Acumatica
The Cloud ERP

T210 Customized Forms and Master-Detail Relationship

Dmitrii Naumov

ISV Solution Architect

Timing and Agenda

July 19, 2022 -10 AM -11:30 AM

Day 1

Lesson 1.1: Adding Custom Fields

July 20, 2022 -10 AM -11:30 AM

Day 2

Lesson 1.2: Configuring the UI—Self-Guided Exercise

Lesson 2.1: Defining a Master-Detail Relationship

Timing and Agenda

July 21, 2022 -10 AM -11:30 AM

Day 3

Lesson 2.2: Defining the Business Logic

Lesson 3.1: Adding a New Tab

July 22, 2022 -10 AM -11:30 AM

Day 4

Lesson 4.1: Calculating Field Values

Lesson 4.2: Inserting a Default Detail Record

Day 1

Part 1: Custom Fields (Stock Items Form)

Cherrylake, Inc.
Acumatica Customer Since 2013

Acumatica
The Cloud ERP

Lesson 1.1: Adding Custom Fields

Learning Objectives

In this lesson, you will learn how to do the following:

- Add a custom column to an Acumatica ERP database table
- Add a custom field to an Acumatica ERP data access class
- Add the control for the custom field to the form

Figure: Custom elements to be added to the Stock Items form

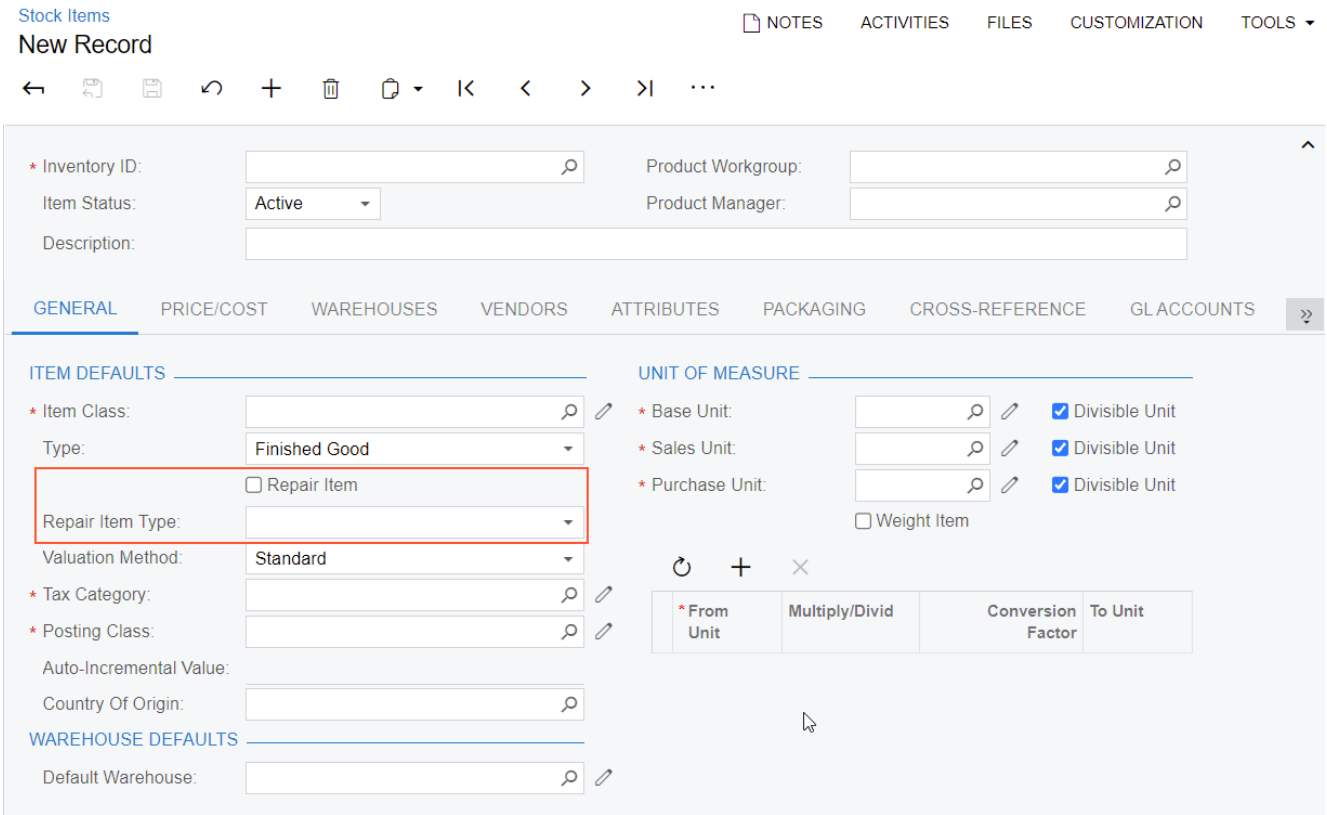


Figure: Customization menu

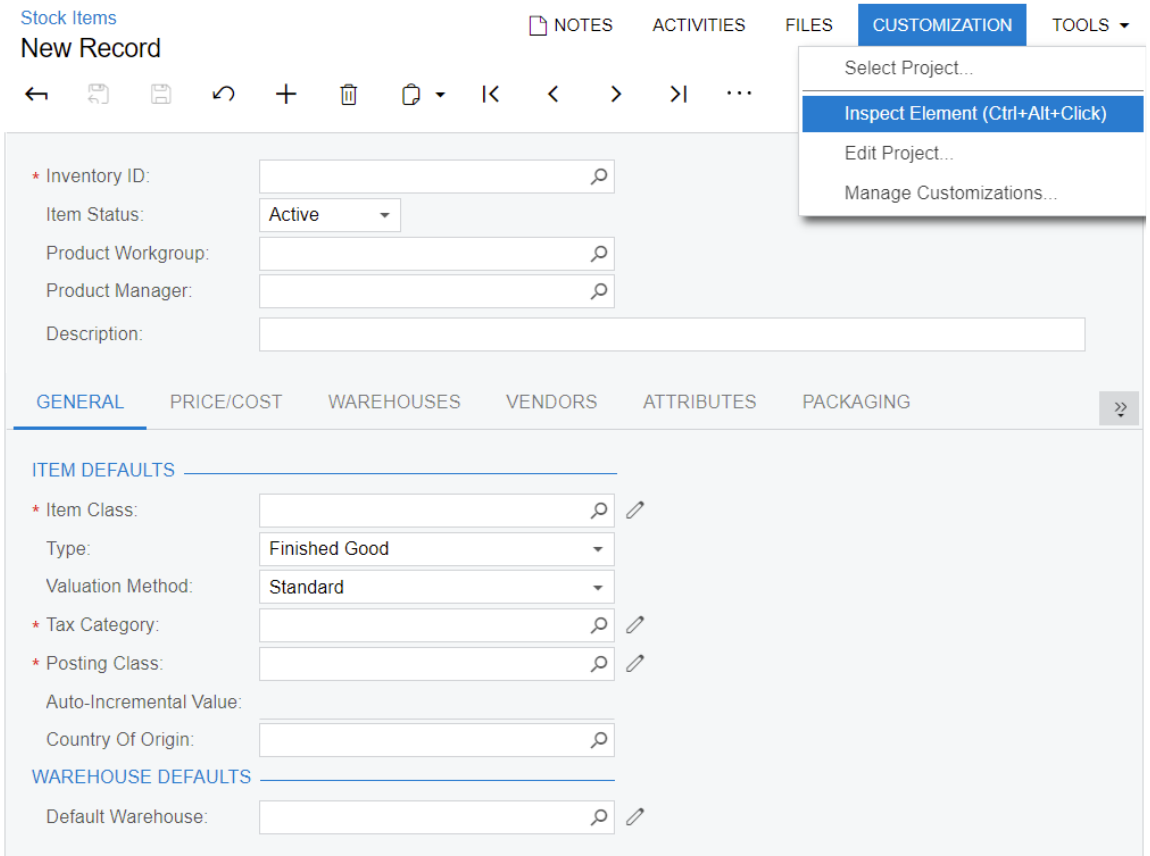


Figure: Element Properties dialog box

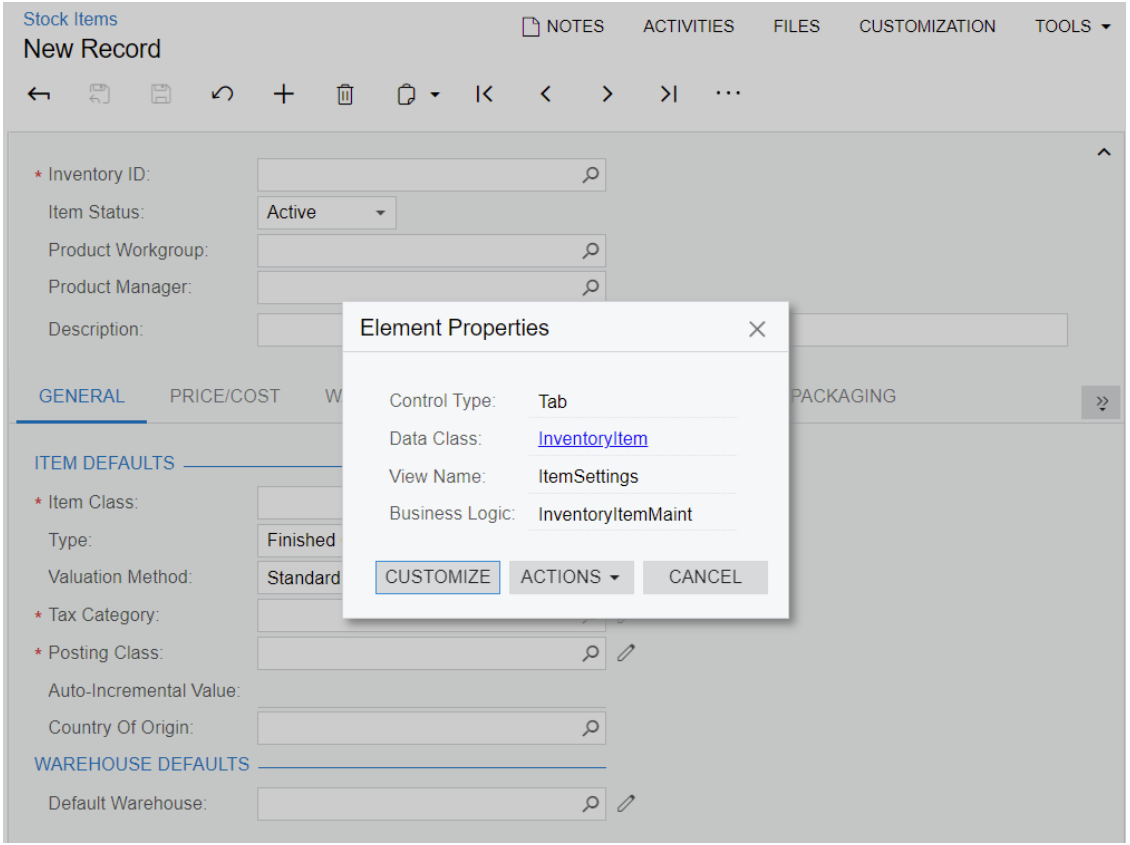


Figure: Suppression of the error in a comment

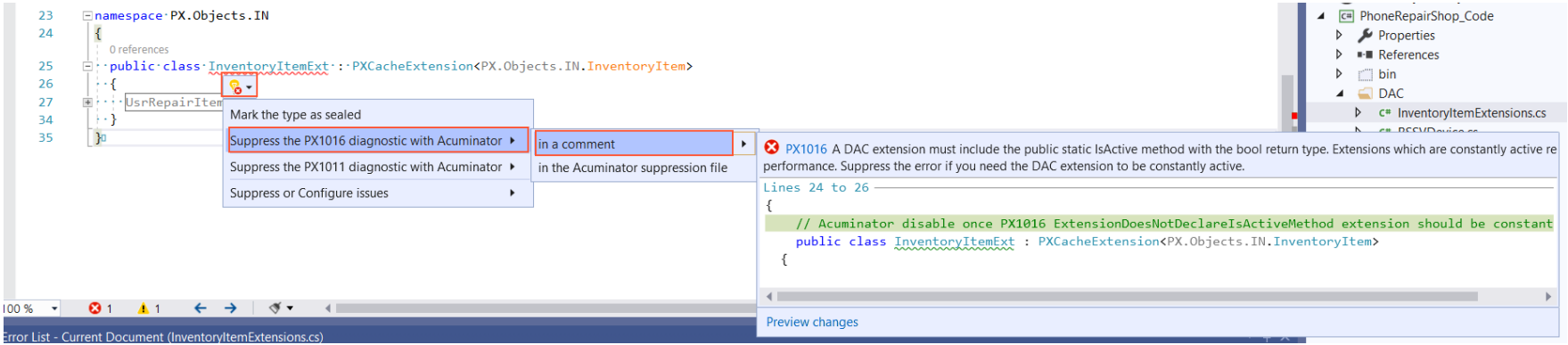


Figure: Fix of the warning

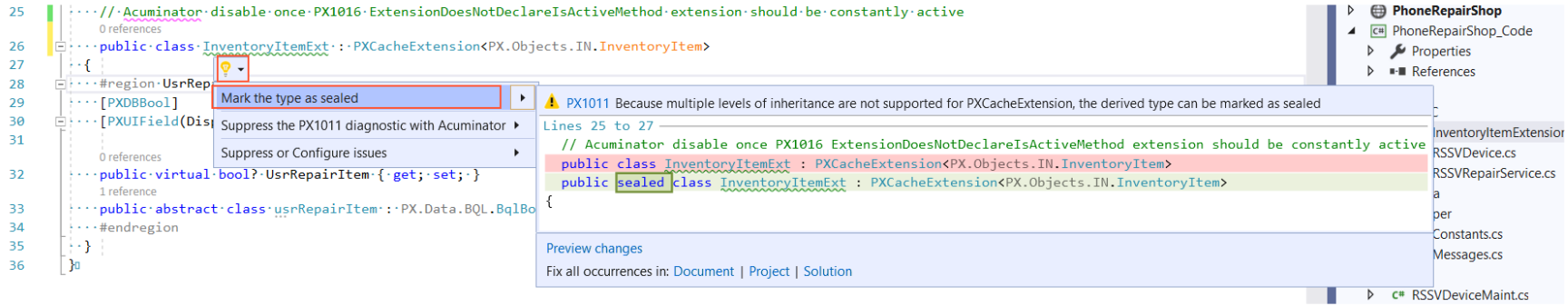


Figure: The Type node in the control tree

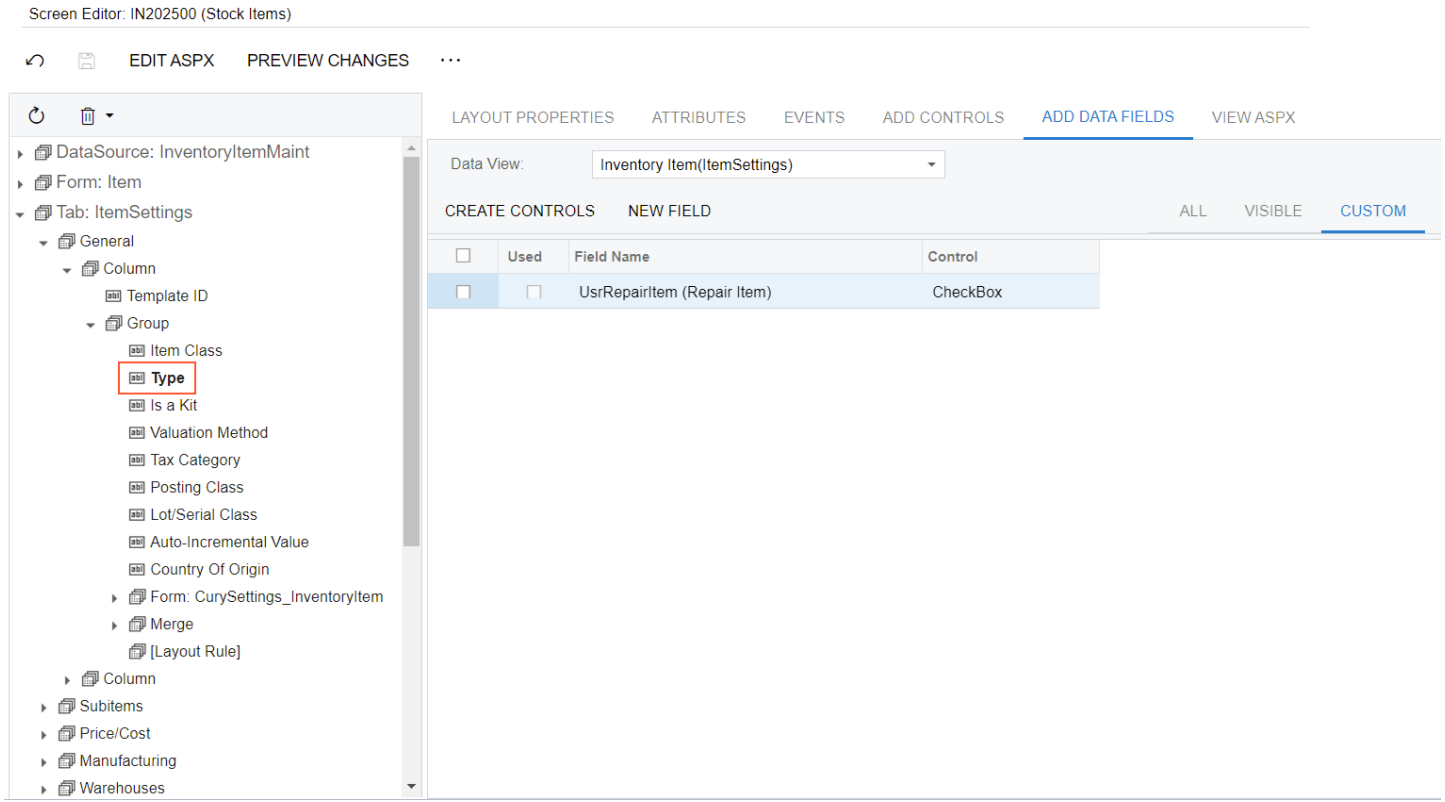


Figure: The added control

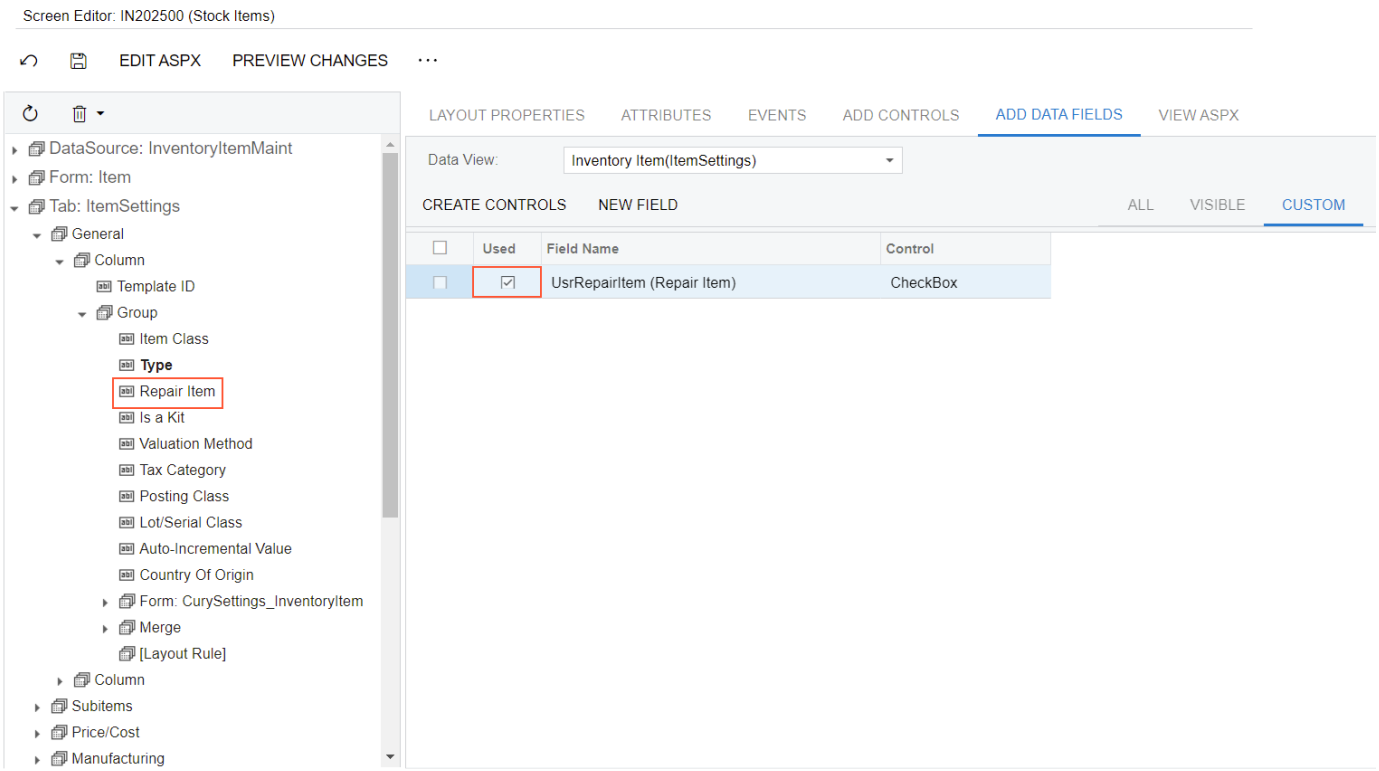


Figure: The Repair Item check box

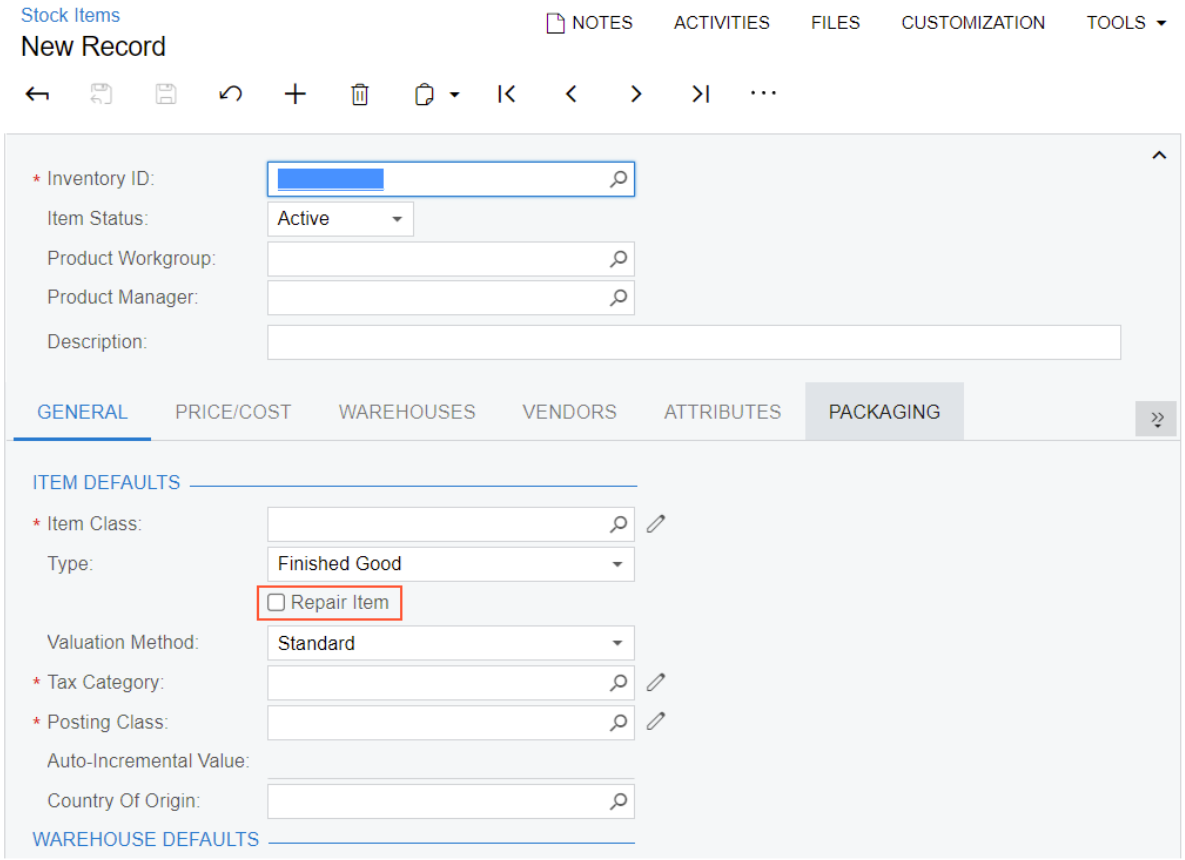


Figure: The list of the project items

Edit Project Items

GET PACKAGE

Object Name	Type	Description	Created By	Creation Date	Last Modified By	Last Modified On
~/pages/in/in202500.aspx	Page		admin admin	10/12/2021	admin admin	10/12/2021
~/pages/rs/rs201000.aspx	Page		admin admin	10/8/2021	admin admin	10/8/2021
~/pages/rs/rs202000.aspx	Page		admin admin	10/8/2021	admin admin	10/8/2021
Bin\PhoneRepairShop_code.dll	File		admin admin	10/8/2021	admin admin	10/8/2021
InputData\RSSVDevice.csv	File		admin admin	10/8/2021	admin admin	10/8/2021
InputData\RSSVRepairService.csv	File		admin admin	10/8/2021	admin admin	10/8/2021
Pages\RS\RS201000.aspx	File		admin admin	10/8/2021	admin admin	10/8/2021
Pages\RS\RS201000.aspx.cs	File		admin admin	10/8/2021	admin admin	10/8/2021
Pages\RS\RS202000.aspx	File		admin admin	10/8/2021	admin admin	10/8/2021
Pages\RS\RS202000.aspx.cs	File		admin admin	10/8/2021	admin admin	10/8/2021
Serviced Devices	GenericInquiryScreen		admin admin	10/8/2021	admin admin	10/8/2021
InventoryItem	Table		admin admin	10/12/2021	admin admin	10/12/2021
Serviced Devices	SiteMapNode		admin admin	10/8/2021	admin admin	10/8/2021
Repair Services	SiteMapNode		admin admin	10/8/2021	admin admin	10/8/2021

Source

```
LoKjLShj41Na1ny5CbPs954Ba67pfKTVOu4A7Ly7Venv9AZY3jcvGSrif4+/8B">
    <PXTabItem Text="General" ParentId="phG_tab_Items#0" TypeFullName="PX.Web.UI.PXTabItem">
        <Children Key="Template">
            <AddItem>
                <PXCheckBox TypeFullName="PX.Web.UI.PXCheckBox">
                    <Prop Key="Virtual:ApplyStylesheetSkin" />
                    <Prop Key="ID" Value="CstPXCheckBox1" />
                    <Prop Key="DataField" Value="UsrRepairItem" />
                </PXCheckBox>
            </AddItem>
            <PXCheckBox DataField="KitItem" OriginalIndex="5" />
        </Children>
    </PXTabItem>
</Page>
```

Figure: The Repair Item Type box

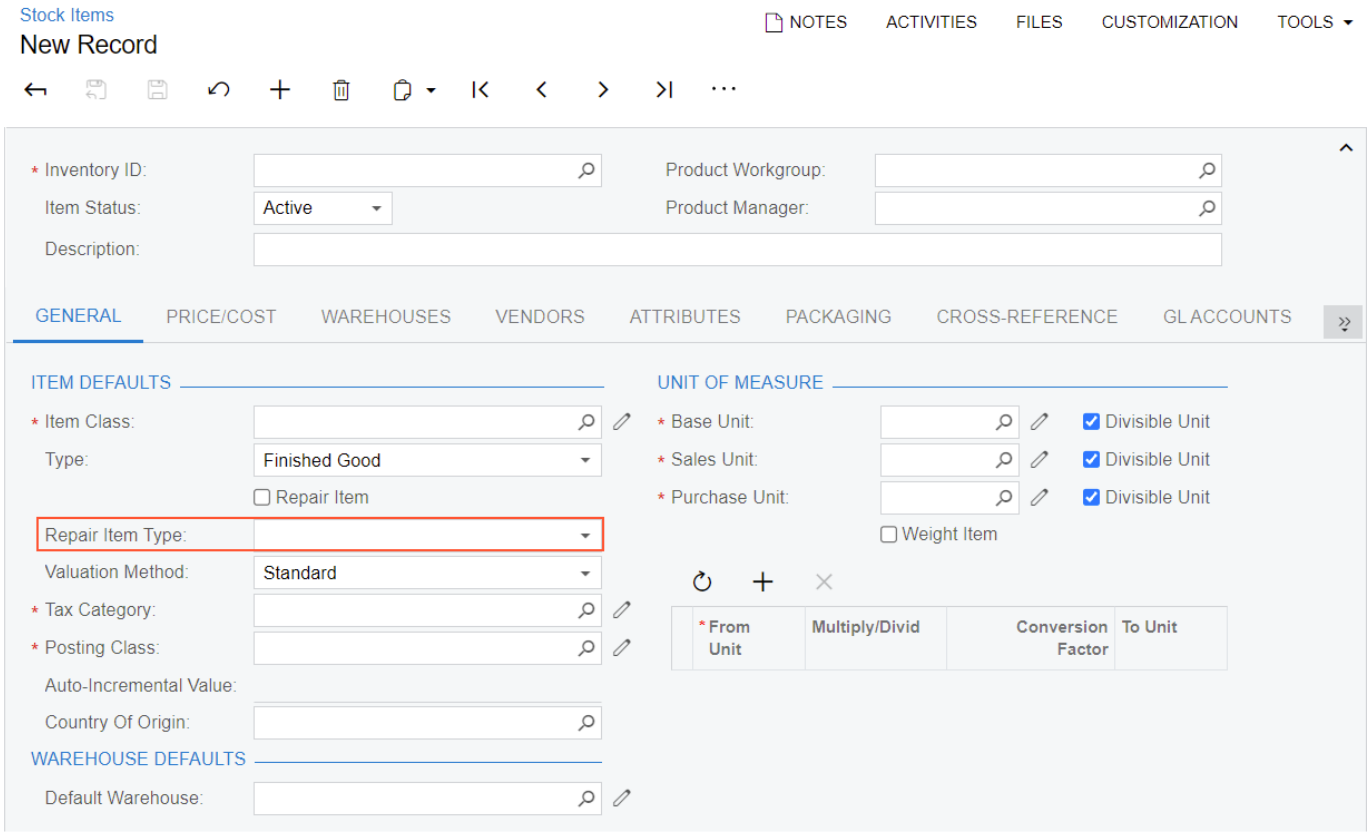


Figure: The generation of the event handler

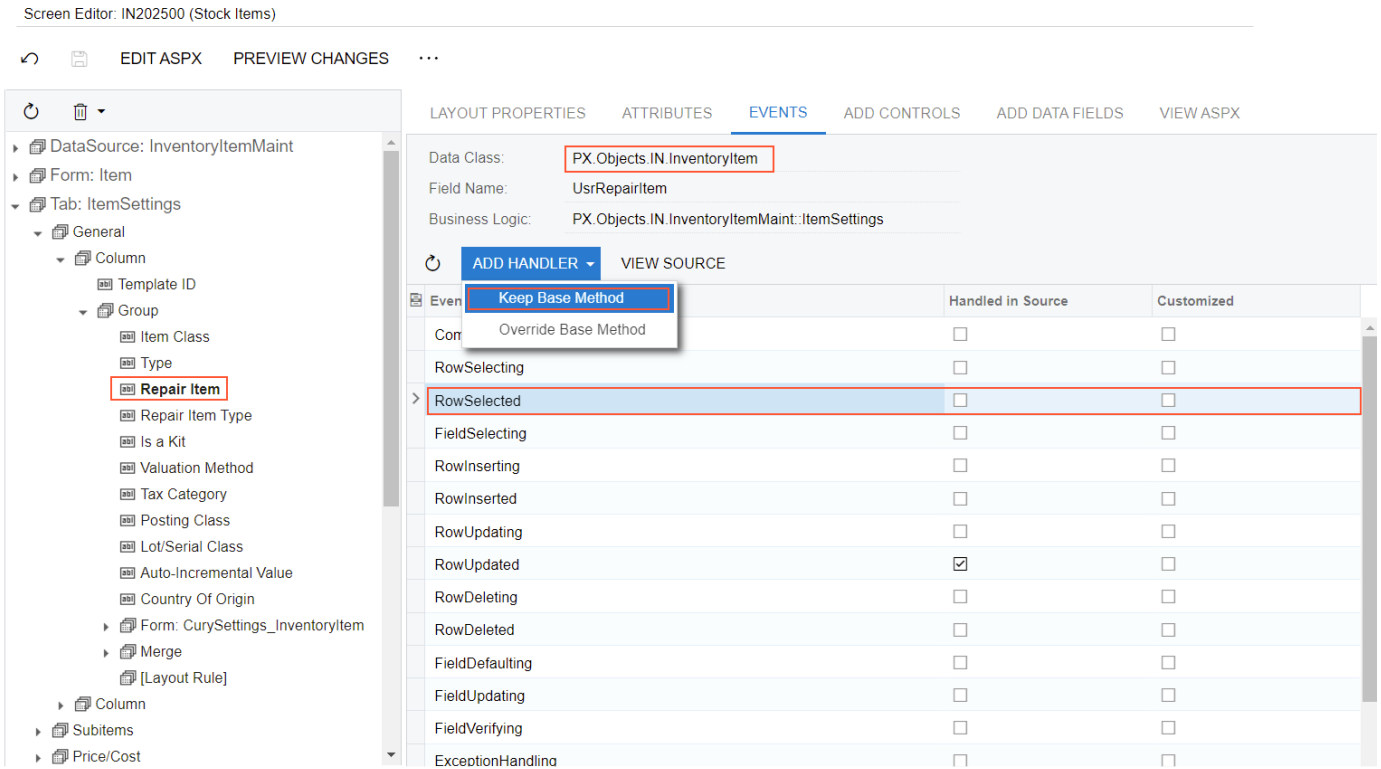
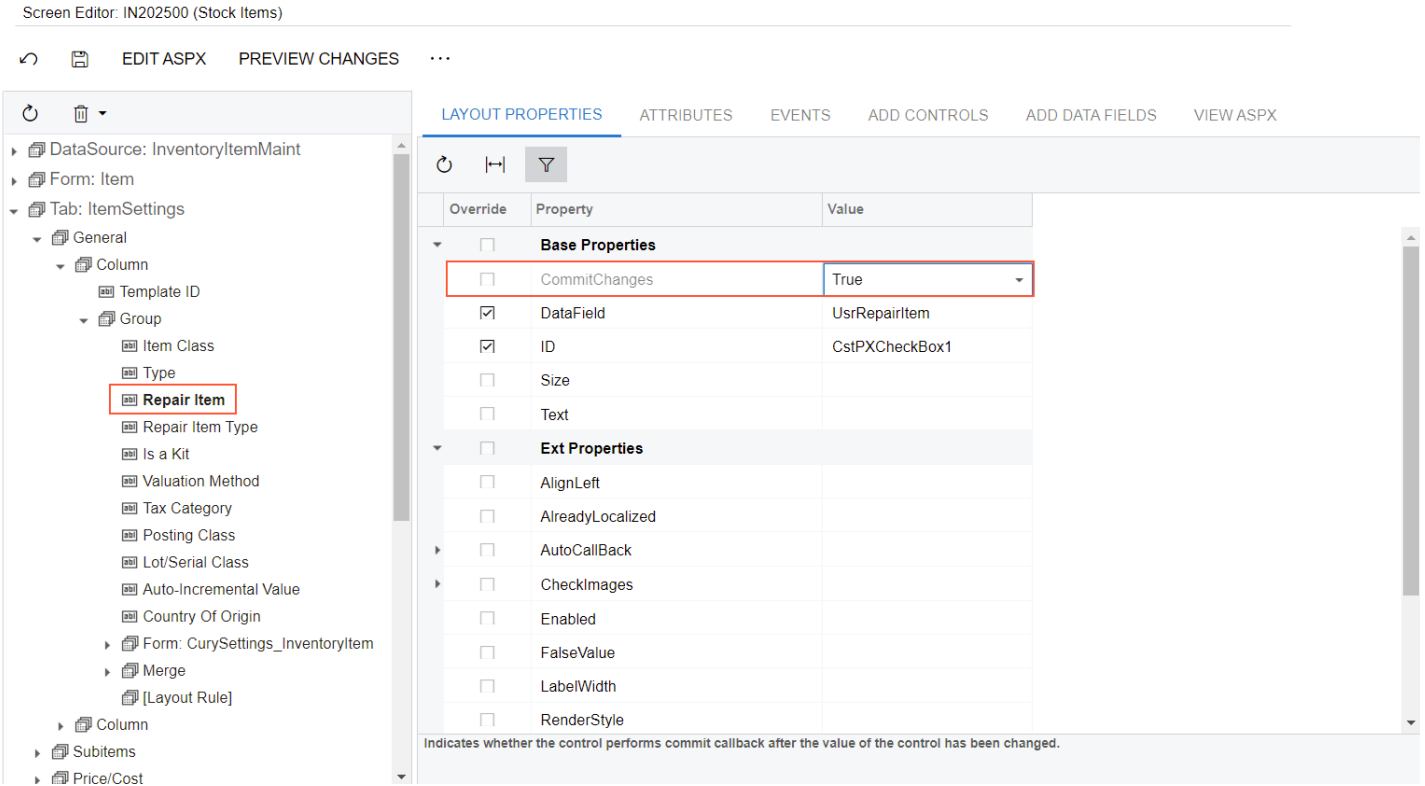


Figure: The CommitChanges property



Lesson Summary

In this lesson, you have learned how to create a control so that you can display on a form a custom field bound to the database. To implement this customization, you have learned how to add the necessary modifications to a customization project and how to publish the project to apply the changes to the system.

As you have completed the lesson, you have added the following elements to the PhoneRepairShop customization project:

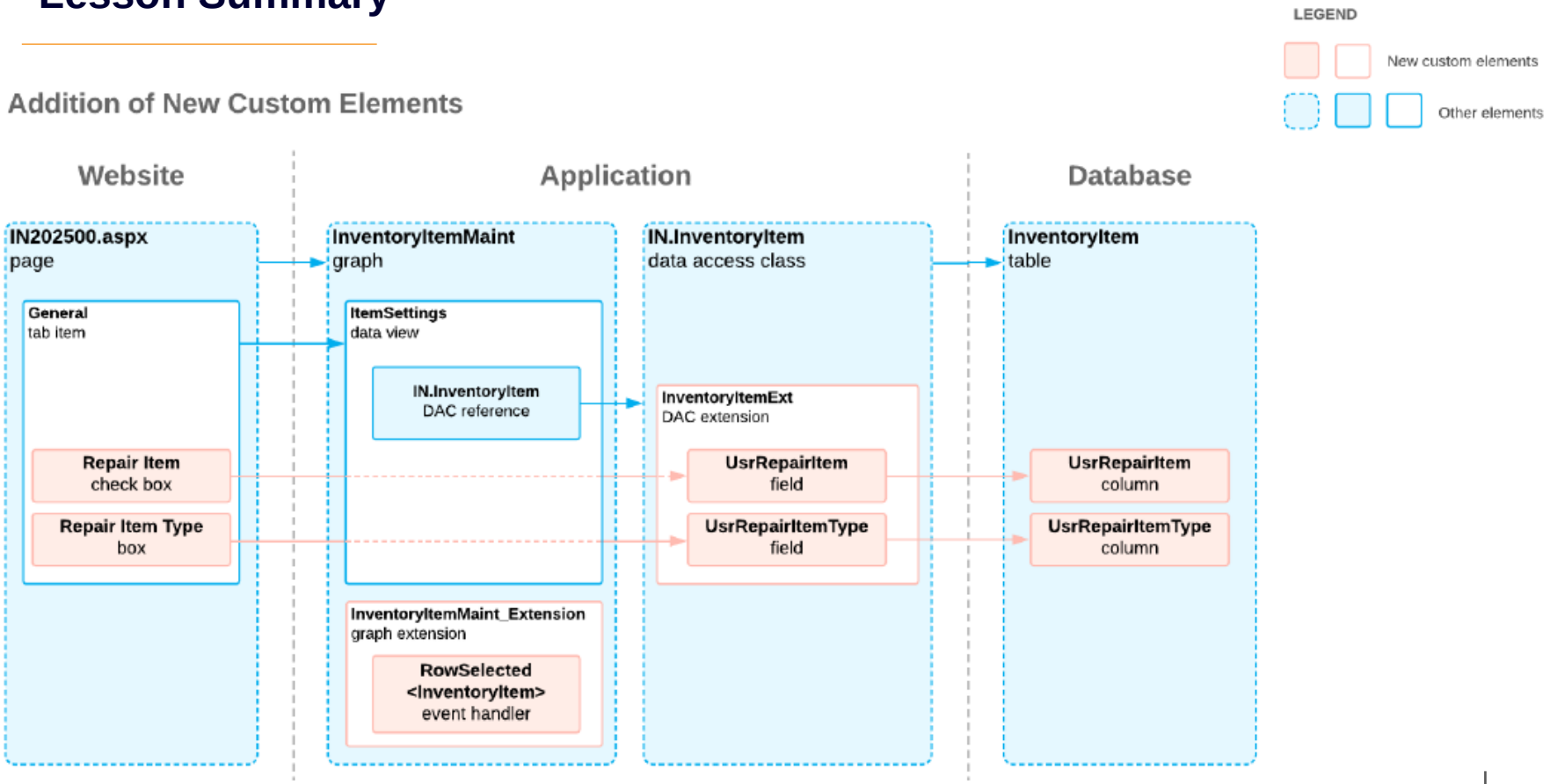
- Two column definitions in the InventoryItem table of the database.

- Two custom field declarations in the extension of the IN.InventoryItem data access class (in the PhoneRepairShop_Code extension library).

- Two controls to display the custom fields on the Stock Items (IN202500) form.

- One custom event handler, which you have added to the InventoryItemMaint graph. You have used the RowSelected event handler to configure the UI presentation logic.

Lesson Summary



Day 2

Lesson 1.2: Configuring the UI—Self-Guided Exercise

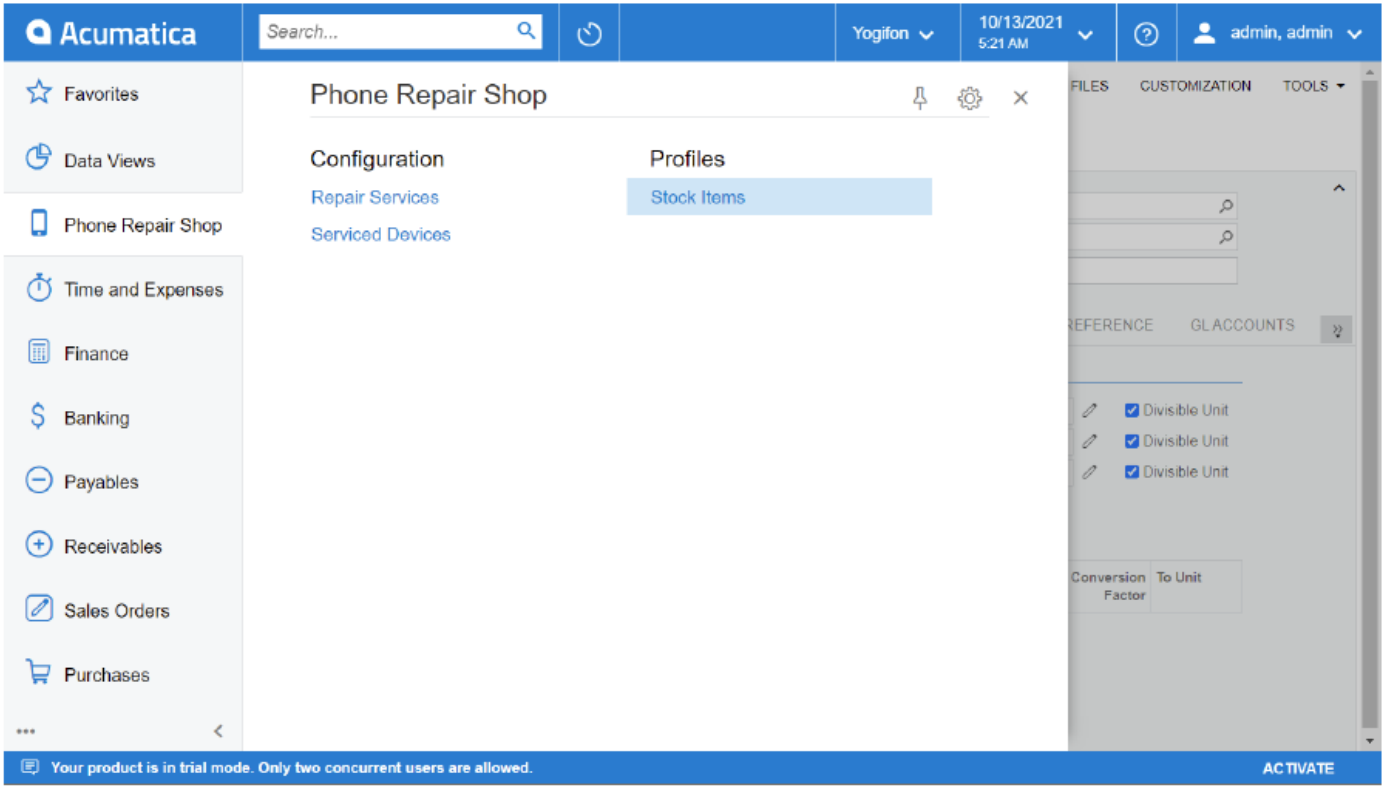
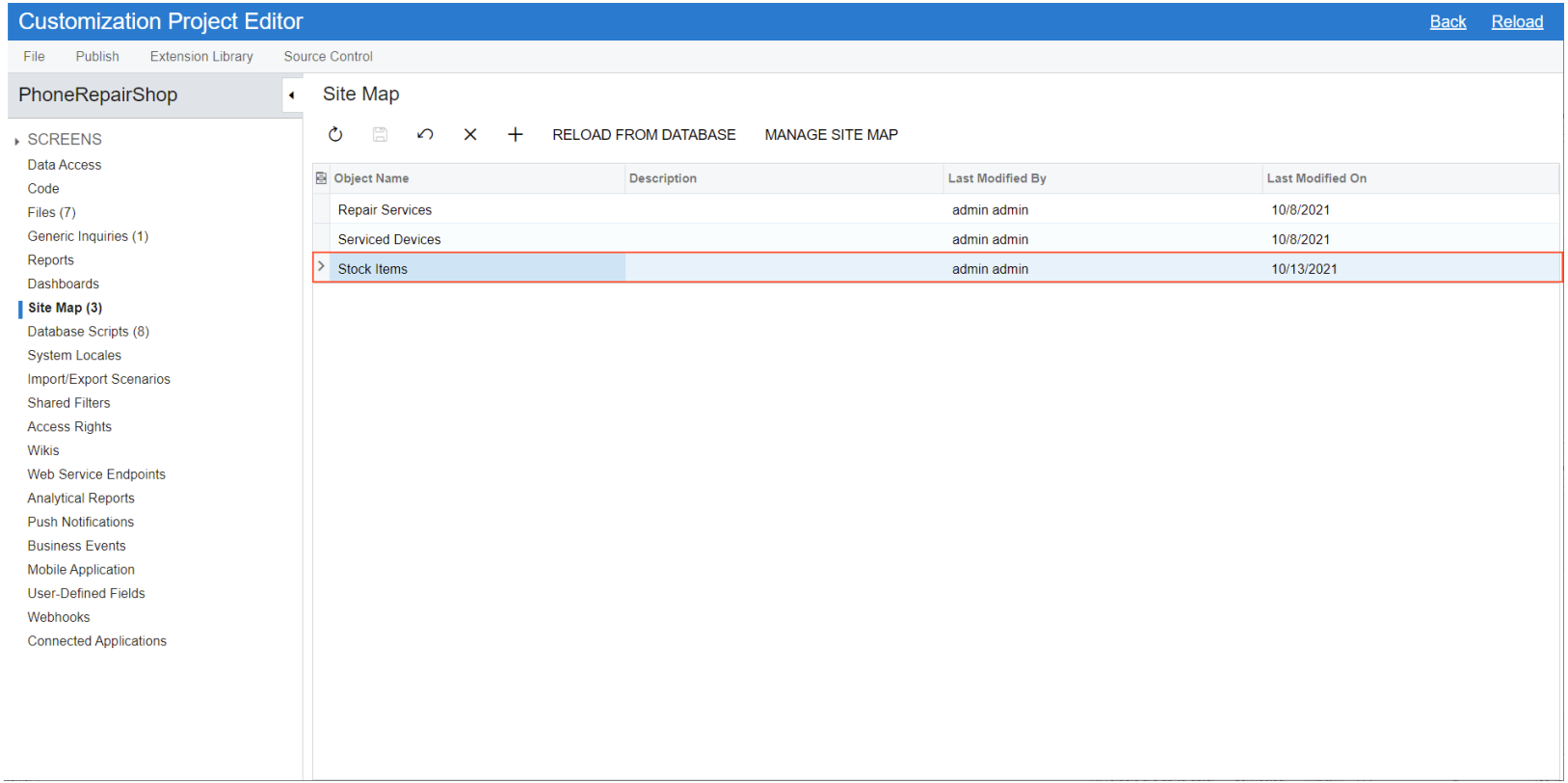


Figure: Stock Items form in the Phone Repair Shop workspace

Figure: SiteMapNode item for the Stock Items form



Part 2: Master-Detail Relationship and Business Logic (Services and Prices Form)

Ray Allen Manufacturing
Acumatica Customer Since 2014

Acumatica
The Cloud ERP

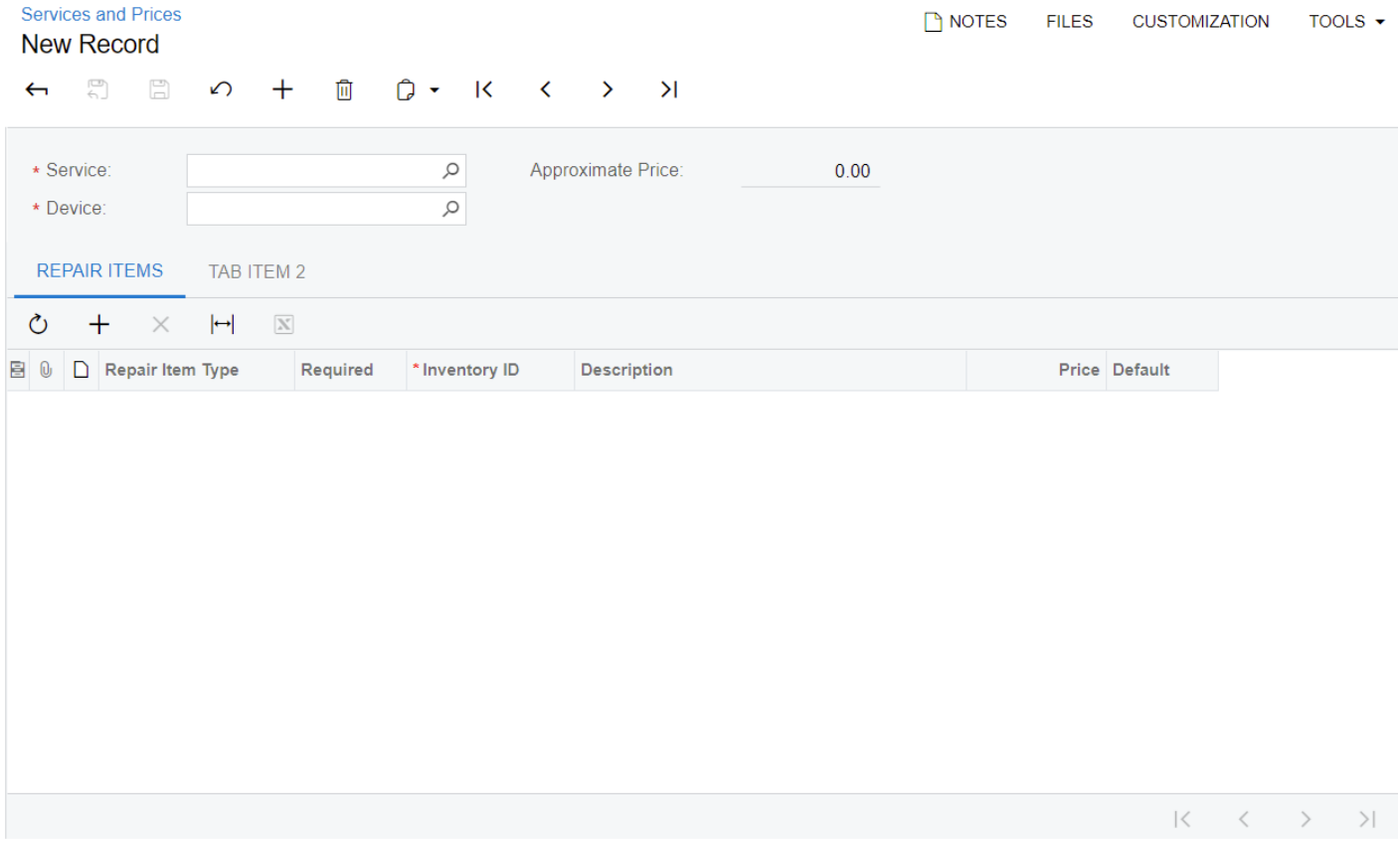
Lesson 2.1: Defining a Master-Detail Relationship

Learning Objectives

In this lesson, you will learn how to do the following:

- Define the master-detail relationship between data
- Implement automatic numbering of detail records

Figure: Services and Prices form



Tables for the Repair Items Tab of the Services and Prices Form

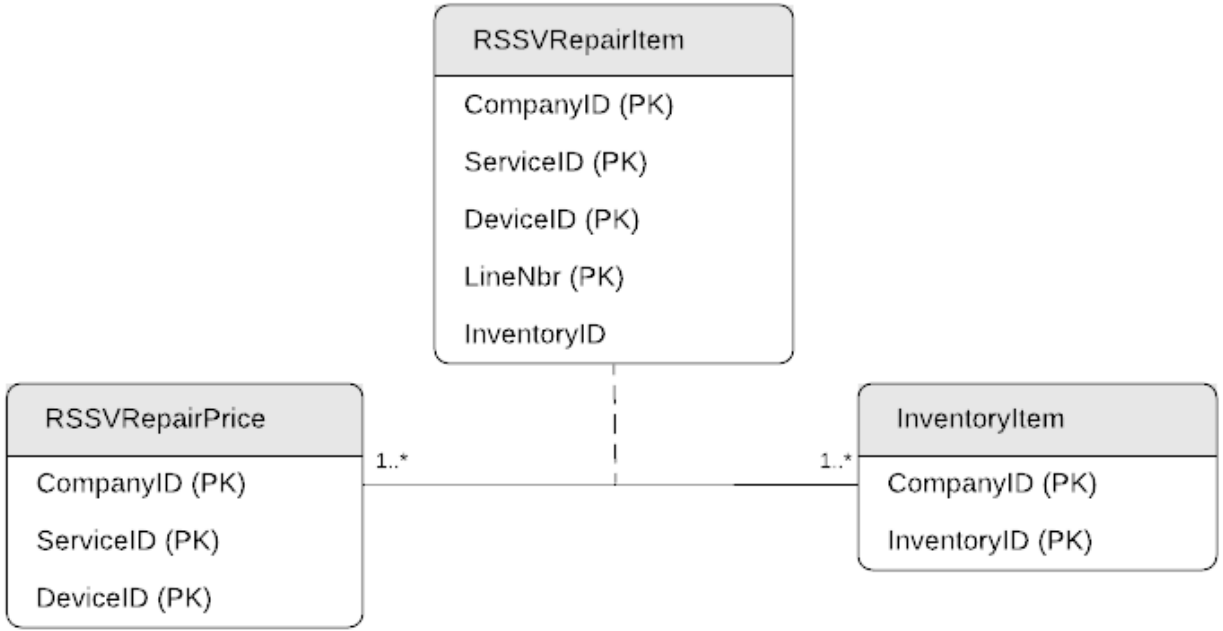


Figure: InventoryID attribute

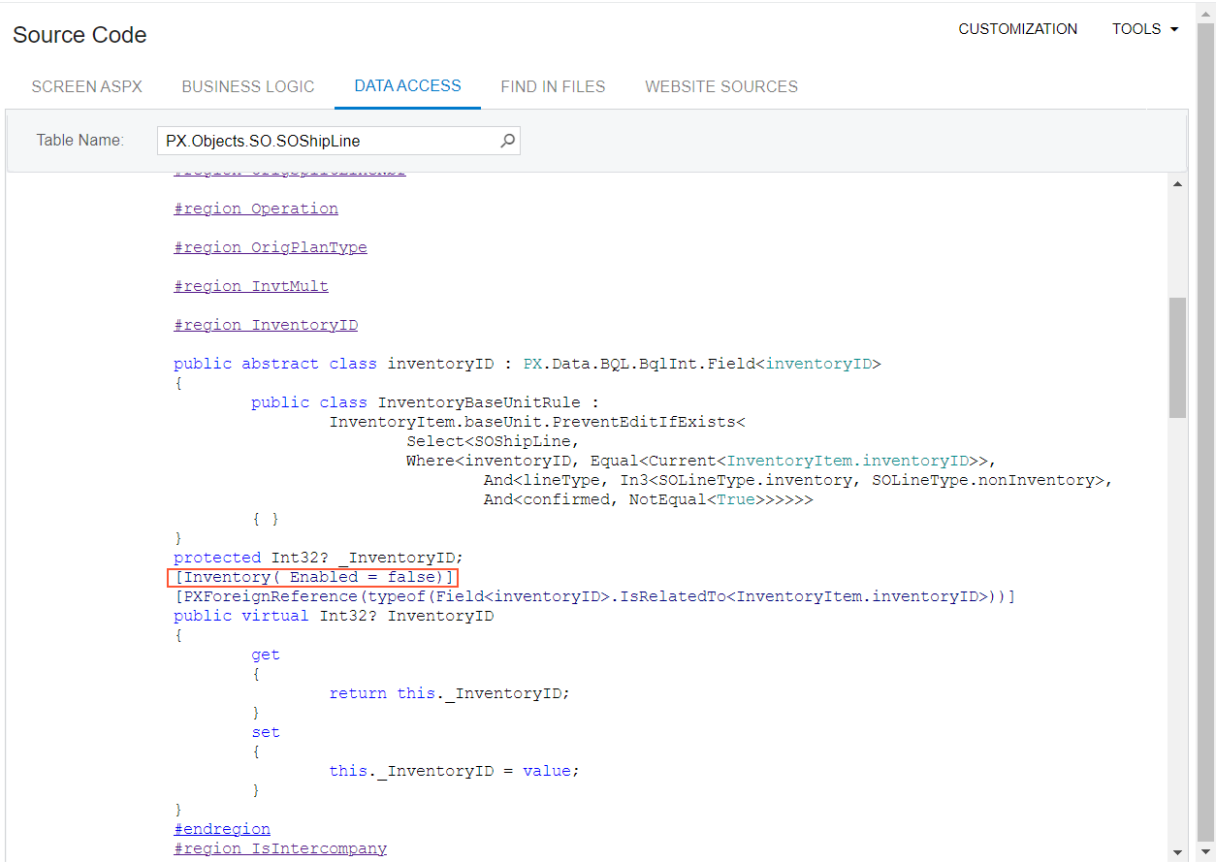
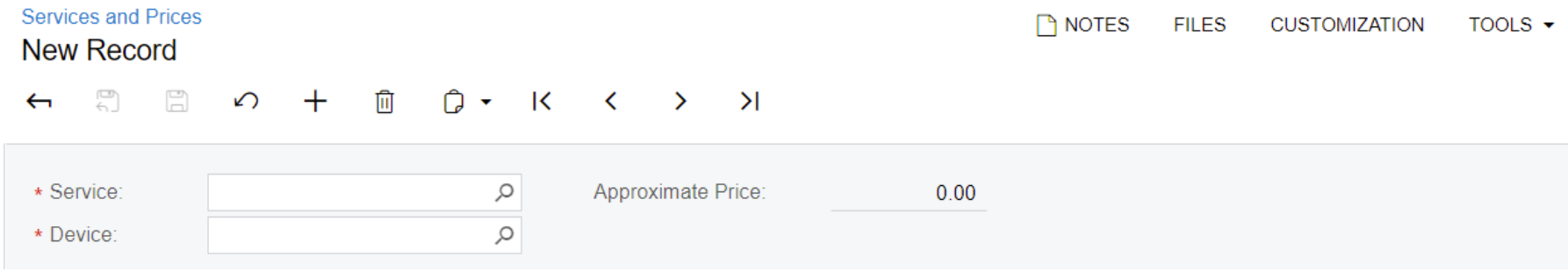


Figure: Two columns in the Summary area



Lesson Summary

In this lesson, you have learned how to set up a master-detail relationship between data.

To create a master-detail form, you have completed the following actions:

1. Set up the master-detail relationship between data access classes as follows:

• Added the PXDBDefault attribute to the key data fields of the detail DAC that are the keys to the master record. The PXDBDefault attribute provides the default value for the key field of the detail DAC.

• Added the PXParent attribute to the first foreign key data field of the detail DAC. The PXParent attribute enables cascading deletion of detail records on deletion of the master record.

2. Defined two data views that select the master-detail data for the form. You have used the FromCurrent parameter in the detail data view type to select records for a particular master record.

Lesson Summary

3. Bound the UI controls that display the data on the ASPX page as follows:

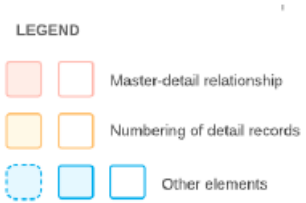
• Specified the master data view for the datasource control on the page (in the PrimaryView property of the control).

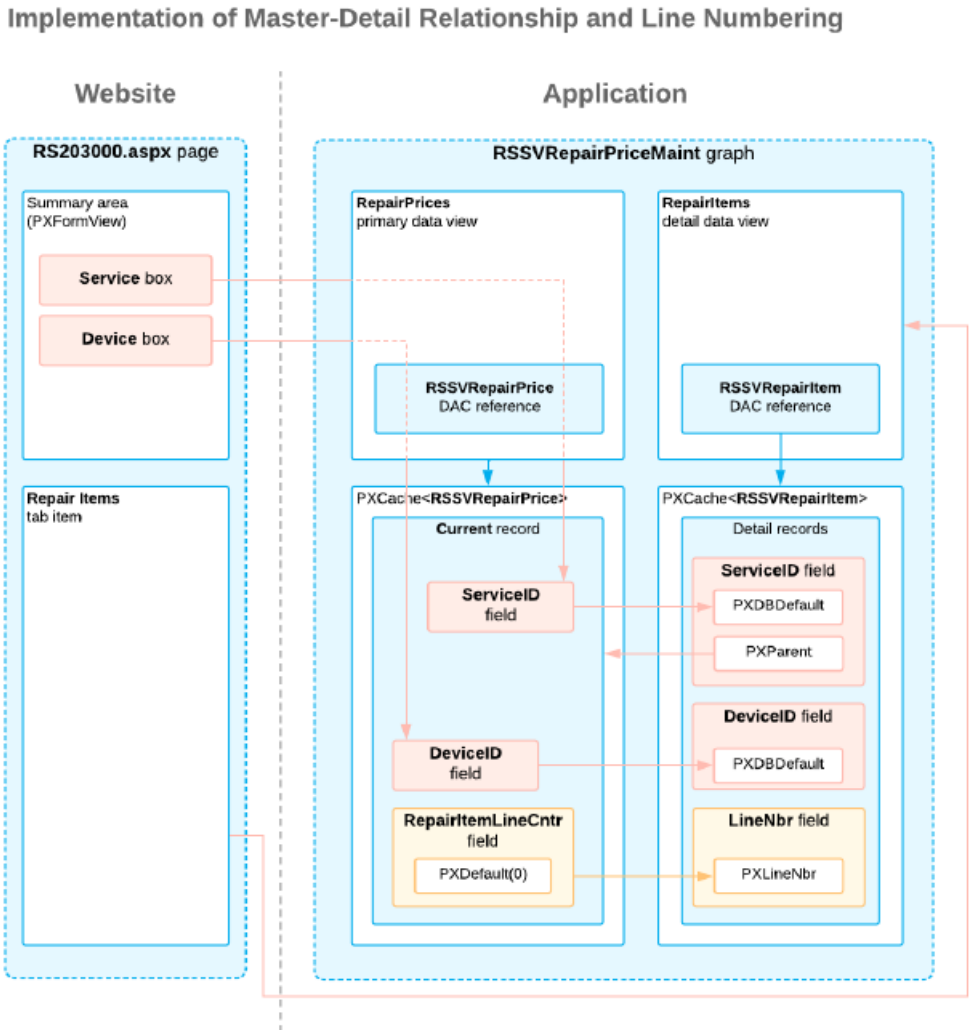
• Specified the master data view for the form (in the DataMember property of the control).

• Specified the detail data view for the grid (in the DataMember property of the control).

In the lesson, you have also implemented the numbering of detail data records by using the PXLineNbr attribute.

Lesson Summary





Day 3

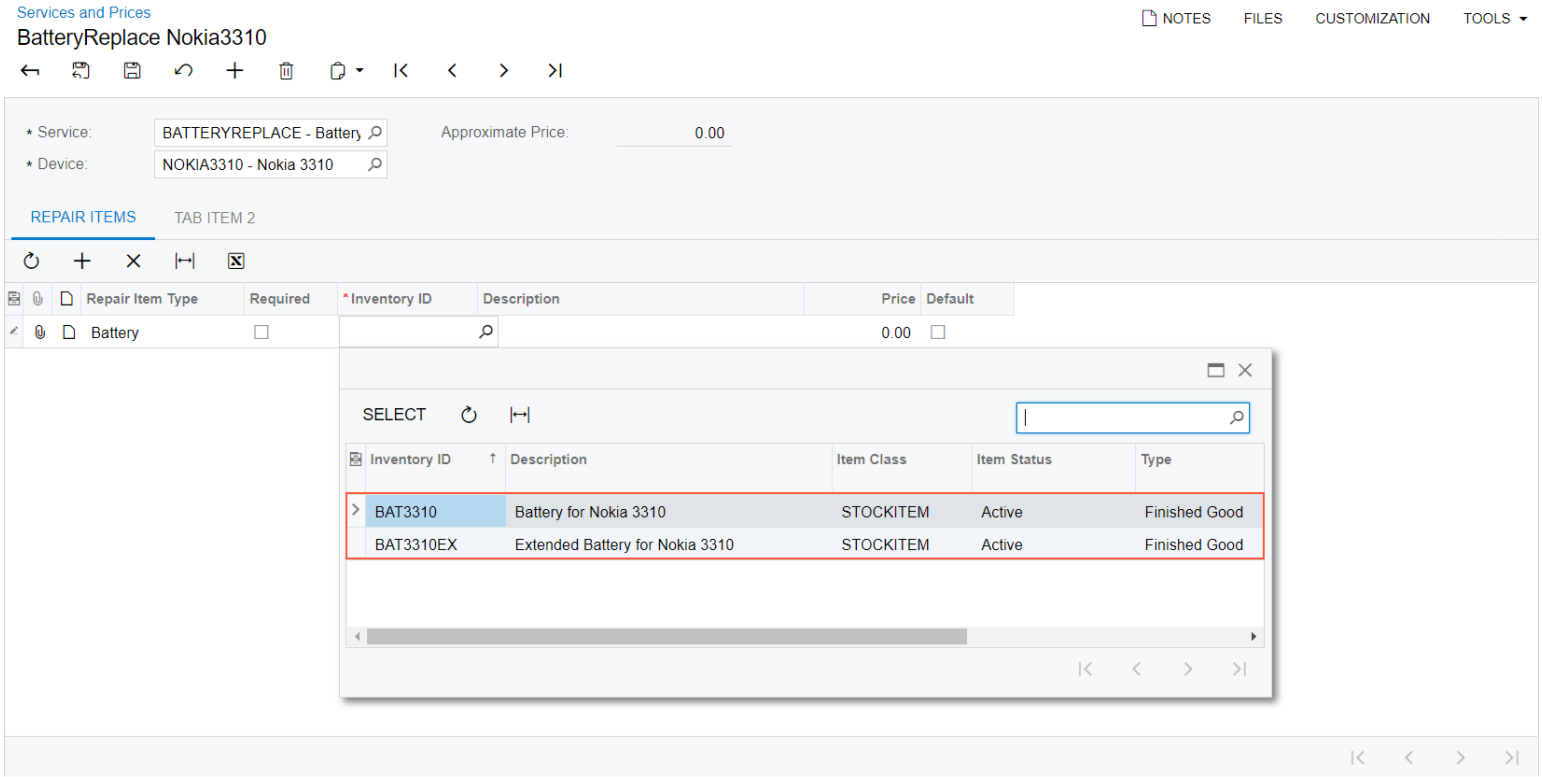
Lesson 2.2: Defining the Business Logic

Learning Objectives

In this lesson, you will learn how to do the following:

- Restrict the possible values of a field by using the PXRestrictor attribute
- Mark localizable messages in code
- Update the fields of the same data record on update of a field of this record
- Update the fields of other records on update of a field
- Learn one of the possible ways to retrieve a data record from the database in code by using the static PXSelectorAttribute.Select<>() method

Figure: Inventory ID selector



Lesson Summary

In this lesson, you have configured the business logic of the Repair Items tab of the Services and Prices (RS203000) form as follows:

• You have used the PXRestrictor attribute to configure a restriction on the values in the Inventory ID column.

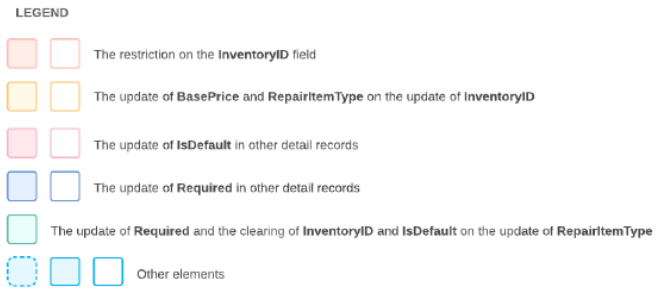
• You have used a class with the PXLocalizable attribute specified to make the text available for localization.

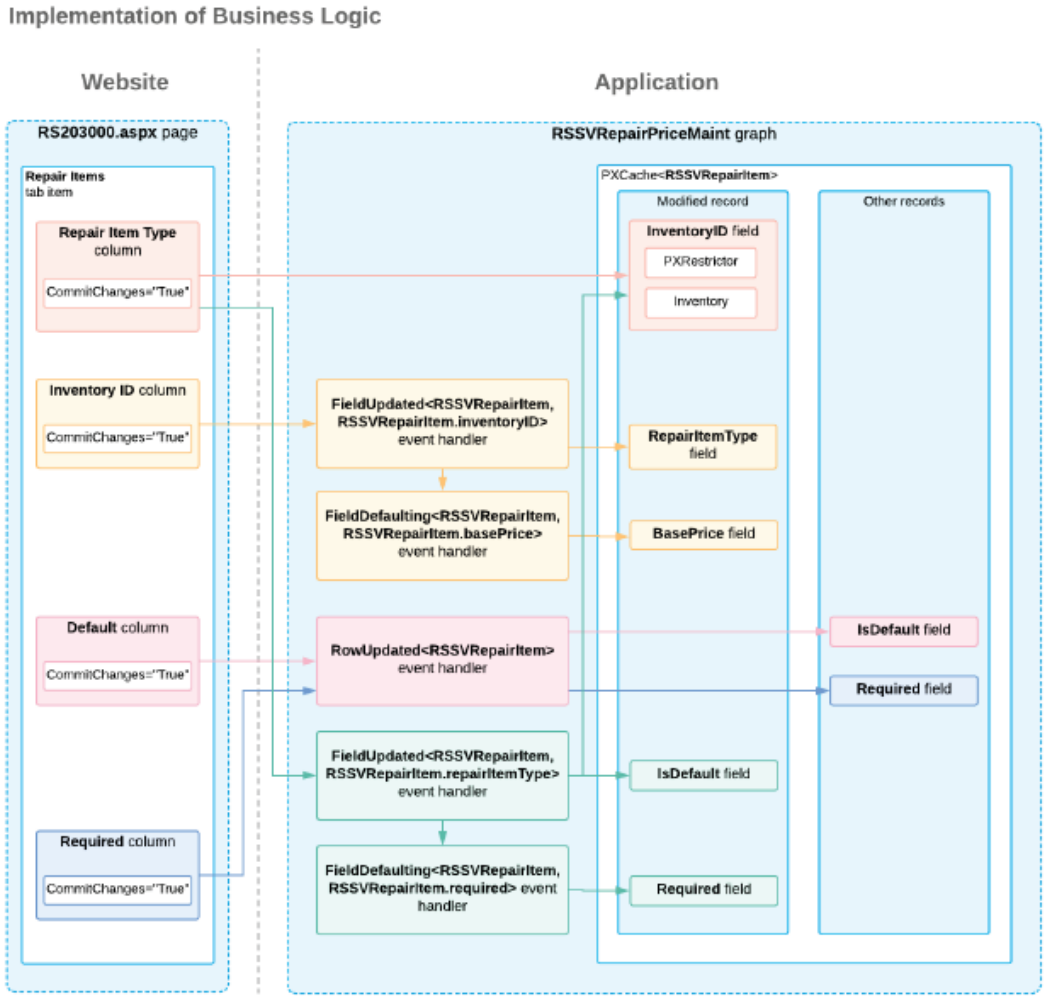
• You have used the FieldUpdated and FieldDefaulting event handlers to modify the values of a detail record on update of the column of this detail record. In the event handlers, you have used the PXSelectorAttribute.Select<>() method to obtain the stock item record with the inventory ID selected in the updated field. You have also used the SetValueExt<field> and SetDefaultExt<field> methods to trigger additional events for particular fields.

Lesson Summary

• You have used the RowUpdated event handler so that when particular fields of one detail record are updated, the values in the other detail records are modified. In this event handler, you have used the PXCache.Update() method to update in PXCache the records other than the record for which the event has been raised. You have also used the PXView.RequestRefresh() method to display in the UI the changes in PXCache that you have made in the event handler. You have also used LINQ for the filtering of the records of the data view.

Lesson Summary





Part 3: Custom Tab (Stock Items Form)

Devil's Peak Brewing Company
Acumatica Customer Since 2016

Acumatica
The Cloud ERP

Lesson 3.1: Adding a New Tab

Learning Objectives

In this lesson, you will learn how to do the following:

- Add a custom data view for an Acumatica ERP form
- Create a custom tab on an Acumatica ERP form
- Conditionally hide a custom tab on an Acumatica ERP form

Figure: The Compatible Devices tab

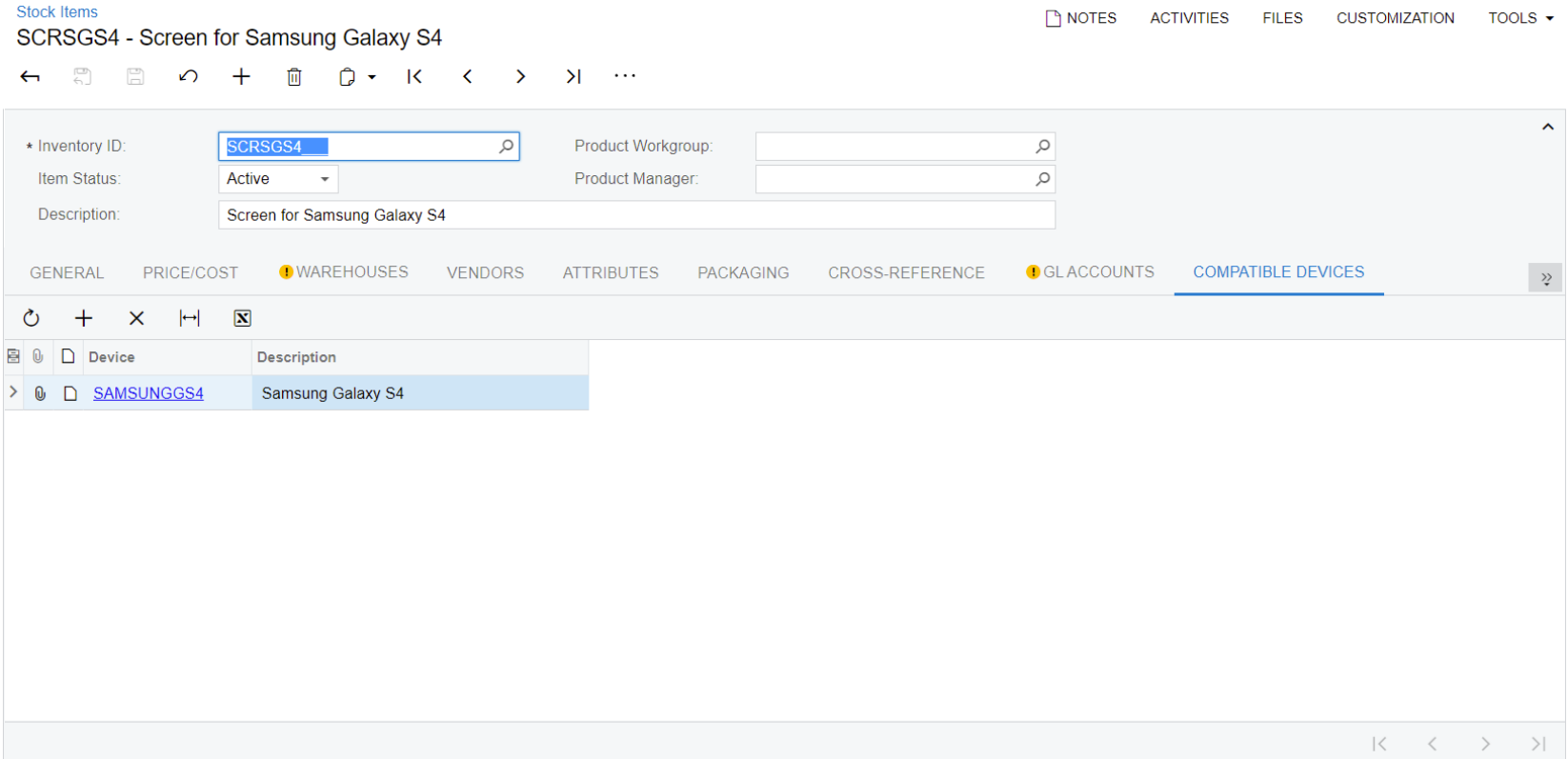


Figure: Addition of the tab item

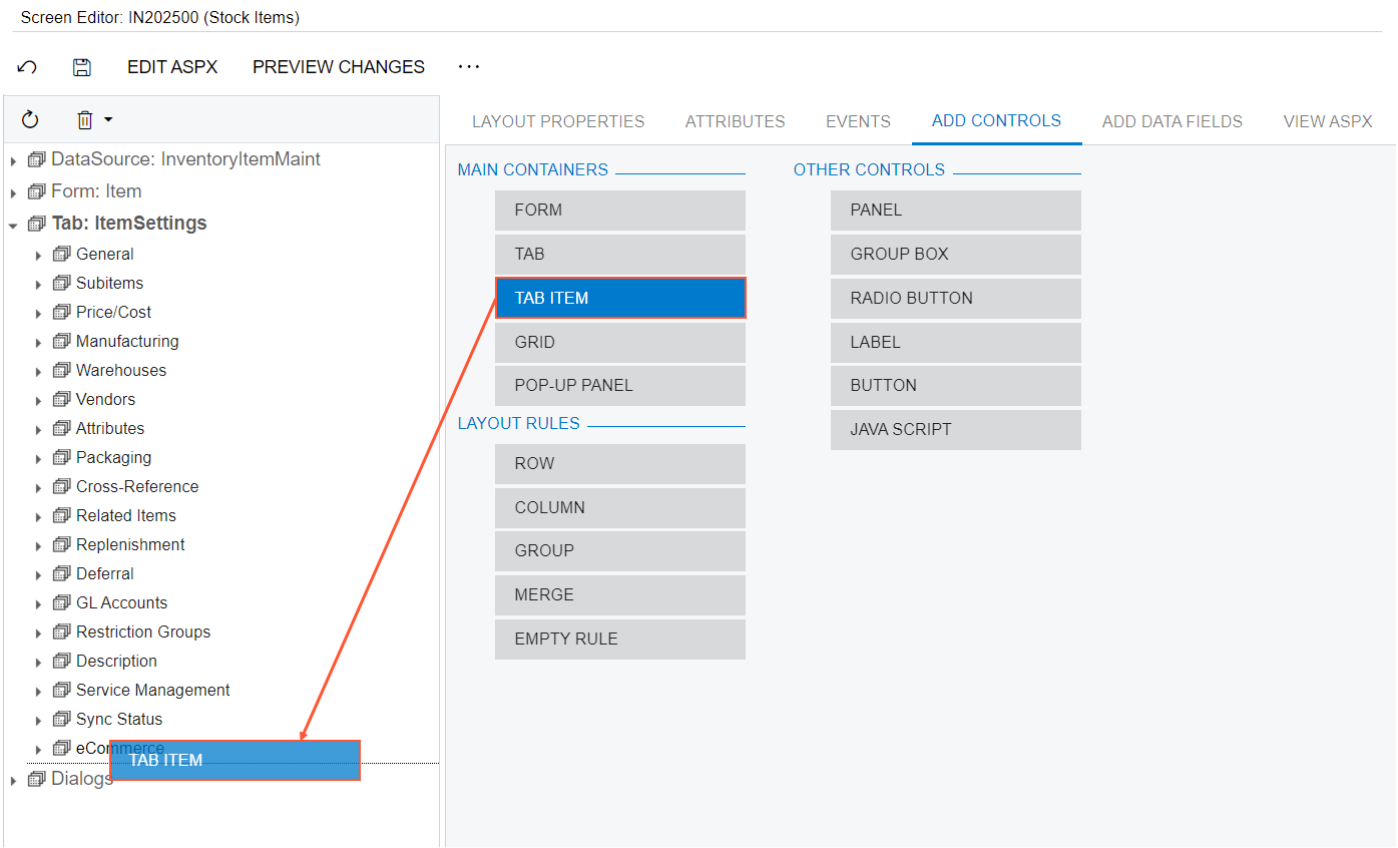


Figure: The Device and Description nodes

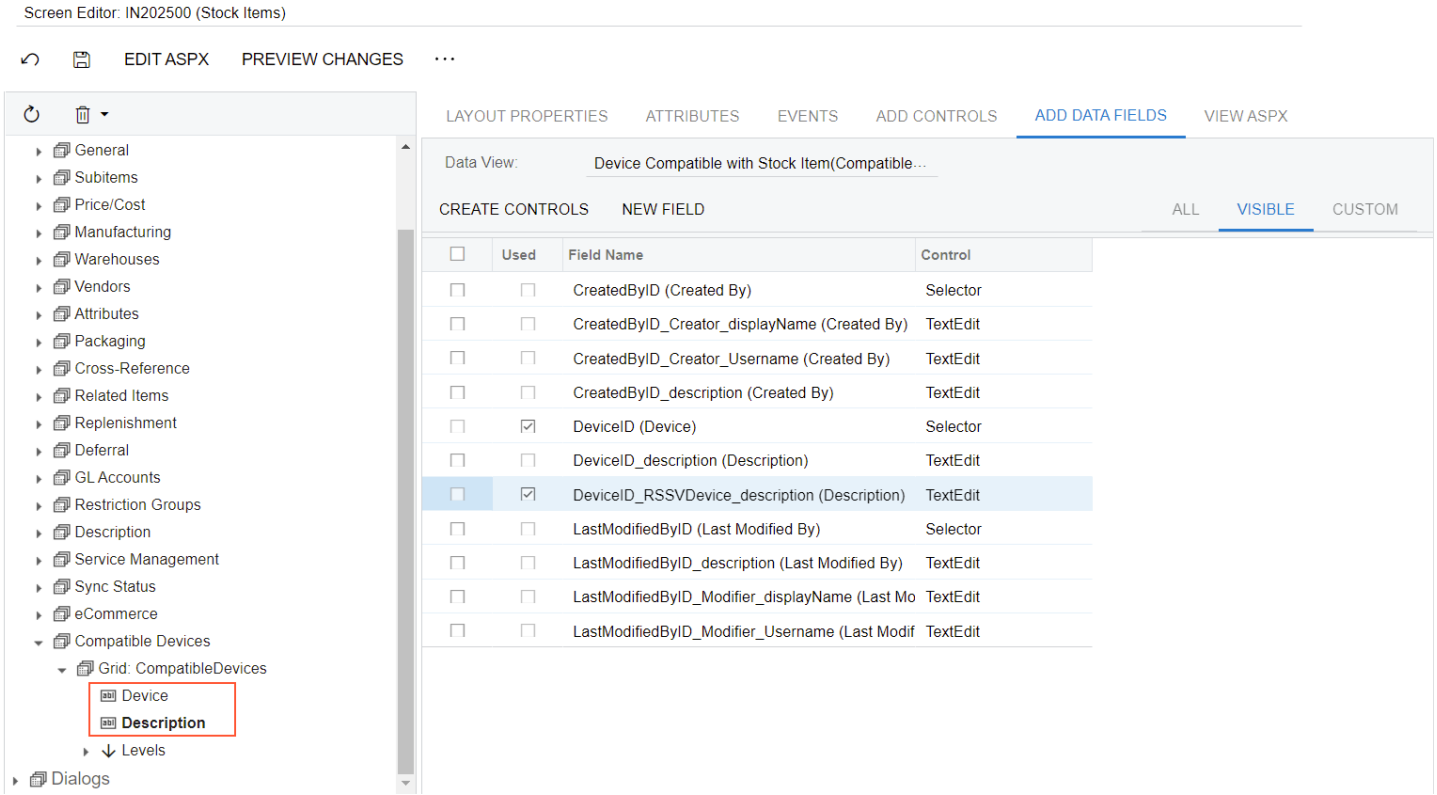


Figure: The new tab

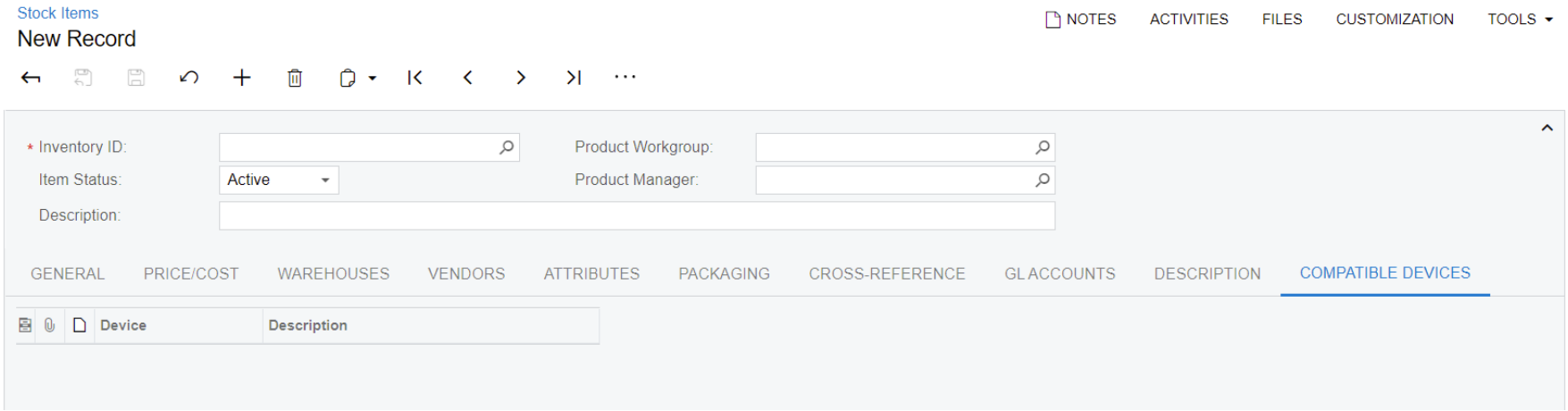


Figure: The final layout of the tab

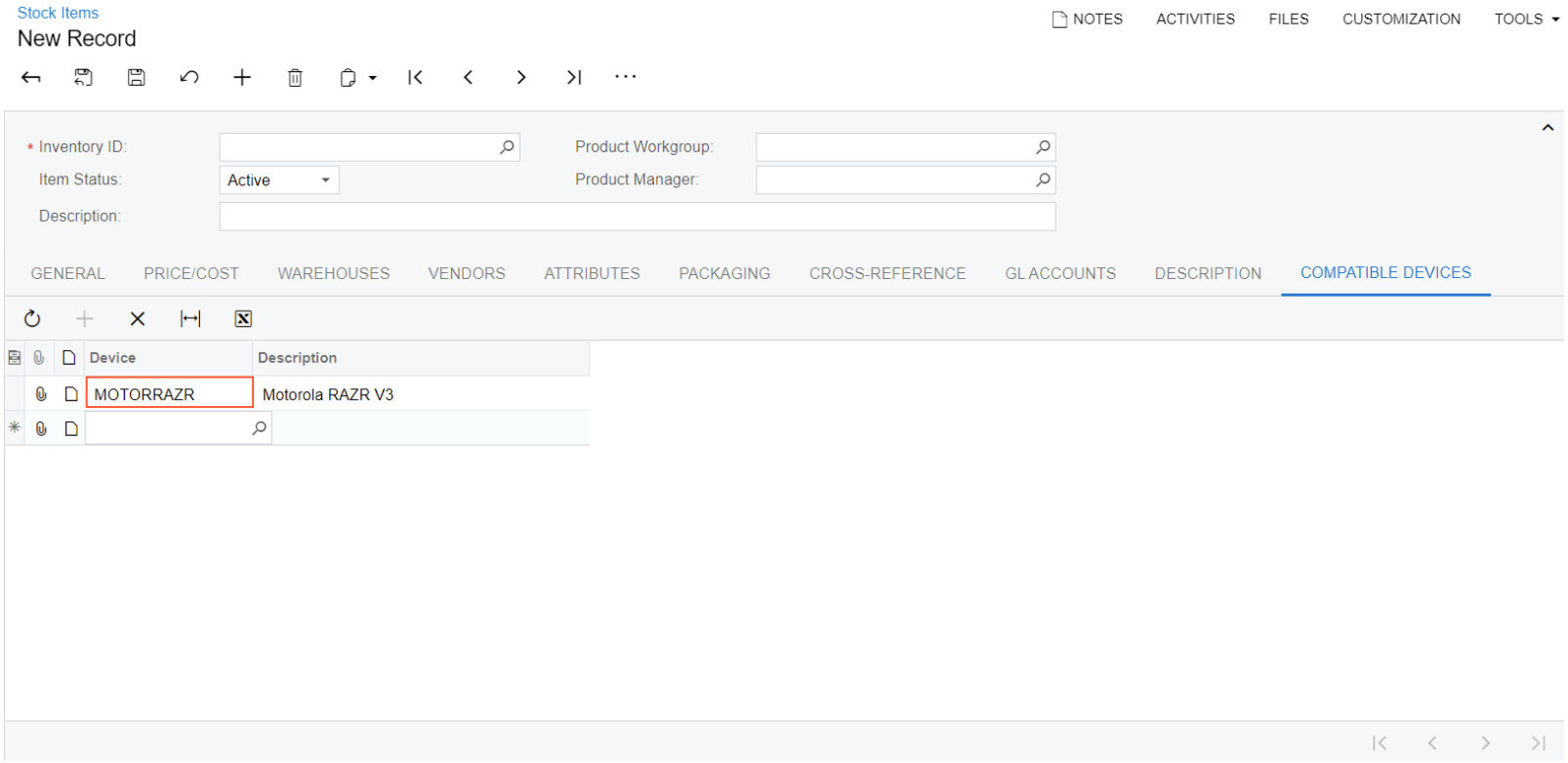


Figure: ASPX code of the tab

Lesson Summary

In this lesson, you have practiced implementing extensions for both the business logic and the user interface of an existing form, and you have learned the following tasks:

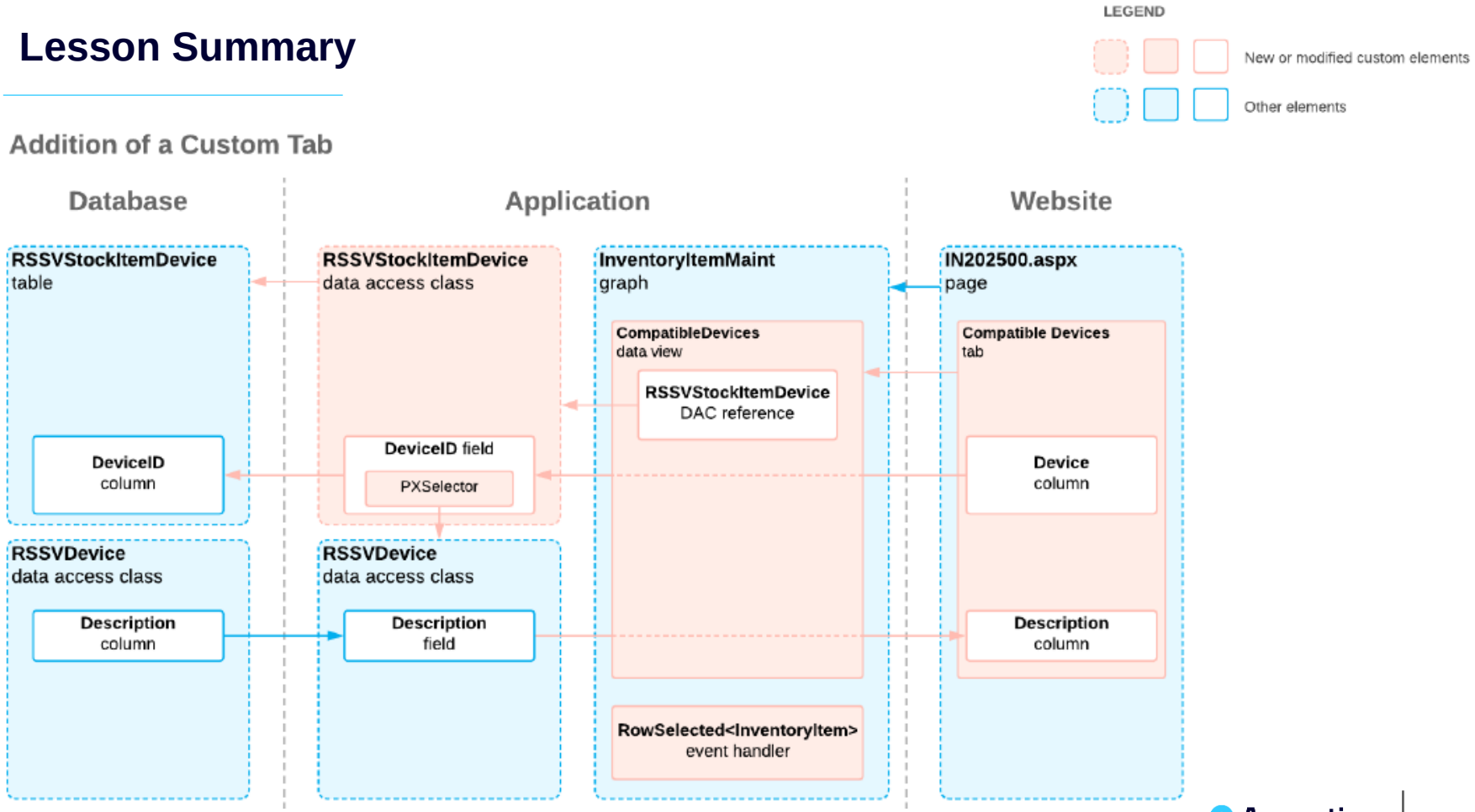
- Creating a control container on a form
- Creating the data view in the extension of the graph for the form to provide data for the container
- Conditionally hiding a tab

During the lesson, you have added to the customization project the following items:

- On the Stock Items (IN202500) form, the Compatible Devices tab, which contains the Device and Description columns
- The CompatibleDevices data view in the InventoryItemMaint_Extension graph
- The RSSVStockItemDevice DAC

You have also modified the RowSelected<InventoryItem> event handler.

Lesson Summary



Day 4

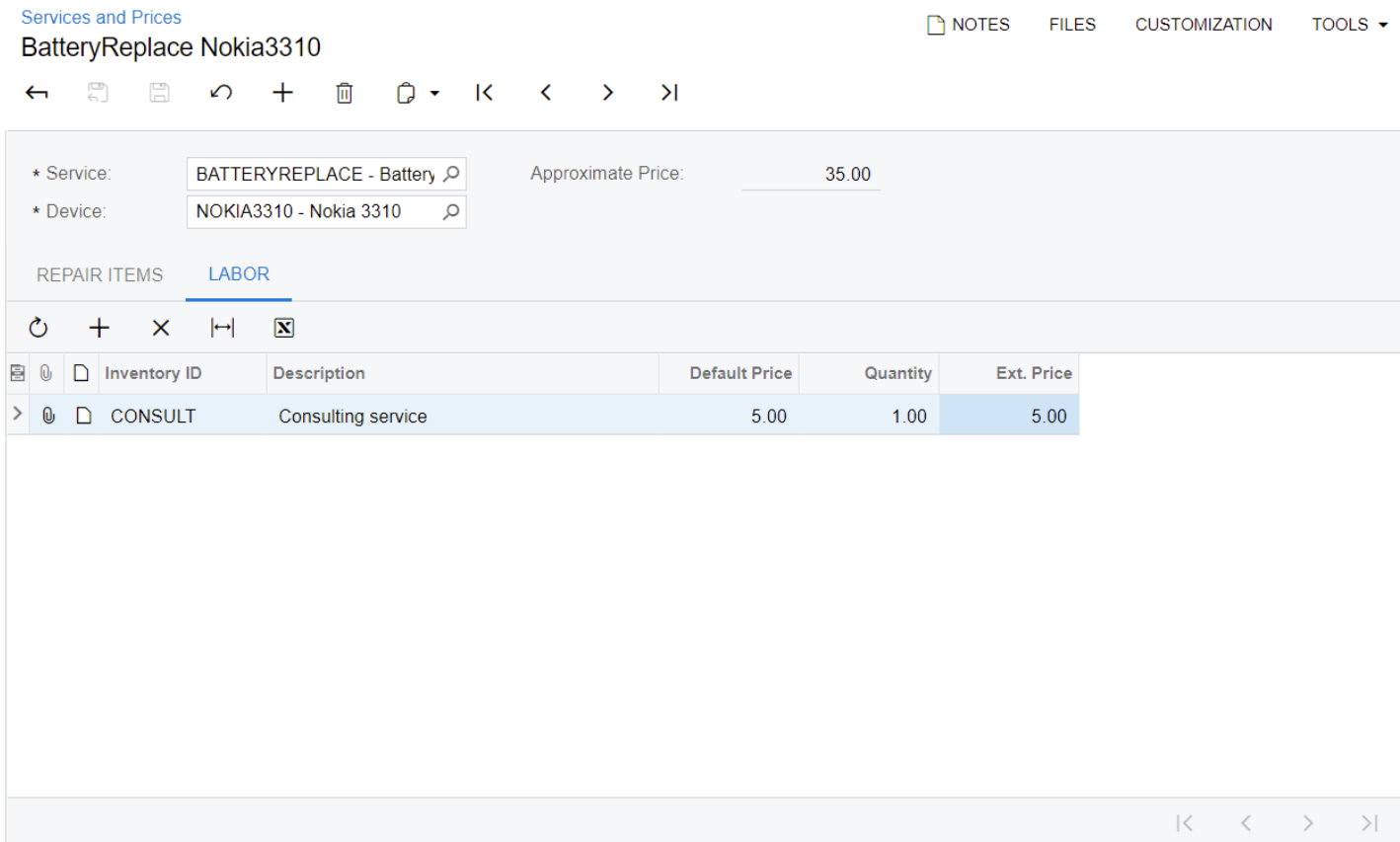
Part 4: Calculations and Insertion of a Default Record (Services and Prices Form)

Lesson 4.1: Calculating Field Values

Learning Objectives

In this lesson, you will learn how to use the PXFormula attribute for calculations.

Figure: The Labor tab



Relationships Between Database Tables

Tables for the Labor Tab of the Services and Prices Form

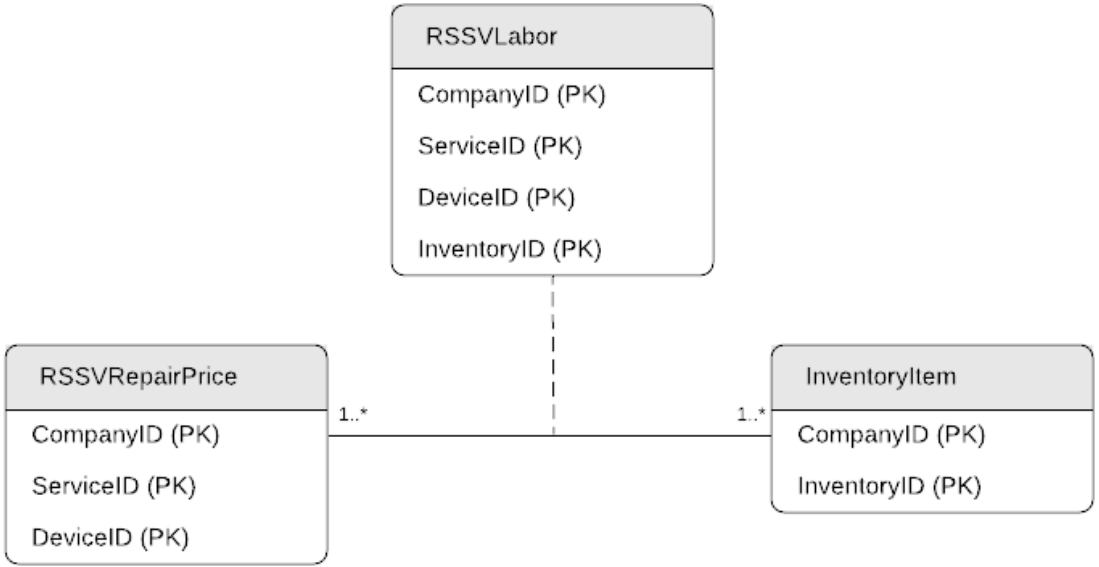
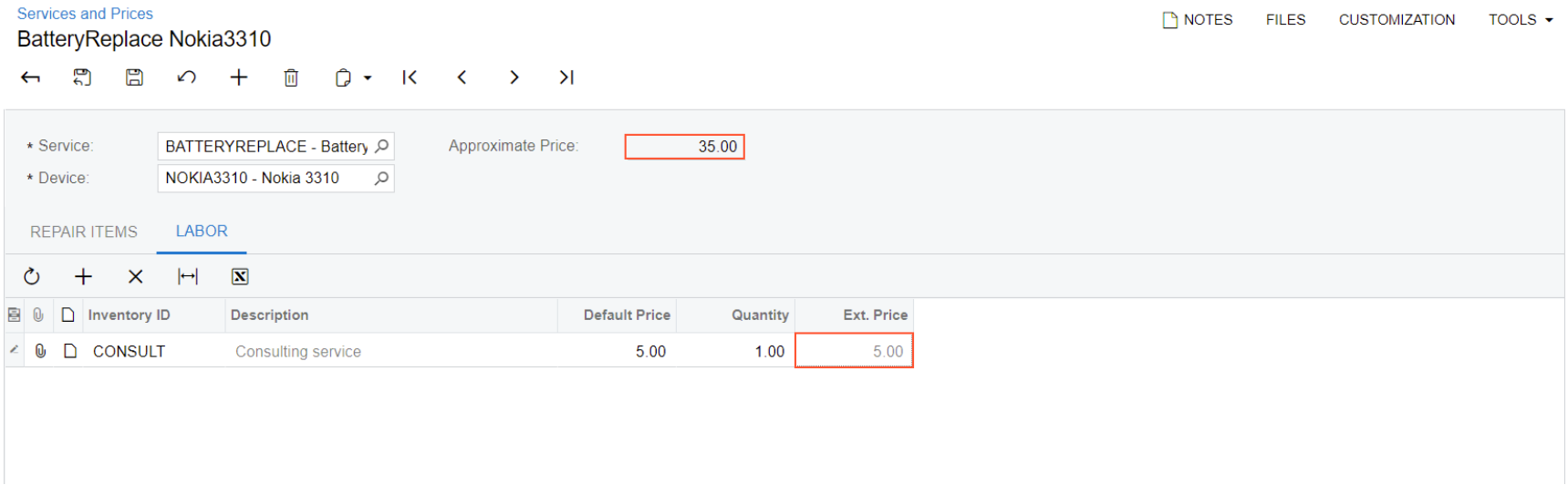


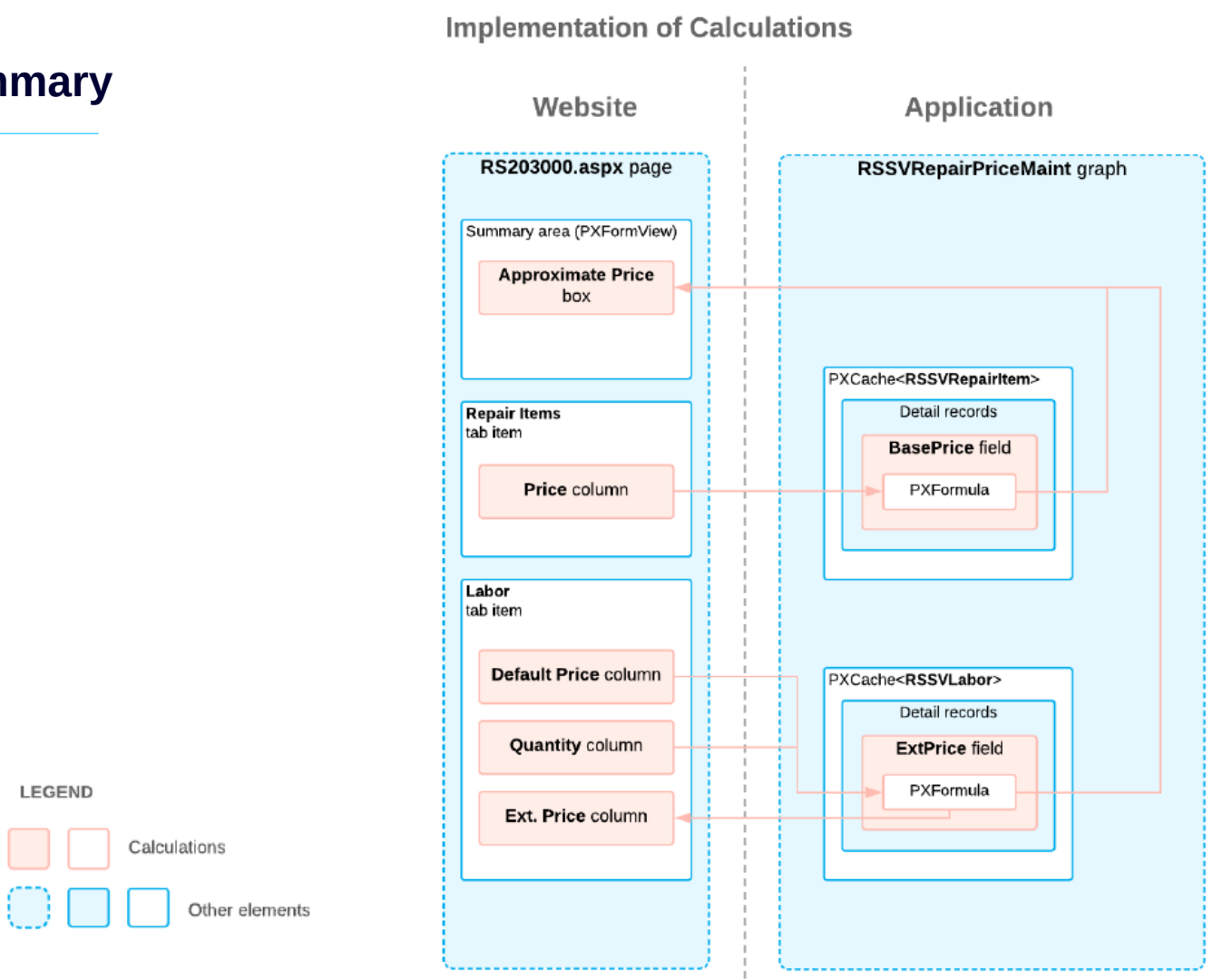
Figure: The calculation of the price



Lesson Summary

In this lesson, you have implemented the calculation of the value of the Approximate Price box on the Services and Prices (RS203000) form. You have used the PXFormula attribute to configure the calculation.

Lesson Summary

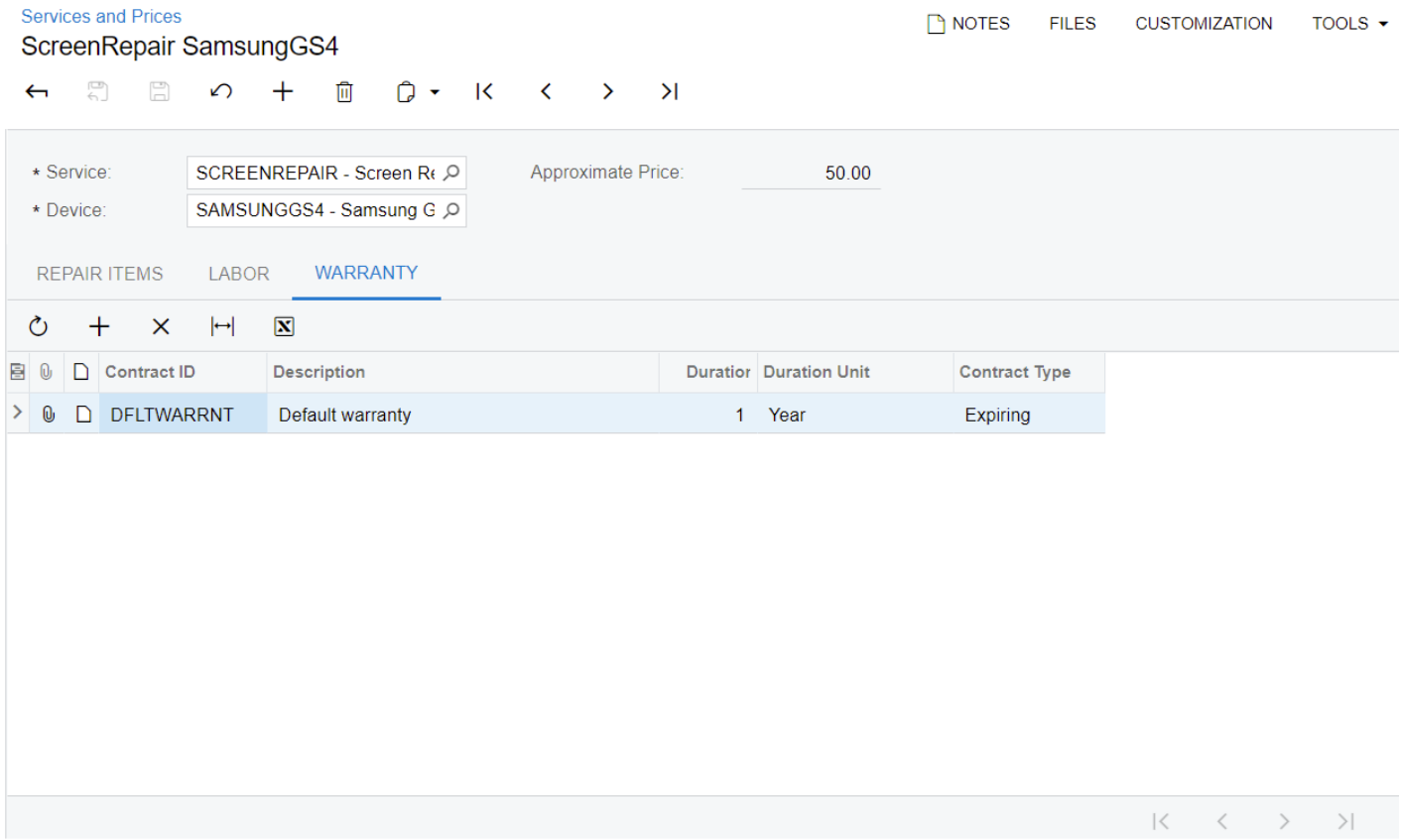


Lesson 4.2: Inserting a Default Detail Record

Learning Objectives

In this lesson, you will learn how to add a default detail record to the grid.

Figure: The Warranty tab



Relationships Between Database Tables

Tables for the Warranty Tab of the Services and Prices Form

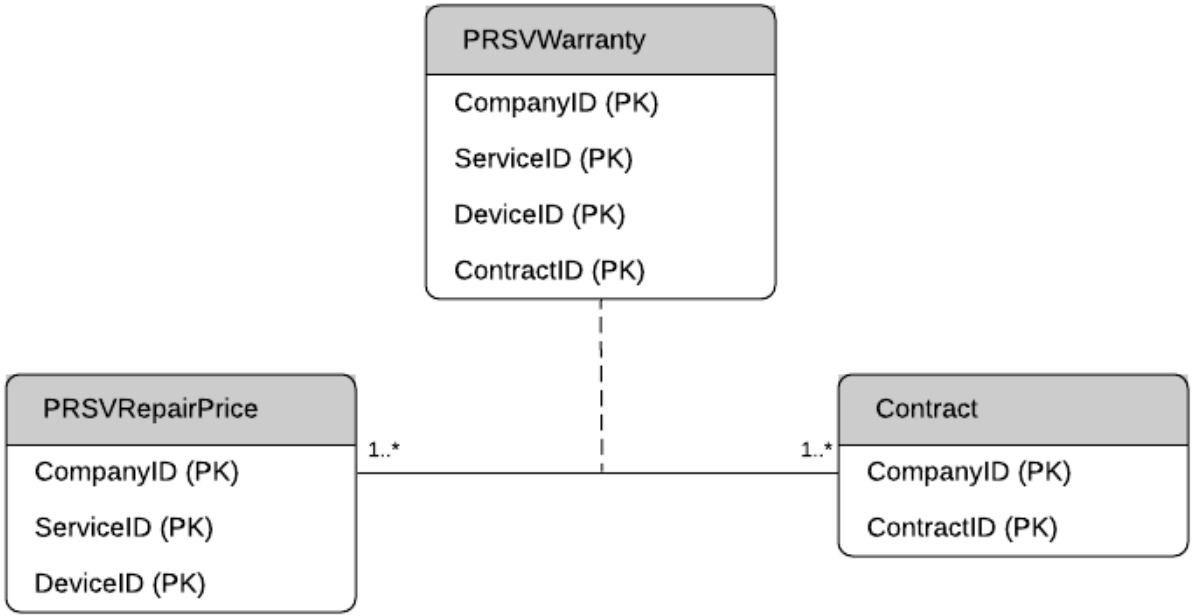
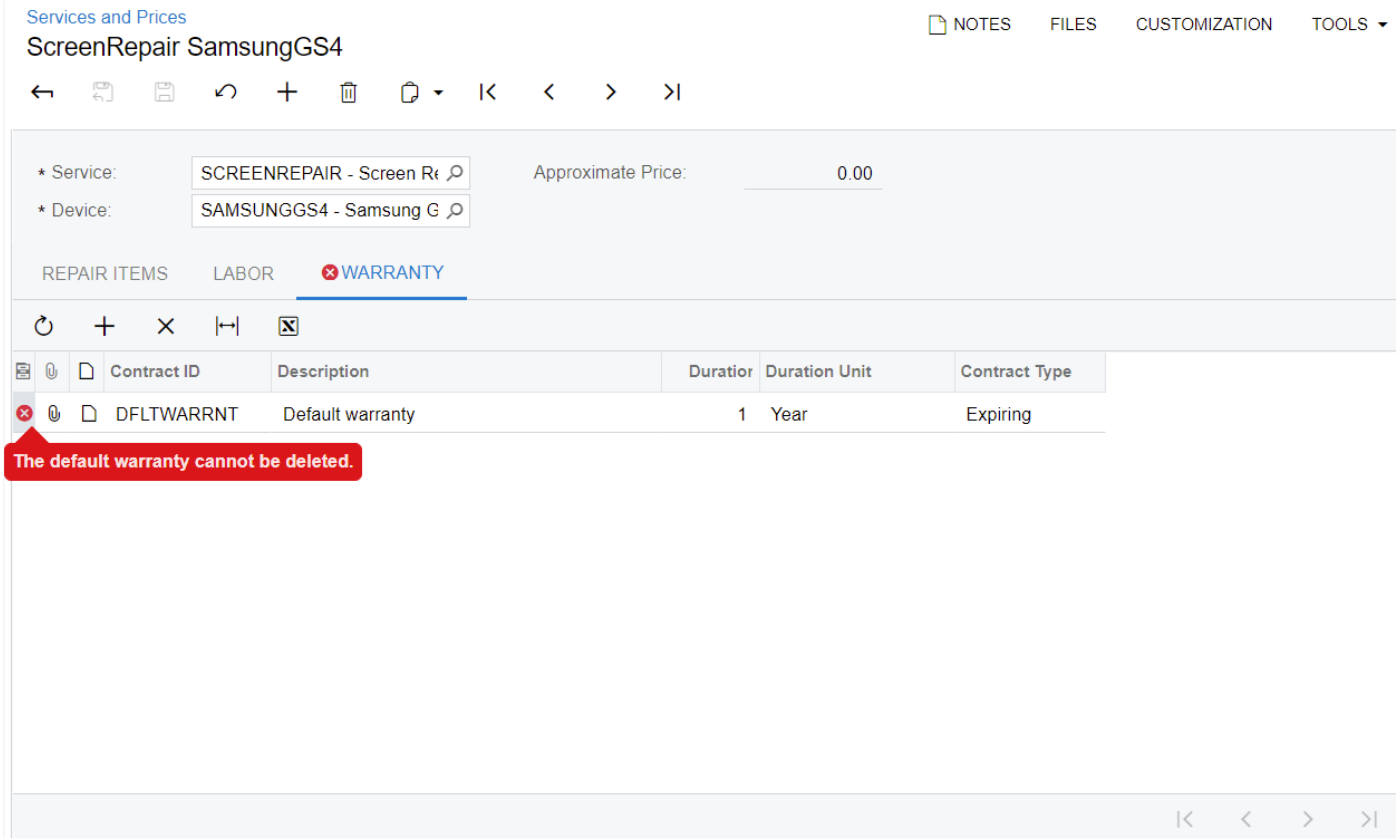


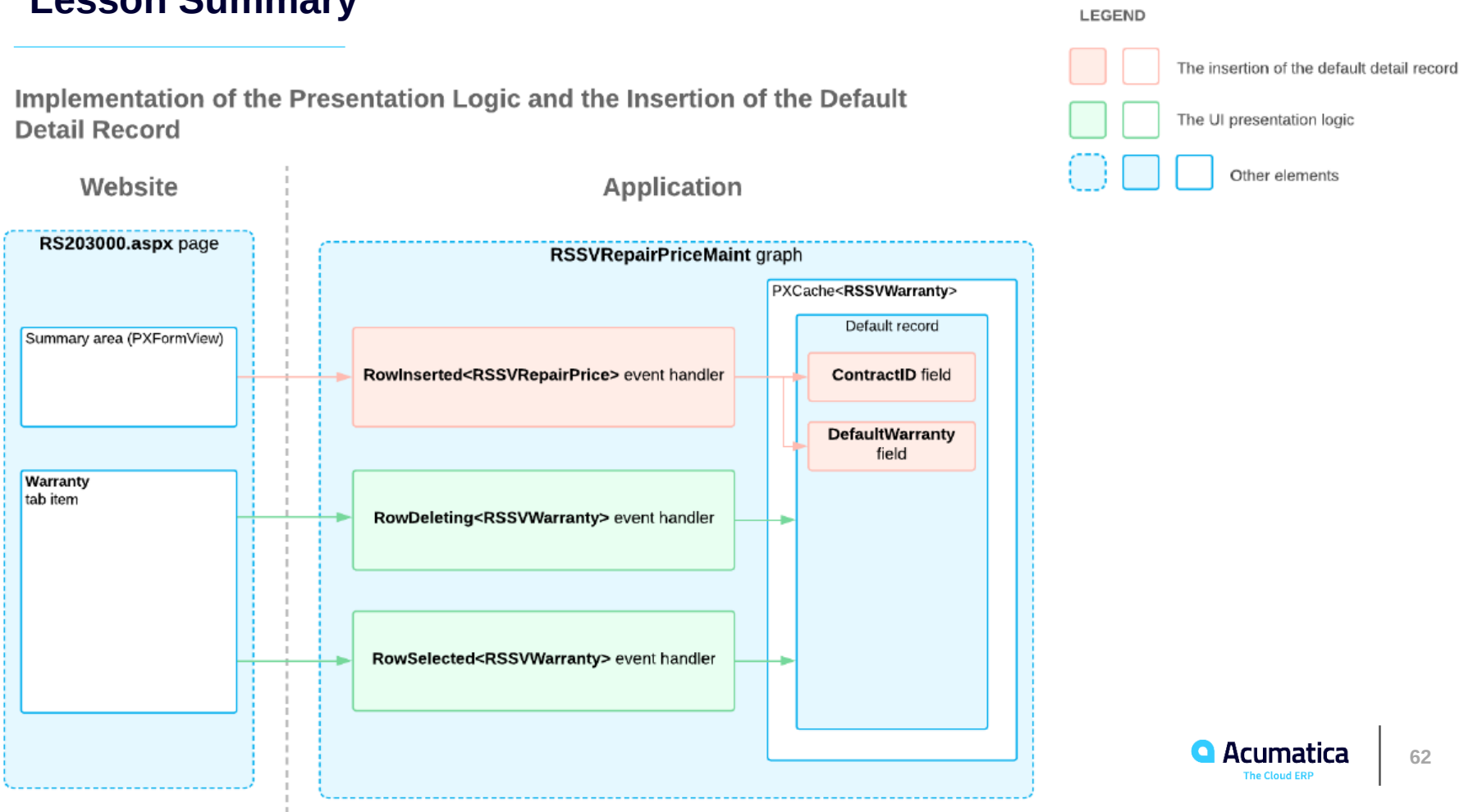
Figure: The error on the form



Lesson Summary

In this lesson, you have learned how to add a default detail record to the grid. To add a default record, you have used the RowInserted event handler for the parent DAC. You have also defined the UI presentation of the default detail record in the RowSelected and RowDeleting event handlers for the child DAC.

Lesson Summary



No Reliance
This document is subject to change without notice. Acumatica cannot guarantee completion of any future products or program features/enhancements described in this document, and no reliance should be placed on their availability.

Confidentiality: This document, including any files contained herein, is confidential information of Acumatica and should not be disclosed to third parties.

Acumatica
The Cloud ERP

Thank you

Dmitrii Naumov