

<https://widgetfactoryext.extranet.3ds.com/api/download/ENOVA/file/service/3DXU.PRDEXT/param/value/action/eno1>

**OUTLINE**

- 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
- LAB 2: Change Container Properties, part 1
- LAB 2: Change Container Properties, part 2
- LAB 2: Add Control, part 1
- LAB 2: Add Control, part 2
- LAB 2: Add Control, part 3
- LAB 2: Add Control, part 4
- LAB 2: Add Control, part 5**

31 / 128    00:00 / 00:00    PREV    NEXT

A © Netvibes [Subscribe to this app](#)

**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

<https://widgetfactoryext.extranet.3ds.com/api/download/ENOVA/file/service/3DXU.PRDEXT/param/value/action/eno1>

**OUTLINE**

- 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
- LAB 2: Change Container Properties, part 1
- LAB 2: Change Container Properties, part 2
- LAB 2: Add Control, part 1
- LAB 2: Add Control, part 2
- LAB 2: Add Control, part 3
- LAB 2: Add Control, part 4
- LAB 2: Add Control, part 5
- LAB 2: Add Control, part 6**

32 / 128    00:00 / 00:00    PREV    NEXT

A © Netvibes [Subscribe to this app](#)

**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

<https://widgetfactoryext.extranet.3ds.com/api/download/ENOVA/file/service/3DXU.PRDEXT/param/value/action/eno1>

File Edit View Favorites Tools Help

## OUTLINE

Search...

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 TRNVOI\_FRM-010  
LAB 2: Change Container Properties, part 1  
LAB 2: Change Container Properties, part 2  
LAB 2: Add Control, part 1  
LAB 2: Add Control, part 2  
LAB 2: Add Control, part 3  
LAB 2: Add Control, part 4  
LAB 2: Add Control, part 5  
LAB 2: Add Control, part 6  
LAB 2: Add Control, part 7

33 / 128 00:00 / 00:00 PREV NEXT

A © Netvibes [Subscribe to this app](#)

**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

<https://widgetfactoryext.extranet.3ds.com/api/download/ENOVA/file/service/3DXU.PRDEXT/param/value/action/eno1>

File Edit View Favorites Tools Help

## OUTLINE

Search...

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 TRNVOI\_FRM-010  
LAB 2: Change Container Properties, part 1  
LAB 2: Change Container Properties, part 2  
LAB 2: Add Control, part 1  
LAB 2: Add Control, part 2  
LAB 2: Add Control, part 3  
LAB 2: Add Control, part 4  
LAB 2: Add Control, part 5  
LAB 2: Add Control, part 6  
LAB 2: Add Control, part 7  
LAB 2: Add Control, part 8

34 / 128 00:00 / 00:00 PREV NEXT

A © Netvibes [Subscribe to this app](#)

**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

<https://widgetfactoryext.extranet.3ds.com/api/download/ENOVA/file/service/3DXU.PRDEXT/param/value/action/eno1>

CSI Online services CSI Online services Launch

File Edit View Favorites Tools Help

OUTLINE

Search...

LAB 1: Test Run TRNXX\_FRM-010  
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  
LAB 2: Change Container Properties, part 1  
LAB 2: Add Control, part 1  
LAB 2: Add Control, part 2  
LAB 2: Add Control, part 3  
LAB 2: Add Control, part 4  
LAB 2: Add Control, part 5  
LAB 2: Add Control, part 6  
LAB 2: Add Control, part 7  
LAB 2: Add Control, part 8  
LAB 2: Add Control, part 9

35 / 128 00:00 / 00:00 PREV NEXT

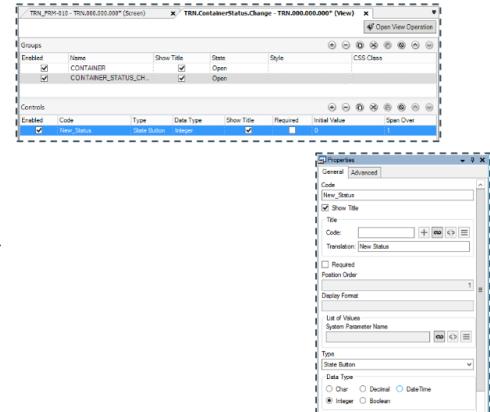
A © Netvibes [Subscribe to this app](#)

**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

<https://widgetfactoryext.extranet.3ds.com/api/download/ENOVA/file/service/3DXU.PRDEXT/param/value/action/eno1>

CSI Online services CSI Online services Launch

File Edit View Favorites Tools Help

OUTLINE

Search...

Container Details  
LAB 2: Create a Screen to Display Container Details  
LAB 2: Create New Screen TRNXX\_FRM-010  
LAB 2: Change Container Properties, part 1  
LAB 2: Change Container Properties, part 2  
LAB 2: Add Control, part 1  
LAB 2: Add Control, part 2  
LAB 2: Add Control, part 3  
LAB 2: Add Control, part 4  
LAB 2: Add Control, part 5  
LAB 2: Add Control, part 6  
LAB 2: Add Control, part 7  
LAB 2: Add Control, part 8  
LAB 2: Add Control, part 9  
End of LAB 2  
LAB 3: Create State Buttons for Container Status

38 / 128 00:00 / 00:00 PREV NEXT

A © Netvibes [Subscribe to this app](#)

**LAB 3: Create State Buttons for Container Status**

**Task:**  

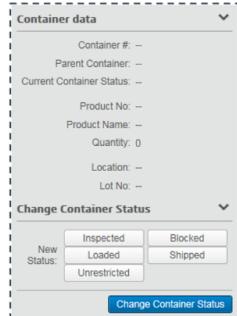
- ▶ Add a System Parameter
- ▶ Link added System Parameter to the correct Screen

**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](mailto:DELMIA.Apriso.training@3ds.com)



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, buttons, will use the List of Values property.

39

**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 do is to write an SQL query to retrieve the values to show.

41

**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.

**CONTAINER\_STATUS**

Description  
Stores the current status of any given Container.

Columns

Column Name	Data Type	Nullable	Default	Description (Custom if Available)	Links
ContainerStatus	SMALLINT(5,0)	No		Used to track the status of a Container (e.g. picked, packed). See Prime Data.	
TextID	INT(10,0)	Yes		Link to the TEXT table.	

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.

**TEXT\_TRANSLATION**

Description  
Contain the various text translation of text table. This table can persist multiple texts (short, medium...) as well as icons for multiple devices types and URLs.

Columns

Column Name	Data Type	Nullable	Default	Description (Custom if Available)	Links
TextID	INT(10,0)	No		Link to Text table	TEXT
LanguageID	INT(10,0)	No		Language of the entity	LANGUAGE
Micro	NVARCHAR(1)	Yes		The micro description (1 character)	
Short	NVARCHAR(80)	Yes		Short description (up to 80 characters)	
Medium	NVARCHAR(255)	Yes		Medium description (up to 255 characters)	
Extended	NVARCHAR(2000)	Yes		Extended description (up to 2000 characters)	

44

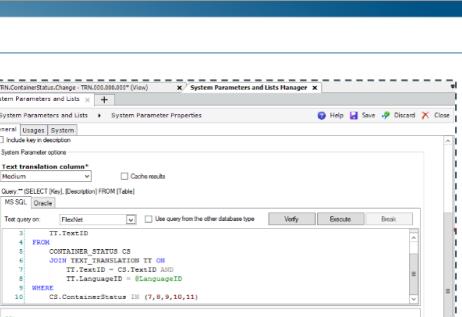
**LAB 3: Add System Parameter, part 3**

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 / 128    00:00 / 00:00    PREV    NEXT

A © NervibesSubscribe to this app

**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

The screenshot shows the 'System Parameters and Lists' manager window. A new system parameter is being created with the following properties:

- Name:** LAB\_ContainerStatusChange
- Type:** Text
- Description:** Container Status Change
- Text translation column:** Medium (selected)
- Query:**

```
3   IT.TextID
4   FROM
5   CONTAINER_STATUS CS
6   JOIN TEXT_TRANSLATION TT ON
7   CS.ContainerStatus = TT.TextID AND
8   TT.LanguageID = $LanguageID
9   WHERE
10  CS.ContainerStatus IN (7,8,9,10,11)
```
- MS SQL:** Oracle

The SQL statement is highlighted with a dashed box, and an arrow points from the text "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." to the 'TextID' column in the SQL code.

**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 / 128    00:00 / 00:00    PREV    NEXT

A © NetvibesSubscribe to this app

**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 / 128    00:00 / 00:00    PREV    NEXT

A © NetvibesSubscribe to this app

**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

**Container Data**

Container #:  
Parent Container:  
ContainerSpec:  
Product No:  
Product Name:  
Quantity: 0  
Location:  
Lot No:

**Change Container Status**

New Status:

49

49 / 128    00:00 / 00:00    PREV    NEXT

A © Netvibes [Subscribe to this app](#)

**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

**Container data**

Container #: --  
Parent Container: --  
Current Container Status: --  
Product No: --  
Product Name: --  
Quantity: 0  
Location: --  
Lot No: --

**Change Container Status**

New Status:

**Properties**

General Advanced

Type:     
 Data Type:  Char  Decimal  Date/Time  
 Integer  Boolean

Initial Value:

Default Value:

Maximum Text Length:

Span Over:

Action:     
 Operation  Action Script Function

Name:

Revision:

50

50 / 128    00:00 / 00:00    PREV    NEXT

A © Netvibes [Subscribe to this app](#)

<https://widgetfactoryext.extranet.3ds.com/api/download/ENOVA/file/service/3DXU.PRDEXT/param/value/action/eno1>

**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.

**Container data**

Container #:	--
Parent Container:	--
Current Container Status:	--
Product No.:	--
Product Name:	--
Quantity:	0
Location:	--
Lot No.:	--

**Change Container Status**

Inspected	Blocked
New Status:	Loaded
Unrestricted	Shipped

**Properties**

- Code: `_control`
- Show Title:
- Title:
- Code:
- Translation:
- Required:
- Position Order:  4
- Display Format:
- List of Values: System Parameter Name
- Type: Label
- Data Type:
  - Char (selected)
  - Decimal
  - Date/Time
  - Integer
  - Boolean

51

<https://widgetfactoryext.extranet.3ds.com/api/download/ENOVA/file/service/3DXU.PRDEXT/param/value/action/eno1>

**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.

**Controls**

Code	Type	Data Type	Show Title	Requir	Initial Value	Span	MoveUp	MoveDown
ContainerNo	Label	Char	<input checked="" type="checkbox"/>	<input type="checkbox"/>		1		
ParentContainer	Label	Char	<input checked="" type="checkbox"/>	<input type="checkbox"/>		1		
ContainerStatusDesc	Label	Char	<input checked="" type="checkbox"/>	<input type="checkbox"/>		1		
<code>_control</code>	Label	Char	<input type="checkbox"/>	<input type="checkbox"/>	1			
ProductNo	Label	Char	<input checked="" type="checkbox"/>	<input type="checkbox"/>		1		
ProductDesc	Label	Char	<input checked="" type="checkbox"/>	<input type="checkbox"/>		1		
QuantityOnHand	Label	Decimal	<input checked="" type="checkbox"/>	<input type="checkbox"/>	0	1		
<code>_control1</code>	Label	Char	<input type="checkbox"/>	<input type="checkbox"/>		1		
Location	Label	Char	<input checked="" type="checkbox"/>	<input type="checkbox"/>		1		
LotNo	Label	Char	<input checked="" type="checkbox"/>	<input type="checkbox"/>		1		

52

<https://widgetfactoryext.extranet.3ds.com/api/download/ENOVA/file/service/3DXU.PRDEXT/param/value/action/eno1>

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**

53 / 128 00:00 / 00:00 PREV NEXT

A © Netvibes [Subscribe to this app](#)

Container data

Container #: --  
Parent Container: --  
Current Container Status: --  
Product No: --  
Product Name: --  
Quantity: 0  
Location: --  
Lot No: --

Change Container Status

New Status:

Change Container Status

53

<https://widgetfactoryext.extranet.3ds.com/api/download/ENOVA/file/service/3DXU.PRDEXT/param/value/action/eno1>

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**

54 / 128 00:00 / 00:00 PREV NEXT

A © Netvibes [Subscribe to this app](#)

Properties of Form Groups and Controls are described in much detail in the Process Builder help.

Help [ ]

DELMIRO Apriso | Process Builder Help

- Contents
- Database Documentation
- Business Components Documentation
- Show Welcome Screen
- About
- Help [ ]
- DELMIRO 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

<https://widgetfactoryext.extranet.3ds.com/api/download/ENOVA/file/service/3DXU.PRDEXT/param/value/action/eno1>

CSI Online services Launch

**OUTLINE**

Search...

- 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
- End of LAB 3
- LAB 4: Pass the ContainerNo
- LAB 4: Pass the ContainerNo

**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

PREV NEXT

A © Netvibes Subscribe to this app

<https://widgetfactoryext.extranet.3ds.com/api/download/ENOVA/file/service/3DXU.PRDEXT/param/value/action/eno1>

CSI Online services Launch

**OUTLINE**

Search...

- 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
- 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: 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.

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

58

PREV NEXT

A © Netvibes Subscribe to this app

**OUTLINE**

Search...

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  
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

59 / 128 00:00 / 00:00 PREV NEXT

A © Netvibes [Subscribe to this app](#)

**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.

59

**OUTLINE**

Search...

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  
End of LAB 3  
LAB 4: Pass the ContainerNo  
LAB 4: Pass the ContainerNo  
LAB 4: Pass the ContainerNo  
LAB 4: Modify Step Name

60 / 128 00:00 / 00:00 PREV NEXT

A © Netvibes [Subscribe to this app](#)

**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

60

[https://widgetfactorytest.extranet.3ds.com/api/download/ENOVIAService/3DXUI\\_PRDEXT/param/value/action/eno1](https://widgetfactorytest.extranet.3ds.com/api/download/ENOVIAService/3DXUI_PRDEXT/param/value/action/eno1)

CSI Online services Launch

OUTLINE

- 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
- LAB 4: Add SQL Query Function**

61 / 128 00:00 / 00:00 PREV NEXT

A Netvibes [Subscribe to this app](#)

[https://widgetfactorytest.extranet.3ds.com/api/download/ENOVIAService/3DXUI\\_PRDEXT/param/value/action/eno1](https://widgetfactorytest.extranet.3ds.com/api/download/ENOVIAService/3DXUI_PRDEXT/param/value/action/eno1)

CSI Online services Launch

OUTLINE

- 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
- LAB 4: Add SQL Query Function
- LAB 4: Provide Script to the Function**

62 / 128 00:00 / 00:00 PREV NEXT

A Netvibes [Subscribe to this app](#)

► 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

**OUTLINE**

Search...

IRNXX.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  
LAB 4: Add SQL Query Function  
LAB 4: Provide Script to the Function  
**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 Languageld (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

**SQL Query Editor**

MS SQL [Oracle] Test query on: [FileNet] Use query from the other database type Verify Execute Break

```

1 SELECT
2   PRD.ProductNo,
3   INV2.LotNo,
4   INV2.QuantityOnHand,
5   COM.InContainer AS ParentContainer,
6   WMS.Location,
7   TT.Medium AS ProductDesc,
8   TT.Medium AS ContainerStatusDesc
9  FROM
10  INVENTORY2 INV
11  JOIN INV2 PRD
12    ON INV2.ProductId = PRD.Id
13  JOIN CONTAINER COM
14    ON INV2.Container = COM.Container

```

Name	Type	Value
LanguageId	Integer	1033
ContainerNo	Char	R1075_CN01

Results Messages

OK Cancel Help

63

**OUTLINE**

Search...

IRNXX.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  
LAB 4: Add SQL Query Function  
LAB 4: Provide Script to the Function  
**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: Languageld and ContainerNo

Container will come from the first Form you created. The Languageld 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.

**GetContainerData**

Inputs: LanguageId, ContainerNo  
Outputs: LotNo, QuantityOnHand, ParentContainer, Location, ProductDesc, ContainerStatus

**Create External Input**

Input Source: Session Variable  
System Variables: LanguageID, Session Variables: User, External Inputs: (Int set)

64

**OUTLINE**

Search...

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  
LAB 4: Add SQL Query Function  
LAB 4: Provide Script to the Function  
LAB 4: Run the Query  
LAB 4: Configure Other Functions  
LAB 4: Check Scalar Output

65 / 128    00:00 / 00:00    PREV    NEXT

A © NetvibesSubscribe to this app

**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.

General

Name: GetContainerData

Description:

Type: SQL Query

Log Events Cache Results

Database: Retailer

SQL Query Editor

```
SELECT P20.ContainerNo,
       P20.LotNo,
       INV.QuantityOnHand,
       COIN.ContainerNo AS ParentContainer,
       INV.ProductID,
       TT.Metric AS ProductType,
       TT.Metric AS ContainerStatus
  FROM INVTRANSINV INV
 WHERE INV.ContainerNo = P20.ContainerNo
```

Report error

No rows are returned

Multiple rows are returned

Scalar output

65

**OUTLINE**

Search...

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  
LAB 4: Add SQL Query Function  
LAB 4: Provide Script to the Function  
LAB 4: Run the Query  
LAB 4: Configure Other Functions  
LAB 4: Check Scalar Output  
LAB 4: Input/Output Indicators

66 / 128    00:00 / 00:00    PREV    NEXT

A © NetvibesSubscribe to this app

**LAB 4: Input/Output Indicators**

After the change, this is 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.

General

Name: GetContainerData

Description:

Type: SQL Query

Log Events Cache Results

Database: Retailer

SQL Query Editor

```
SELECT P20.ContainerNo,
       P20.LotNo,
       INV.QuantityOnHand,
       COIN.ContainerNo AS ParentContainer,
       INV.ProductID,
       TT.Metric AS ProductType,
       TT.Metric AS ContainerStatus
  FROM INVTRANSINV INV
 WHERE INV.ContainerNo = P20.ContainerNo
```

Report error

No rows are returned

Multiple rows are returned

Scalar output

66

<https://widgetfactorytest.extranet.3ds.com/api/download/ENOVIAM/file/service/3DXU.PRDEXT/param/value/action/eno1>

CSI Online services Launch

OUTLINE

Search...

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

LAB 4: Add SQL Query Function

LAB 4: Provide Script to the Function

LAB 4: Run the Query

LAB 4: Configure Other Functions

LAB 4: Check Scalar Output

LAB 4: Input/Output Indicators

LAB 4: Enable Interface

67 / 128 00:00 / 00:00 PREV NEXT

A © Netvibes [Subscribe to this app](#)

**LAB 4: Enable Interface**

In the Entry 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

<https://widgetfactorytest.extranet.3ds.com/api/download/ENOVIAM/file/service/3DXU.PRDEXT/param/value/action/eno1>

CSI Online services Launch

OUTLINE

Search...

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

LAB 4: Add SQL Query Function

LAB 4: Provide Script to the Function

LAB 4: Run the Query

LAB 4: Configure Other Functions

LAB 4: Check Scalar Output

LAB 4: Input/Output Indicators

LAB 4: Enable Interface

LAB 4: Modify Operation Properties, part 1

68 / 128 00:00 / 00:00 PREV NEXT

A © Netvibes [Subscribe to this app](#)

**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

<https://widgetfactorytest.extranet.3ds.com/api/download/ENOVIAM/file/service/3DXU.PRDEXT/param/value/action/enov>

CSI Online services

Launch

File Edit View Favorites Tools Help

OUTLINE

Search...

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

LAB 4: Add SQL Query Function

LAB 4: Provide Script to the Function

LAB 4: Run the Query

LAB 4: Configure Other Functions

LAB 4: Check Scalar Output

LAB 4: Input/Output Indicators

LAB 4: Enable Interface

LAB 4: Modify Operation Properties, part 1

LAB 4: Modify Operation Properties, part 2

69 / 128 00:00 / 00:00 PREV NEXT

A NetvibesSubscribe to this app

655 PM

**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

<https://widgetfactorytest.extranet.3ds.com/api/download/ENOVIAM/file/service/3DXU.PRDEXT/param/value/action/enov>

CSI Online services

Launch

File Edit View Favorites Tools Help

OUTLINE

Search...

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

LAB 4: Add SQL Query Function

LAB 4: Provide Script to the Function

LAB 4: Run the Query

LAB 4: Configure Other Functions

LAB 4: Check Scalar Output

LAB 4: Input/Output Indicators

LAB 4: Enable Interface

LAB 4: Modify Operation Properties, part 1

LAB 4: Modify GetContainerData, part 1

70 / 128 00:00 / 00:00 PREV NEXT

A NetvibesSubscribe to this app

656 PM

**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