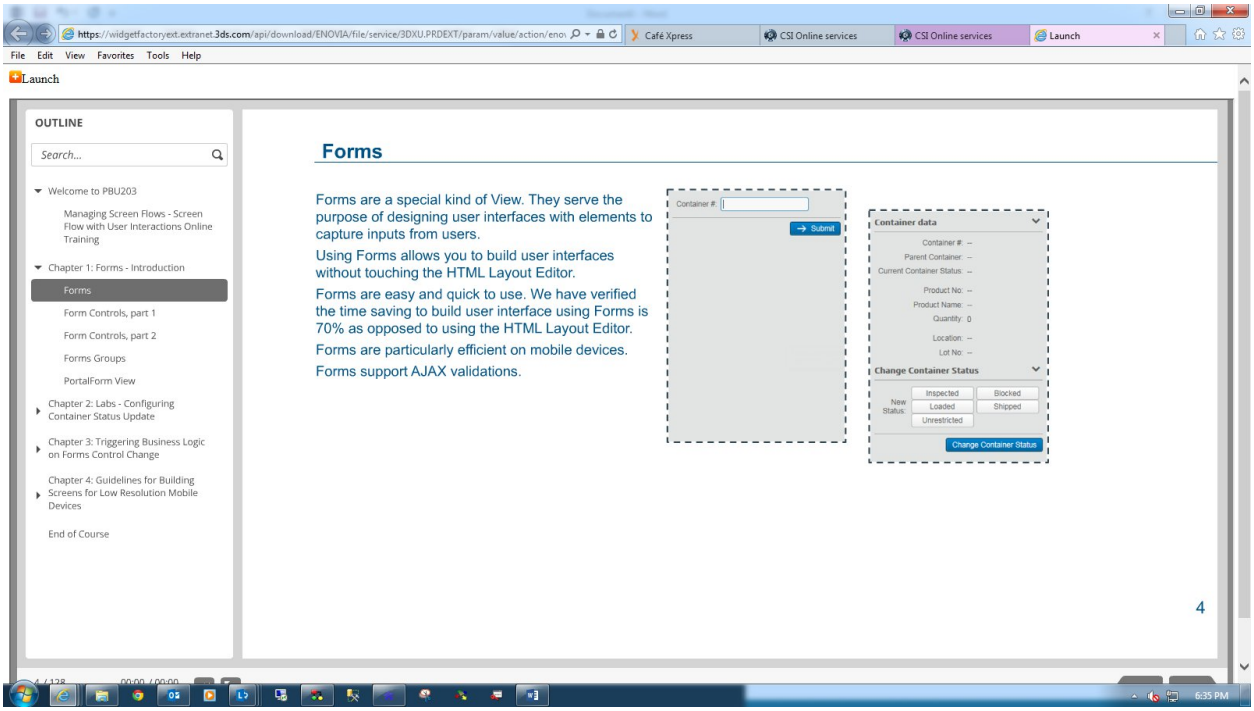

File Edit View Favorites Tools Help
Launch
OUTLINE
Search...
Welcome to PBU203
Managing Screen Flows - Screen Flow with User Interactions Online Training
Chapter 1: Forms - Introduction
Chapter 2: Labs - Configuring Container Status Update
Chapter 3: Triggering Business Logic on Forms Control Change
Chapter 4: Guidelines for Building Screens for Low Resolution Mobile Devices
End of Course
Managing Screen Flows - Screen Flow with User Interactions Online Training
Abstract:
This training will teach you how to create a simple user interface to capture information from users. You will learn how to configure text box and state button input types, as well as how to display information as labels. You will also learn how to configure one of the standard DELMIA Apriso template views, the PortalForm View.
Detailed objectives. After the training you will know:
How to build user interfaces to capture and display input from users
How to pass inputs from a Form to other Screens (standard and alternative ways)
How to work with OnChange Operations, which trigger some business logic
How to configure Screens to work with low resolution mobile devices
Target audience:
People who need to configure screen flows as part of their customer projects
Requirements:
Working knowledge of Process Builder
Role and level:
DELMIA Apriso Process Authors
DELMIA Apriso users that will be creating or managing screens
Duration: 3,5 hour
2

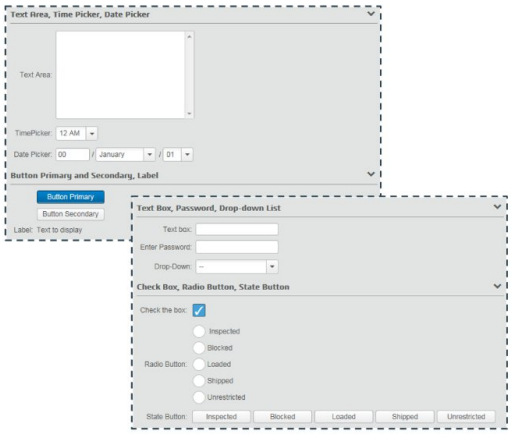

File Edit View Favorites Tools Help
Launch
OUTLINE
Search...
Welcome to PBU203
Managing Screen Flows - Screen Flow with User Interactions Online Training
Chapter 1: Forms - Introduction
Forms
Form Controls, part 1
Form Controls, part 2
Forms Groups
PortalForm View
Chapter 2: Labs - Configuring Container Status Update
Chapter 3: Triggering Business Logic on Forms Control Change
Chapter 4: Guidelines for Building Screens for Low Resolution Mobile Devices
End of Course
Forms
Forms are a special kind of View. They serve the purpose of designing user interfaces with elements to capture inputs from users.
Using Forms allows you to build user interfaces without touching the HTML Layout Editor.
Forms are easy and quick to use. We have verified the time saving to build user interface using Forms is 70% as opposed to using the HTML Layout Editor.
Forms are particularly efficient on mobile devices.
Forms support AJAX validations.
Container #:
Submit
Container data
Container #: --
Parent Container: --
Current Container Status: --
Product No: --
Product Name: --
Quantity: 0
Location: --
Lot No: --
Change Container Status
New Status:
Inspected
Blocked
Loaded
Shipped
Unrestricted
Change Container Status
4

Form Controls, part 1

The main granular element of a Form is called a Control.

A Control is a user interface element that can be used to collect input from or display information to a user. In this case a Control can use one of the following types:

- an input box
- a drop down
- a radio button
- a label
- and more



Form Controls, part 2

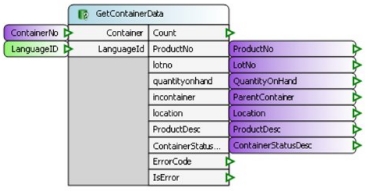
Since Form Controls capture user inputs, their values must be available for other Views or Operations.

Therefore once you add a Control, it will automatically be available as an External Variable. The name of this External Variable must be identical to the Control Code.

If you want to display a value from the system in a Control, similar logic applies. You will need to pass External Variables to your Form with names identical to the Form Controls.

Controls

Code	Type	Data Type
ContainerNo	Label	Char
ParentContainer	Label	Char
ContainerStatusDesc	Label	Char
control	Label	Char
ProductNo	Label	Char
ProductDesc	Label	Char
QuantityOnHand	Label	Decimal
Location	Label	Char
LotNo	Label	Char



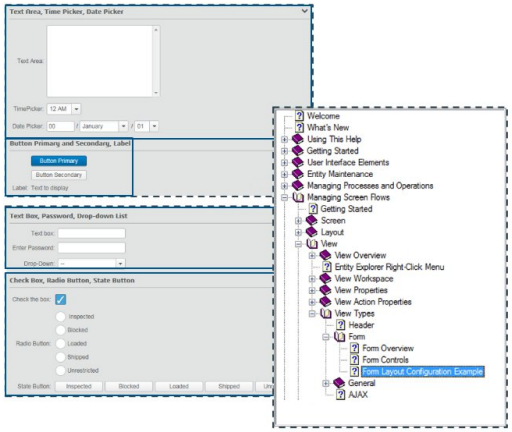
Forms Groups

For a better user experience, Controls can be pulled together into Groups.

A Group is used to display logically connected Controls together, and it can be expanded or collapsed.

The title of a Group can be shown or hidden.

Process Builder Help includes a layout configuration example of a Form View.



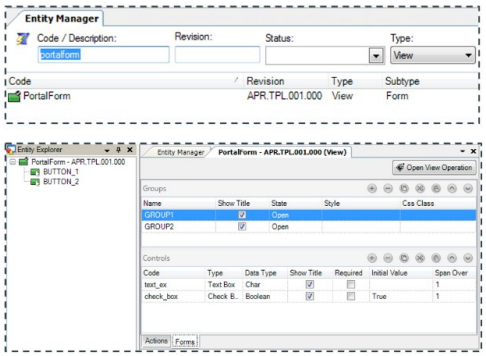
7

PortalForm View

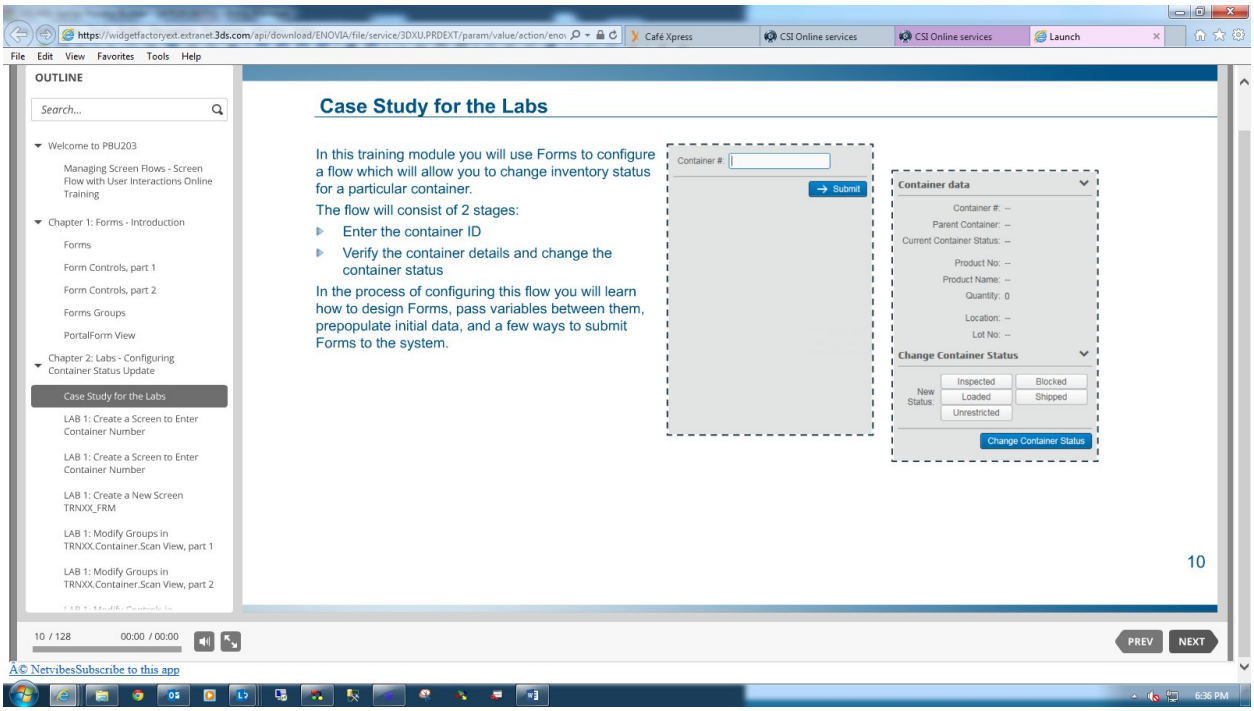
Since Forms are a special kind of View, if you want to use a Form in your Screen, always use the **PortalForm** View as your starting point.

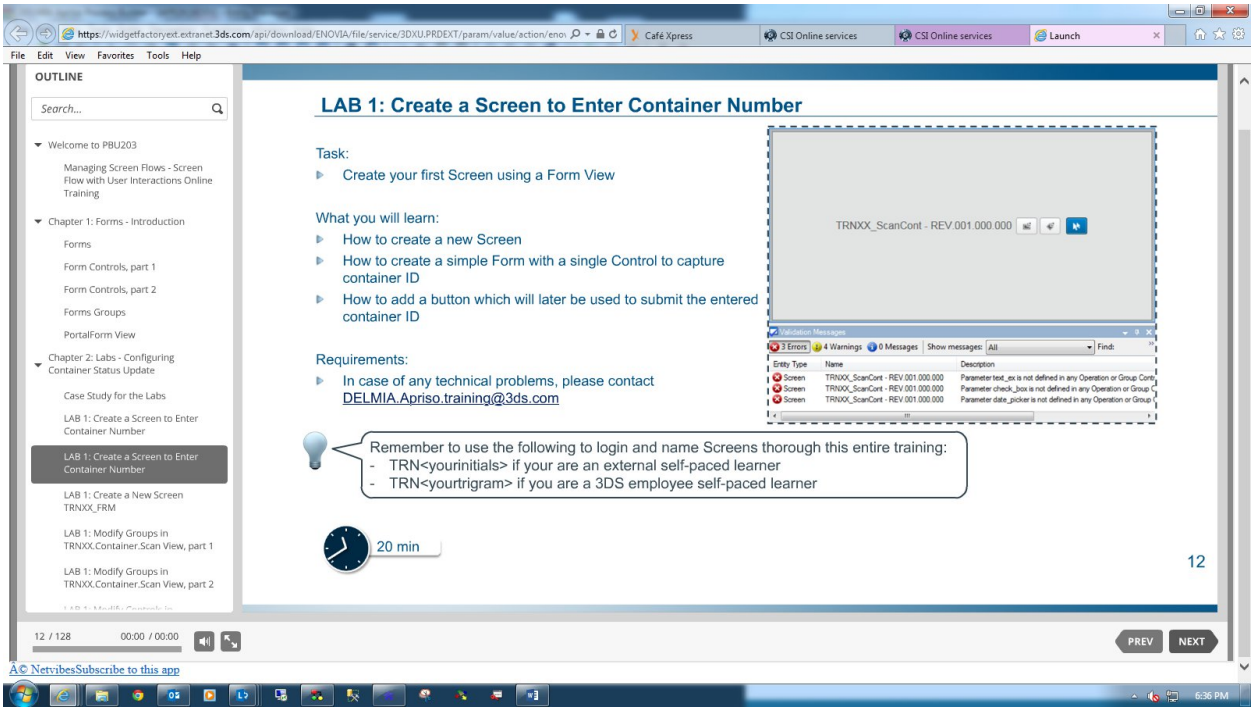
This View comes with a Forms tab, where 2 Groups with a few sample Controls are pre-configured.

Also, the PortalForm View has two buttons for potential use.



8

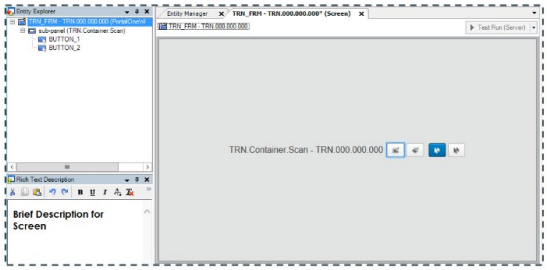

OUTLINE
Search...
Welcome to PBU203
Managing Screen Flows - Screen Flow with User Interactions Online Training
Chapter 1: Forms - Introduction
Forms
Form Controls, part 1
Form Controls, part 2
Forms Groups
PortalForm View
Chapter 2: Labs - Configuring Container Status Update
Case Study for the Labs
LAB 1: Create a Screen to Enter Container Number
LAB 1: Create a Screen to Enter Container Number
LAB 1: Create a New Screen TRNXX_FRM
LAB 1: Modify Groups in TRNXX.Container.Scan View, part 1
LAB 1: Modify Groups in TRNXX.Container.Scan View, part 2
Case Study for the Labs
In this training module you will use Forms to configure a flow which will allow you to change inventory status for a particular container.
The flow will consist of 2 stages:
Enter the container ID
Verify the container details and change the container status
In the process of configuring this flow you will learn how to design Forms, pass variables between them, prepopulate initial data, and a few ways to submit Forms to the system.
Container #:
Submit
Container data
Container #: --
Parent Container: --
Current Container Status: --
Product No: --
Product Name: --
Quantity: 0
Location: --
Lot No: --
Change Container Status
New Status:
Inspected
Blocked
Loaded
Shipped
Unrestricted
Change Container Status
10
10 / 128
00:00 / 00:00
PREV
NEXT


OUTLINE
Search...
Welcome to PBU203
Managing Screen Flows - Screen Flow with User Interactions Online Training
Chapter 1: Forms - Introduction
Forms
Form Controls, part 1
Form Controls, part 2
Forms Groups
PortalForm View
Chapter 2: Labs - Configuring Container Status Update
Case Study for the Labs
LAB 1: Create a Screen to Enter Container Number
LAB 1: Create a Screen to Enter Container Number
LAB 1: Create a New Screen TRNXX_FRM
LAB 1: Modify Groups in TRNXX.Container.Scan View, part 1
LAB 1: Modify Groups in TRNXX.Container.Scan View, part 2
LAB 1: Create a Screen to Enter Container Number
Task:
Create your first Screen using a Form View
What you will learn:
How to create a new Screen
How to create a simple Form with a single Control to capture container ID
How to add a button which will later be used to submit the entered container ID
Requirements:
In case of any technical problems, please contact DELMIA.Apriso.training@3ds.com
TRNXX_ScanCont - REV.001.000.000
Validation Messages
3 Errors
4 Warnings
0 Messages
Show messages:
All
Find:
Entity Type
Name
Description
Screen
TRNXX_ScanCont - REV.001.000.000
Parameter text_ex is not defined in any Operation or Group Cont
Parameter check_box is not defined in any Operation or Group C
Parameter date_picker is not defined in any Operation or Group C
Remember to use the following to login and name Screens thorough this entire training:
- TRN<yourinitials> if your are an external self-paced learner
- TRN<yourtrigram> if you are a 3DS employee self-paced learner
20 min
12
12 / 128
00:00 / 00:00
PREV
NEXT

LAB 1: Create a New Screen TRNXX_FRM

The first Screen you will need will be a simple Form to scan the container ID.

- Create a new Screen using the PortalOneWindow Layout, and give it a name of **TRNXX_FRM**
- Unlink the Header (you will not use any)
- For the Panel, copy and link the PortalForm View to a new View, and name it **TRNXX.Container.Scan**



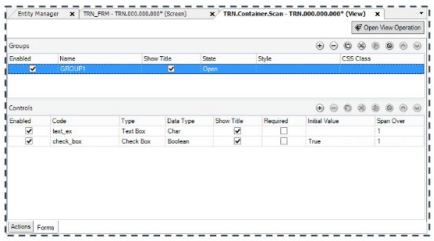
13

LAB 1: Modify Groups in TRNXX.Container.Scan View, part 1

Go to **TRNXX.Container.Scan** View.

- In the Forms tab, click to mark the GROUP2 Group and remove it

In the new Screen, there will be only one Control, so we need only one Group.

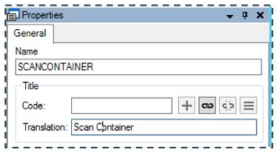


14

LAB 1: Modify Groups in TRNXX.Container.Scan View, part 2

- Click on GROUP1 Group
- Rename it to SCANCONTAINER
- Uncheck Show Title box

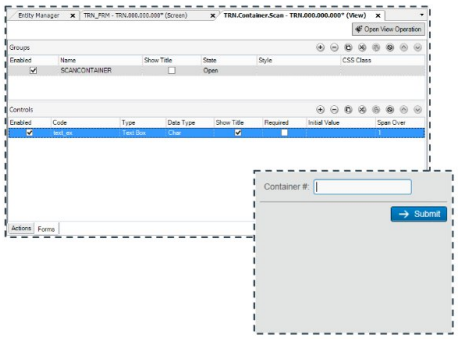
Now you have only one Group, and it will not show the title.

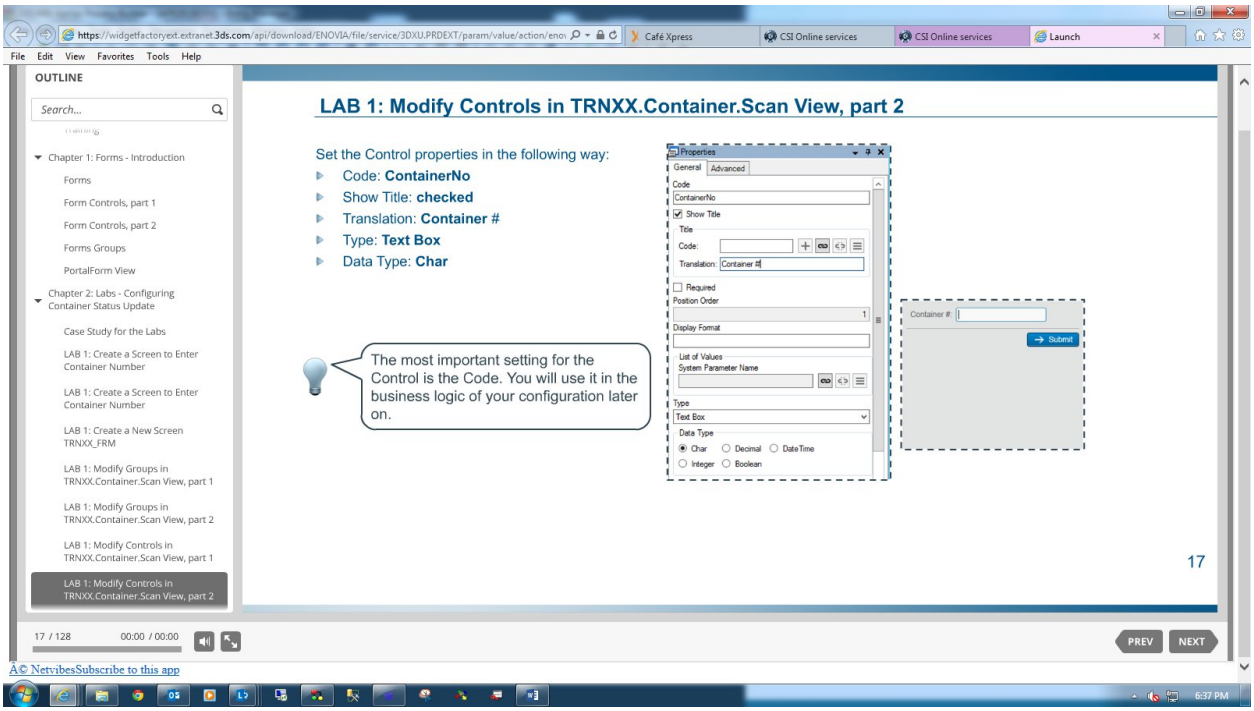


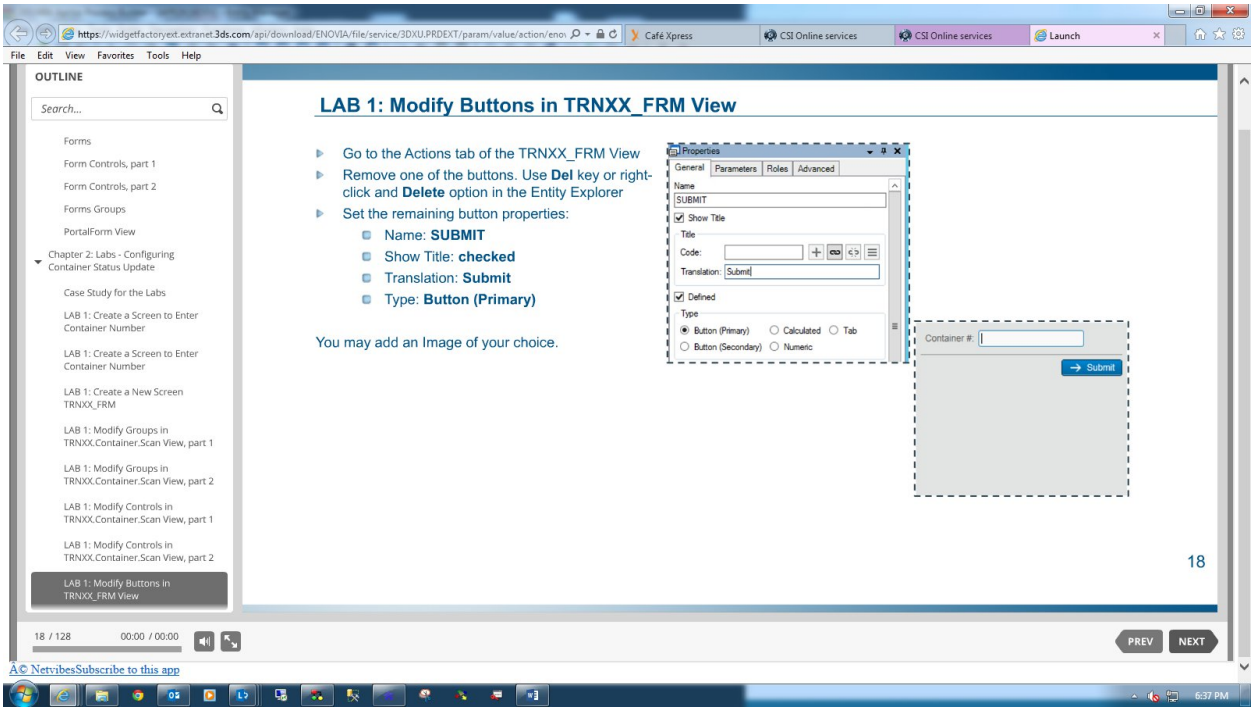
LAB 1: Modify Controls in TRNXX.Container.Scan View, part 1

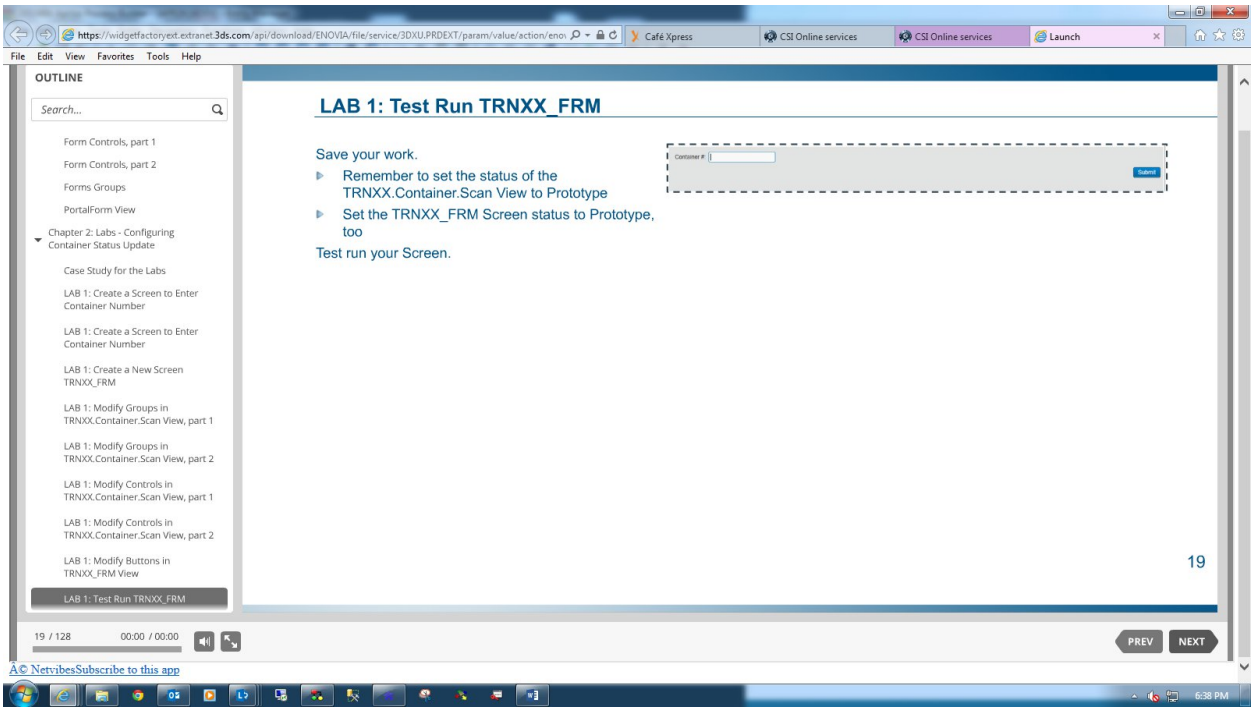
In this step, you will adjust the Controls. The SCANCONTAINER Group has two controls, and you need only one.

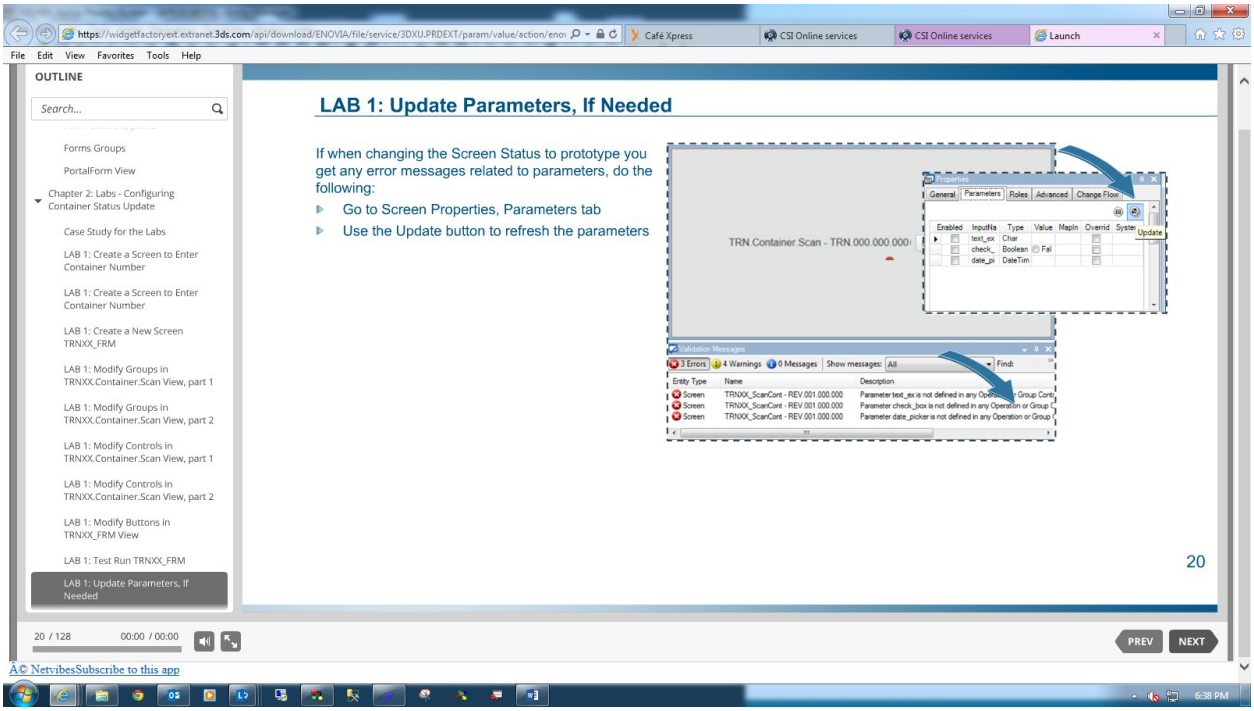
- Click on the Control you want to delete. Use the Del key, or the Remove button
- Click on the remaining Control, because in the next step you will change its properties

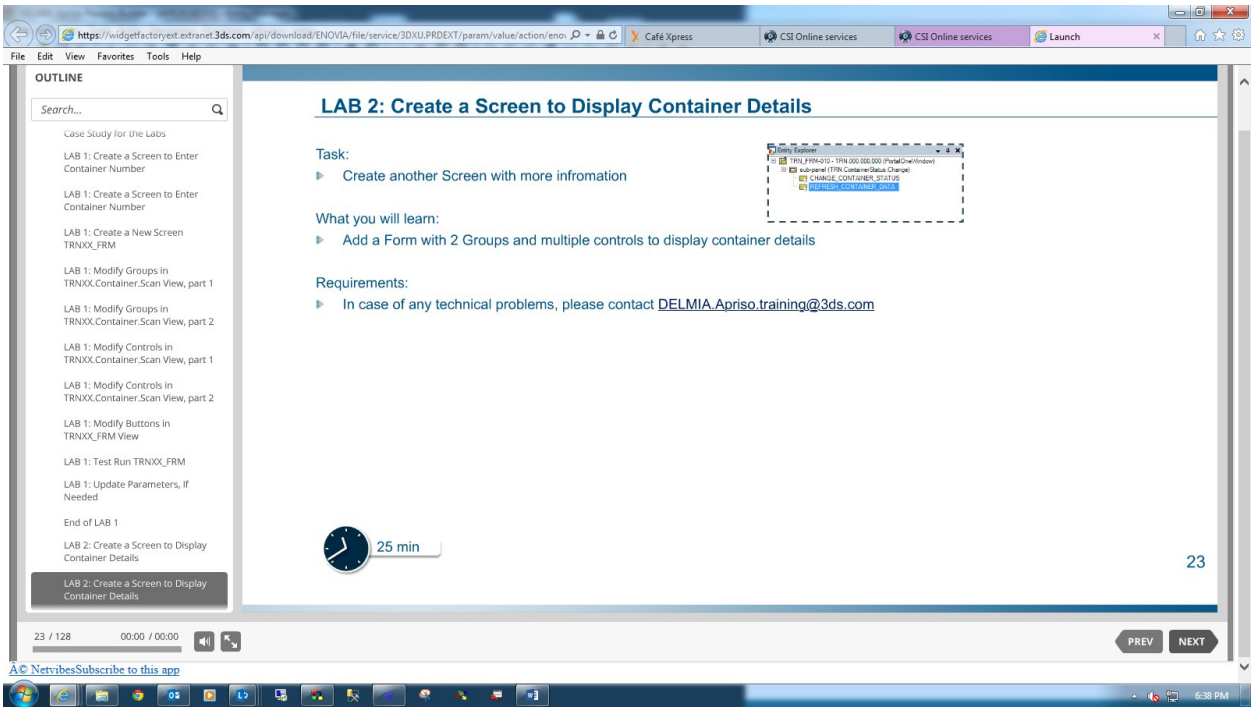


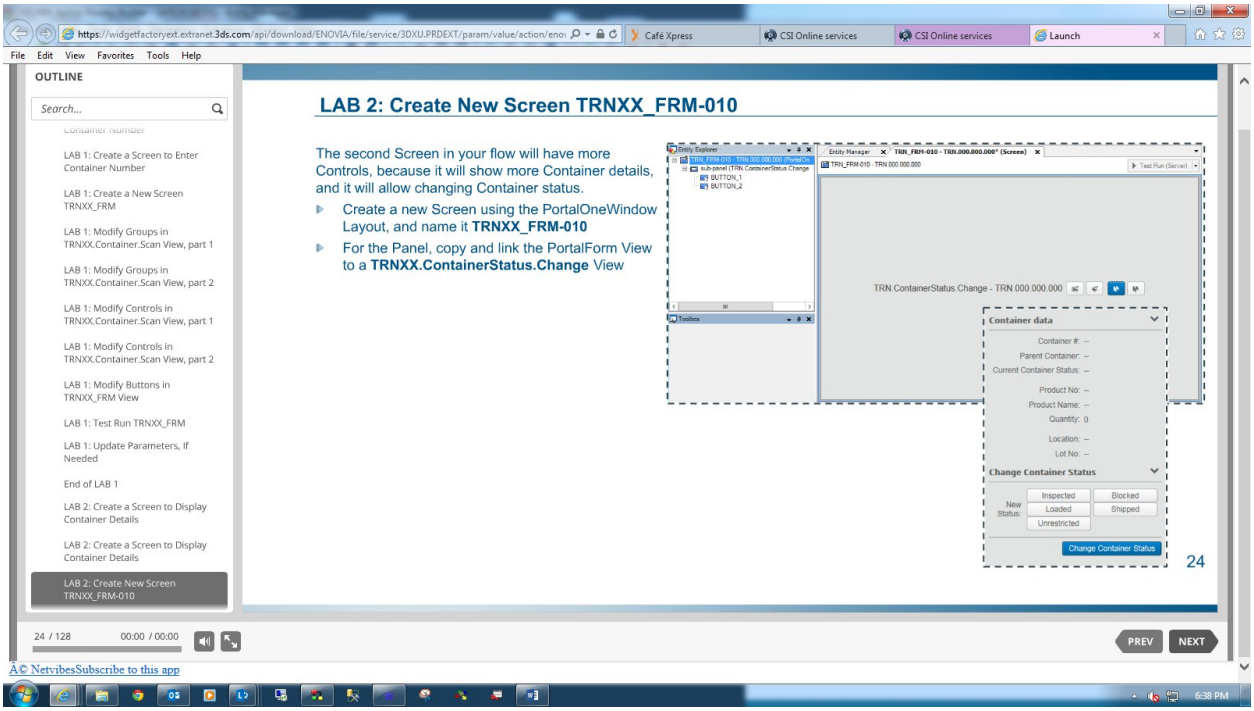

File Edit View Favorites Tools Help
OUTLINE
Search...
Chapter 1: Forms - Introduction
Forms
Form Controls, part 1
Form Controls, part 2
Forms Groups
PortalForm View
Chapter 2: Labs - Configuring Container Status Update
Case Study for the Labs
LAB 1: Create a Screen to Enter Container Number
LAB 1: Create a Screen to Enter Container Number
LAB 1: Create a New Screen TRNXX_FRM
LAB 1: Modify Groups in TRNXX.Container.Scan View, part 1
LAB 1: Modify Groups in TRNXX.Container.Scan View, part 2
LAB 1: Modify Controls in TRNXX.Container.Scan View, part 1
LAB 1: Modify Controls in TRNXX.Container.Scan View, part 2
LAB 1: Modify Controls in TRNXX.Container.Scan View, part 2
Set the Control properties in the following way:
Code: ContainerNo
Show Title: checked
Translation: Container #
Type: Text Box
Data Type: Char
The most important setting for the Control is the Code. You will use it in the business logic of your configuration later on.
Properties
General
Advanced
Code
ContainerNo
Show Title
Title
Code:
Translation:
Container #
Required
Position Order
Display Format
List of Values
System Parameter Name
Type
Text Box
Data Type
Char
Decimal
DateTime
Integer
Boolean
Container #:
Submit
17
17 / 128
00:00 / 00:00
PREV
NEXT


File Edit View Favorites Tools Help
OUTLINE
Search...
Forms
Form Controls, part 1
Form Controls, part 2
Forms Groups
PortalForm View
Chapter 2: Labs - Configuring Container Status Update
Case Study for the Labs
LAB 1: Create a Screen to Enter Container Number
LAB 1: Create a Screen to Enter Container Number
LAB 1: Create a New Screen TRNXX_FRM
LAB 1: Modify Groups in TRNXX.Container.Scan View, part 1
LAB 1: Modify Groups in TRNXX.Container.Scan View, part 2
LAB 1: Modify Controls in TRNXX.Container.Scan View, part 1
LAB 1: Modify Controls in TRNXX.Container.Scan View, part 2
LAB 1: Modify Buttons in TRNXX_FRM View
Go to the Actions tab of the TRNXX_FRM View
Remove one of the buttons. Use Del key or right-click and Delete option in the Entity Explorer
Set the remaining button properties:
Name: SUBMIT
Show Title: checked
Translation: Submit
Type: Button (Primary)
You may add an Image of your choice.
Properties
General
Parameters
Roles
Advanced
Name
SUBMIT
Show Title
Title
Code:
Translation:
Submit
Defined
Type
Button (Primary)
Calculated
Tab
Button (Secondary)
Numeric
Container #:
Submit
18
18 / 128
00:00 / 00:00
PREV
NEXT


File Edit View Favorites Tools Help
OUTLINE
Search...
Form Controls, part 1
Form Controls, part 2
Forms Groups
PortalForm View
Chapter 2: Labs - Configuring Container Status Update
Case Study for the Labs
LAB 1: Create a Screen to Enter Container Number
LAB 1: Create a Screen to Enter Container Number
LAB 1: Create a New Screen TRNXX_FRM
LAB 1: Modify Groups in TRNXX.Container.Scan View, part 1
LAB 1: Modify Groups in TRNXX.Container.Scan View, part 2
LAB 1: Modify Controls in TRNXX.Container.Scan View, part 1
LAB 1: Modify Controls in TRNXX.Container.Scan View, part 2
LAB 1: Modify Buttons in TRNXX_FRM View
LAB 1: Test Run TRNXX_FRM
LAB 1: Test Run TRNXX_FRM
Save your work.
Remember to set the status of the TRNXX.Container.Scan View to Prototype
Set the TRNXX_FRM Screen status to Prototype, too
Test run your Screen.
Container #
Submit
19
19 / 128
00:00 / 00:00
PREV
NEXT
Â© NetvibesSubscribe to this app


File Edit View Favorites Tools Help
OUTLINE
Search...
Forms Groups
PortalForm View
Chapter 2: Labs - Configuring Container Status Update
Case Study for the Labs
LAB 1: Create a Screen to Enter Container Number
LAB 1: Create a Screen to Enter Container Number
LAB 1: Create a New Screen TRNXX_FRM
LAB 1: Modify Groups in TRNXX.Container.Scan View, part 1
LAB 1: Modify Groups in TRNXX.Container.Scan View, part 2
LAB 1: Modify Controls in TRNXX.Container.Scan View, part 1
LAB 1: Modify Controls in TRNXX.Container.Scan View, part 2
LAB 1: Modify Buttons in TRNXX_FRM View
LAB 1: Test Run TRNXX_FRM
LAB 1: Update Parameters, If Needed
LAB 1: Update Parameters, If Needed
If when changing the Screen Status to prototype you get any error messages related to parameters, do the following:
Go to Screen Properties, Parameters tab
Use the Update button to refresh the parameters
TRN.Container.Scan - TRN.000.000.000
Properties
General
Parameters
Roles
Advanced
Change Flow
Update
Validation Messages
3 Errors
4 Warnings
0 Messages
Show messages:
Find:
Entity Type
Name
Description
20
20 / 128
00:00 / 00:00
PREV
NEXT
Â© NetvibesSubscribe to this app

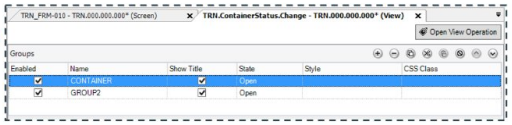

File Edit View Favorites Tools Help
OUTLINE
Search...
Case Study for the Labs
LAB 1: Create a Screen to Enter Container Number
LAB 1: Create a Screen to Enter Container Number
LAB 1: Create a New Screen TRNXX_FRM
LAB 1: Modify Groups in TRNXX.Container.Scan View, part 1
LAB 1: Modify Groups in TRNXX.Container.Scan View, part 2
LAB 1: Modify Controls in TRNXX.Container.Scan View, part 1
LAB 1: Modify Controls in TRNXX.Container.Scan View, part 2
LAB 1: Modify Buttons in TRNXX_FRM View
LAB 1: Test Run TRNXX_FRM
LAB 1: Update Parameters, If Needed
End of LAB 1
LAB 2: Create a Screen to Display Container Details
LAB 2: Create a Screen to Display Container Details
LAB 2: Create a Screen to Display Container Details
Task:
Create another Screen with more infromation
What you will learn:
Add a Form with 2 Groups and multiple controls to display container details
Requirements:
In case of any technical problems, please contact DELMIA.Apriso.training@3ds.com
25 min
23
23 / 128
00:00 / 00:00
PREV
NEXT

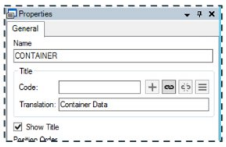

File Edit View Favorites Tools Help
OUTLINE
Search...
LAB 1: Create a Screen to Enter Container Number
LAB 1: Create a New Screen TRNXX_FRM
LAB 1: Modify Groups in TRNXX.Container.Scan View, part 1
LAB 1: Modify Groups in TRNXX.Container.Scan View, part 2
LAB 1: Modify Controls in TRNXX.Container.Scan View, part 1
LAB 1: Modify Controls in TRNXX.Container.Scan View, part 2
LAB 1: Modify Buttons in TRNXX_FRM View
LAB 1: Test Run TRNXX_FRM
LAB 1: Update Parameters, If Needed
End of LAB 1
LAB 2: Create a Screen to Display Container Details
LAB 2: Create a Screen to Display Container Details
LAB 2: Create New Screen TRNXX_FRM-010
The second Screen in your flow will have more Controls, because it will show more Container details, and it will allow changing Container status.
Create a new Screen using the PortalOneWindow Layout, and name it TRNXX_FRM-010
For the Panel, copy and link the PortalForm View to a TRNXX.ContainerStatus.Change View
TRN.ContainerStatus.Change - TRN.000.000.000
Container data
Container #:
Parent Container:
Current Container Status:
Product No:
Product Name:
Quantity: 0
Location:
Lot No:
Change Container Status
New Status:
Inspected
Blocked
Loaded
Shipped
Unrestricted
Change Container Status
24
24 / 128
00:00 / 00:00
PREV
NEXT

LAB 2: Change Container Properties, part 1

- Go to TRNXX.ContainerStatus.Change View
- In the Forms tab, change the properties of the first Group:
  - Name: **CONTAINER**
  - Translation: **Container Data**
  - Show Title: checked

This Group will be used to display container information.



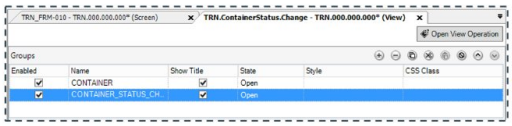


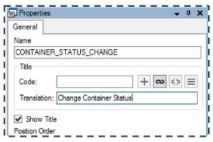
25

LAB 2: Change Container Properties, part 2

- Now, change the properties of the second Group:
  - Name: **CONTAINER_STATUS_CHANGE**
  - Translation: **Change Container Status**
  - Show Title: checked

This Group will be used to display available status values, and to take action from the screen user.
The next step will be to add Controls to both Groups.



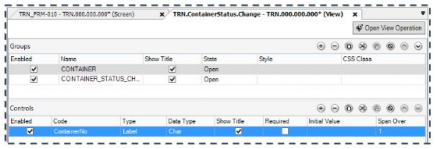


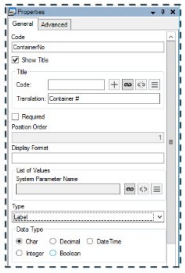
26

LAB 2: Add Control, part 1

- Go to the CONTAINER Group.
- Add the ContainerNo Control:
  - Code: **ContainerNo**
  - Show Title: **checked**
  - Translation: **Container #**
  - Type: **Label**
  - Data Type: **Char**

The Control Type „Label" is used to add a display only Control. All Controls in this Group will be labels, because you are using the Group to show selected container details.



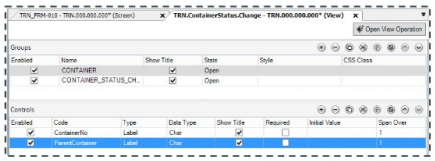


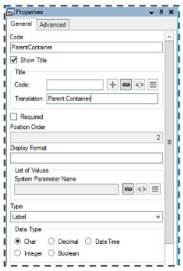
27

LAB 2: Add Control, part 2

- Add the ParentContainer Control:
  - Code: **ParentContainer**
  - Show Title: **checked**
  - Translation: **Parent Container**
  - Type: **Label**
  - Data Type: **Char**

This will be the information about a container, in which the selected container is stored. An example will be a box (child container) stored on a pallet (parent container).



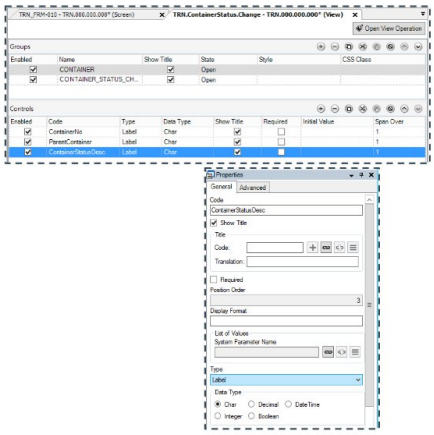


28

LAB 2: Add Control, part 3

- Add the ContainerStatusDesc Control:
  - Code: **ContainerStatusDesc**
  - Show Title: **checked**
  - Translation: **Current Container Status**
  - Type: **Label**
  - Data Type: **Char**

You will display the current status of the selected container.

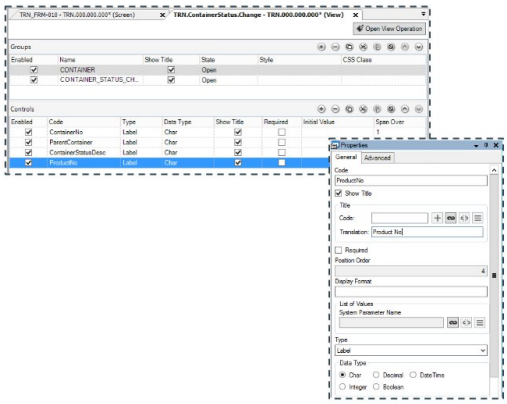


29

LAB 2: Add Control, part 4

- Add the ProductNo Control:
  - Code: **ProductNo**
  - Show Title: **checked**
  - Translation: **Product No**
  - Type: **Label**
  - Data Type: **Char**

This will be information about the product code stored in the container. We will work with containers which have just one type of a product.

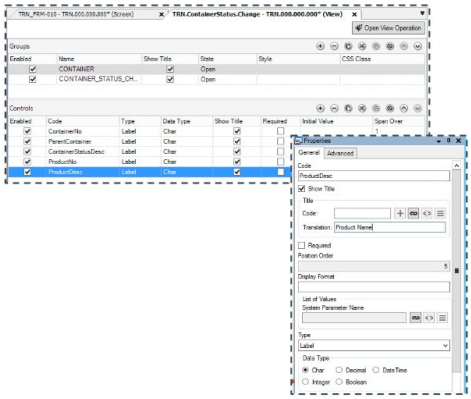


30

LAB 2: Add Control, part 5

- Add the ProductDesc Control:
  - Code: **ProductDesc**
  - Show Title: **checked**
  - Translation: **Product Name**
  - Type: **Label**
  - Data Type: **Char**

The previous control will show the product code, and this one will show its name.



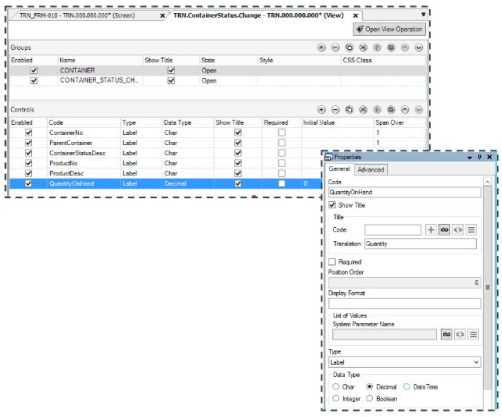
31

LAB 2: Add Control, part 6

- Add the QuantityOnHand Control:
  - Code: **QuantityOnHand**
  - Show Title: **checked**
  - Translation: **Quantity**
  - Type: **Label**
  - Data Type: **Decimal**

This will be the information about the product quantity in the container.

NOTE: Pay attention to the data type of this Control. It has to be Decimal, while all other Controls will be Char.

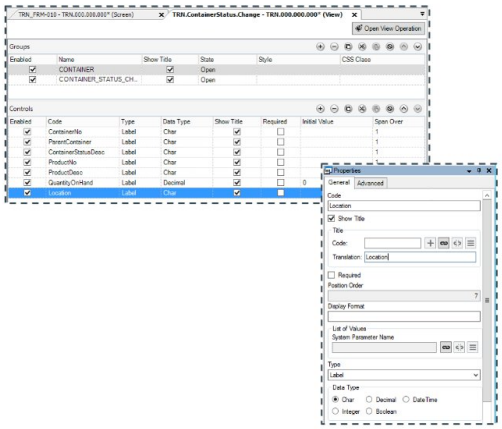


32

LAB 2: Add Control, part 7

- Add the Location Control:
  - Code: **Location**
  - Show Title: **checked**
  - Translation: **Location**
  - Type: **Label**
  - Data Type: **Char**

This will be information about the warehouse location in which the container is stored.

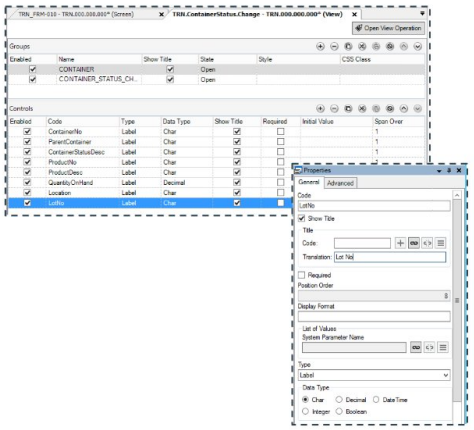


33

LAB 2: Add Control, part 8

- Add the LotNo Control:
  - Code: **LotNo**
  - Show Title: **checked**
  - Translation: **Lot No**
  - Type: **Label**
  - Data Type: **Char**

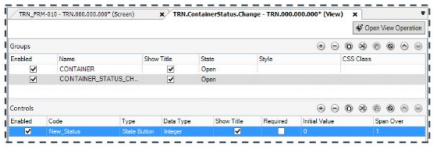
This will be information about the lot number that the container and its content belong to.

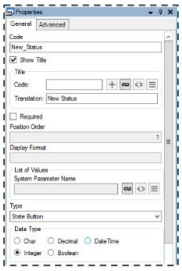


34

LAB 2: Add Control, part 9

- Go to the CONTAINER_STATUS_CHANGE Group
- Add the New_Status Control:
  - Code: **New_Status**
  - Show Title: **checked**
  - Translation: **New Status**
  - Type: **State Button**
  - Data Type: **Integer**





This will be the Control in which the user will select a new container status, and make a change.

Don't test run this Screen yet, as you will get an error. You need to add values for the State Button, which you will do in the next lab.

35

LAB 3: Create State Buttons for Container Status

Task:
- Add a System Parameter
- Link added System Parameter to the correct Screen

What you will learn:
- Add a new System Parameter to define the list of available container status values
- Link the System Parameter to New Status state buttons
- Add a button to submit new container status value
- Modify the look of the TRNXX_FRM-010 Screen

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

30 min

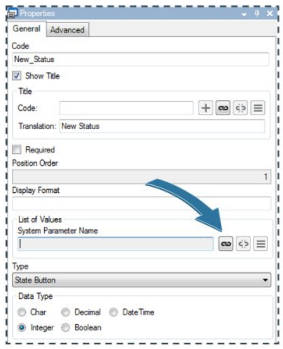
38

LAB 3: Check List of Values, part 1

The New_Status Control will show buttons with available status options. This is why you chose **State Button** as its Type. You need now to define what these status options will be.

This can be done using the List of Values property:

- Click on the Link button at this property



Any Control which collects user input, but is supposed to give a choice of possible selections, like a drop down, or buttons, will use the List of Values property.

39

https://widgetfactoryext.extranet.3ds.com/api/download/ENOVIA/file/service/3DXU.PRDEXT/param/value/action/enov
Café Xpress | CSI Online services | CSI Online services | Launch
File Edit View Favorites Tools Help

LAB 3: Check List of Values, part 2

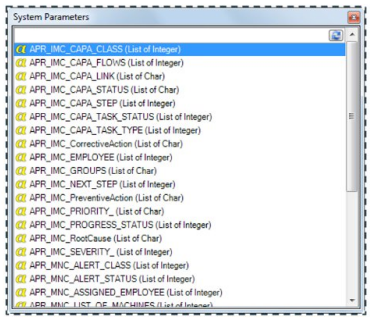
This button will show a popup with a list of available system parameters.

In your real life projects, you will use existing System Parameters list, either with default or custom values.

In many cases you will need to create your own System Parameter. This is a case in this training.

- Close the popup

Next few slides will take you through the steps how to create your own System Parameter.

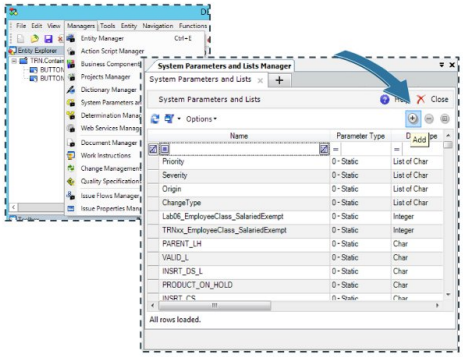


40

LAB 3: Add System Parameter, part 1

- In Process Builder, go to the menu bar
- Go to Managers, and select **System Parameters and Lists Manager**
- In order to create your own System Parameter, click on the Add button

This System Parameter will display container status values. The DELMIA Apriso system has a table for these values. Therefore a part of the setup you will do is to write an SQL query to retrieve the values to show.



41

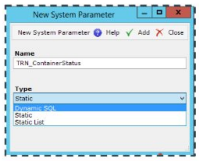
https://widgetfactoryext.extranet.3ds.com/api/download/ENOVIA/file/service/3DXU.PRDEXT/param/value/action/enov
Café Xpress | CSI Online services | CSI Online services | Launch
File Edit View Favorites Tools Help

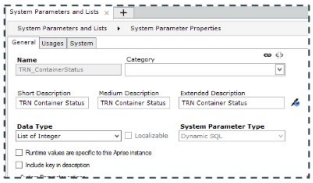
LAB 3: Add System Parameter, part 2

- Set the properties as follows:
  - Name: **TRNXX_ContainerStatus**
  - Short Description: **TRNXX Container Status**
  - Data Type: **List of Integer**
  - System Parameter Type: **Dynamic SQL**

In the DELMIA Apriso database, container status is a number, this is why the Data Type of this System Parameter is List of Integer. Status descriptions will be shown to the user, but the system will operate on the integer codes.

You will paste a query script in a moment, but first you need to understand 2 DELMIA Apriso tables which will be used by the query.



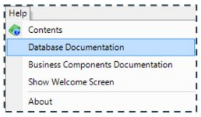


42

LAB 3: Container_Status Help Entry

The query will need to retrieve the container status integer code, and its text description.

To see which table and which columns to use you can use the database documentation, which is available in Process Builder in the Help menu.



As you can see here, ContainerStatus is indeed an integer, and the table does not hold text descriptions, but it has a TextID column.

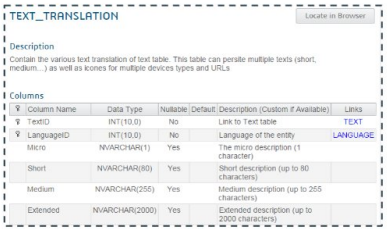
43

LAB 3: Text_Translation Help Entry

You will find a TextID column in many database tables. It links these tables to the TEXT_TRANSLATION table, where multilingual translations of all objects are maintained.

Therefore the key of this table consists of 2 columns: the TextID, and LanguageID, so that you can retrieve the description in the desired language.

There are different lengths of description can be configured, as DELMIA Apriso works with many devices, which sometimes have limited graphical capabilities.



44

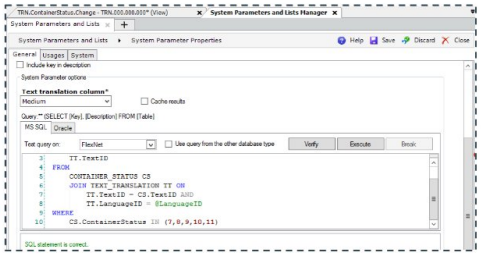
LAB 3: Add System Parameter, part 3

Now you are ready to paste the SQL query into the System Parameter you created. Use the scripts:

SCRIPT FILE: Desktop/Training Materials/Level 1

The LanguageID is a System Variable. It is populated at the login to the system. The @ before the variable name tells the query the value needs to be retrieved from the DELMIA Apriso system.

You do not want to list all statuses. A few will be enough. Therefore you will limit the container status to be between 7 and 11.



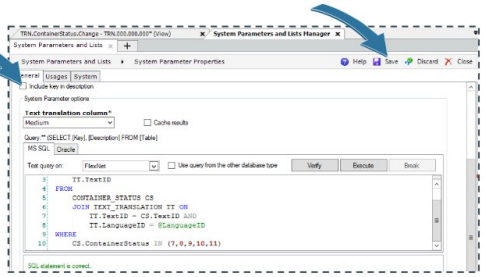
45

LAB 3: Add System Parameter, part 4

One of the System Parameter options is „Text translation column."

Rather than selecting your description in SQL, you can use the drop down to determine which description you want to use, and leave the selected column as TT.TextID.

- Save the System Parameter

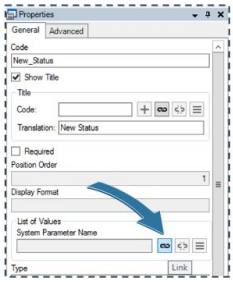


46

LAB 3: Link the System Parameter

Back in the New_Status Control, click the Link button at the List of Values again.

- In the drop-down, select your **TRNXX_ContainerStatus** System Parameter

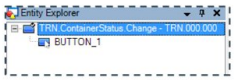


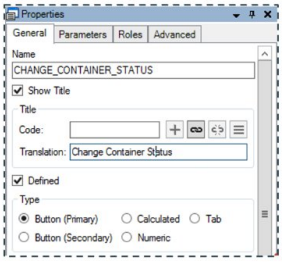
47

LAB 3: Modify Buttons in TRNXX.ContainerStatus.Change View

- Go to the Actions tab of the **TRNXX.ContainerStatus.Change** View
- Remove one of the buttons
- Set the remaining button properties:
  - Name: **CHANGE_CONTAINER_STATUS**
  - Translation: **Change Container Status**
  - Show Title: **checked**
  - Type: **Button (Primary)**

This will be a button to confirm user's choice of the new container status.





48

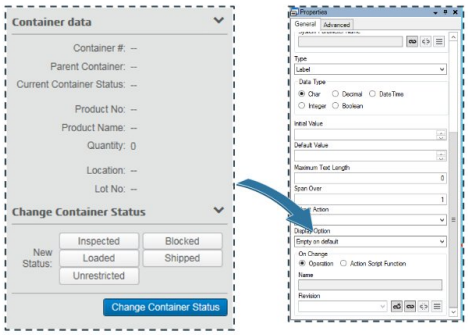
LAB 3: Test Run TRNXX_FRM-010

- Save your work
- Remember to set the statuses of the TRNXX_FRM-010 View and Screen to Prototype
- Test run your Screen

LAB 3: Make Visual Modifications to TRNXX_FRM-010, part 1

Make now some visual modifications to this Screen.

- Go to **TRNXX.ContainerStatus.Change** View
- Go to the CONTAINER Group
- Go to ContainerNo Control properties
- Set the Display Option to Empty on Default, to eliminate empty spaces after the labels
- Set the Display Option in the same way for all other Controls in the CONTAINER Group:
  - ParentContainer
  - ContainerStatusDesc
  - ProductNo
  - ProductDesc
  - QuantityOnHand
  - Location
  - LotNo

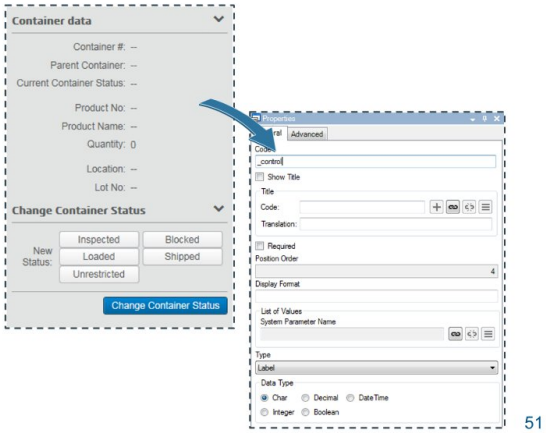


LAB 3: Make Visual Modifications to TRNXX_FRM-010, part 2

You may also want to visually separate container information, from product, and from location details.

For this purpose, you can add new Controls, which will be empty and will have Show Title unchecked.

Such controls have to have the Show Title unchecked, and the Code needs to have the „_" prefix. This prefix disables the Control as an external variable. And you do not want this Control to be such variable, as you use it only to improve the visual look of the Form.



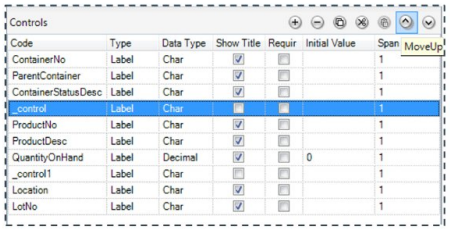
51

https://widgetfactoryext.extranet.3ds.com/api/download/ENOVIA/file/service/3DXU.PRDEXT/param/value/action/enov
Café Xpress | CSI Online services | CSI Online services | Launch
File Edit View Favorites Tools Help

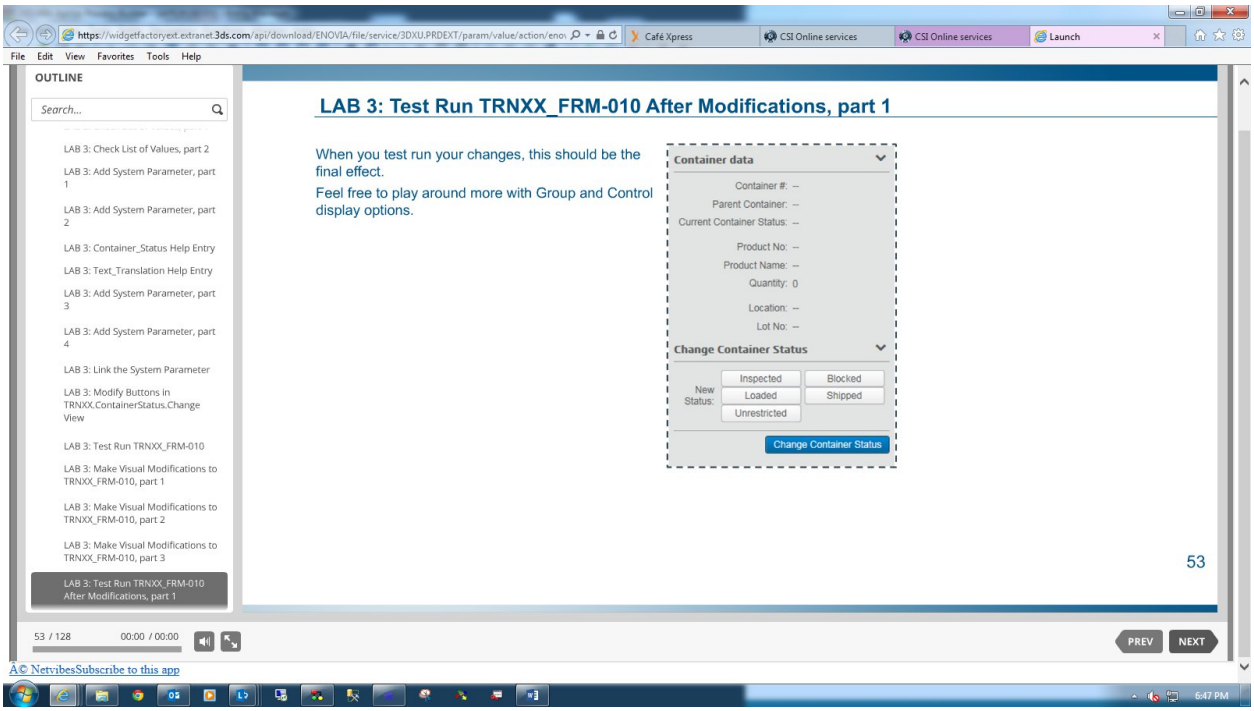
LAB 3: Make Visual Modifications to TRNXX_FRM-010, part 3

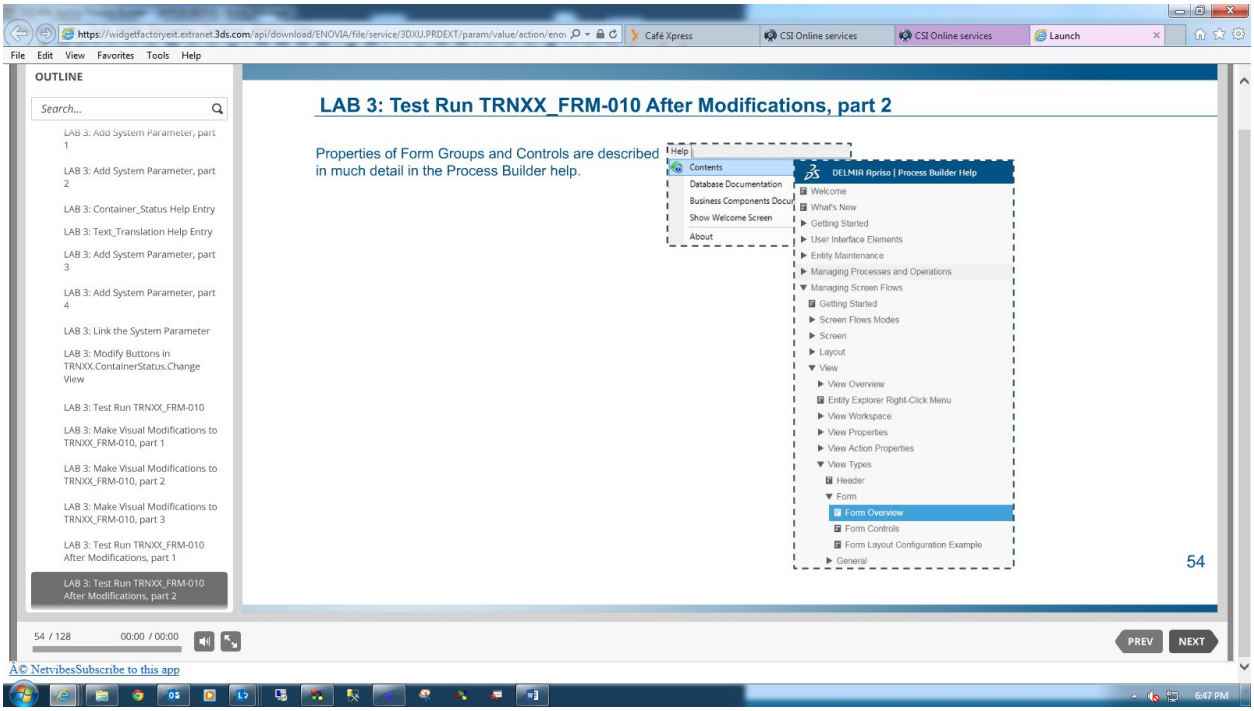
Because of the external variables, one Form must have unique Control Codes, so if you add more empty Controls, name them differently.

You can use MoveUp and MoveDown buttons to change the display sequence of the Controls.



52

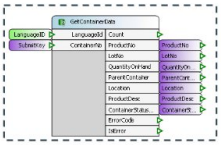

File Edit View Favorites Tools Help
OUTLINE
Search...
LAB 3: Check List of Values, part 2
LAB 3: Add System Parameter, part 1
LAB 3: Add System Parameter, part 2
LAB 3: Container_Status Help Entry
LAB 3: Text_Translation Help Entry
LAB 3: Add System Parameter, part 3
LAB 3: Add System Parameter, part 4
LAB 3: Link the System Parameter
LAB 3: Modify Buttons in TRNXX.ContainerStatus.Change View
LAB 3: Test Run TRNXX_FRM-010
LAB 3: Make Visual Modifications to TRNXX_FRM-010, part 1
LAB 3: Make Visual Modifications to TRNXX_FRM-010, part 2
LAB 3: Make Visual Modifications to TRNXX_FRM-010, part 3
LAB 3: Test Run TRNXX_FRM-010 After Modifications, part 1
LAB 3: Test Run TRNXX_FRM-010 After Modifications, part 1
When you test run your changes, this should be the final effect.
Feel free to play around more with Group and Control display options.
Container data
Container #: --
Parent Container: --
Current Container Status: --
Product No: --
Product Name: --
Quantity: 0
Location: --
Lot No: --
Change Container Status
New Status:
Inspected
Blocked
Loaded
Shipped
Unrestricted
Change Container Status
53
53 / 128
00:00 / 00:00
PREV
NEXT
Â© NetvibesSubscribe to this app


File Edit View Favorites Tools Help
OUTLINE
Search...
LAB 3: Add System Parameter, part 1
LAB 3: Add System Parameter, part 2
LAB 3: Container_Status Help Entry
LAB 3: Text_Translation Help Entry
LAB 3: Add System Parameter, part 3
LAB 3: Add System Parameter, part 4
LAB 3: Link the System Parameter
LAB 3: Modify Buttons in TRNXX.ContainerStatus.Change View
LAB 3: Test Run TRNXX_FRM-010
LAB 3: Make Visual Modifications to TRNXX_FRM-010, part 1
LAB 3: Make Visual Modifications to TRNXX_FRM-010, part 2
LAB 3: Make Visual Modifications to TRNXX_FRM-010, part 3
LAB 3: Test Run TRNXX_FRM-010 After Modifications, part 1
LAB 3: Test Run TRNXX_FRM-010 After Modifications, part 2
LAB 3: Test Run TRNXX_FRM-010 After Modifications, part 2
Properties of Form Groups and Controls are described in much detail in the Process Builder help.
Help
Contents
Database Documentation
Business Components Docu
Show Welcome Screen
About
DELMIA Apriso | Process Builder Help
Welcome
What's New
Getting Started
User Interface Elements
Entity Maintenance
Managing Processes and Operations
Managing Screen Flows
Getting Started
Screen Flows Modes
Screen
Layout
View
View Overview
Entity Explorer Right-Click Menu
View Workspace
View Properties
View Action Properties
View Types
Header
Form
Form Overview
Form Controls
Form Layout Configuration Example
General
54
54 / 128
00:00 / 00:00
PREV
NEXT
Â© NetvibesSubscribe to this app

LAB 4: Pass the ContainerNo

Task:

- Add business logic to capture the ContainerNo
- Display this containers details in the second Screen



What you will learn:

- How to create a new Screen using an existing Layout
- How to use existing Views to provide content to your Screens
- How to test run your Screens

Requirements:

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

45 min

57

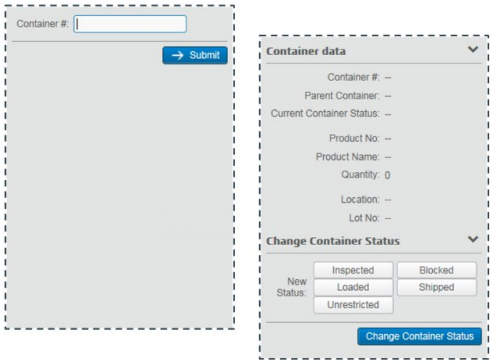
LAB 4: Pass the ContainerNo

In order to navigate between the Forms, the Submit button will need to direct the user to the second Screen.

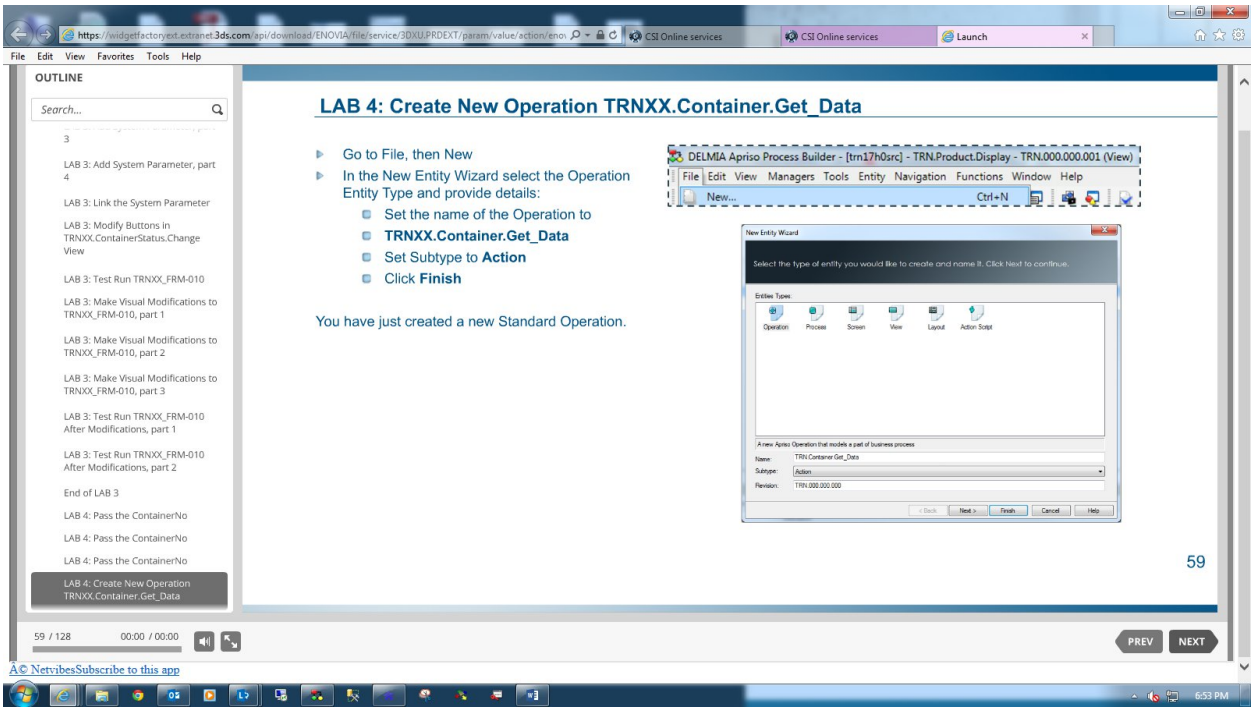
Also, we need business logic to extract the container details from the system based on the input in the first Form, and display them in the second one.

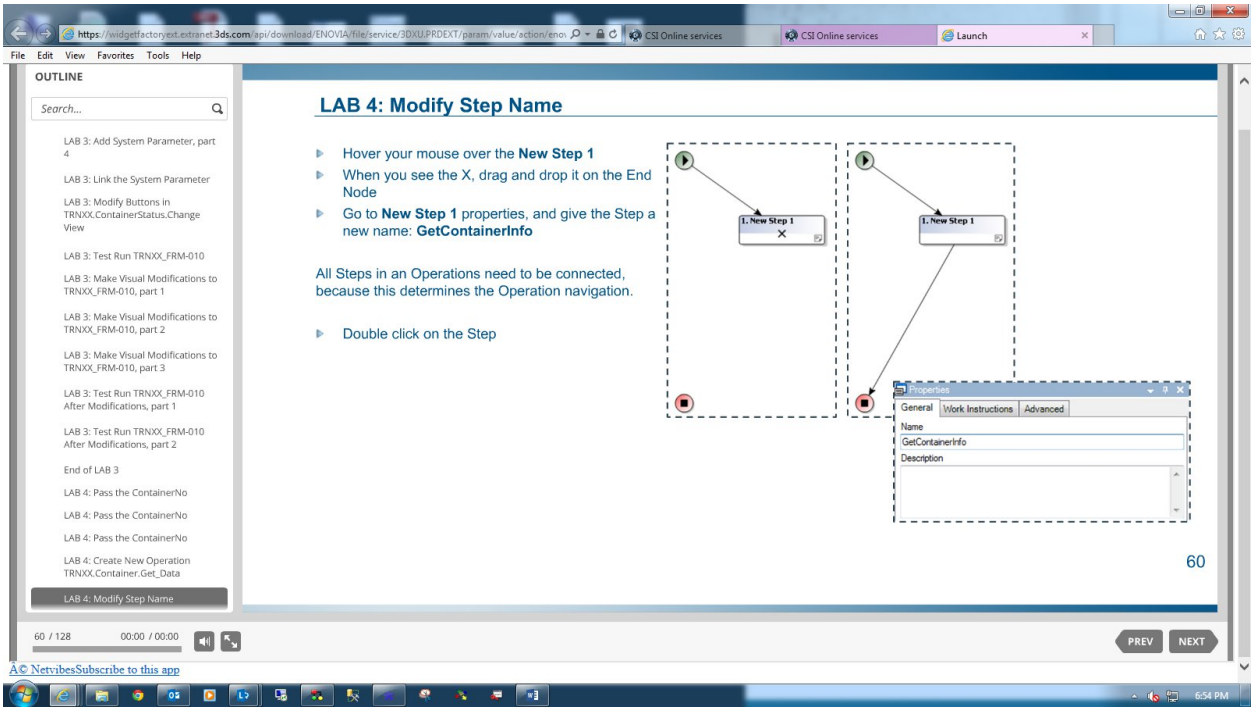
This business logic will be a Standard Operation attached to the Submit button.

Let's start the configuration.



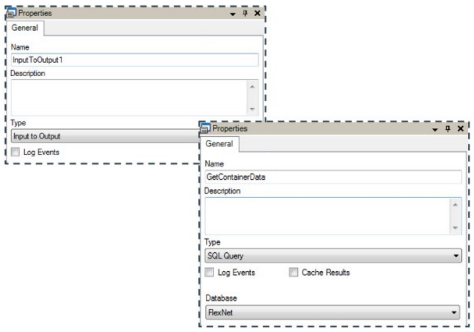
58


File Edit View Favorites Tools Help
OUTLINE
Search...
LAB 3: Add System Parameter, part 4
LAB 3: Link the System Parameter
LAB 3: Modify Buttons in TRNXX.ContainerStatus.Change View
LAB 3: Test Run TRNXX_FRM-010
LAB 3: Make Visual Modifications to TRNXX_FRM-010, part 1
LAB 3: Make Visual Modifications to TRNXX_FRM-010, part 2
LAB 3: Make Visual Modifications to TRNXX_FRM-010, part 3
LAB 3: Test Run TRNXX_FRM-010 After Modifications, part 1
LAB 3: Test Run TRNXX_FRM-010 After Modifications, part 2
End of LAB 3
LAB 4: Pass the ContainerNo
LAB 4: Pass the ContainerNo
LAB 4: Pass the ContainerNo
LAB 4: Create New Operation TRNXX.Container.Get_Data
LAB 4: Create New Operation TRNXX.Container.Get_Data
Go to File, then New
In the New Entity Wizard select the Operation Entity Type and provide details:
Set the name of the Operation to
TRNXX.Container.Get_Data
Set Subtype to Action
Click Finish
You have just created a new Standard Operation.
DELMIA Apriso Process Builder - [trn17h0src] - TRN.Product.Display - TRN.000.000.001 (View)
File Edit View Managers Tools Entity Navigation Functions Window Help
New... Ctrl+N
New Entity Wizard
Select the type of entity you would like to create and name it. Click Next to continue.
Entities Types:
Operation
Process
Screen
View
Layout
Action Script
A new Apriso Operation that models a part of business process
Name: TRN.Container.Get_Data
Subtype: Action
Revision: TRN.000.000.000
< Back
Next >
Finish
Cancel
Help
59
59 / 128
00:00 / 00:00
PREV
NEXT
© NetvibesSubscribe to this app
6:53 PM


File Edit View Favorites Tools Help
OUTLINE
Search...
LAB 3: Add System Parameter, part 4
LAB 3: Link the System Parameter
LAB 3: Modify Buttons in TRNXX.ContainerStatus.Change View
LAB 3: Test Run TRNXX_FRM-010
LAB 3: Make Visual Modifications to TRNXX_FRM-010, part 1
LAB 3: Make Visual Modifications to TRNXX_FRM-010, part 2
LAB 3: Make Visual Modifications to TRNXX_FRM-010, part 3
LAB 3: Test Run TRNXX_FRM-010 After Modifications, part 1
LAB 3: Test Run TRNXX_FRM-010 After Modifications, part 2
End of LAB 3
LAB 4: Pass the ContainerNo
LAB 4: Pass the ContainerNo
LAB 4: Pass the ContainerNo
LAB 4: Create New Operation TRNXX.Container.Get_Data
LAB 4: Modify Step Name
Hover your mouse over the New Step 1
When you see the X, drag and drop it on the End Node
Go to New Step 1 properties, and give the Step a new name: GetContainerInfo
All Steps in an Operations need to be connected, because this determines the Operation navigation.
Double click on the Step
1. New Step 1
1. New Step 1
Properties
General
Work Instructions
Advanced
Name
GetContainerInfo
Description
60
60 / 128
00:00 / 00:00
PREV
NEXT
© NetvibesSubscribe to this app
6:54 PM

LAB 4: Add SQL Query Function

- In the Step, right-click and Add Function
- Go to the Function properties
- Change the name to **GetContainerData**
- Change the Type to **SQL Query**
- Click on the SQL Query Editor button



61

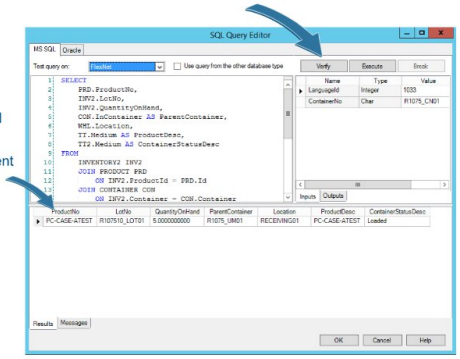
https://widgetfactoryext.extranet.3ds.com/api/download/ENOVIA/file/service/3DXU.PRDEXT/param/value/action/enov
CSI Online services
CSI Online services
Launch
File Edit View Favorites Tools Help

LAB 4: Provide Script to the Function

- Make sure you are in the MS SQL tab, and the **Test query on** points to FlexNet
- Paste the query from the scripts:

SCRIPT FILE: Desktop/Training Materials/Level 1

- Click on the Verify button and check the comment in the Messages tab below

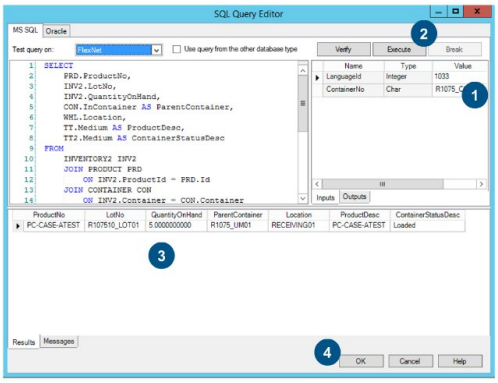


62

LAB 4: Run the Query

To test run the query:

1. Use **R1075_CN01** as Container in the Inputs on the right. Use „1033" as LanguageId (this is Windows Language Code ID for English)
2. Click on the Execute button
3. You should see results at the bottom of the window
4. Click OK



LAB 4: Configure Other Functions

You can see the Function has Inputs and Outputs based on the SQL script you used.

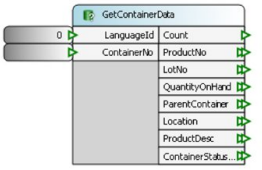
You need to let the function know where the input values come from, and how the outputs will be used.

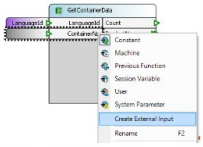
- Create two external inputs: LanguageId and ContainerNo

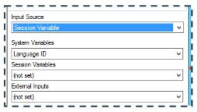
Container will come from the first Form you created. The LanguageId is a System Variable.

The Outputs will need to be passed to the second Form. They are the container details you need.

Inputs and Outputs need to be defined in the Operation Interface.







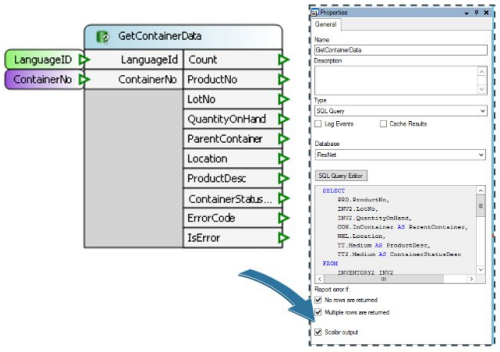
LAB 4: Check Scalar Output

For the purpose of this lab, we are assuming containers have just one product each, so there will be only one record returned by the query.

This needs to be reflected in the data type of GetContainerData function outputs.

- Go to the function properties, and at the bottom check the **Scalar** output box

Error boxes are automatically checked. They allow configuring error messages if the query output is not scalar. You will not configure this in this lab.



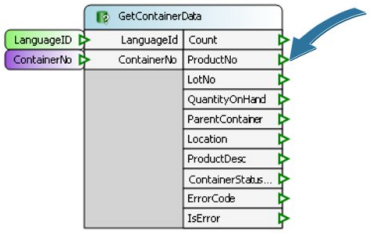
LAB 4: Input/Output Indicators

After the change, this how your query function should look like.

Green triangles at the function inputs/outputs indicate if they are scalar, or not.

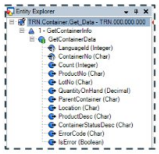
A single triangle means a scalar input or output.

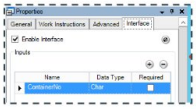
A double triangle (like you had before) indicates a list.



LAB 4: Enable Interface

- In the Entiry Explorer, make sure the top node (the Operation) is marked
- Go to Operation Properties and select the **Interface** Tab
- Check the **Enable Interface** box





In this way you allow the Operation to receive inputs from other entities, and send outputs to them.

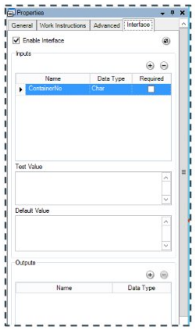
The next step will be to define which inputs and outputs you want the Operation to work with. They will be the ones used in the SQL query function.

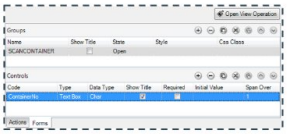
67

LAB 4: Modify Operation Properties, part 1

There are further properties in the Interface tab.

- In the Inputs, add (if it's not there yet):
  - ContainerNo with Data Type **Char**





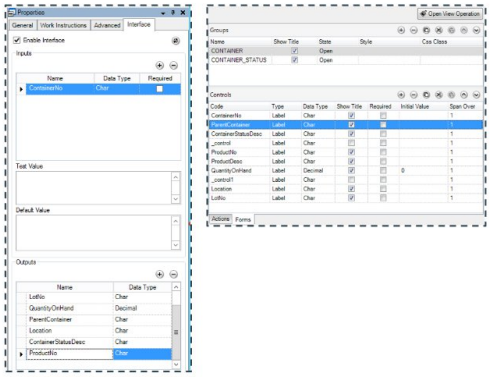
The ContainerNo input will be fed from the ContainerNo Control in your first Form, this is why the names and data types must match.

68

LAB 4: Modify Operation Properties, part 2

- In the Outputs, add:
  - ProductDesc as **Char**
  - LotNo as **Char**
  - QuantityOnHand as **Decimal**
  - ParentContainer as **Char**
  - Location as **Char**
  - ContainerStatusDesc as **Char**
  - ProductNo as **Char**

You may recognize these are names of the Controls in your second Screen. These variables will feed the data into that Screen. Therefore the names and data types must match, too.

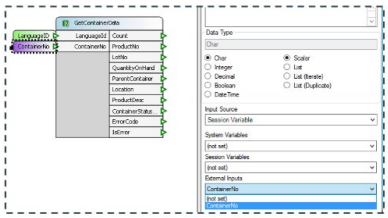


69

LAB 4: Modify GetContainerData, part 1

- Go back to the Operation Step, and map the inputs and outputs to the GetContainerData Operation
- Right-click on ContainerNo Input
- In the Input Properties, expand the External Inputs list, and select **ContainerNo**

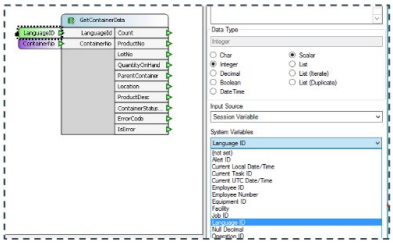
Once the Inputs and Outputs are defined in the Operation Interface, they should be mapped to the Operations's business logic, i.e. to the Functions.



70

LAB 4: Modify GetContainerData, part 2

- For the LanguageId Input, in the Properties, expand the System Variable drop-down and select LanguageID



System Variables are variables defined in the system, and are populated automatically when the DELMIA Apriso session is started. E.g., LanguageID is populated when you logon to the system. The system recognizes your user ID, and the preferred language associated with it.

71

LAB 4: Modify GetContainerData, part 3

To add external output to ProductNo output in the GetContainerData Function:

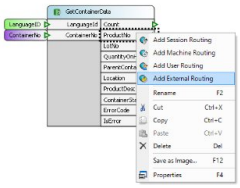
- Right-click on the ProductNo output, and select **Add External Routing**

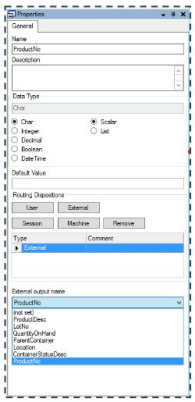
The system will automatically give the external routing the name of the Function output, i.e. ProductNo.

To make sure you have the correct mapping:

- Go to Function Properties, and select the correct External output name

The outputs available there are the ones you added in the Operation Interface.



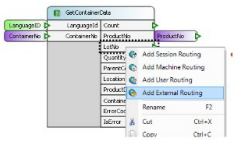


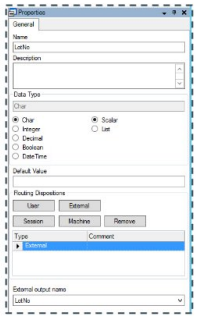
72

LAB 4: Modify GetContainerData, part 4

- Add external routings to the remaining Function outputs:
  - LotNo as LotNo
  - QuantityOnHand as QuantityOnHand
  - ParentContainer as ParentContainer
  - Location as Location
  - ProductDesc as ProductDesc
  - ContainerStatusDesc as ContainerStatusDesc

The Count output remains unmapped as you are not using it at this moment.

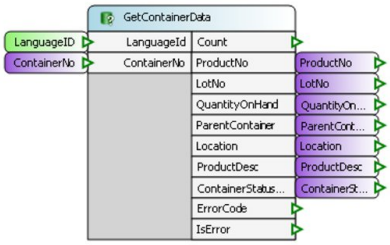




LAB 4: Modify GetContainerData, part 5

The GetContainerData Function should look like this now.

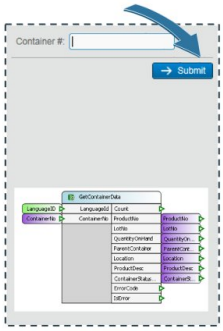
- Save your work
- Change the Operation Status to Prototype



LAB 4: Introduction to Linking Screens, part 1

You have now 2 Screens, and an Operation which uses the Container number as input and returns container details which you want to show in the second Screen.

The next step you need to do is to link the Screens and the Operation into a working flow.

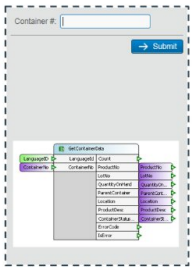


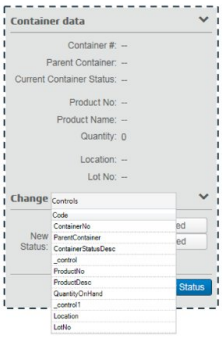
LAB 4: Introduction to Linking Screens, part 2

Your Operation will be added as an On Action Operation to the Submit button on the first screen.

On the second screen no configuration will be needed because the container detail Controls have names identical to the external variables populated by the Operation linked to the Submit button.

The system will recognize it and it will show the data when the second screen is loaded.

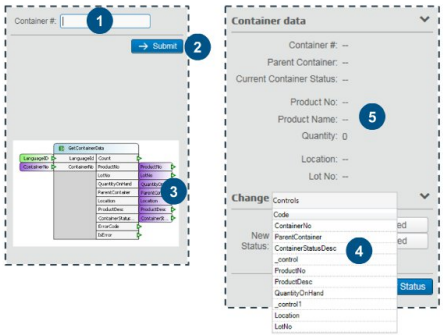




LAB 4: Lab Steps Execution

When fully configured, the sequence of execution will be as follows:

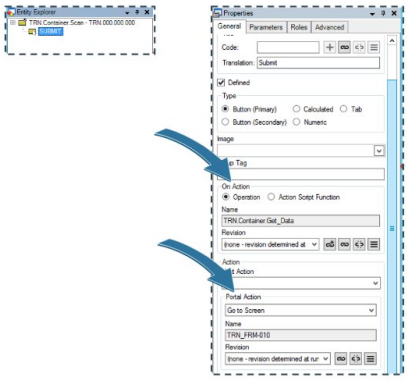
1. User enters container information
2. User clicks on Submit
3. When Submit button is clicked, the Operation is executed and external variables with container details are populated
4. The second Screen is triggered, and the Form Controls take the values from the respective external variables
5. These values are displayed on the second Screen



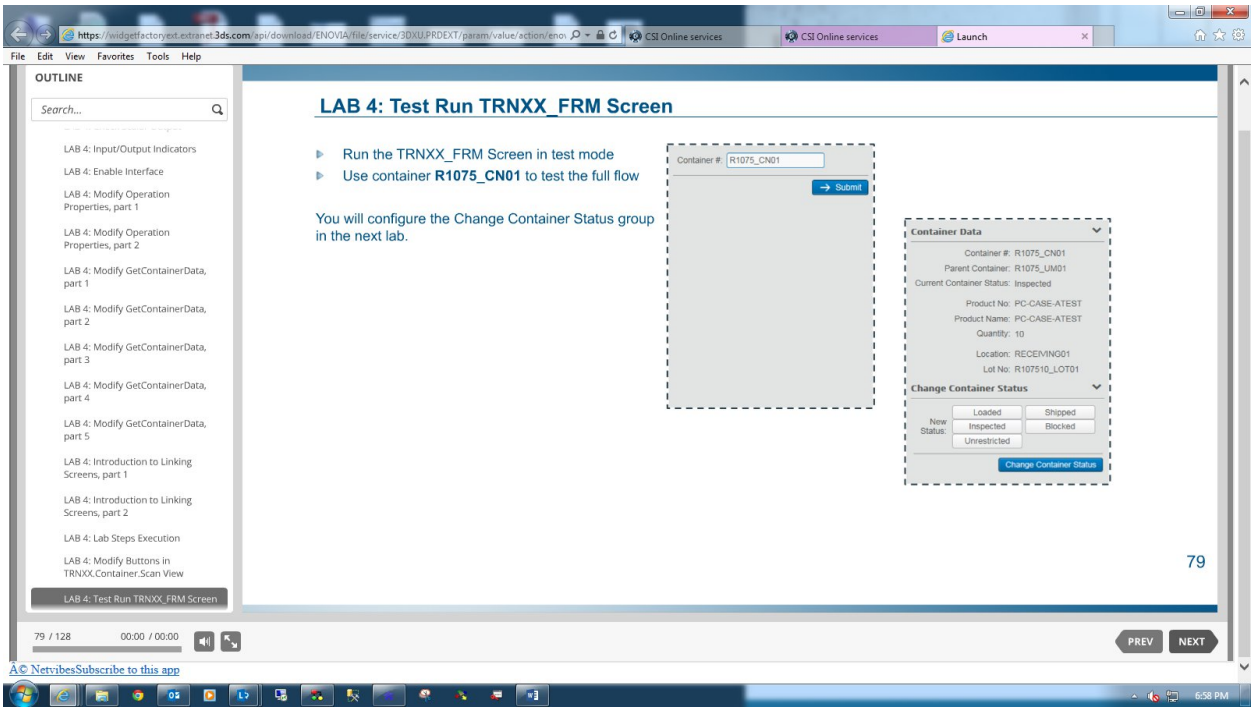
77

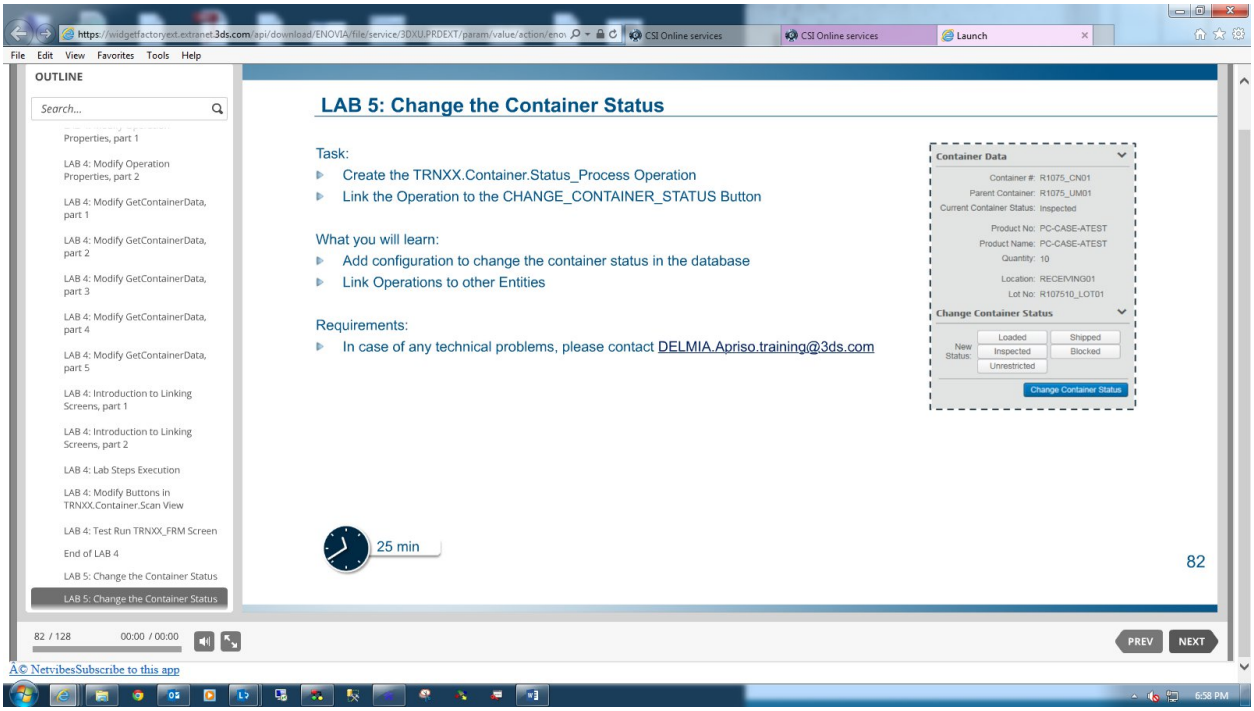
LAB 4: Modify Buttons in TRNXX.Container.Scan View

- Go to TRNXX_FRM Screen, and open the **TRNXX.Container.Scan** View
- Go to the **SUBMIT** Button properties
- In the On Action Operation property, link the **TRNXX.Container.Get_Data** Operation
- In the **Portal Action** property, use the „Go to Screen" Portal Action to link the TRNXX_FRM-010 Screen
- Save your work
- Make sure both your Screens and the Operation are in the Prototype status



78


File Edit View Favorites Tools Help
OUTLINE
Search...
LAB 4: Input/Output Indicators
LAB 4: Enable Interface
LAB 4: Modify Operation Properties, part 1
LAB 4: Modify Operation Properties, part 2
LAB 4: Modify GetContainerData, part 1
LAB 4: Modify GetContainerData, part 2
LAB 4: Modify GetContainerData, part 3
LAB 4: Modify GetContainerData, part 4
LAB 4: Modify GetContainerData, part 5
LAB 4: Introduction to Linking Screens, part 1
LAB 4: Introduction to Linking Screens, part 2
LAB 4: Lab Steps Execution
LAB 4: Modify Buttons in TRNXX.Container.Scan View
LAB 4: Test Run TRNXX_FRM Screen
LAB 4: Test Run TRNXX_FRM Screen
Run the TRNXX_FRM Screen in test mode
Use container R1075_CN01 to test the full flow
You will configure the Change Container Status group in the next lab.
Container #: R1075_CN01
Submit
Container Data
Container #: R1075_CN01
Parent Container: R1075_UM01
Current Container Status: Inspected
Product No: PC-CASE-ATEST
Product Name: PC-CASE-ATEST
Quantity: 10
Location: RECEIVING01
Lot No: R107510_LOT01
Change Container Status
New Status:
Loaded
Shipped
Inspected
Blocked
Unrestricted
Change Container Status
79
79 / 128
00:00 / 00:00
PREV
NEXT
Â© NetvibesSubscribe to this app
6:58 PM


File Edit View Favorites Tools Help
OUTLINE
Search...
Properties, part 1
LAB 4: Modify Operation Properties, part 2
LAB 4: Modify GetContainerData, part 1
LAB 4: Modify GetContainerData, part 2
LAB 4: Modify GetContainerData, part 3
LAB 4: Modify GetContainerData, part 4
LAB 4: Modify GetContainerData, part 5
LAB 4: Introduction to Linking Screens, part 1
LAB 4: Introduction to Linking Screens, part 2
LAB 4: Lab Steps Execution
LAB 4: Modify Buttons in TRNXX.Container.Scan View
LAB 4: Test Run TRNXX_FRM Screen
End of LAB 4
LAB 5: Change the Container Status
LAB 5: Change the Container Status
Task:
Create the TRNXX.Container.Status_Process Operation
Link the Operation to the CHANGE_CONTAINER_STATUS Button
What you will learn:
Add configuration to change the container status in the database
Link Operations to other Entities
Requirements:
In case of any technical problems, please contact DELMIA.Apriso.training@3ds.com
Container Data
Container #: R1075_CN01
Parent Container: R1075_UM01
Current Container Status: Inspected
Product No: PC-CASE-ATEST
Product Name: PC-CASE-ATEST
Quantity: 10
Location: RECEIVING01
Lot No: R107510_LOT01
Change Container Status
New Status:
Loaded
Shipped
Inspected
Blocked
Unrestricted
Change Container Status
25 min
82
82 / 128
00:00 / 00:00
PREV
NEXT
Â© NetvibesSubscribe to this app
6:58 PM

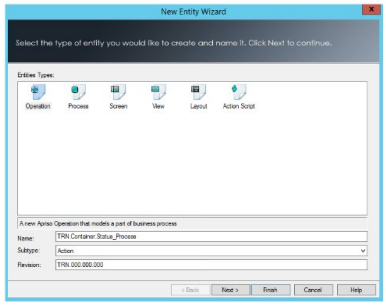
LAB 5: Introduction to Container Status

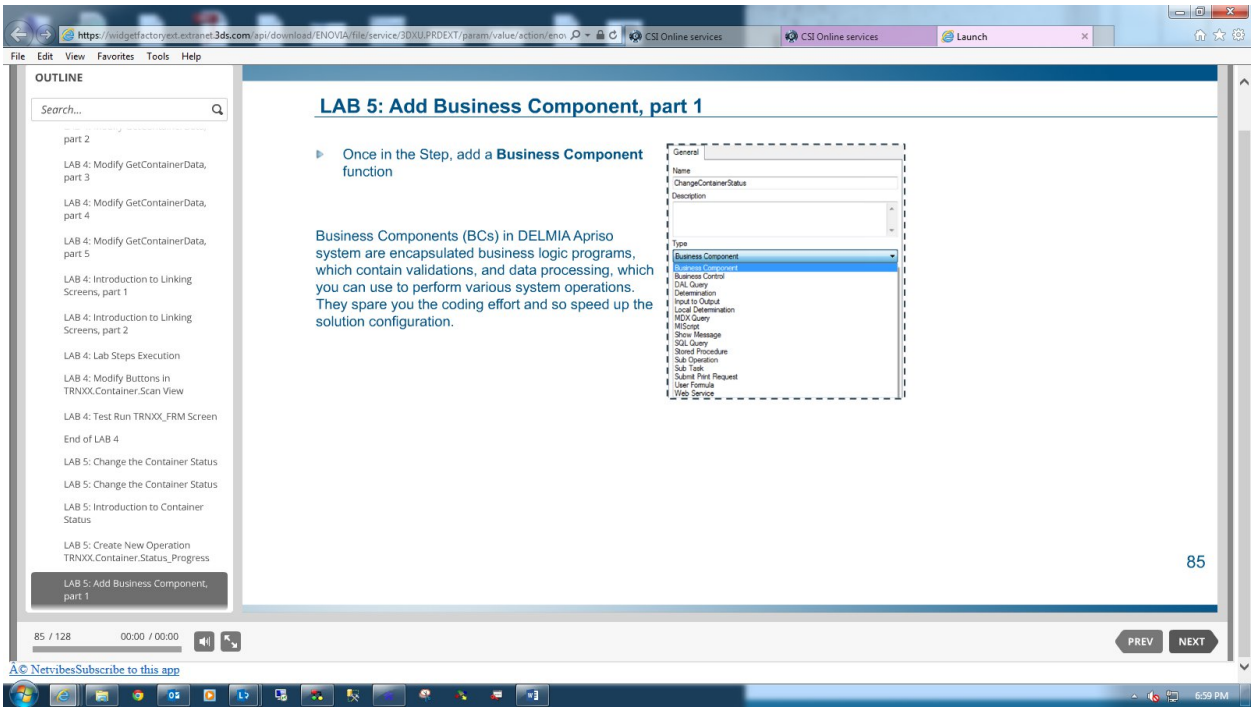
The new container status will be determined by clicking on one of the State Buttons.

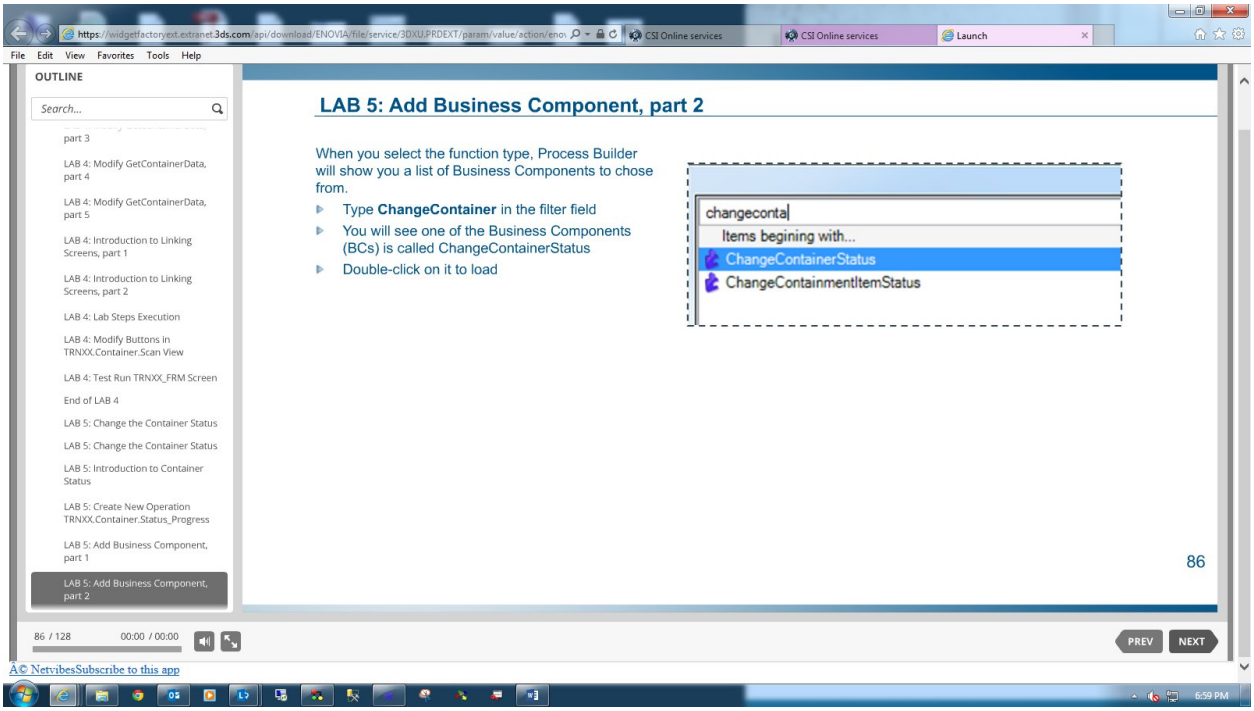
To save the container status change in the database, you will need a Standard Operation, which will take the Container number and the new status, and update the system.

LAB 5: Create New Operation TRNXX.Container.Status_Progress

- Create a new Operation, and name it **TRNXX.Container.Status_Process**
- Make sure the subtype is **Action**
- Rename the **Step** to **UpdateContainerStatus**
- Make sure the Step is connected to the End Node
- Double-click on the Step to open it




File Edit View Favorites Tools Help
OUTLINE
Search...
part 2
LAB 4: Modify GetContainerData, part 3
LAB 4: Modify GetContainerData, part 4
LAB 4: Modify GetContainerData, part 5
LAB 4: Introduction to Linking Screens, part 1
LAB 4: Introduction to Linking Screens, part 2
LAB 4: Lab Steps Execution
LAB 4: Modify Buttons in TRNXX.Container.Scan View
LAB 4: Test Run TRNXX_FRM Screen
End of LAB 4
LAB 5: Change the Container Status
LAB 5: Change the Container Status
LAB 5: Introduction to Container Status
LAB 5: Create New Operation TRNXX.Container.Status_Progress
LAB 5: Add Business Component, part 1
LAB 5: Add Business Component, part 1
Once in the Step, add a Business Component function
Business Components (BCs) in DELMIA Apriso system are encapsulated business logic programs, which contain validations, and data processing, which you can use to perform various system operations. They spare you the coding effort and so speed up the solution configuration.
General
Name
ChangeContainerStatus
Description
Type
Business Component
Business Component
Business Control
DAL Query
Determination
Input to Output
Local Determination
MDX Query
MEScript
Show Message
SQL Query
Stored Procedure
Sub Operation
Sub Task
Submit Print Request
User Formula
Web Service
85
85 / 128
00:00 / 00:00
PREV
NEXT
A© NetvibesSubscribe to this app


File Edit View Favorites Tools Help
OUTLINE
Search...
part 3
LAB 4: Modify GetContainerData, part 4
LAB 4: Modify GetContainerData, part 5
LAB 4: Introduction to Linking Screens, part 1
LAB 4: Introduction to Linking Screens, part 2
LAB 4: Lab Steps Execution
LAB 4: Modify Buttons in TRNXX.Container.Scan View
LAB 4: Test Run TRNXX_FRM Screen
End of LAB 4
LAB 5: Change the Container Status
LAB 5: Change the Container Status
LAB 5: Introduction to Container Status
LAB 5: Create New Operation TRNXX.Container.Status_Progress
LAB 5: Add Business Component, part 1
LAB 5: Add Business Component, part 2
LAB 5: Add Business Component, part 2
When you select the function type, Process Builder will show you a list of Business Components to chose from.
Type ChangeContainer in the filter field
You will see one of the Business Components (BCs) is called ChangeContainerStatus
Double-click on it to load
changeconta
Items begining with...
ChangeContainerStatus
ChangeContainmentItemStatus
86
86 / 128
00:00 / 00:00
PREV
NEXT
A© NetvibesSubscribe to this app

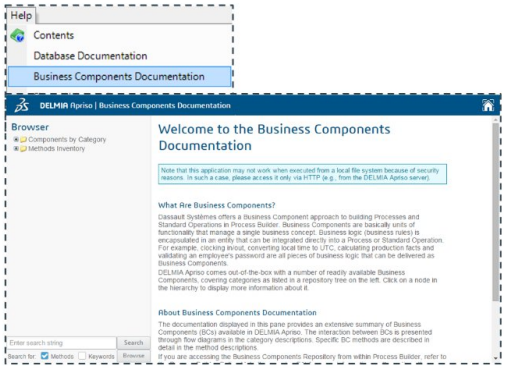
LAB 5: Add Business Component, part 3

In this training, the BC to use is given in the instruction, but Apriso has several hundred Business Components, and how do you which one does what?

The answer is in the Business Components Documentation.

You can access it from the Process Builder menu bar, in the Help section.

Business Components will be addressed in more detail in later training modules.



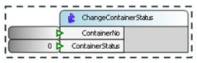
87

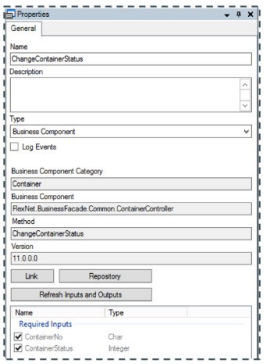
LAB 5: Function Details

The Function you added has just two inputs.

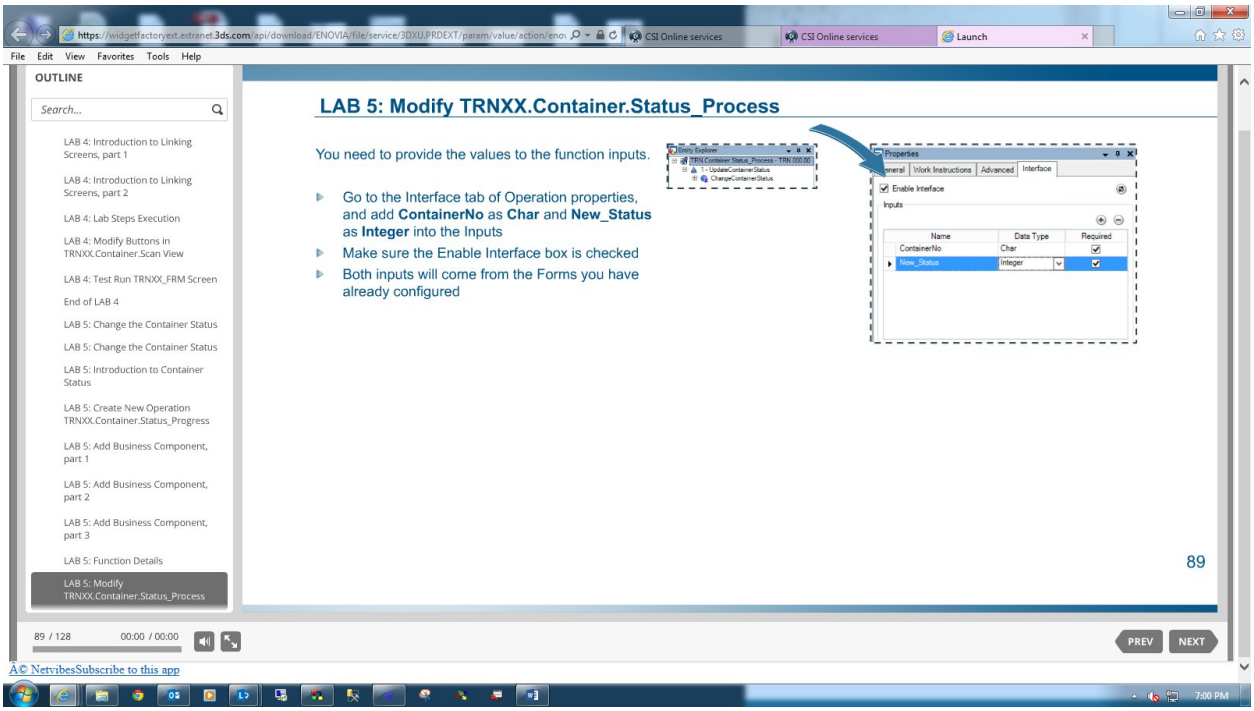
The System applied the BC name to name your function. You don't need to change this.

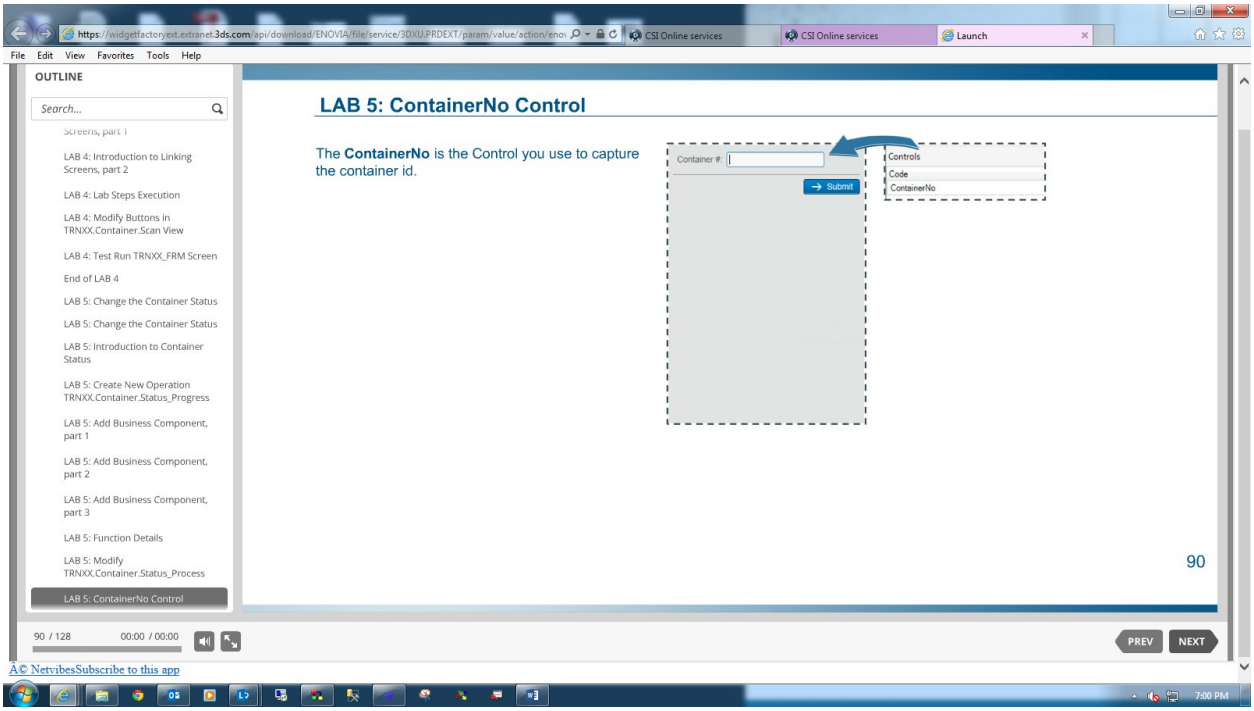
If you want to check what this BC does, use the Repository button in the function properties. It will take you to the Business Components Documentation which was just mentioned on previous screen.





88

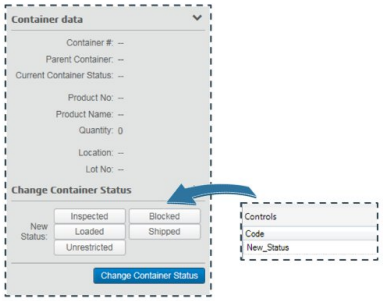

File Edit View Favorites Tools Help
OUTLINE
Search...
LAB 4: Introduction to Linking Screens, part 1
LAB 4: Introduction to Linking Screens, part 2
LAB 4: Lab Steps Execution
LAB 4: Modify Buttons in TRNXX.Container.Scan View
LAB 4: Test Run TRNXX_FRM Screen
End of LAB 4
LAB 5: Change the Container Status
LAB 5: Change the Container Status
LAB 5: Introduction to Container Status
LAB 5: Create New Operation TRNXX.Container.Status_Progress
LAB 5: Add Business Component, part 1
LAB 5: Add Business Component, part 2
LAB 5: Add Business Component, part 3
LAB 5: Function Details
LAB 5: Modify TRNXX.Container.Status_Process
LAB 5: Modify TRNXX.Container.Status_Process
You need to provide the values to the function inputs.
Go to the Interface tab of Operation properties, and add ContainerNo as Char and New_Status as Integer into the Inputs
Make sure the Enable Interface box is checked
Both inputs will come from the Forms you have already configured
Properties
General
Work Instructions
Advanced
Interface
Enable Interface
Inputs
Name
Data Type
Required
ContainerNo
Char
New_Status
Integer
89
89 / 128
00:00 / 00:00
PREV
NEXT
7:00 PM


File Edit View Favorites Tools Help
OUTLINE
Search...
LAB 4: Introduction to Linking Screens, part 2
LAB 4: Lab Steps Execution
LAB 4: Modify Buttons in TRNXX.Container.Scan View
LAB 4: Test Run TRNXX_FRM Screen
End of LAB 4
LAB 5: Change the Container Status
LAB 5: Change the Container Status
LAB 5: Introduction to Container Status
LAB 5: Create New Operation TRNXX.Container.Status_Progress
LAB 5: Add Business Component, part 1
LAB 5: Add Business Component, part 2
LAB 5: Add Business Component, part 3
LAB 5: Function Details
LAB 5: Modify TRNXX.Container.Status_Process
LAB 5: ContainerNo Control
LAB 5: ContainerNo Control
The ContainerNo is the Control you use to capture the container id.
Container #:
Submit
Controls
Code
ContainerNo
90
90 / 128
00:00 / 00:00
PREV
NEXT
7:00 PM

LAB 5: New_Status Control

The **New_Status** is the name of the Control with State Buttons.

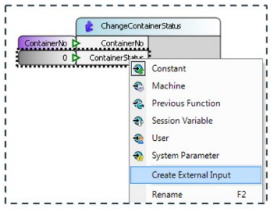
The input to the Operation needs to be Integer, because this is the data type of container status in the database.

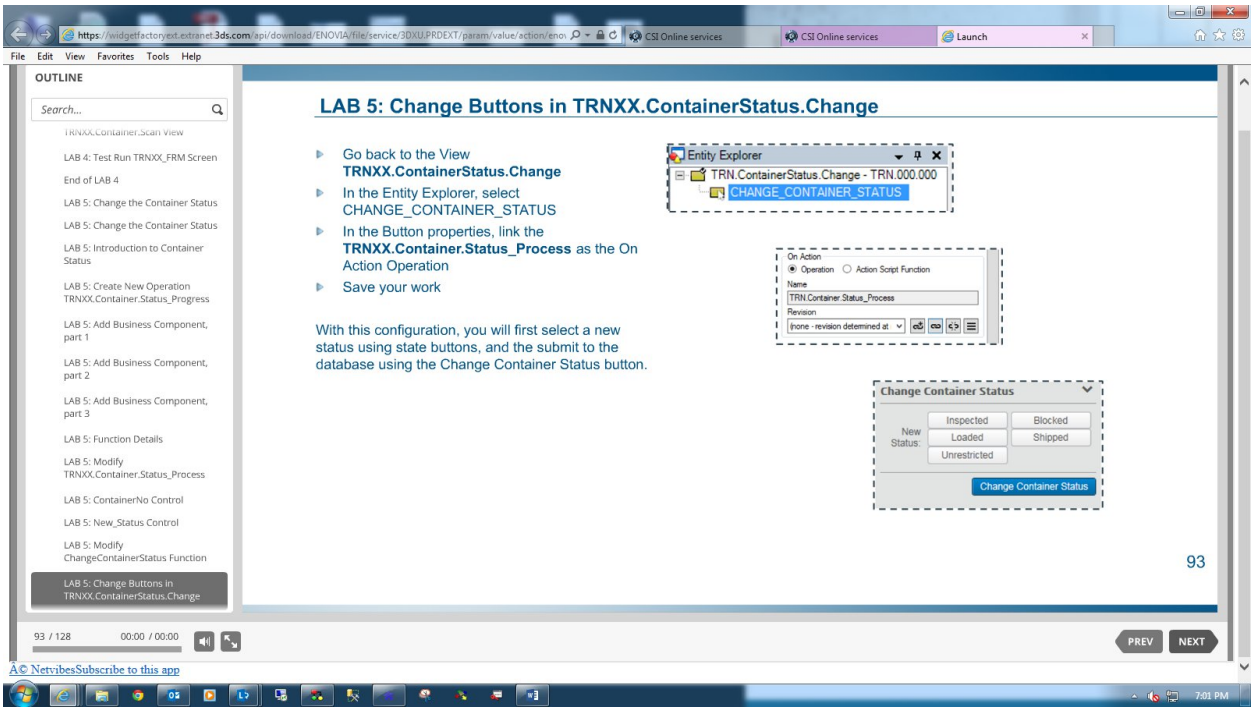


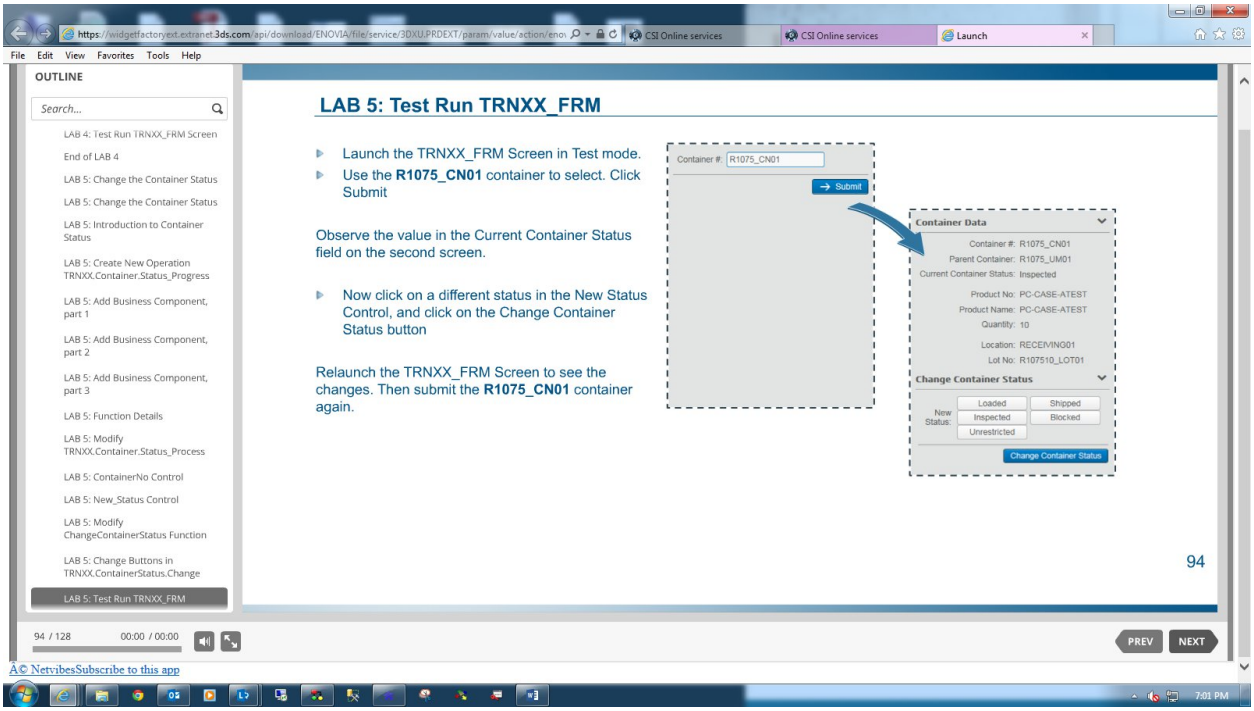
LAB 5: Modify ChangeContainerStatus Function

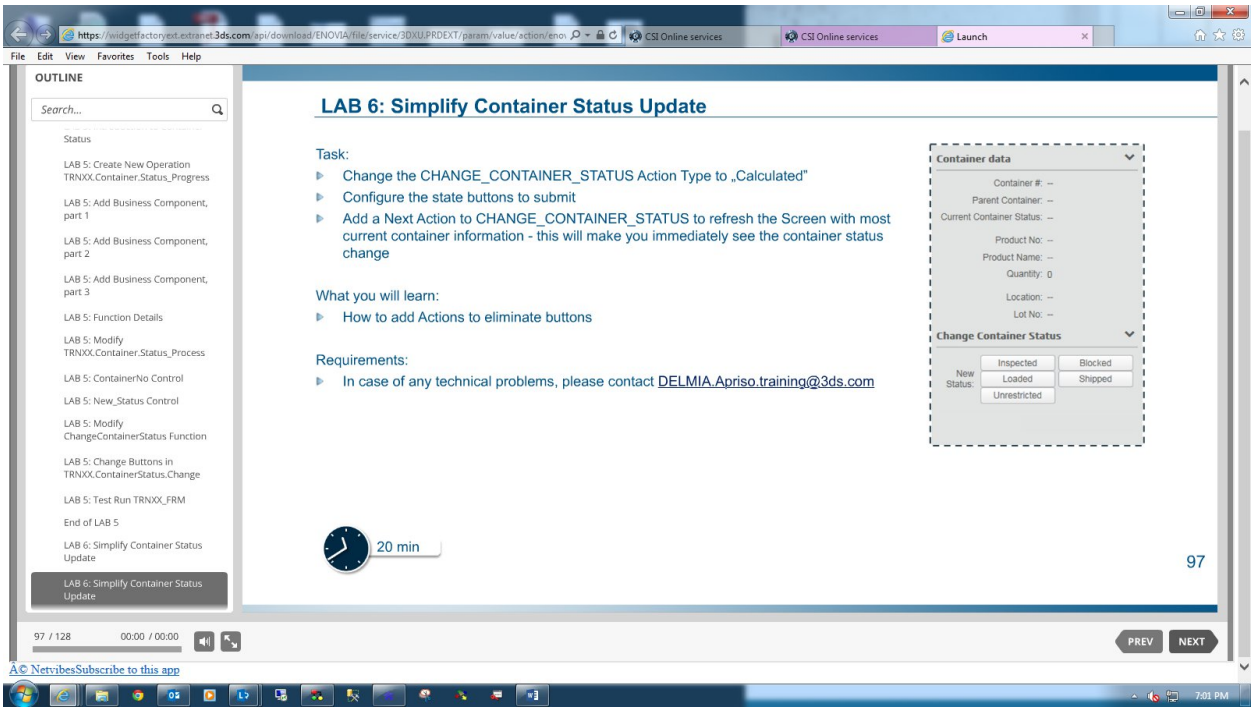
Now map the external inputs into the function:

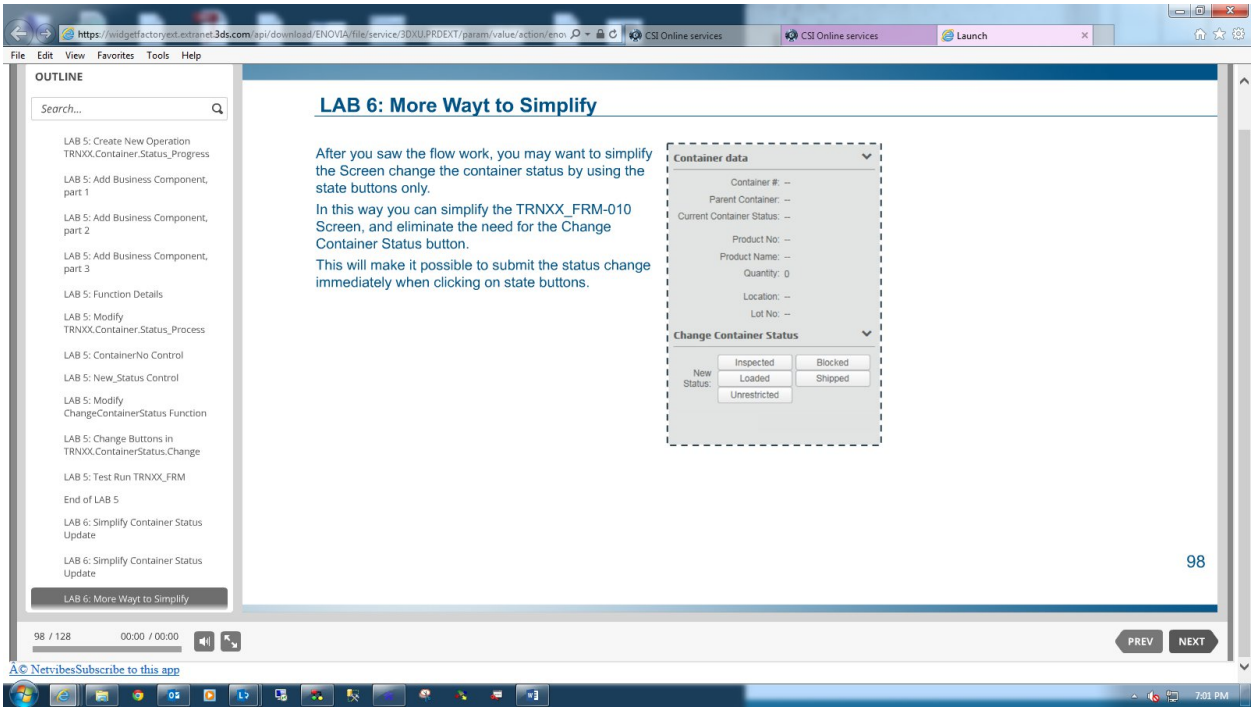
- Right-click on each input, and select **Create External Input**
- Then, in the Input properties, select New_Status external input from the External Inputs drop-down list
- Save the Operation
- Change to Prototype Status

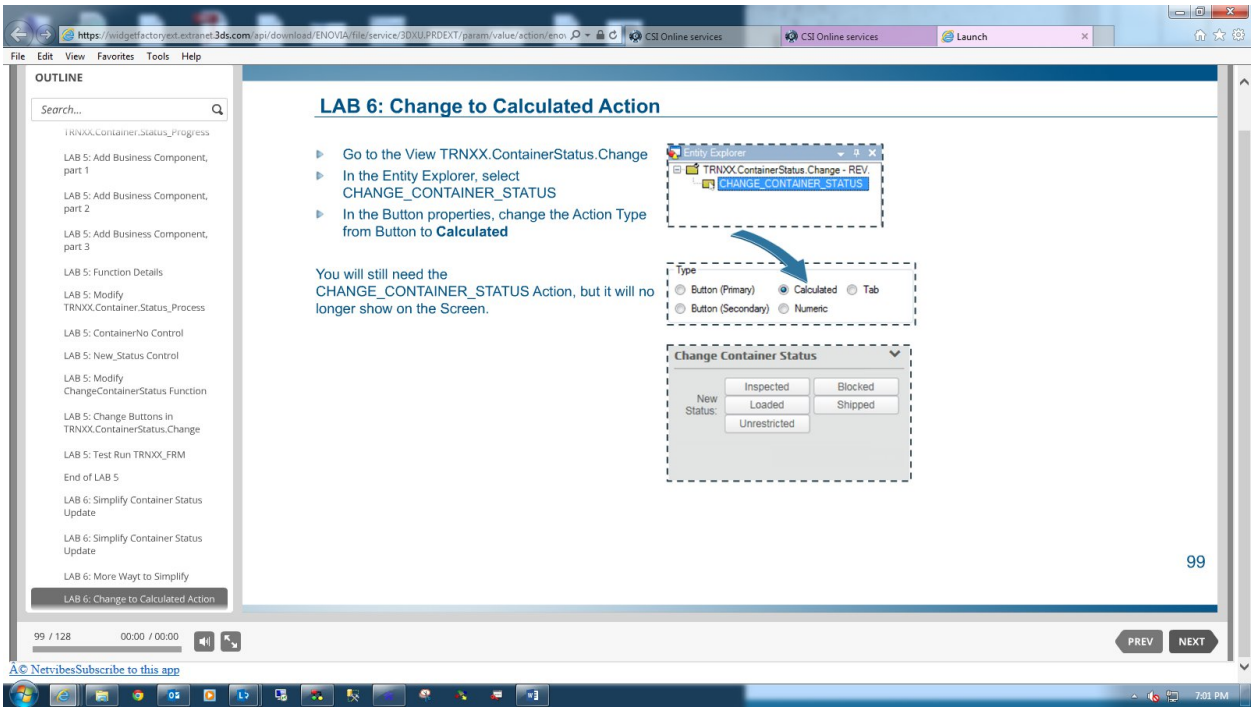


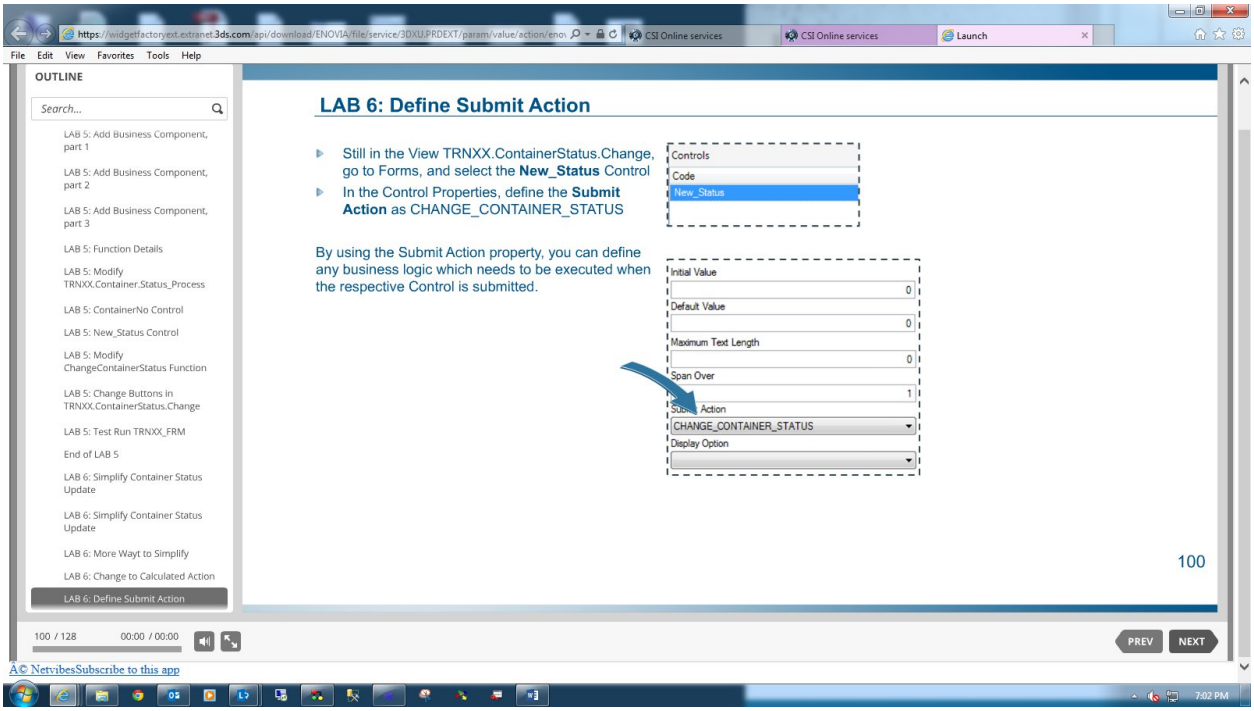

File Edit View Favorites Tools Help
OUTLINE
Search...
LAB 4: Test Run TRNXX_FRM Screen
End of LAB 4
LAB 5: Change the Container Status
LAB 5: Change the Container Status
LAB 5: Introduction to Container Status
LAB 5: Create New Operation TRNXX.Container.Status_Progress
LAB 5: Add Business Component, part 1
LAB 5: Add Business Component, part 2
LAB 5: Add Business Component, part 3
LAB 5: Function Details
LAB 5: Modify TRNXX.Container.Status_Process
LAB 5: ContainerNo Control
LAB 5: New_Status Control
LAB 5: Modify ChangeContainerStatus Function
LAB 5: Change Buttons in TRNXX.ContainerStatus.Change
LAB 5: Change Buttons in TRNXX.ContainerStatus.Change
Go back to the View TRNXX.ContainerStatus.Change
In the Entity Explorer, select CHANGE_CONTAINER_STATUS
In the Button properties, link the TRNXX.Container.Status_Process as the On Action Operation
Save your work
With this configuration, you will first select a new status using state buttons, and the submit to the database using the Change Container Status button.
Entity Explorer
TRN.ContainerStatus.Change - TRN.000.000
CHANGE_CONTAINER_STATUS
On Action
Operation
Action Script Function
Name
TRN.Container.Status_Process
Revision
Change Container Status
New Status:
Inspected
Blocked
Loaded
Shipped
Unrestricted
Change Container Status
93
93 / 128
00:00 / 00:00
PREV
NEXT
© NetvibesSubscribe to this app
7:01 PM

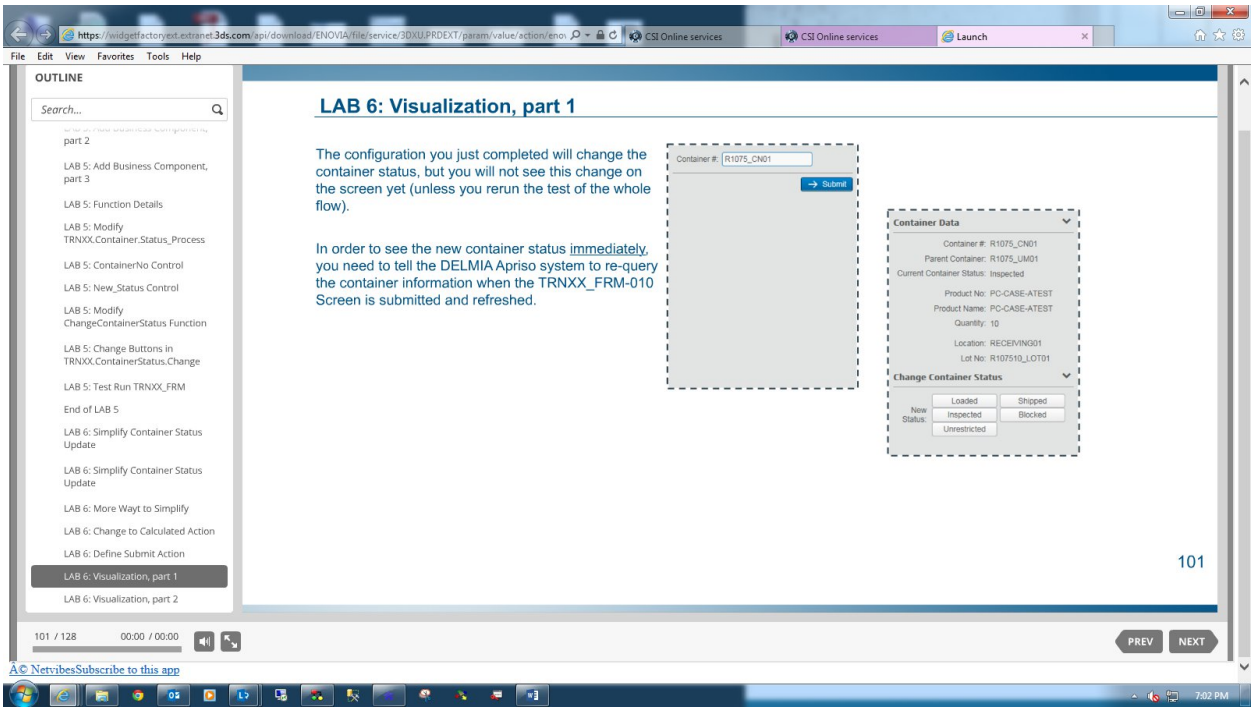

File Edit View Favorites Tools Help
OUTLINE
Search...
LAB 4: Test Run TRNXX_FRM Screen
End of LAB 4
LAB 5: Change the Container Status
LAB 5: Change the Container Status
LAB 5: Introduction to Container Status
LAB 5: Create New Operation TRNXX.Container.Status_Progress
LAB 5: Add Business Component, part 1
LAB 5: Add Business Component, part 2
LAB 5: Add Business Component, part 3
LAB 5: Function Details
LAB 5: Modify TRNXX.Container.Status_Process
LAB 5: ContainerNo Control
LAB 5: New_Status Control
LAB 5: Modify ChangeContainerStatus Function
LAB 5: Change Buttons in TRNXX.ContainerStatus.Change
LAB 5: Test Run TRNXX_FRM
LAB 5: Test Run TRNXX_FRM
Launch the TRNXX_FRM Screen in Test mode.
Use the R1075_CN01 container to select. Click Submit
Observe the value in the Current Container Status field on the second screen.
Now click on a different status in the New Status Control, and click on the Change Container Status button
Relaunch the TRNXX_FRM Screen to see the changes. Then submit the R1075_CN01 container again.
Container #: R1075_CN01
Submit
Container Data
Container #: R1075_CN01
Parent Container: R1075_UM01
Current Container Status: Inspected
Product No: PC-CASE-ATEST
Product Name: PC-CASE-ATEST
Quantity: 10
Location: RECEIVING01
Lot No: R107510_LOT01
Change Container Status
New Status:
Loaded
Shipped
Inspected
Blocked
Unrestricted
Change Container Status
94
94 / 128
00:00 / 00:00
PREV
NEXT
© NetvibesSubscribe to this app
7:01 PM

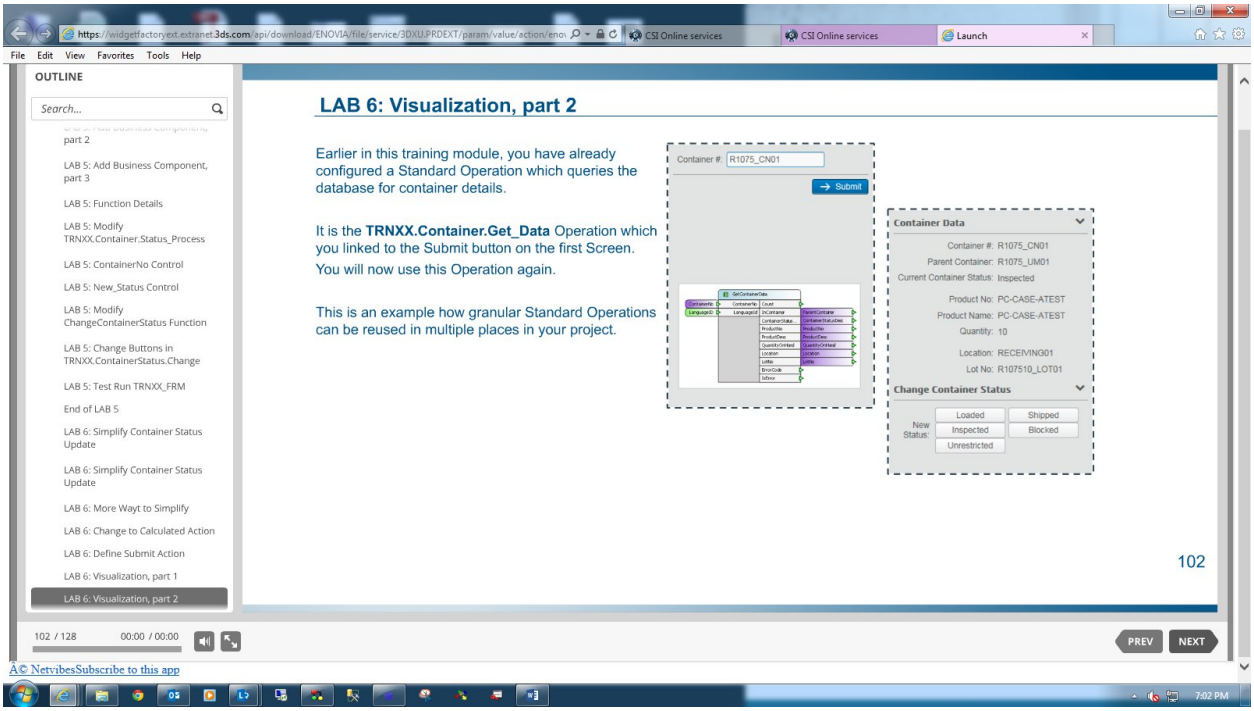

File Edit View Favorites Tools Help
OUTLINE
Search...
Status
LAB 5: Create New Operation TRNXX.Container.Status_Progress
LAB 5: Add Business Component, part 1
LAB 5: Add Business Component, part 2
LAB 5: Add Business Component, part 3
LAB 5: Function Details
LAB 5: Modify TRNXX.Container.Status_Process
LAB 5: ContainerNo Control
LAB 5: New_Status Control
LAB 5: Modify ChangeContainerStatus Function
LAB 5: Change Buttons in TRNXX.ContainerStatus.Change
LAB 5: Test Run TRNXX_FRM
End of LAB 5
LAB 6: Simplify Container Status Update
LAB 6: Simplify Container Status Update
LAB 6: Simplify Container Status Update
LAB 6: Simplify Container Status Update
Task:
Change the CHANGE_CONTAINER_STATUS Action Type to „Calculated"
Configure the state buttons to submit
Add a Next Action to CHANGE_CONTAINER_STATUS to refresh the Screen with most current container information - this will make you immediately see the container status change
What you will learn:
How to add Actions to eliminate buttons
Requirements:
In case of any technical problems, please contact DELMIA.Apriso.training@3ds.com
Container data
Container #: --
Parent Container: --
Current Container Status: --
Product No: --
Product Name: --
Quantity: 0
Location: --
Lot No: --
Change Container Status
New Status:
Inspected
Blocked
Loaded
Shipped
Unrestricted
20 min
97
97 / 128
00:00 / 00:00
PREV
NEXT
Â© NetvibesSubscribe to this app


File Edit View Favorites Tools Help
OUTLINE
Search...
LAB 5: Create New Operation TRNXX.Container.Status_Progress
LAB 5: Add Business Component, part 1
LAB 5: Add Business Component, part 2
LAB 5: Add Business Component, part 3
LAB 5: Function Details
LAB 5: Modify TRNXX.Container.Status_Process
LAB 5: ContainerNo Control
LAB 5: New_Status Control
LAB 5: Modify ChangeContainerStatus Function
LAB 5: Change Buttons in TRNXX.ContainerStatus.Change
LAB 5: Test Run TRNXX_FRM
End of LAB 5
LAB 6: Simplify Container Status Update
LAB 6: Simplify Container Status Update
LAB 6: More Wayt to Simplify
LAB 6: More Wayt to Simplify
After you saw the flow work, you may want to simplify the Screen change the container status by using the state buttons only.
In this way you can simplify the TRNXX_FRM-010 Screen, and eliminate the need for the Change Container Status button.
This will make it possible to submit the status change immediately when clicking on state buttons.
Container data
Container #: --
Parent Container: --
Current Container Status: --
Product No: --
Product Name: --
Quantity: 0
Location: --
Lot No: --
Change Container Status
New Status:
Inspected
Blocked
Loaded
Shipped
Unrestricted
98
98 / 128
00:00 / 00:00
PREV
NEXT
Â© NetvibesSubscribe to this app


File Edit View Favorites Tools Help
OUTLINE
Search...
LAB 5: Add Business Component, part 1
LAB 5: Add Business Component, part 2
LAB 5: Add Business Component, part 3
LAB 5: Function Details
LAB 5: Modify TRNXX.Container.Status_Process
LAB 5: ContainerNo Control
LAB 5: New_Status Control
LAB 5: Modify ChangeContainerStatus Function
LAB 5: Change Buttons in TRNXX.ContainerStatus.Change
LAB 5: Test Run TRNXX_FRM
End of LAB 5
LAB 6: Simplify Container Status Update
LAB 6: Simplify Container Status Update
LAB 6: More Wayt to Simplify
LAB 6: Change to Calculated Action
LAB 6: Change to Calculated Action
Go to the View TRNXX.ContainerStatus.Change
In the Entity Explorer, select CHANGE_CONTAINER_STATUS
In the Button properties, change the Action Type from Button to Calculated
You will still need the CHANGE_CONTAINER_STATUS Action, but it will no longer show on the Screen.
Entity Explorer
TRNXX.ContainerStatus.Change - REV.
CHANGE_CONTAINER_STATUS
Type
Button (Primary)
Calculated
Tab
Button (Secondary)
Numeric
Change Container Status
New Status:
Inspected
Blocked
Loaded
Shipped
Unrestricted
99
99 / 128
00:00 / 00:00
PREV
NEXT
7:01 PM

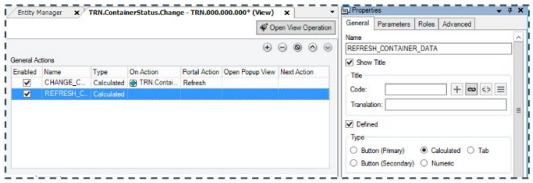

File Edit View Favorites Tools Help
OUTLINE
Search...
LAB 5: Add Business Component, part 1
LAB 5: Add Business Component, part 2
LAB 5: Add Business Component, part 3
LAB 5: Function Details
LAB 5: Modify TRNXX.Container.Status_Process
LAB 5: ContainerNo Control
LAB 5: New_Status Control
LAB 5: Modify ChangeContainerStatus Function
LAB 5: Change Buttons in TRNXX.ContainerStatus.Change
LAB 5: Test Run TRNXX_FRM
End of LAB 5
LAB 6: Simplify Container Status Update
LAB 6: Simplify Container Status Update
LAB 6: More Wayt to Simplify
LAB 6: Change to Calculated Action
LAB 6: Define Submit Action
LAB 6: Define Submit Action
Still in the View TRNXX.ContainerStatus.Change, go to Forms, and select the New_Status Control
In the Control Properties, define the Submit Action as CHANGE_CONTAINER_STATUS
By using the Submit Action property, you can define any business logic which needs to be executed when the respective Control is submitted.
Controls
Code
New_Status
Initial Value
Default Value
Maximum Text Length
Span Over
Submit Action
CHANGE_CONTAINER_STATUS
Display Option
100
100 / 128
00:00 / 00:00
PREV
NEXT
7:02 PM


File Edit View Favorites Tools Help
OUTLINE
Search...
part 2
LAB 5: Add Business Component, part 3
LAB 5: Function Details
LAB 5: Modify TRNXX.Container.Status_Process
LAB 5: ContainerNo Control
LAB 5: New_Status Control
LAB 5: Modify ChangeContainerStatus Function
LAB 5: Change Buttons in TRNXX.ContainerStatus.Change
LAB 5: Test Run TRNXX_FRM
End of LAB 5
LAB 6: Simplify Container Status Update
LAB 6: Simplify Container Status Update
LAB 6: More Wayt to Simplify
LAB 6: Change to Calculated Action
LAB 6: Define Submit Action
LAB 6: Visualization, part 1
LAB 6: Visualization, part 2
LAB 6: Visualization, part 1
The configuration you just completed will change the container status, but you will not see this change on the screen yet (unless you rerun the test of the whole flow).
In order to see the new container status immediately, you need to tell the DELMIA Apriso system to re-query the container information when the TRNXX_FRM-010 Screen is submitted and refreshed.
Container #: R1075_CN01
Submit
Container Data
Container #: R1075_CN01
Parent Container: R1075_UM01
Current Container Status: Inspected
Product No: PC-CASE-ATEST
Product Name: PC-CASE-ATEST
Quantity: 10
Location: RECEIVING01
Lot No: R107510_LOT01
Change Container Status
New Status:
Loaded
Shipped
Inspected
Blocked
Unrestricted
101
101 / 128
00:00 / 00:00
PREV
NEXT
Â© NetvibesSubscribe to this app
7:02 PM


File Edit View Favorites Tools Help
OUTLINE
Search...
part 2
LAB 5: Add Business Component, part 3
LAB 5: Function Details
LAB 5: Modify TRNXX.Container.Status_Process
LAB 5: ContainerNo Control
LAB 5: New_Status Control
LAB 5: Modify ChangeContainerStatus Function
LAB 5: Change Buttons in TRNXX.ContainerStatus.Change
LAB 5: Test Run TRNXX_FRM
End of LAB 5
LAB 6: Simplify Container Status Update
LAB 6: Simplify Container Status Update
LAB 6: More Wayt to Simplify
LAB 6: Change to Calculated Action
LAB 6: Define Submit Action
LAB 6: Visualization, part 1
LAB 6: Visualization, part 2
LAB 6: Visualization, part 2
Earlier in this training module, you have already configured a Standard Operation which queries the database for container details.
It is the TRNXX.Container.Get_Data Operation which you linked to the Submit button on the first Screen.
You will now use this Operation again.
This is an example how granular Standard Operations can be reused in multiple places in your project.
Container #: R1075_CN01
Submit
Container Data
Container #: R1075_CN01
Parent Container: R1075_UM01
Current Container Status: Inspected
Product No: PC-CASE-ATEST
Product Name: PC-CASE-ATEST
Quantity: 10
Location: RECEIVING01
Lot No: R107510_LOT01
Change Container Status
New Status:
Loaded
Shipped
Inspected
Blocked
Unrestricted
102
102 / 128
00:00 / 00:00
PREV
NEXT
Â© NetvibesSubscribe to this app
7:02 PM

LAB 6: Add New Action (Refresh)

- Go to View TRNXX.ContainerStatus.Change
- Go to the Actions Tab
- Add a new Action after the CHANGE_CONTAINER_STATUS Action.
  - Name the new Action **REFRESH_CONTAINER_DATA**
  - Set the Type to **Calculated**

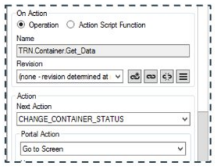


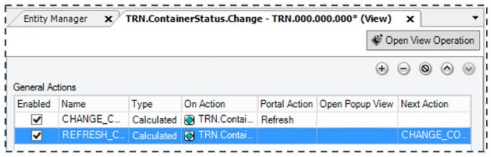
LAB 6: Refresh Action Details

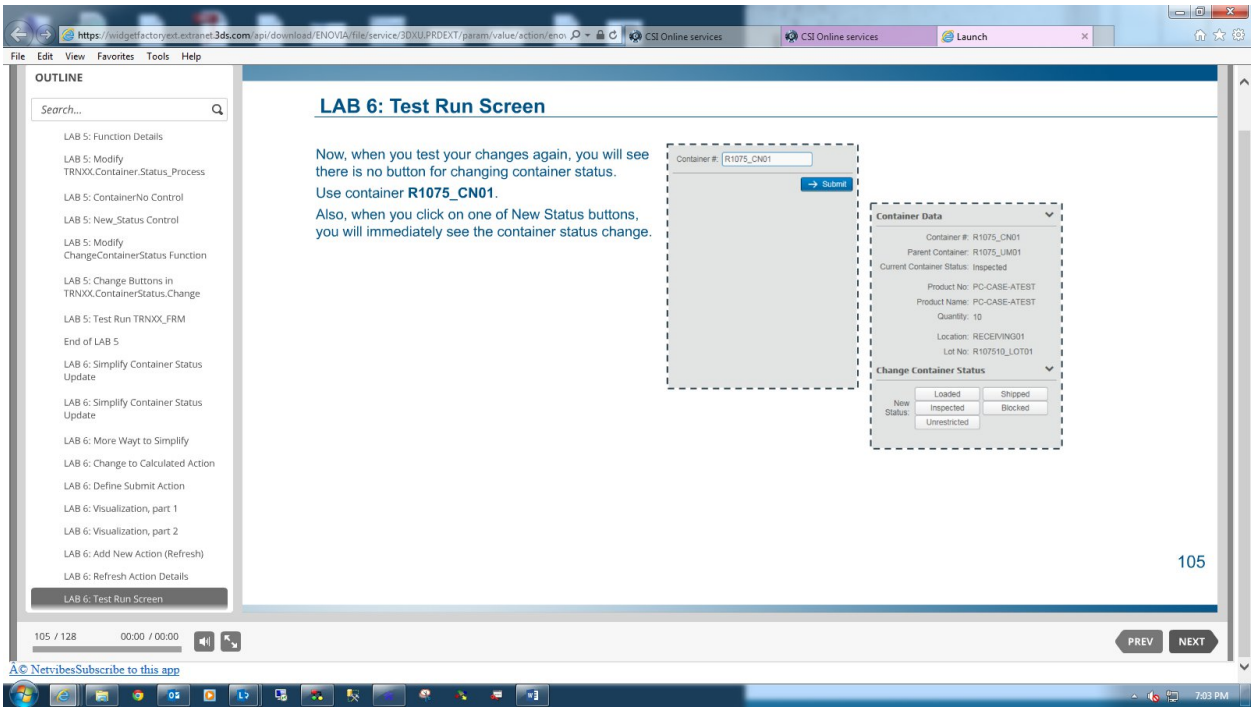
- In the REFRESH_CONTAINER_DATA Action properties, link the **TRNXX.Container.Get_Data** Operation as an On Action Operation
- In the properties of the Action CHANGE_CONTAINER_STATUS, define the **Next Action** as **REFRESH_CONTAINER_DATA**

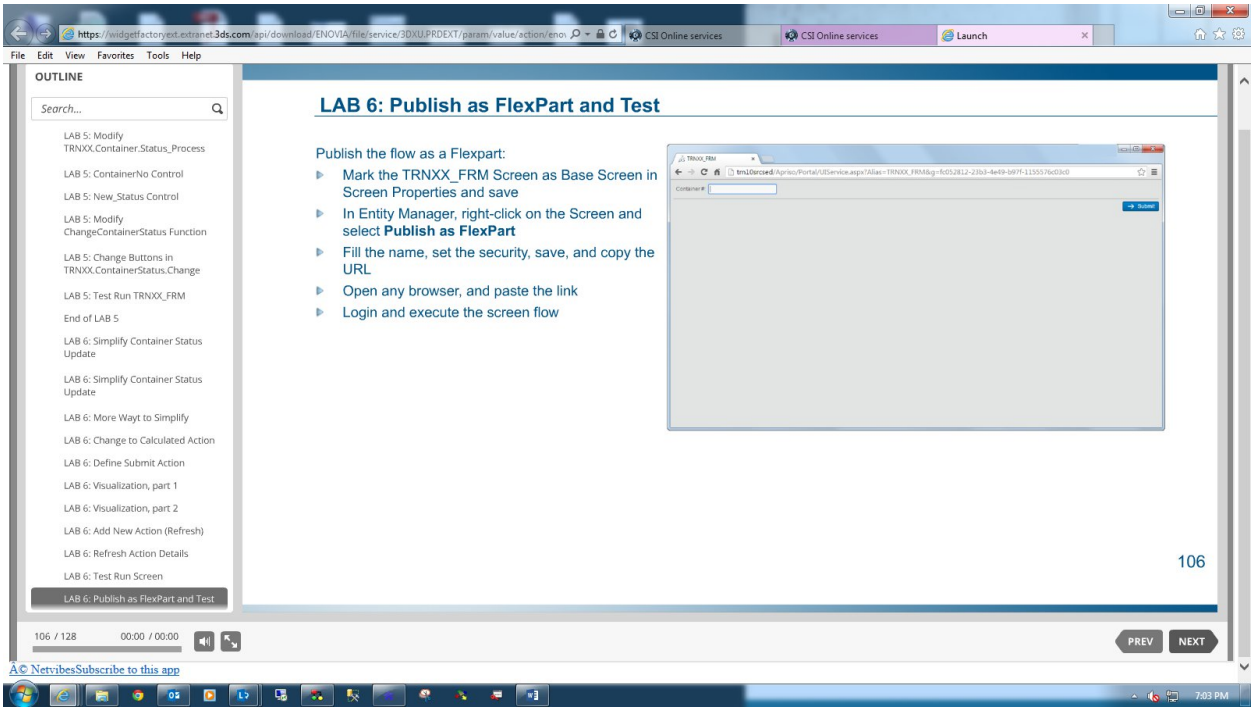
After making these changes, the Actions tab of your View should look like the one here.

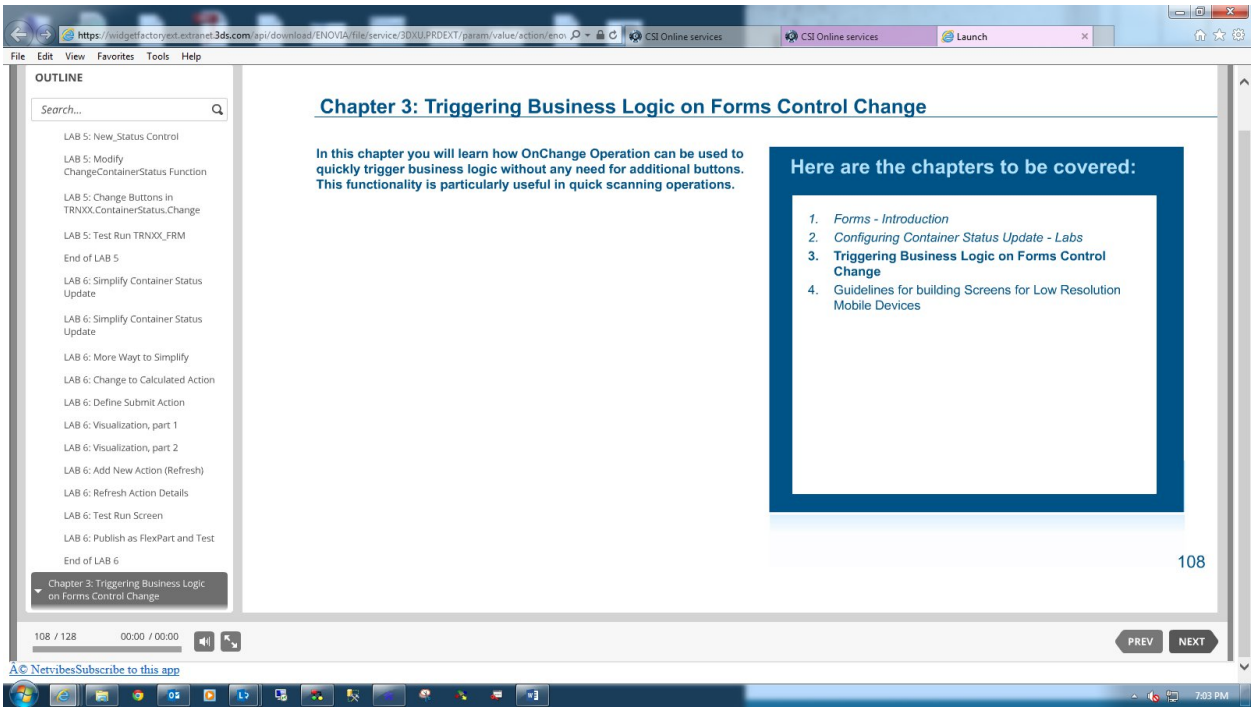
- Save your work, and remember about the Prototype status

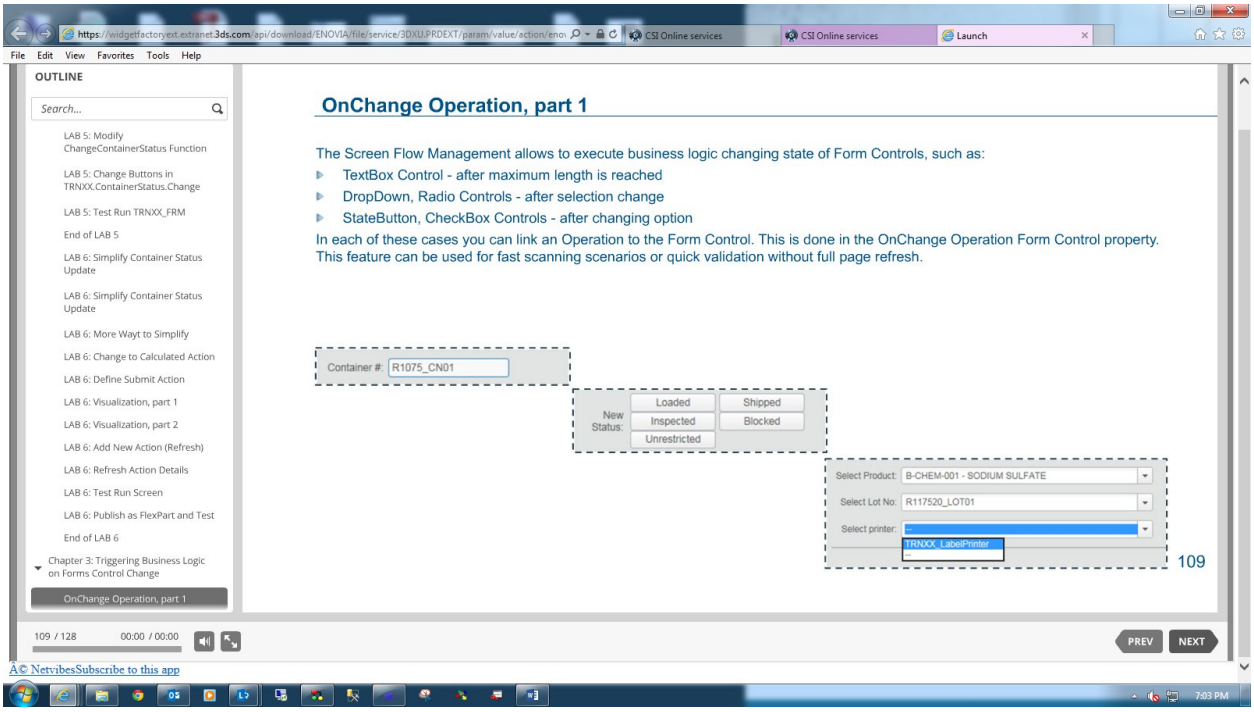





File Edit View Favorites Tools Help
OUTLINE
Search...
LAB 5: Function Details
LAB 5: Modify TRNXX.Container.Status_Process
LAB 5: ContainerNo Control
LAB 5: New_Status Control
LAB 5: Modify ChangeContainerStatus Function
LAB 5: Change Buttons in TRNXX.ContainerStatus.Change
LAB 5: Test Run TRNXX_FRM
End of LAB 5
LAB 6: Simplify Container Status Update
LAB 6: Simplify Container Status Update
LAB 6: More Wayt to Simplify
LAB 6: Change to Calculated Action
LAB 6: Define Submit Action
LAB 6: Visualization, part 1
LAB 6: Visualization, part 2
LAB 6: Add New Action (Refresh)
LAB 6: Refresh Action Details
LAB 6: Test Run Screen
LAB 6: Test Run Screen
Now, when you test your changes again, you will see there is no button for changing container status.
Use container R1075_CN01.
Also, when you click on one of New Status buttons, you will immediately see the container status change.
Container #: R1075_CN01
Submit
Container Data
Container #: R1075_CN01
Parent Container: R1075_UM01
Current Container Status: Inspected
Product No: PC-CASE-ATEST
Product Name: PC-CASE-ATEST
Quantity: 10
Location: RECEIVING01
Lot No: R107510_LOT01
Change Container Status
New Status:
Loaded
Shipped
Inspected
Blocked
Unrestricted
105
105 / 128
00:00 / 00:00
PREV
NEXT
7:03 PM


File Edit View Favorites Tools Help
OUTLINE
Search...
LAB 5: Modify TRNXX.Container.Status_Process
LAB 5: ContainerNo Control
LAB 5: New_Status Control
LAB 5: Modify ChangeContainerStatus Function
LAB 5: Change Buttons in TRNXX.ContainerStatus.Change
LAB 5: Test Run TRNXX_FRM
End of LAB 5
LAB 6: Simplify Container Status Update
LAB 6: Simplify Container Status Update
LAB 6: More Wayt to Simplify
LAB 6: Change to Calculated Action
LAB 6: Define Submit Action
LAB 6: Visualization, part 1
LAB 6: Visualization, part 2
LAB 6: Add New Action (Refresh)
LAB 6: Refresh Action Details
LAB 6: Test Run Screen
LAB 6: Publish as FlexPart and Test
LAB 6: Publish as FlexPart and Test
Publish the flow as a Flexpart:
Mark the TRNXX_FRM Screen as Base Screen in Screen Properties and save
In Entity Manager, right-click on the Screen and select Publish as FlexPart
Fill the name, set the security, save, and copy the URL
Open any browser, and paste the link
Login and execute the screen flow
106
106 / 128
00:00 / 00:00
PREV
NEXT
7:03 PM

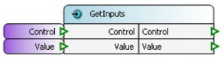

File Edit View Favorites Tools Help
OUTLINE
Search...
LAB 5: New_Status Control
LAB 5: Modify ChangeContainerStatus Function
LAB 5: Change Buttons in TRNXX.ContainerStatus.Change
LAB 5: Test Run TRNXX_FRM
End of LAB 5
LAB 6: Simplify Container Status Update
LAB 6: Simplify Container Status Update
LAB 6: More Wayt to Simplify
LAB 6: Change to Calculated Action
LAB 6: Define Submit Action
LAB 6: Visualization, part 1
LAB 6: Visualization, part 2
LAB 6: Add New Action (Refresh)
LAB 6: Refresh Action Details
LAB 6: Test Run Screen
LAB 6: Publish as FlexPart and Test
End of LAB 6
Chapter 3: Triggering Business Logic on Forms Control Change
Chapter 3: Triggering Business Logic on Forms Control Change
In this chapter you will learn how OnChange Operation can be used to quickly trigger business logic without any need for additional buttons. This functionality is particularly useful in quick scanning operations.
Here are the chapters to be covered:
1. Forms - Introduction
2. Configuring Container Status Update - Labs
3. Triggering Business Logic on Forms Control Change
4. Guidelines for building Screens for Low Resolution Mobile Devices
108
108 / 128
00:00 / 00:00
PREV
NEXT


File Edit View Favorites Tools Help
OUTLINE
Search...
LAB 5: Modify ChangeContainerStatus Function
LAB 5: Change Buttons in TRNXX.ContainerStatus.Change
LAB 5: Test Run TRNXX_FRM
End of LAB 5
LAB 6: Simplify Container Status Update
LAB 6: Simplify Container Status Update
LAB 6: More Wayt to Simplify
LAB 6: Change to Calculated Action
LAB 6: Define Submit Action
LAB 6: Visualization, part 1
LAB 6: Visualization, part 2
LAB 6: Add New Action (Refresh)
LAB 6: Refresh Action Details
LAB 6: Test Run Screen
LAB 6: Publish as FlexPart and Test
End of LAB 6
Chapter 3: Triggering Business Logic on Forms Control Change
OnChange Operation, part 1
OnChange Operation, part 1
The Screen Flow Management allows to execute business logic changing state of Form Controls, such as:
TextBox Control - after maximum length is reached
DropDown, Radio Controls - after selection change
StateButton, CheckBox Controls - after changing option
In each of these cases you can link an Operation to the Form Control. This is done in the OnChange Operation Form Control property. This feature can be used for fast scanning scenarios or quick validation without full page refresh.
Container #: R1075_CN01
New Status:
Loaded
Shipped
Inspected
Blocked
Unrestricted
Select Product: B-CHEM-001 - SODIUM SULFATE
Select Lot No: R117520_LOT01
Select printer:
TRNXX_LabelPrinter
109
109 / 128
00:00 / 00:00
PREV
NEXT

OnChange Operation, part 2

The OnChange Operation needs to be built in a special way:

- Firstly, the Operation subtype needs to be „Change"
- Then it needs to have the following Inputs and Outputs:
  - External Inputs:
    - **Control** - the unique name of the control that triggered the change
    - **Value** - the new value of the control
  - External Outputs
    - **Value** - the value to be put back inside the control
    - **KeepFocus** - if set to True, the focus will be kept on the control (otherwise, it will follow the Autoadvance settings)

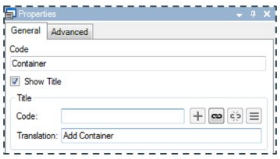


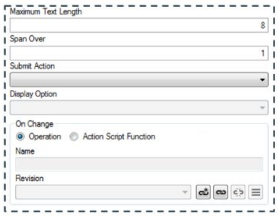
110

Max Length

This is a property used for a Form Control of the Text Box type. It controls the maximum length of the entry.

- The following can happen upon reaching the maximum text length:
  - If the Submit Action property is not set, the focus will be moved to the next Form Control
  - If the Submit Action property is set, the View will submit with the given action
  - If the OnChange Operation is set, the OnChange Operation will be called by AJAX



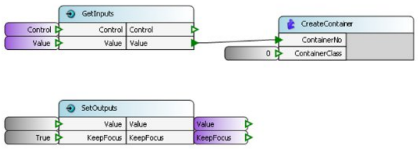


111

LAB 7: OnChange Operation and Max Length Property

Task:

- Create a simple Form Screen to enter new container name
- Create a Standard Operation which will use this container name to create the container in the system
- Use OnChange Operation which will trigger once the maximum length of the container name text field is reached



What you will learn:

- How to create a new Screen using an existing Layout
- How to use existing Views to provide content to your Screens
- How to test run your Screens

Requirements:

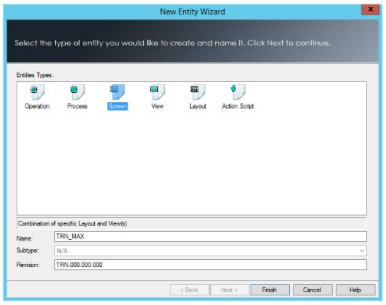
- In case of any technical problems, please contact DELMIA.Apriso.training@3ds.com

15 min

113

LAB 7: Create New Screen TRNXX_MAX

- Create Screen TRNXX_MAX
  - Base Screen: **checked**
  - Layout: **PortalOneWindow**
  - Header: **none**

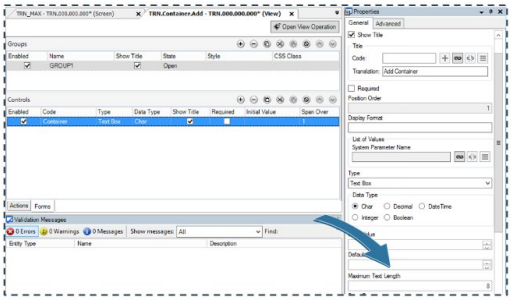


114

LAB 7: Configure TRNXX_MAX

- In the main panel, copy and link PortalForm View to **TRNXX.Container.Add** View
- Go to the new View and delete GROUP2
- In GROUP1, uncheck the **Show Title** box
- In GROUP1, change the **text_ex** Control to:
  - Code: **Container**
  - Title Translation: **Add Container**
  - Type: **Text Box**
  - Data Type: **Char**
  - Max Length: **8**
- In GROUP1, delete the „check_box" Control
- **Delete** the BUTTON_1 Action
- **Delete** the BUTTON_2 Action
- Change the View and the Screen to Prototype

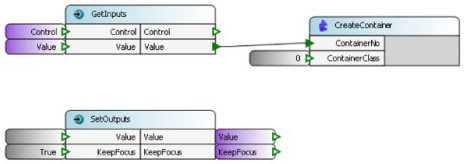
In this case, the Max Length number means, that the container ID needs to have exactly 8 characters, no less and no more. You will see during test run that PB will automatically add the container when the name reaches 8 characters.



115

LAB 7: Configure TRNXX.Container_Add Operation

- Copy the OnChange_Template Operation into a new **TRNXX.Container_Add** Operation
- Make sure it is of Change Subtype
- Add a new Business Component Function
- Link the **CreateContainer** Business Component
- Link the Value output from GetInputs function to the ContainerNo input of the CreateContainer function

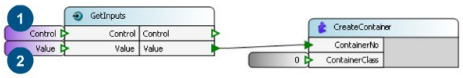


The OnChange Operation is different from other Operations, because other Operations are called on the server side, and OnChange Operations are called on the client side (by AJAX). The technical operation interface requires this Operation to have two inputs per Control it is called from:
1. Control - the Control Code
2. Value - the actual Control's value

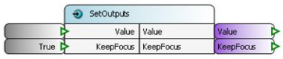
116

LAB 7: GetInputs Controls

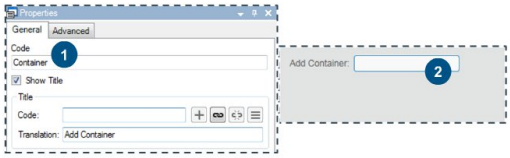
1. The Control input of the GetInputs Function will be the name of the Form Control you are going to use this Operation on, the Container
2. The Value input of this function will be what you will enter in the Add Container box



You don't need to map it manually, because when you attach this Operation to the Container Control, it will automatically recognize where the Control and Value input values come from and what they are

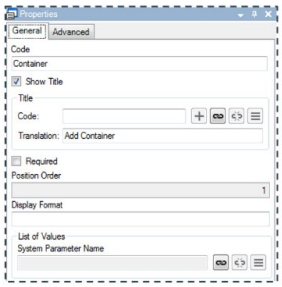


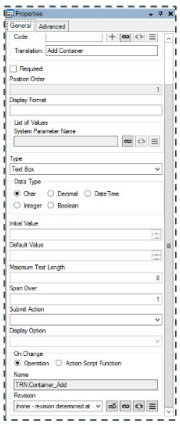
- Save the Operation and set to Prototype



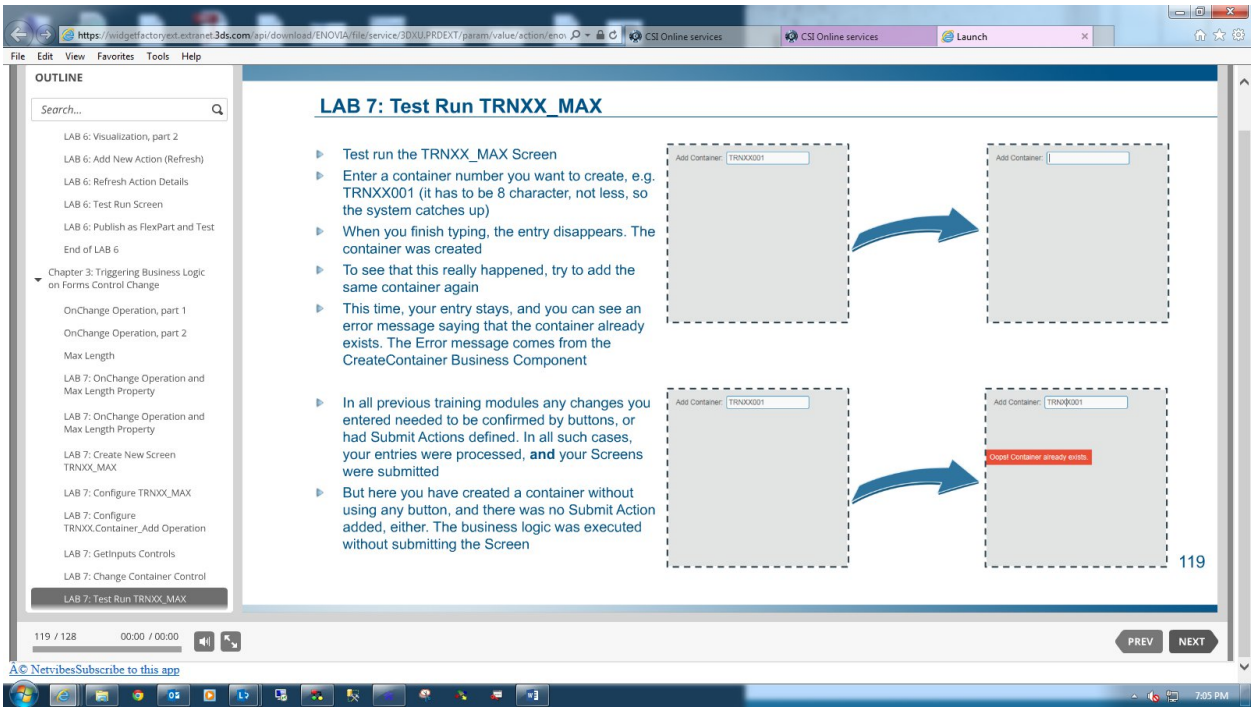
LAB 7: Change Container Control

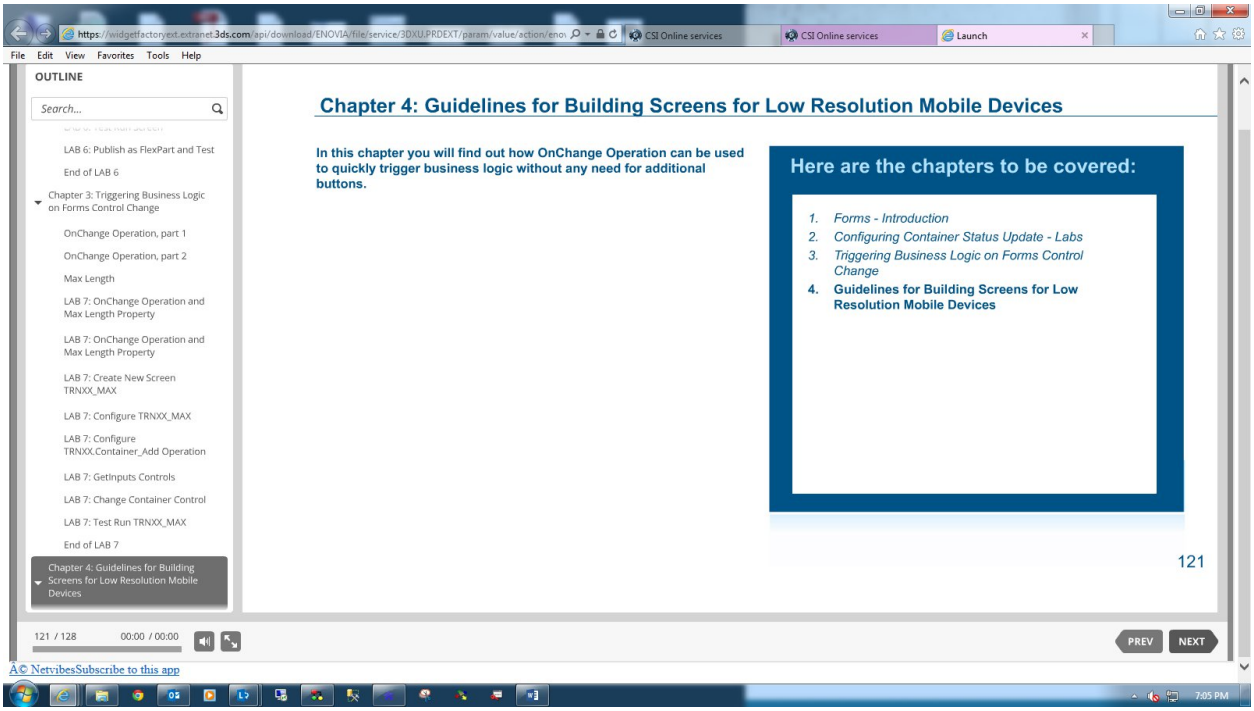
- Go to the TRNXX.Container.Add View
- For the Container Control, link the TRNXX.Container_Add Operation as the OnChange Operation





- Save the View and set it to Prototype
- Save and set to Prototype the TRNXX_MAX Screen, too


File Edit View Favorites Tools Help
OUTLINE
Search...
LAB 6: Visualization, part 2
LAB 6: Add New Action (Refresh)
LAB 6: Refresh Action Details
LAB 6: Test Run Screen
LAB 6: Publish as FlexPart and Test
End of LAB 6
Chapter 3: Triggering Business Logic on Forms Control Change
OnChange Operation, part 1
OnChange Operation, part 2
Max Length
LAB 7: OnChange Operation and Max Length Property
LAB 7: OnChange Operation and Max Length Property
LAB 7: Create New Screen TRNXX_MAX
LAB 7: Configure TRNXX_MAX
LAB 7: Configure TRNXX.Container_Add Operation
LAB 7: GetInputs Controls
LAB 7: Change Container Control
LAB 7: Test Run TRNXX_MAX
LAB 7: Test Run TRNXX_MAX
Test run the TRNXX_MAX Screen
Enter a container number you want to create, e.g. TRNXX001 (it has to be 8 character, not less, so the system catches up)
When you finish typing, the entry disappears. The container was created
To see that this really happened, try to add the same container again
This time, your entry stays, and you can see an error message saying that the container already exists. The Error message comes from the CreateContainer Business Component
Add Container: TRNXX001
Add Container:
In all previous training modules any changes you entered needed to be confirmed by buttons, or had Submit Actions defined. In all such cases, your entries were processed, and your Screens were submitted
But here you have created a container without using any button, and there was no Submit Action added, either. The business logic was executed without submitting the Screen
Add Container: TRNXX001
Add Container: TRNXX001
Oops! Container already exists.
119
119 / 128
00:00 / 00:00
PREV
NEXT
7:05 PM

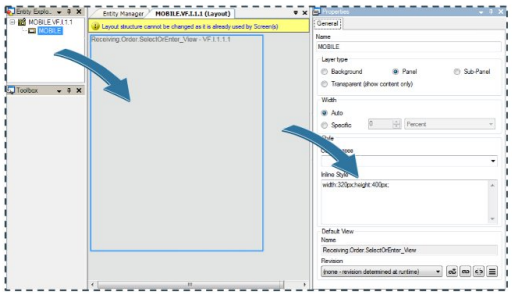

File Edit View Favorites Tools Help
OUTLINE
Search...
LAB 6: Publish as FlexPart and Test
End of LAB 6
Chapter 3: Triggering Business Logic on Forms Control Change
OnChange Operation, part 1
OnChange Operation, part 2
Max Length
LAB 7: OnChange Operation and Max Length Property
LAB 7: OnChange Operation and Max Length Property
LAB 7: Create New Screen TRNXX_MAX
LAB 7: Configure TRNXX_MAX
LAB 7: Configure TRNXX.Container_Add Operation
LAB 7: GetInputs Controls
LAB 7: Change Container Control
LAB 7: Test Run TRNXX_MAX
End of LAB 7
Chapter 4: Guidelines for Building Screens for Low Resolution Mobile Devices
Chapter 4: Guidelines for Building Screens for Low Resolution Mobile Devices
In this chapter you will find out how OnChange Operation can be used to quickly trigger business logic without any need for additional buttons.
Here are the chapters to be covered:
1. Forms - Introduction
2. Configuring Container Status Update - Labs
3. Triggering Business Logic on Forms Control Change
4. Guidelines for Building Screens for Low Resolution Mobile Devices
121
121 / 128
00:00 / 00:00
PREV
NEXT
7:05 PM

Main Principles

- Low resolution mobile primarily refers to devices running Windows CE
- Many manufacturers still use such devices, and DELMIA Apriso has to be able to work with them
- When designing Screens for such devices, use the following principles:
  - Always start with a Layout configured for the type of device
  - The header should have these elements:
    - Back & Exit buttons
    - Screen information
    - Paging information (if needed)
  - The context panel should have only 1 input (if any)
  - The footer should have paging actions (if needed), and user actions
  - Any message should be displayed across the top of the Screen
- This will make sure all mobile device screens look consistent and use DELMIA Apriso product in the best way

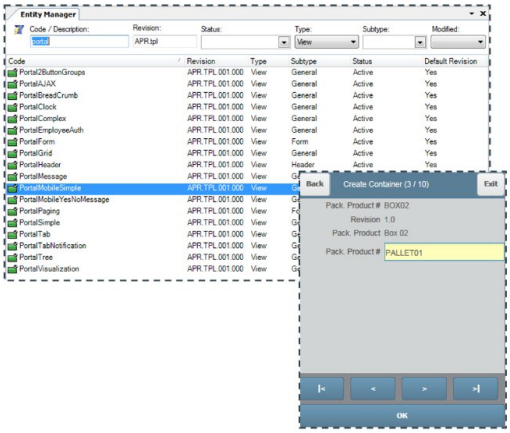
Layout Dedicated to the Device

- All the Layouts you have used in the Screen Flow Management training so far were designed primarily for desktops
- A Windows CE device will have a limited screen space, so you need to use a dedicated Layout, which will have:
  - Just one panel
  - Screen size definition in the Inline Style
  - Any additional properties reflecting the dedicated device constraints



Product View Templates for Low Resolution Devices

- The DELMIA Apriso product provides a few View Templates, so that you don't need to create mobile devices Views from scratch. They are:
  - **PortalMobileSimple** - a simple View for mobile devices
  - **PortalPaging** - a paging Template for mobile devices
  - **PortalMobileYesNoMessage** - a View that displays a message with YES and NO Buttons
  - **PortalMessage** - a View for mobile devices that displays a message



124

Example of Configuration for Low Resolution Mobile Devices

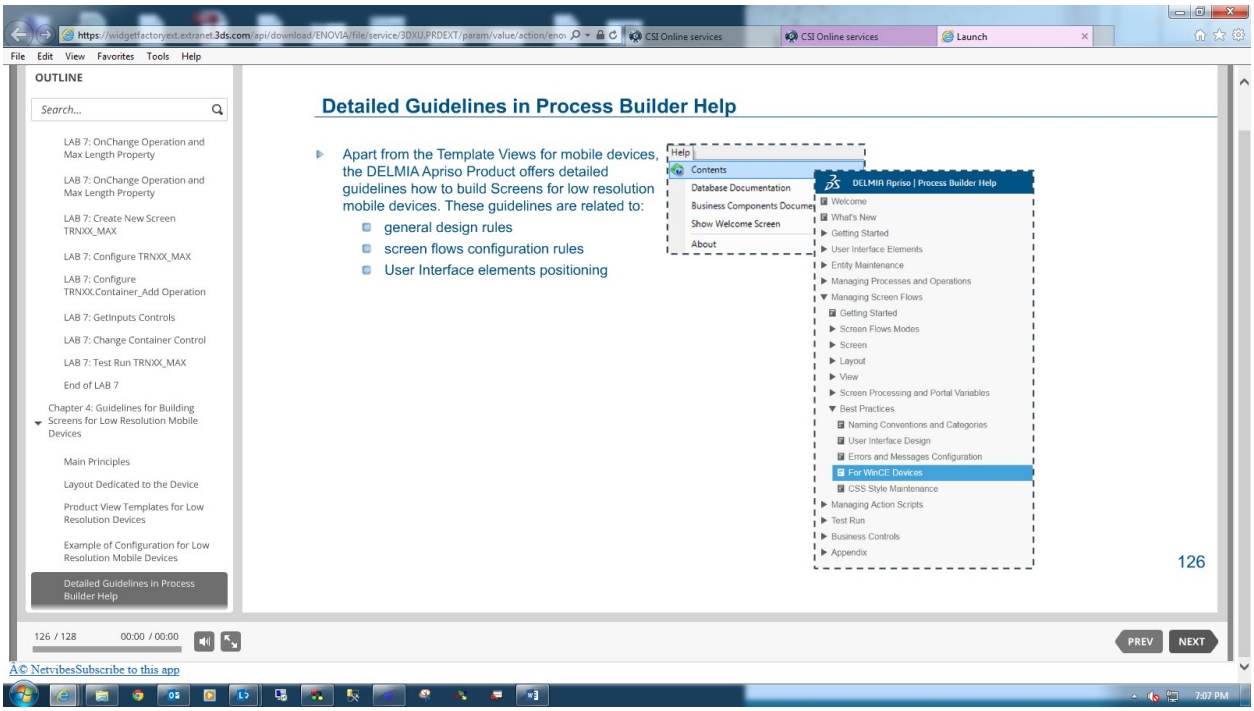
The best DELMIA Apriso example for configuration for low resolution mobile devices is the new Warehouse Management System solution (WMS SVL).

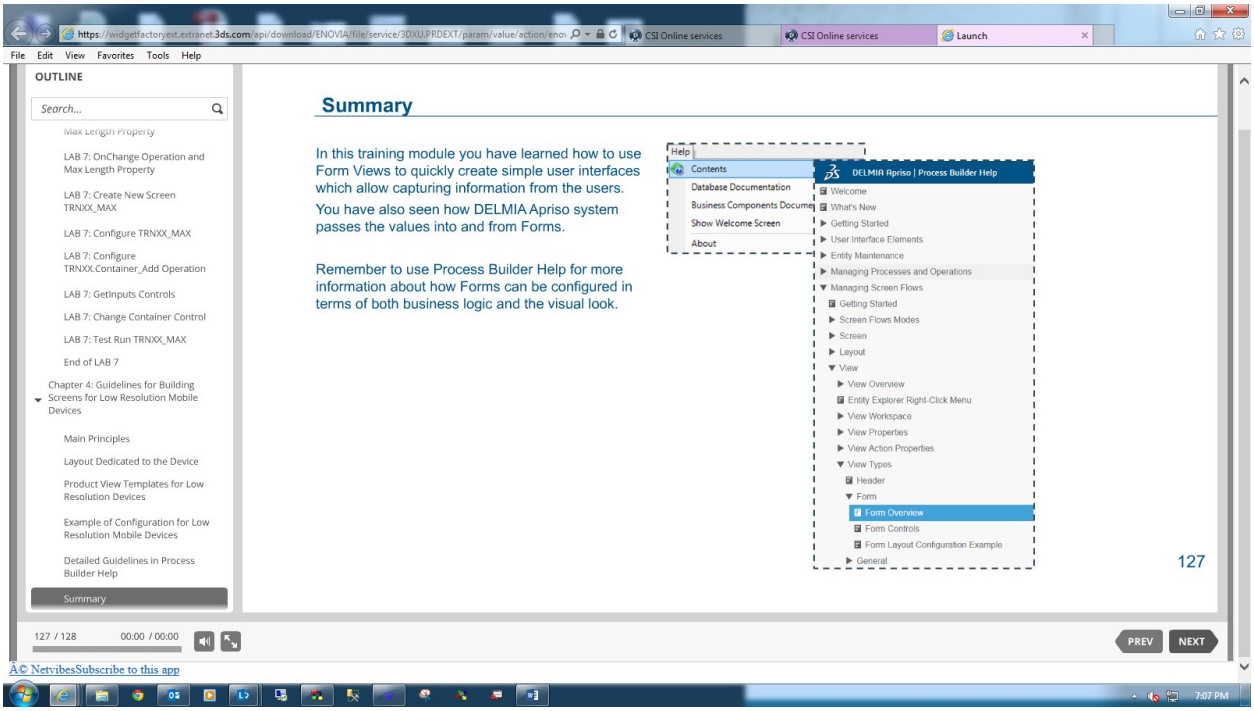
Several standard warehouse related business flows were created with extension points for customer specific modifications.

The whole user interface was designed entirely for low resolution mobile devices.

Launch the video to see how it looks in action!

125


File Edit View Favorites Tools Help
OUTLINE
Search...
LAB 7: OnChange Operation and Max Length Property
LAB 7: OnChange Operation and Max Length Property
LAB 7: Create New Screen TRNXX_MAX
LAB 7: Configure TRNXX_MAX
LAB 7: Configure TRNXX.Container_Add Operation
LAB 7: GetInputs Controls
LAB 7: Change Container Control
LAB 7: Test Run TRNXX_MAX
End of LAB 7
Chapter 4: Guidelines for Building Screens for Low Resolution Mobile Devices
Main Principles
Layout Dedicated to the Device
Product View Templates for Low Resolution Devices
Example of Configuration for Low Resolution Mobile Devices
Detailed Guidelines in Process Builder Help
Detailed Guidelines in Process Builder Help
Apart from the Template Views for mobile devices, the DELMIA Apriso Product offers detailed guidelines how to build Screens for low resolution mobile devices. These guidelines are related to:
general design rules
screen flows configuration rules
User Interface elements positioning
Help
Contents
Database Documentation
Business Components Documentation
Show Welcome Screen
About
DELMIA Apriso | Process Builder Help
Welcome
What's New
Getting Started
User Interface Elements
Entity Maintenance
Managing Processes and Operations
Managing Screen Flows
Getting Started
Screen Flows Modes
Screen
Layout
View
Screen Processing and Portal Variables
Best Practices
Naming Conventions and Categories
User Interface Design
Errors and Messages Configuration
For WinCE Devices
CSS Style Maintenance
Managing Action Scripts
Test Run
Business Controls
Appendix
126
126 / 128
00:00 / 00:00
PREV
NEXT
Â© NetvibesSubscribe to this app
7:07 PM


File Edit View Favorites Tools Help
OUTLINE
Search...
LAB 7: OnChange Operation and Max Length Property
LAB 7: Create New Screen TRNXX_MAX
LAB 7: Configure TRNXX_MAX
LAB 7: Configure TRNXX.Container_Add Operation
LAB 7: GetInputs Controls
LAB 7: Change Container Control
LAB 7: Test Run TRNXX_MAX
End of LAB 7
Chapter 4: Guidelines for Building Screens for Low Resolution Mobile Devices
Main Principles
Layout Dedicated to the Device
Product View Templates for Low Resolution Devices
Example of Configuration for Low Resolution Mobile Devices
Detailed Guidelines in Process Builder Help
Summary
Summary
In this training module you have learned how to use Form Views to quickly create simple user interfaces which allow capturing information from the users.
You have also seen how DELMIA Apriso system passes the values into and from Forms.
Remember to use Process Builder Help for more information about how Forms can be configured in terms of both business logic and the visual look.
Help
Contents
Database Documentation
Business Components Documentation
Show Welcome Screen
About
DELMIA Apriso | Process Builder Help
Welcome
What's New
Getting Started
User Interface Elements
Entity Maintenance
Managing Processes and Operations
Managing Screen Flows
Getting Started
Screen Flows Modes
Screen
Layout
View
View Overview
Entity Explorer Right-Click Menu
View Workspace
View Properties
View Action Properties
View Types
Header
Form
Form Overview
Form Controls
Form Layout Configuration Example
General
127
127 / 128
00:00 / 00:00
PREV
NEXT
Â© NetvibesSubscribe to this app
7:07 PM