Acumatica
The Cloud ERP

T200 Maintenance Forms

Vidhyalakshmi Hariharasubramanian

Technical Account Manager

Timing and Agenda

June 27, 2022 -10 AM -11 AM

Day 1

Lesson 1.1: Prepare a Customization Project

Lesson 1.2: Create a Form

Lesson 1.3: Make the New Form Visible in the UI

June 28, 2022 -10 AM -11 AM

Day 2

Lesson 1.4: Configure the Data Access Class

Lesson 1.5: Configure the Form

Lesson 1.6: Add an Event Handler to the Walk-In Service Check Box

Lesson 1.7: Debug the Customization Code

Timing and Agenda

June 29, 2022 -10 AM -11 AM

Day 3

Lesson 1.8: Move the Customization Code to an Extension Library

Lesson 1.9: Add an Event Handler In Visual Studio

Lesson 2.1: Create a Graph and a DAC in Visual Studio

June 30, 2022 -10 AM -11 AM

Day 4

Lesson 2.2: Create an ASPX Page in Visual Studio

Lesson 2.3: Configure a Form in Visual Studio

Lesson 2.4: Add the Form to the Site Map and Workspace

Lesson 2.5: Create a Substitute Form

Day 1

Cherrylake, Inc.
Acumatica Customer Since 2013

Acumatica
The Cloud ERP

Part 1: Repair Services Maintenance Form

Lesson 1.1: Prepare a Customization Project

Learning Objectives

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

- What customization project is
- How to create a customization project
- How to add a database script for a database table to the customization project

Lesson Summary

In this lesson, you have learned how to create customization projects and add database scripts to the project.

As you have completed the lesson, you have created the *PhoneRepairShop* customization project on the Customization Projects (SM204505) form. You have used the Customization Project Editor to add database scripts for custom tables to the customization project.

Lesson 1.2: Create a Form

Learning Objectives

In this lesson, you will learn how to create a form of the application by generating the needed items with the New Screen wizard.

Lesson 1.2: Create a Form

Column Name	Data Type	Description
Service ID	String	The identifier of the service
Description	String	The description of the service
Active	Boolean	An indicator of whether a service is currently provided by the shop
Walk-In Service	Boolean	An indicator of whether this is a walk-in service
Requires Preliminary Check	Boolean	An indicator of whether the service requires diagnostic checks
Requires Prepayment	Boolean	An indicator of whether this service should be prepaid

Figure: The Customized Screen page of the Customization Project Editor

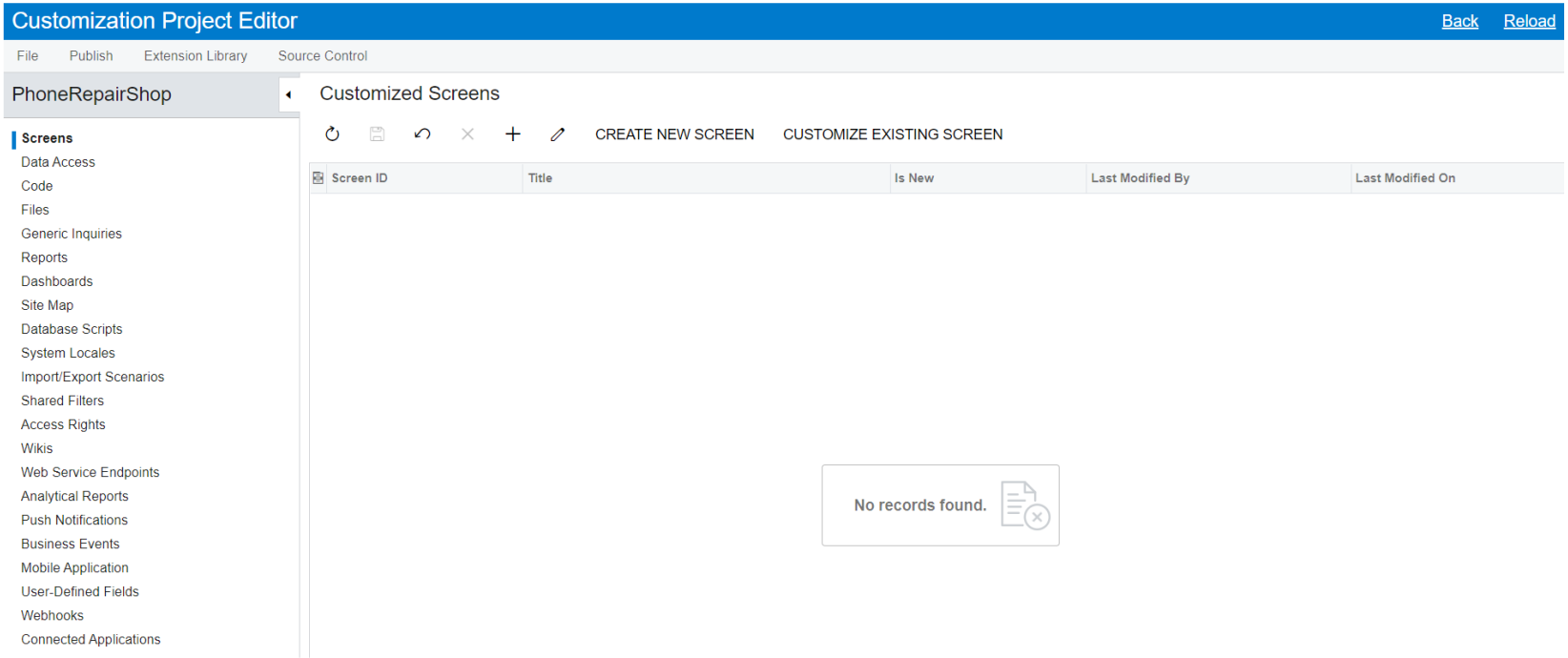
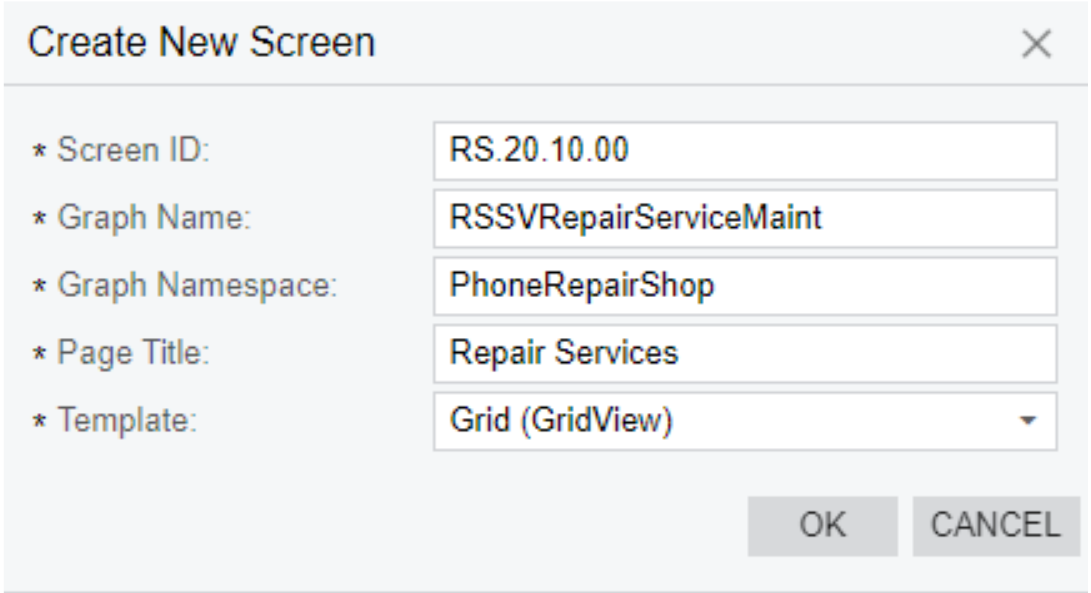


Figure: The Create New Screen dialog box



Lesson 1.2: Create a Form

Item	Description
RS201000	This *Screen* item contains the new page content.
RSSVRepairService-Maint	This *Code* item contains the code template of the graph for the new form. This item is saved in the database. When you publish the project, the platform creates a copy of the code in the `RSSVRepairServiceMaint.cs` file in the `App_RuntimeCode` folder of the Acumatica ERP application instance.
Pages\RS\RS201000.aspx **Pages\RS\RS201000.aspx.cs**	These *File* items contain ASPX page code for the new form. When you publish the project for the first time, the platform creates the files in the `Pages\RS` folder of the Acumatica ERP application instance, and the platform creates copies of these files in the `pages_RS` subfolder of the `CstPublished` folder of the instance.
Repair Services	This *SiteMapNode* item contains the site map object of the new form.

Lesson Summary

In this lesson, you've learned how to create a new form by using the New Screen wizard. From the example of the Repair Services form you created, you have learned about the following basic components of a form:

- The ASPX page
- The graph
- The site map node

Lesson 1.3: Make the New Form Visible in the UI

Learning Objectives

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

- Create a workspace
- Add a link to a custom form to the workspace
- Update the *SiteMapNode* item in the customization project

Figure: The Phone Repair Shop workspace in Menu Editing mode

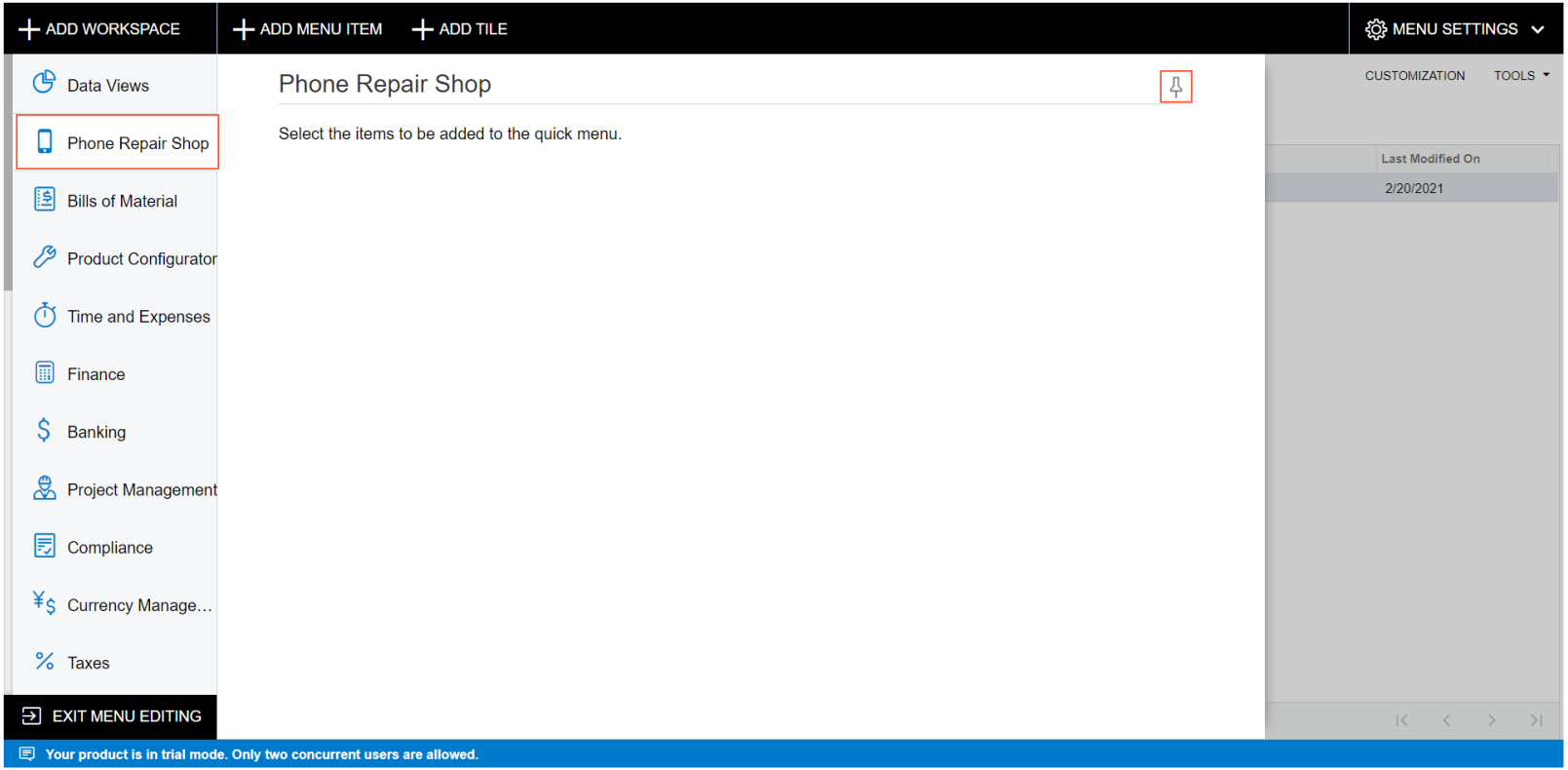


Figure: The Select Forms dialog box

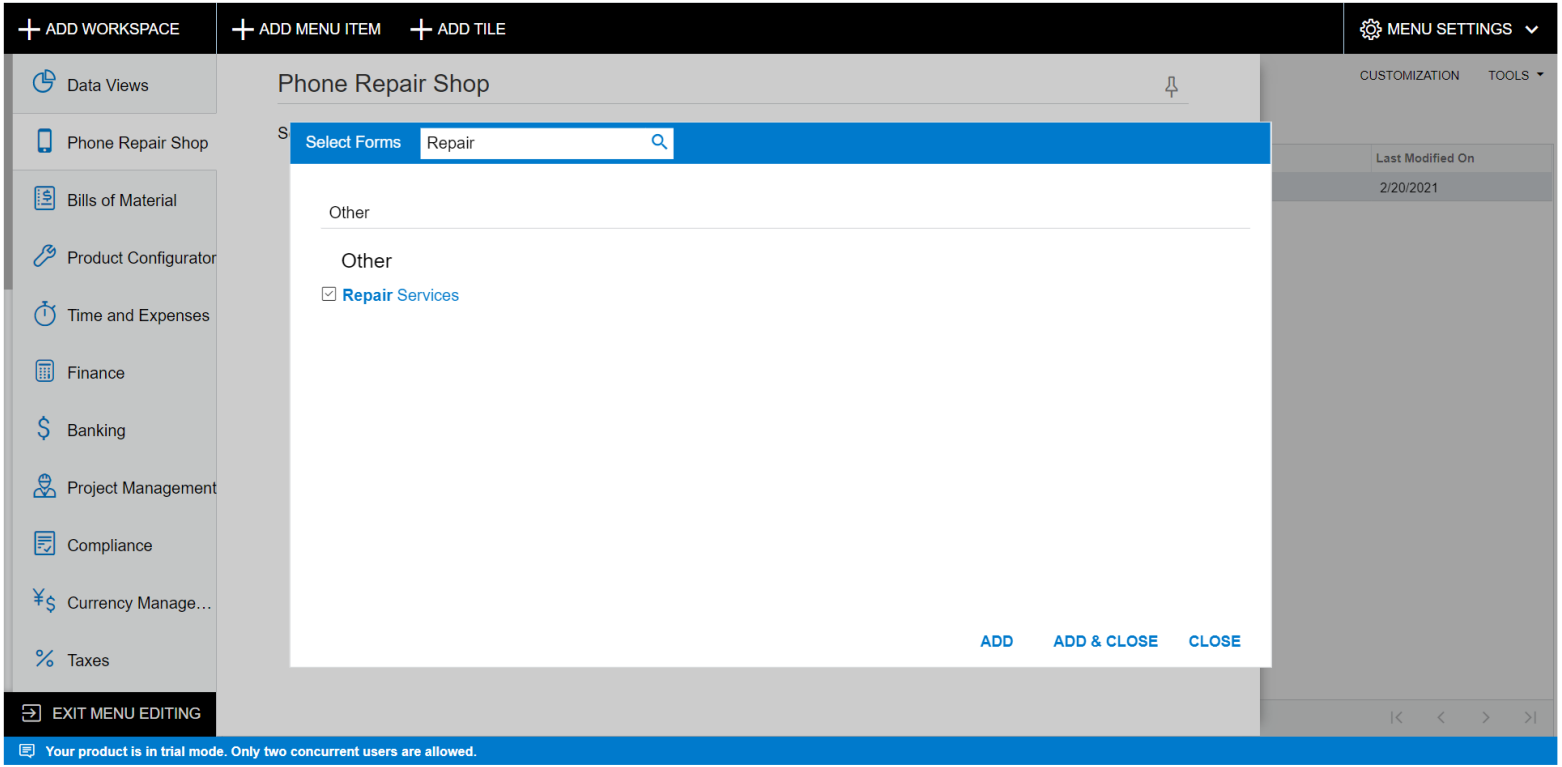


Figure: The Edit button

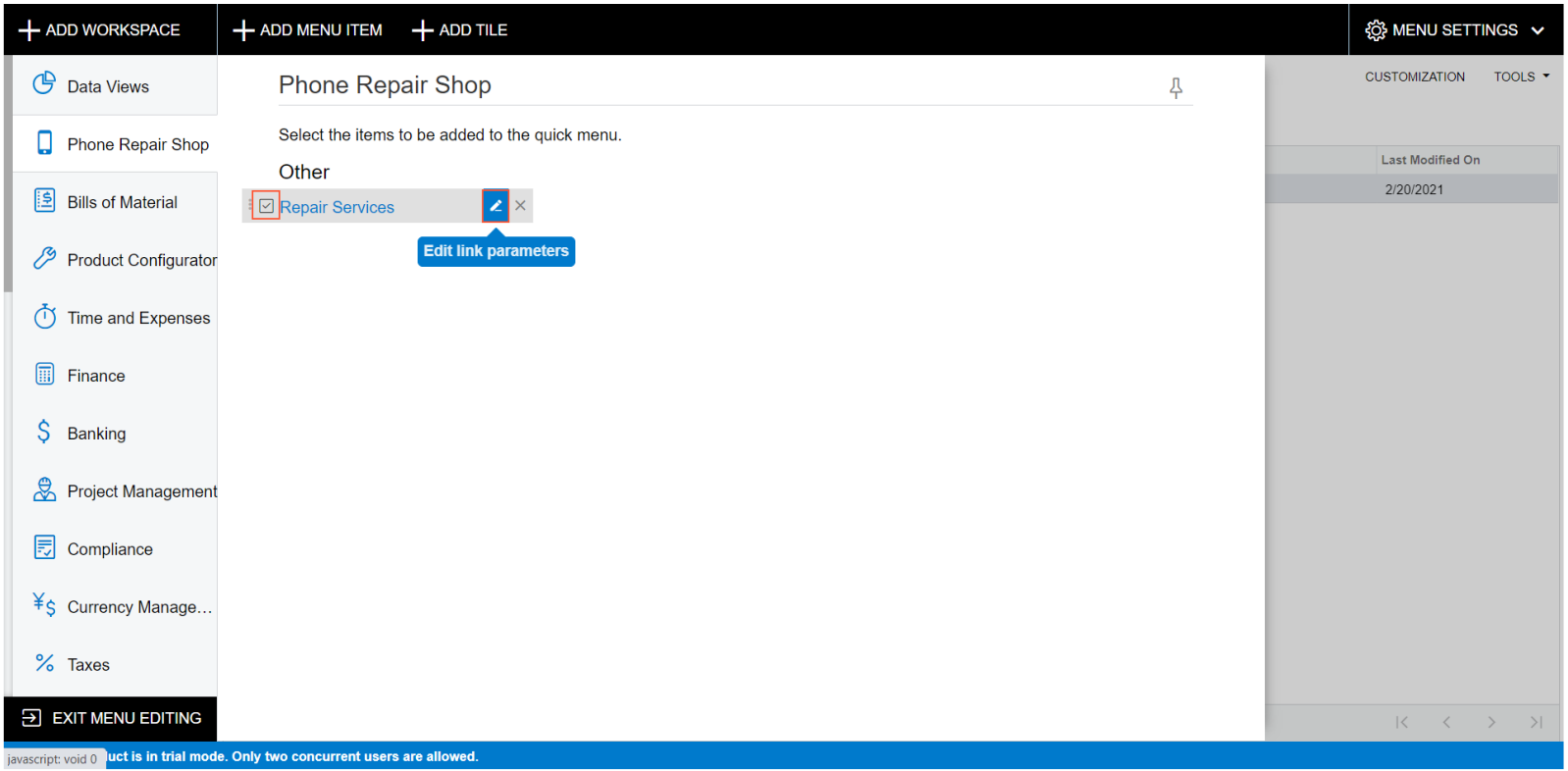


Figure: The Phone Repair Shop workspace

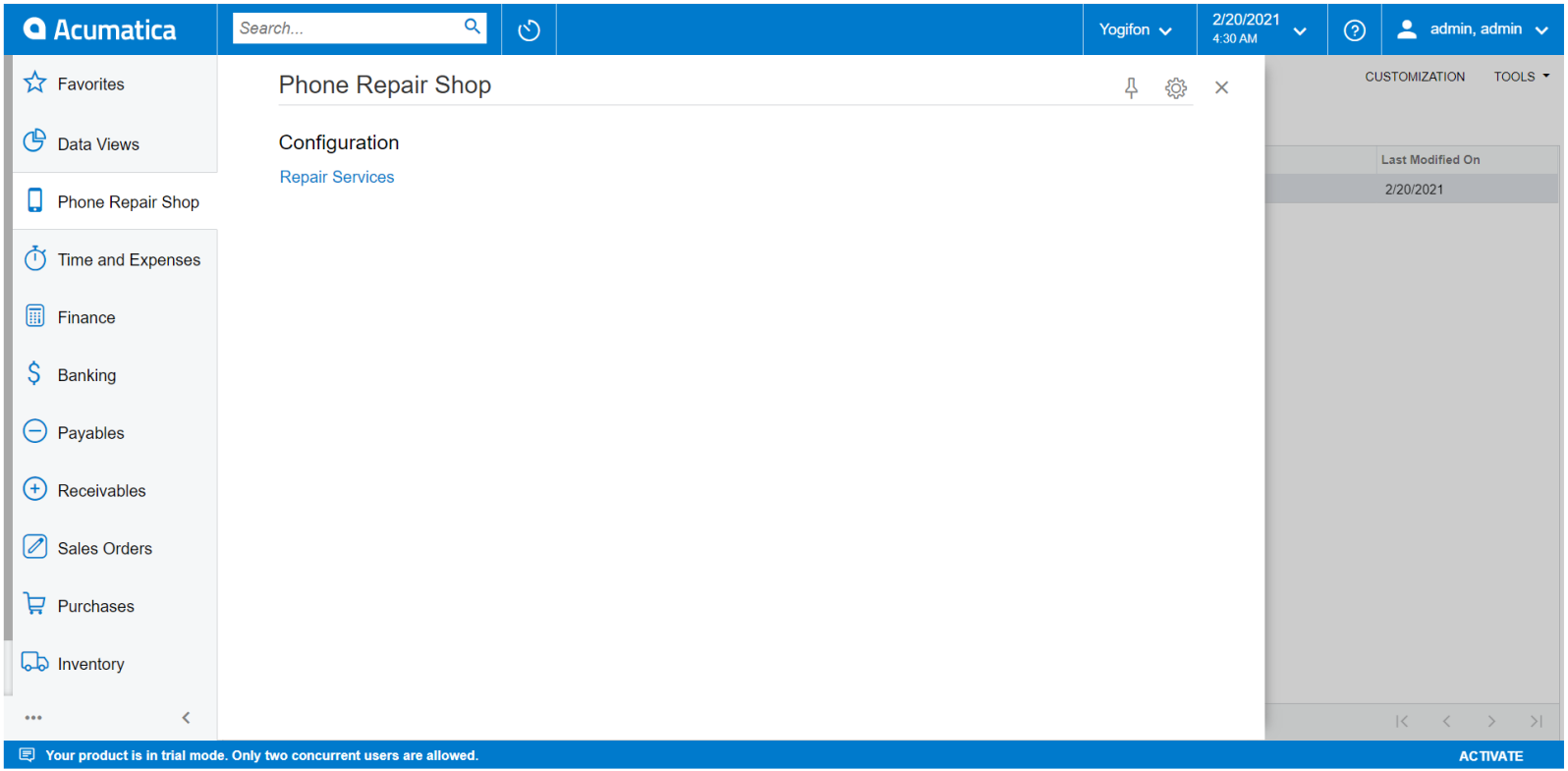
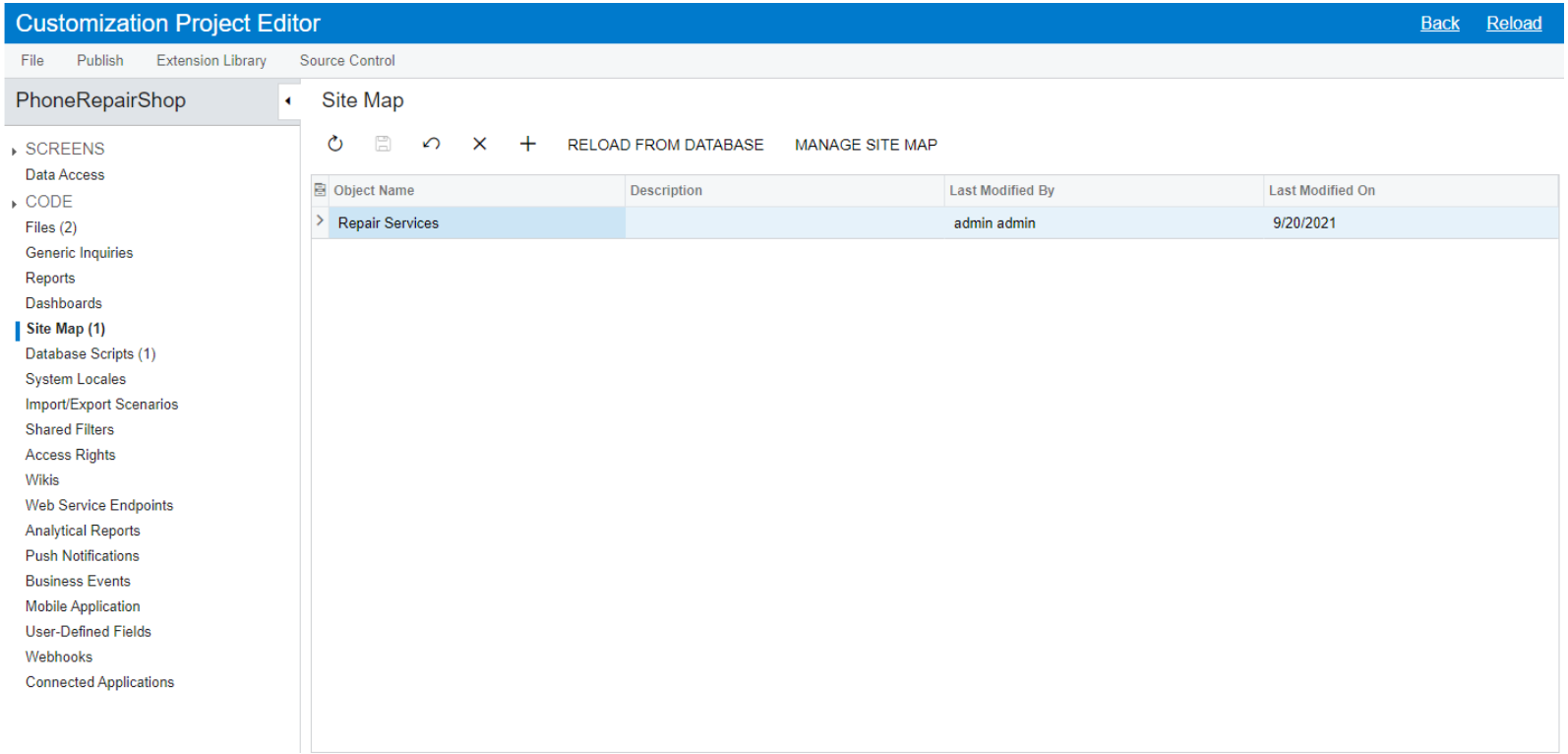


Figure: The Site Map page



Lesson Summary

In this lesson, you have learned how to add a new workspace to the Acumatica ERP UI, add links to a workspace, and update the *SiteMapNode* item of the customization project.

Day 2

Devil's Peak Brewing Company
Acumatica Customer Since 2016

Lesson 1.4: Configure the Data Access Class

Learning Objectives

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

- Generate and configure a DAC
- Configure a view in a graph

The connection between DAC fields and database fields

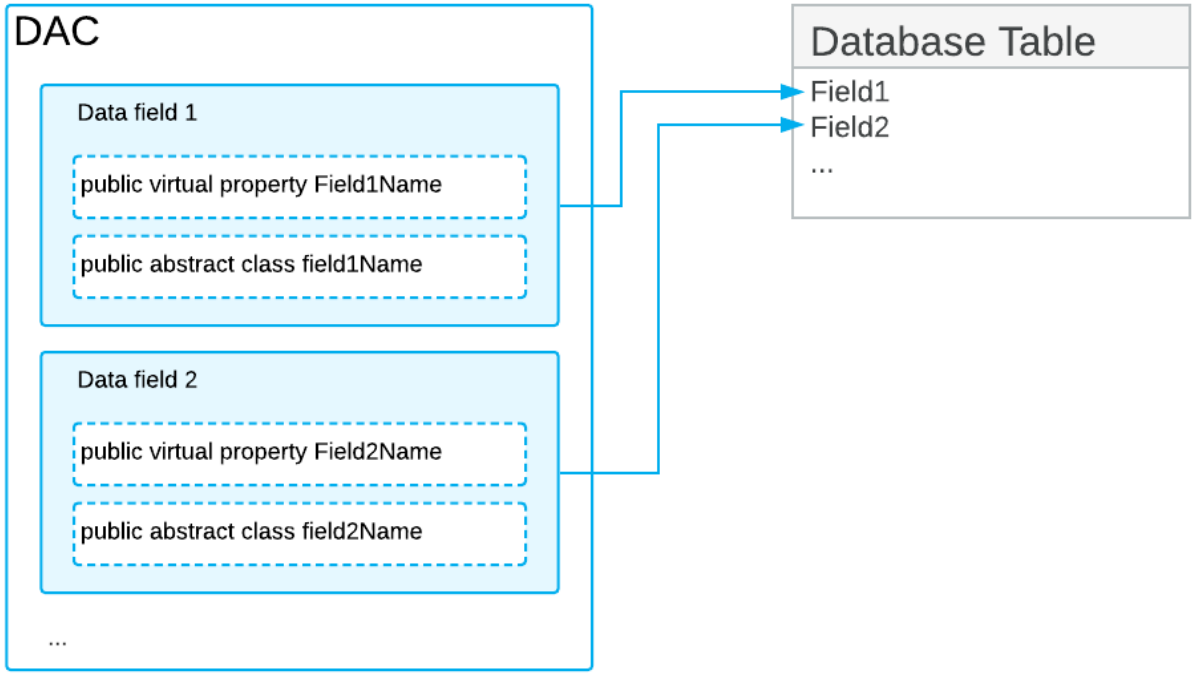


Figure: The Create Code File dialog box

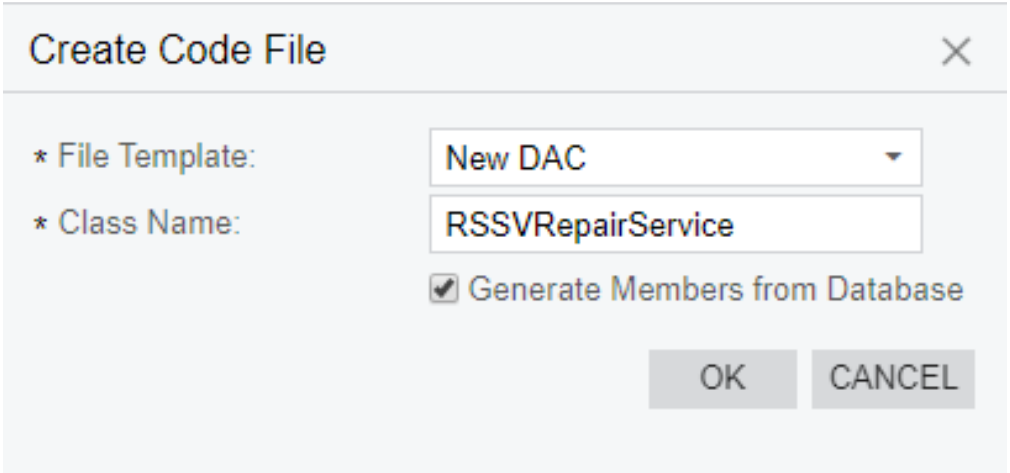
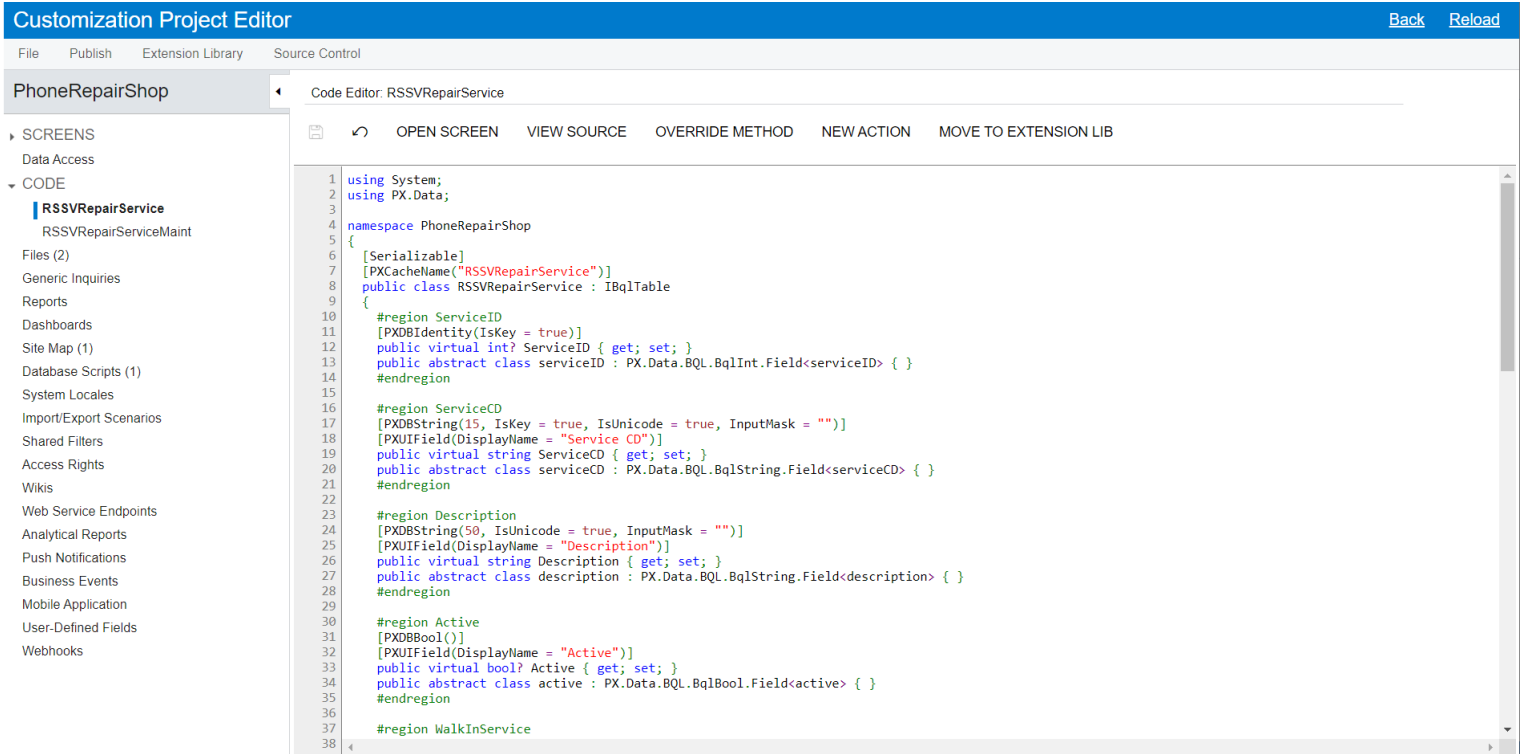


Figure: The RSSVRepairService DAC code



Field	Attribute
CreatedDateTime	[PXDBCreatedDateTime()]
CreatedByID	[PXDBCreatedByID()]
CreatedByScreenID	[PXDBCreatedByScreenID()]
LastModifiedDateTime	[PXDBLastModifiedDateTime()]
LastModifiedByID	[PXDBLastModifiedByID()]
LastModifiedByScreenID	[PXDBLastModifiedByScreenID()]
Tstamp	[PXDBTimestamp()]
Noteid	[PXNote()]

Figure: Specifying the PrimaryView property in the Screen Editor

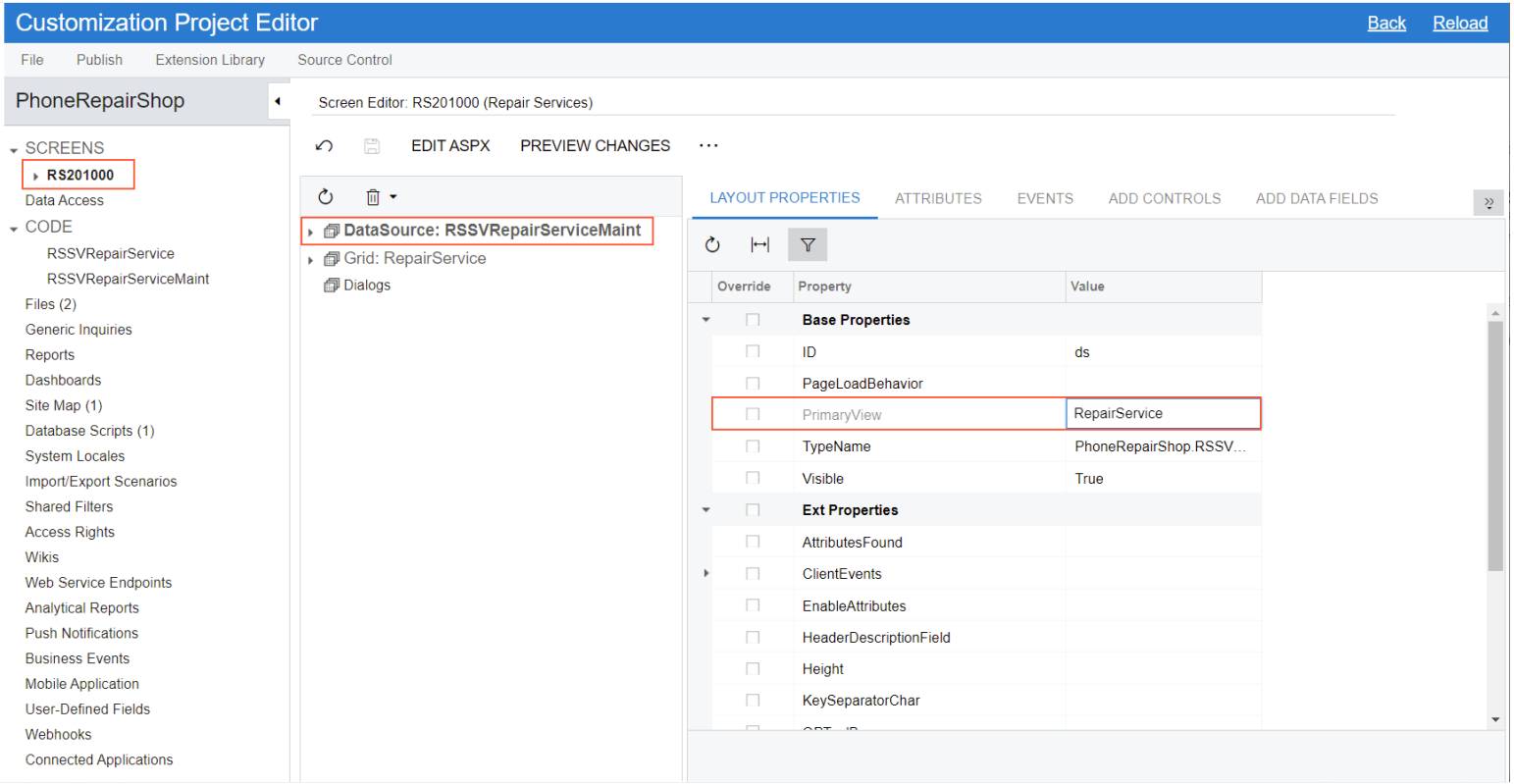
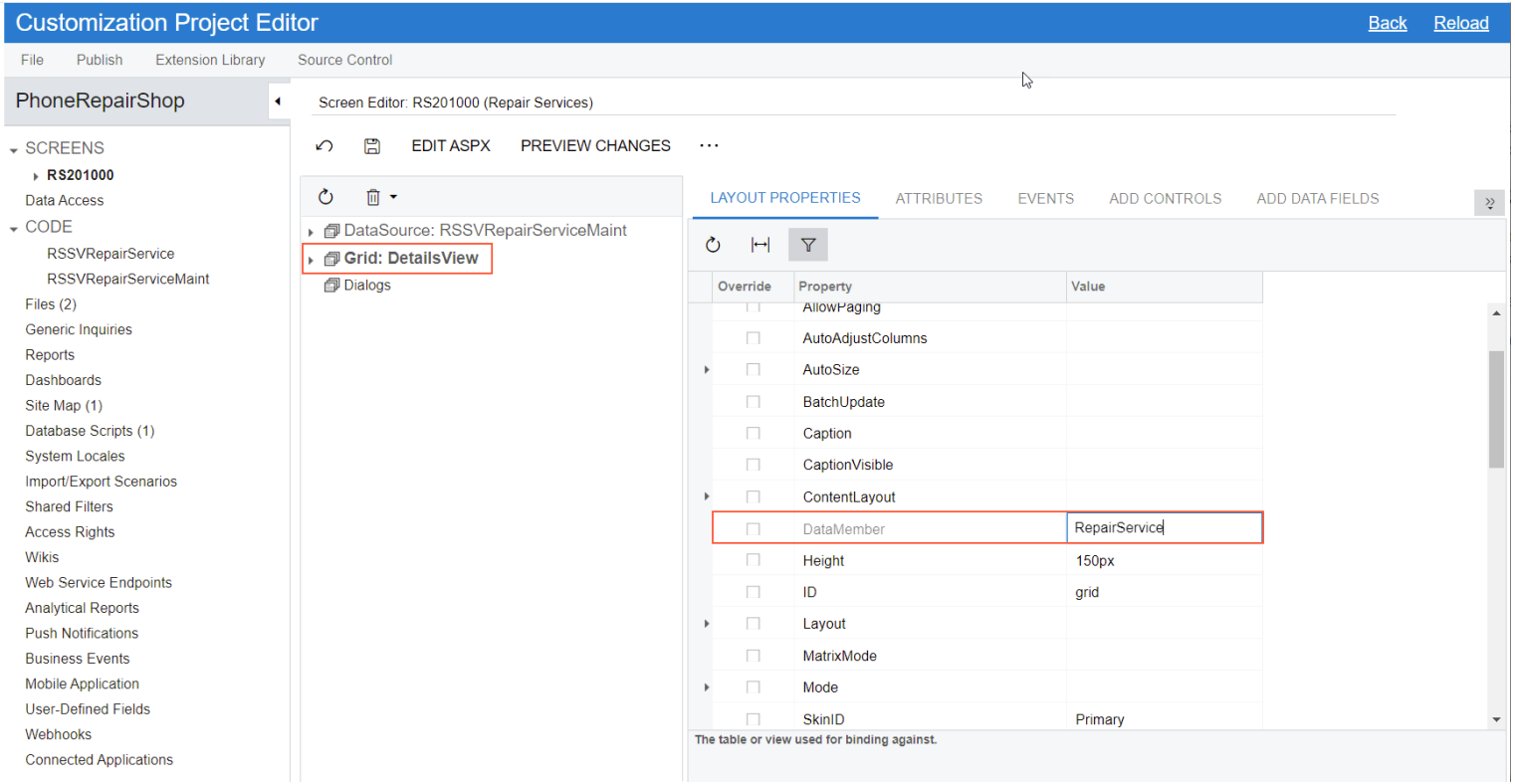


Figure: Specifying the DataMember property in the Screen Editor



Lesson Summary

In this lesson, you have learned the concept of a data access class (DAC), and have generated and configured a DAC.

You have learned about the attributes that are required for DAC fields and have configured a view by using a fluent BQL query.

Lesson 1.5: Configure the Form

Learning Objectives

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

- Add columns to a form grid
- Configure the appearance of the columns in the grid

Figure: Columns to be added

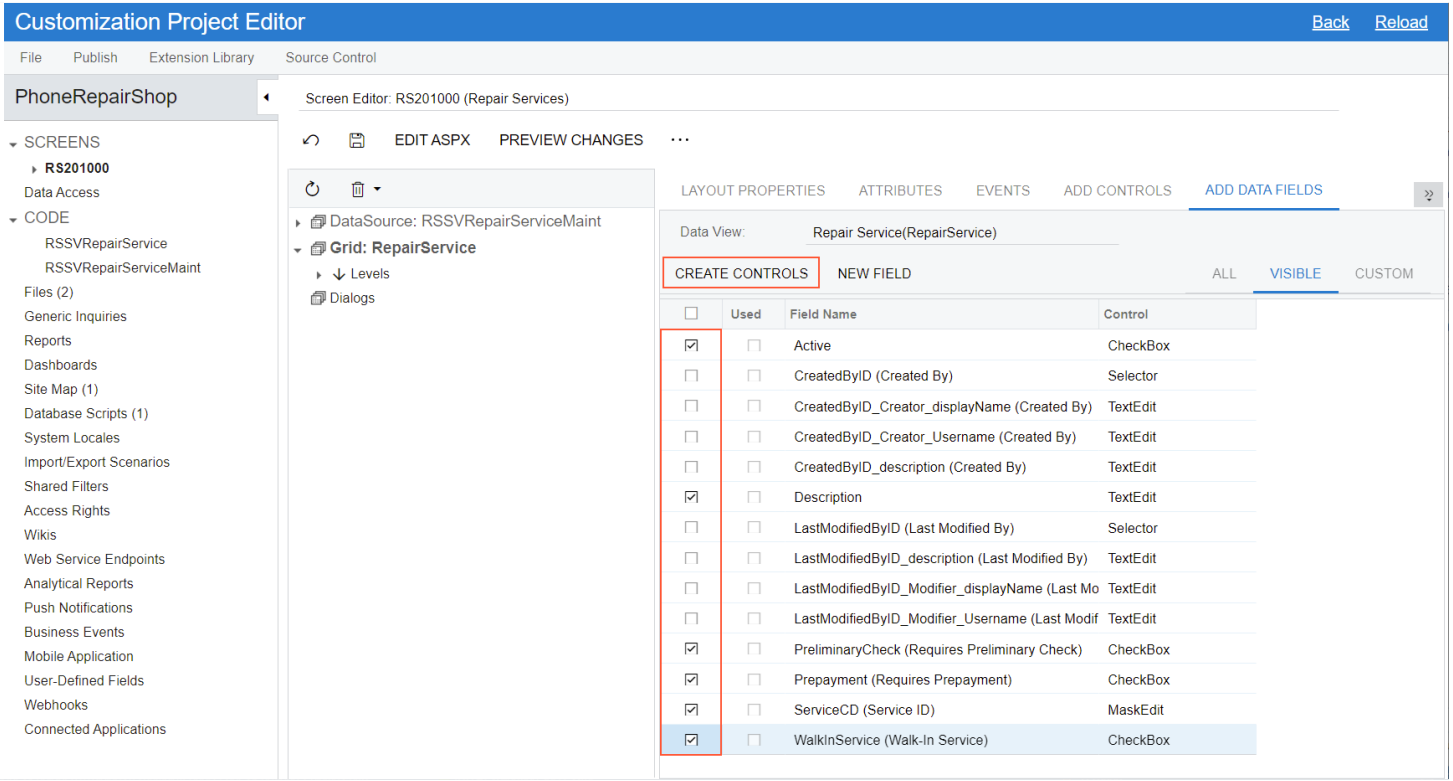


Figure: Control tree with new columns

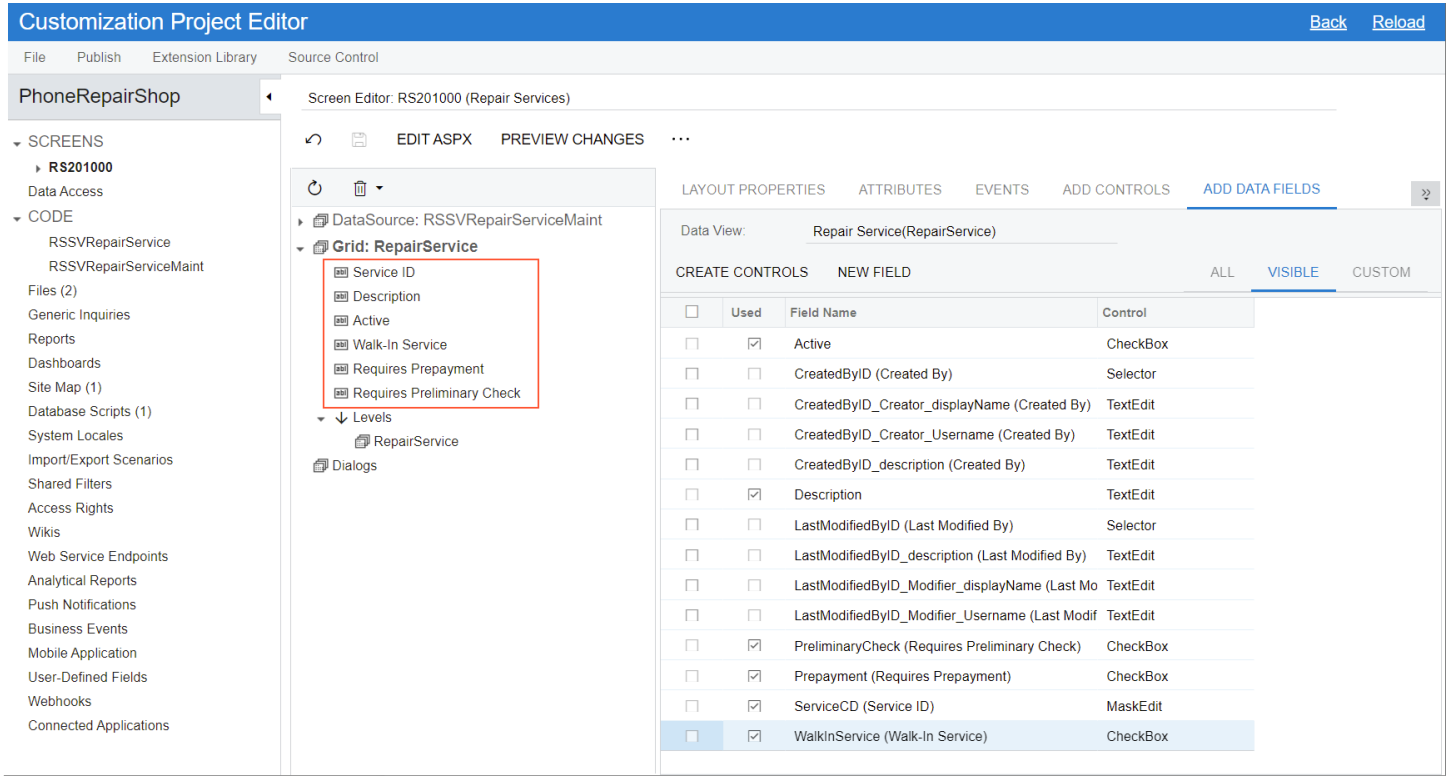


Figure: Property to define the Active column as a check box

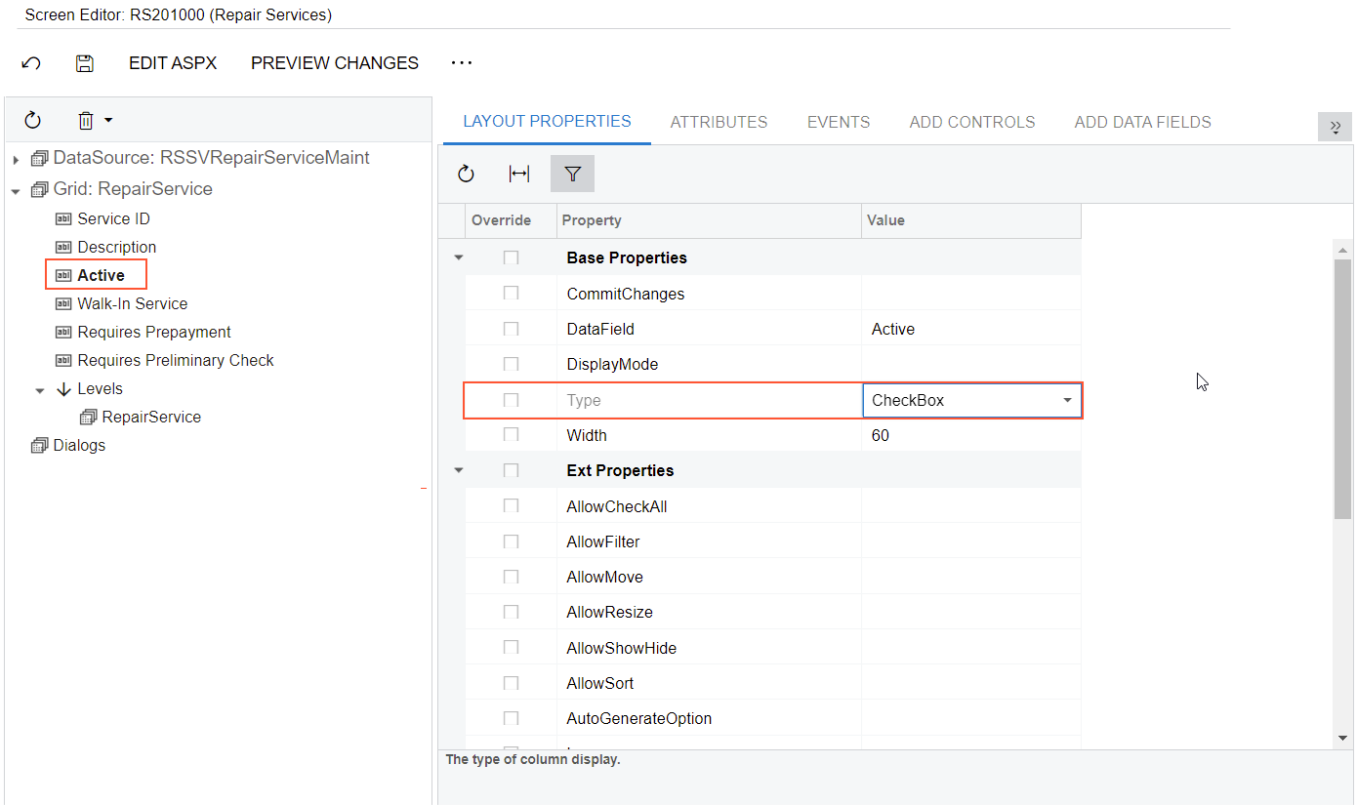


Figure: The Repair Services form in the preview mode

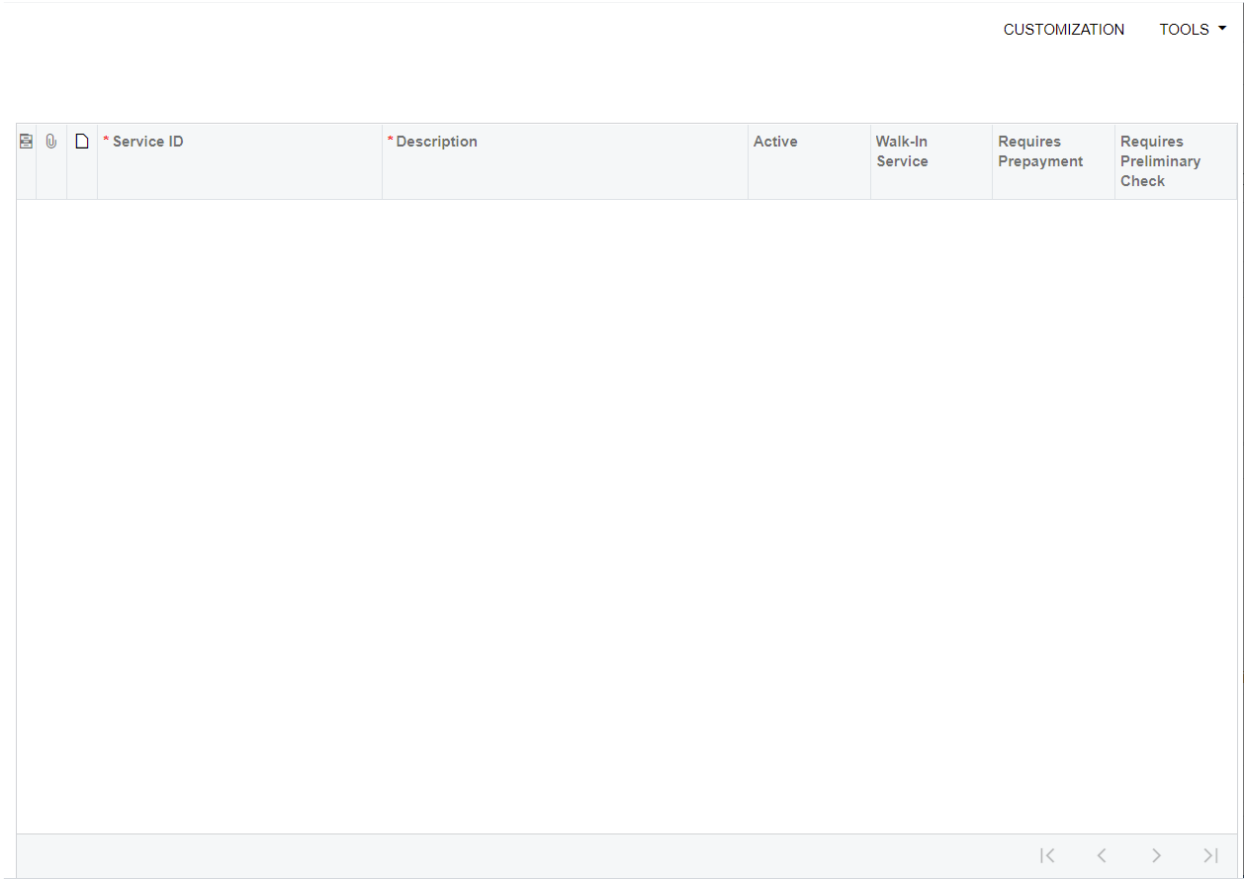


Figure: The Repair Services form

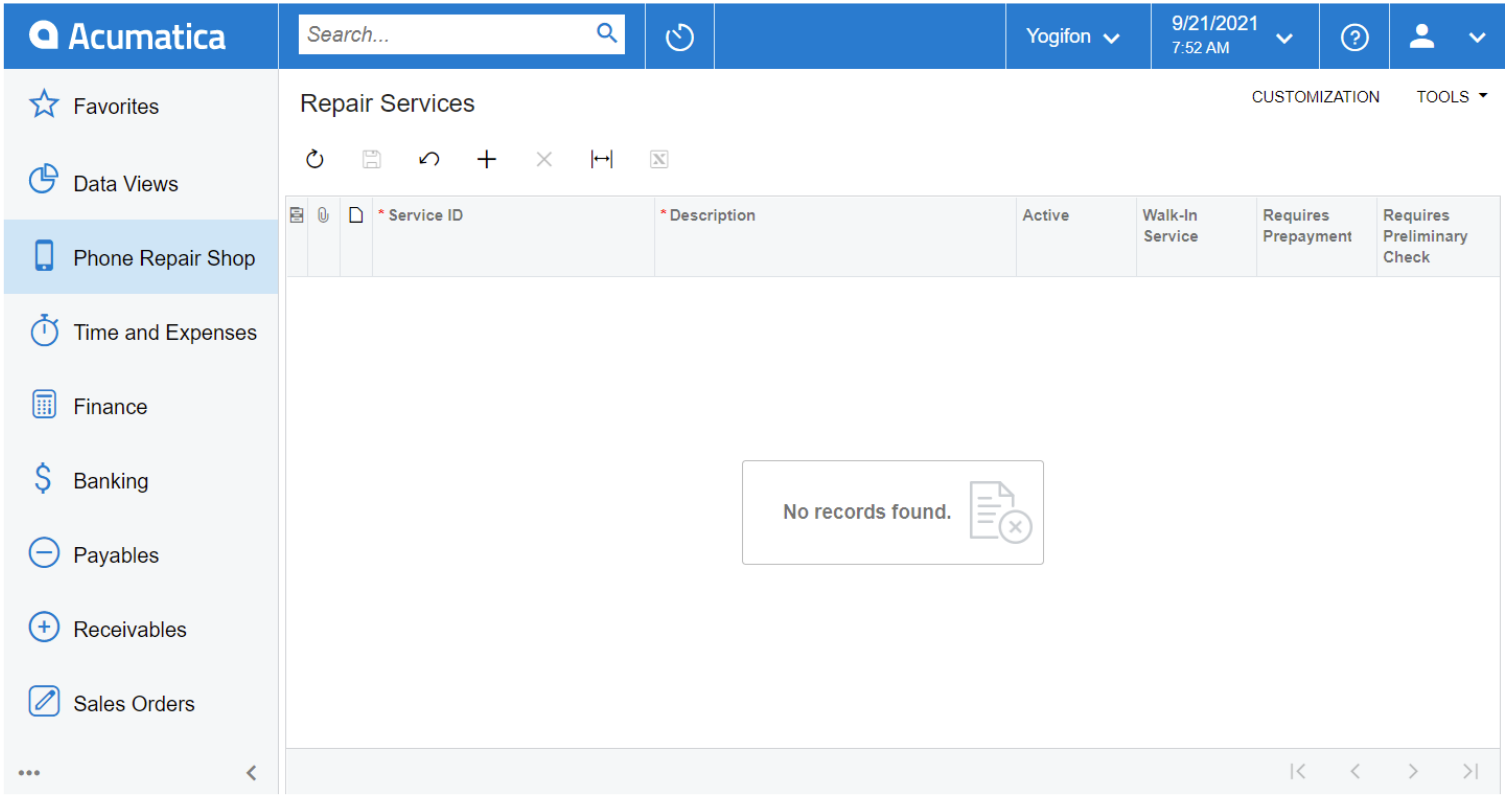
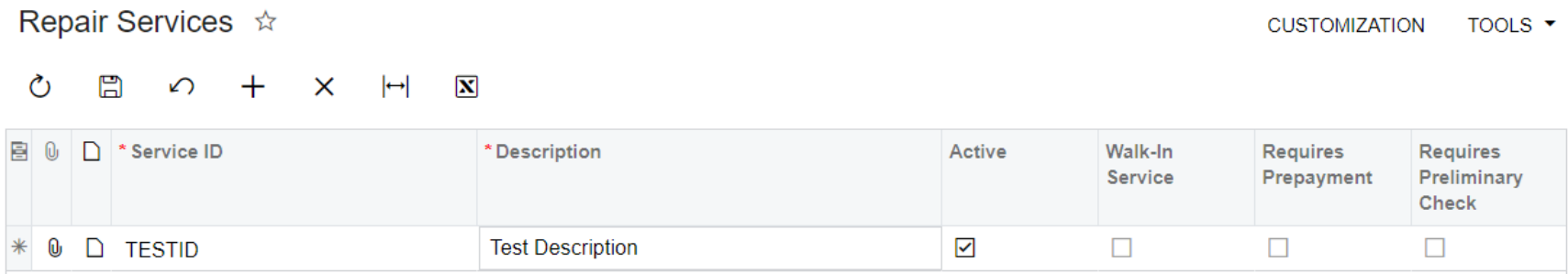


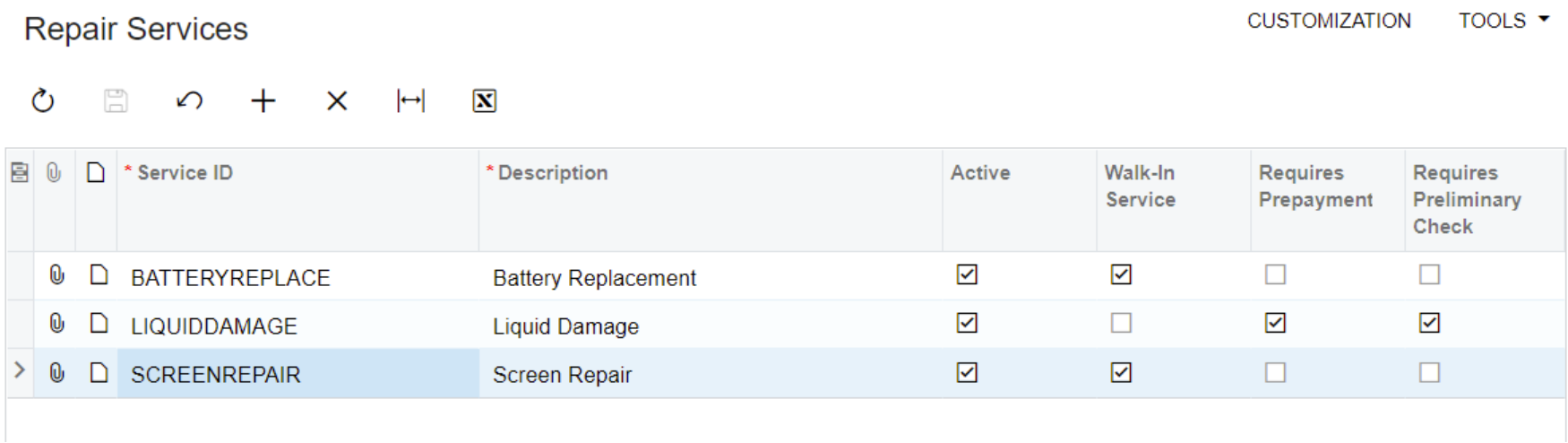
Figure: The new row on the Repair Services form



Add the following rows to the table

Service ID	Description	Active	Walk-In Service	Requires Prepayment	Requires Preliminary Check
BatteryReplace	Battery Replacement	Selected	Selected	Cleared	Cleared
LiquidDamage	Liquid Damage	Selected	Cleared	Selected	Selected
ScreenRepair	Screen Repair	Selected	Selected	Cleared	Cleared

Figure: New rows on the Repair Services form



Lesson Summary

In this lesson, you have learned how to configure the elements on a form by using the Screen Editor; you have also tested the created form.

Lesson 1.6: Add an Event Handler to the Walk-In Service Check Box

Learning Objectives

In this lesson, you will learn how to add an event handler for a check box.

Figure: The FieldUpdated event

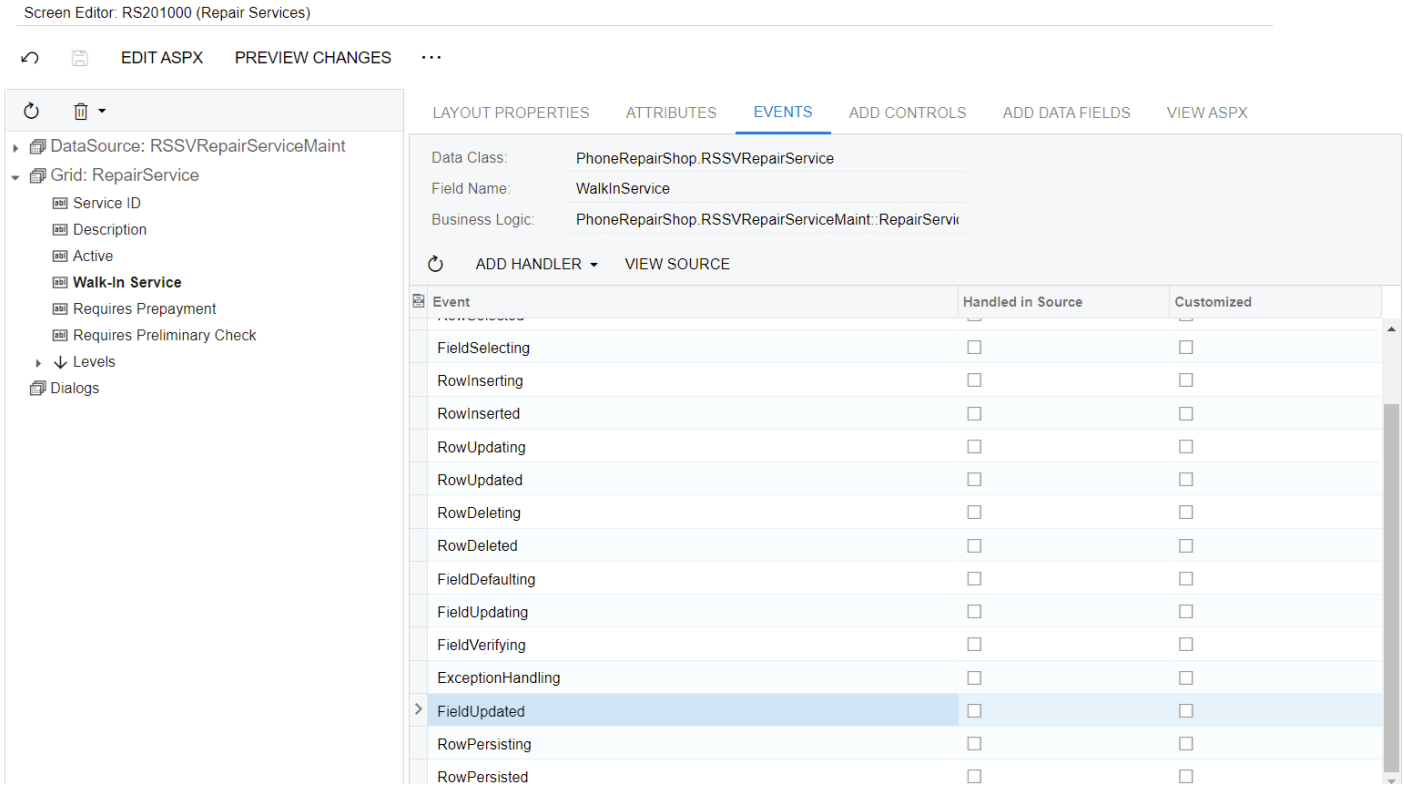
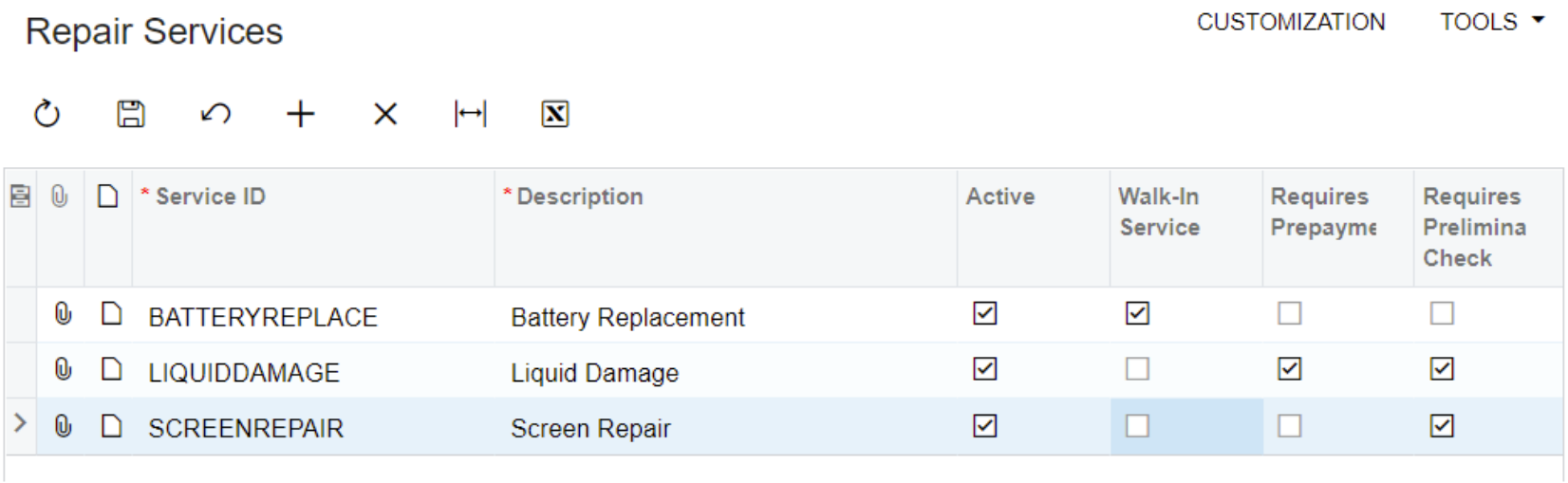


Figure: Clearing the Walk-In Service check box



Lesson Summary

In this lesson, you learned how to add and enable an event handler by using the Customization Project Editor. You have added code to the generated event handler and tested it on the Repair Services (RS201000) form to be sure it works as intended.

To add and enable an event handler, you have done the following:

1. In the Screen Editor, configured the CommitChanges property of the control for which an event handler should work.

2. Generated code for the event handler by using the Screen Editor. You have added the FieldUpdated event, which is raised for each field of a record that is currently updated or inserted.

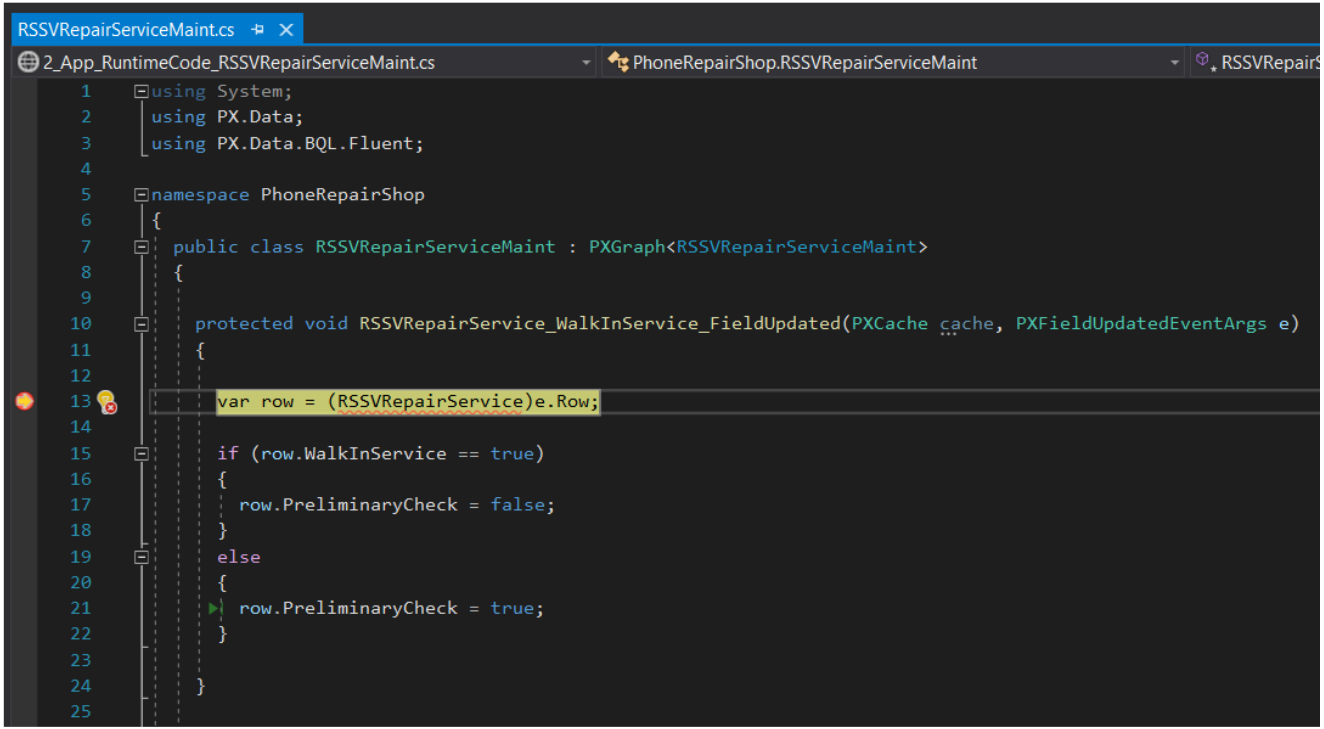
3. Added code to the event handler by using the Code Editor.

Lesson 1.7: Debug the Customization Code

Learning Objectives

In this lesson, you will learn how to debug the code of a customization project by using Visual Studio.

Figure: Breakpoint hit



Lesson Summary

In this lesson, you have learned how to debug customization code by using Visual Studio.

To debug customization code, you have completed the following steps:

1. Configured the web.config file of the Acumatica ERP instance.
2. In Visual Studio, added a breakpoint.
3. Attached the Visual Studio debugger to the w3wp.exe process.
4. Performed the debugging of the instance in Visual Studio.

Day 3

Ray Allen Manufacturing
Acumatica Customer Since 2014

Acumatica
The Cloud ERP

Lesson 1.8: Move the Customization Code to an Extension Library

Learning Objectives

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

- Create an extension library
- Open the created extension library in Visual Studio
- Build the created extension library in Visual Studio

Figure: Creation of the PhoneRepairShop_Code extension library

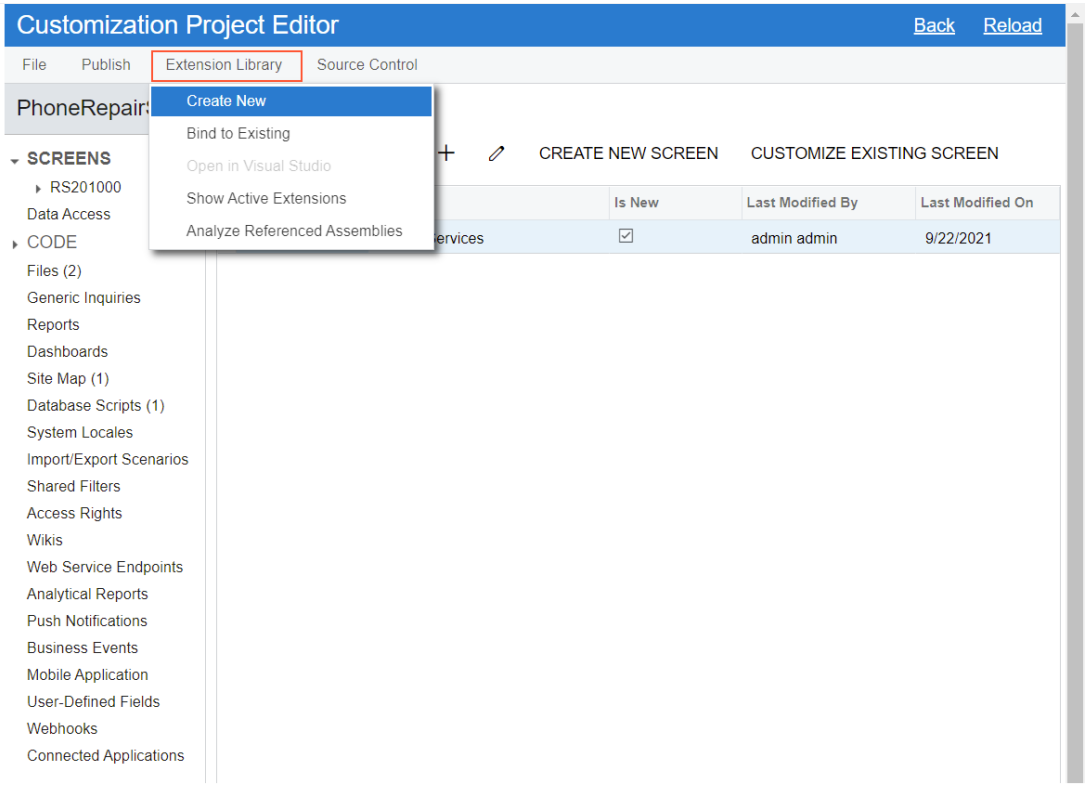


Figure: Button to move the Code item to the extension library

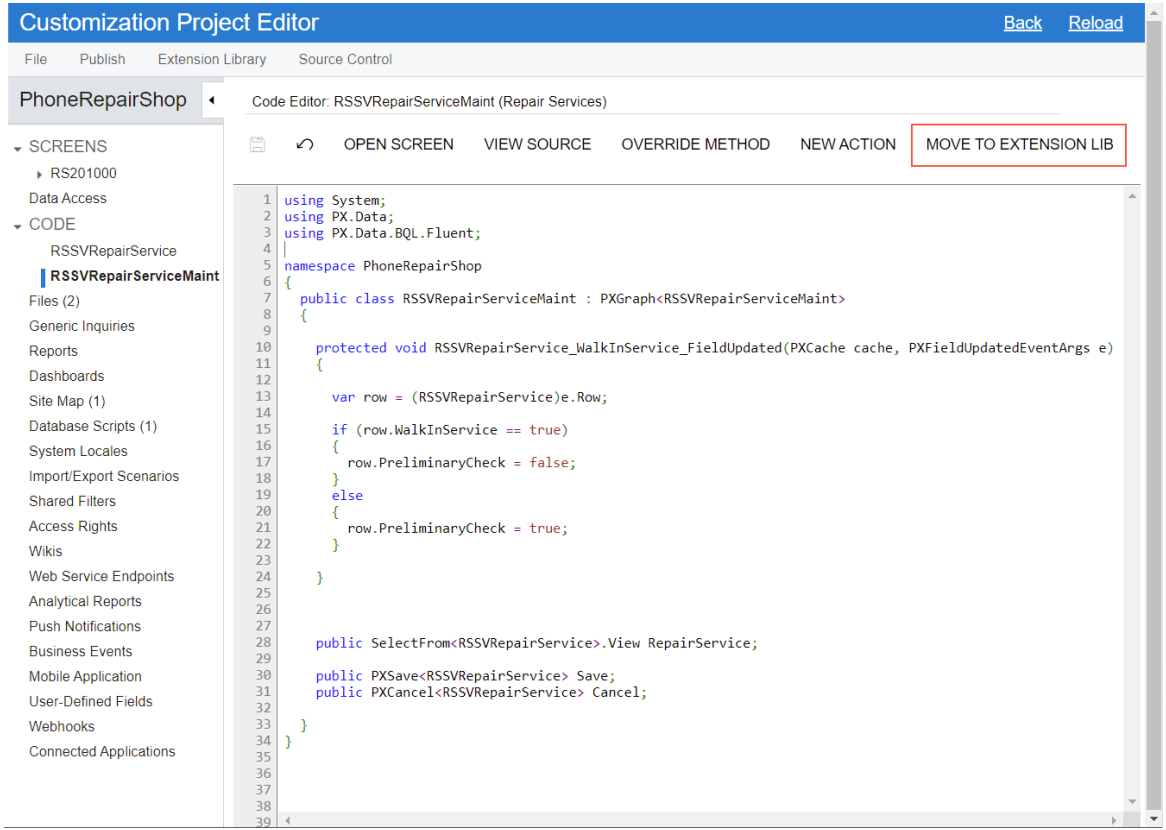


Figure: The empty list of the Code page

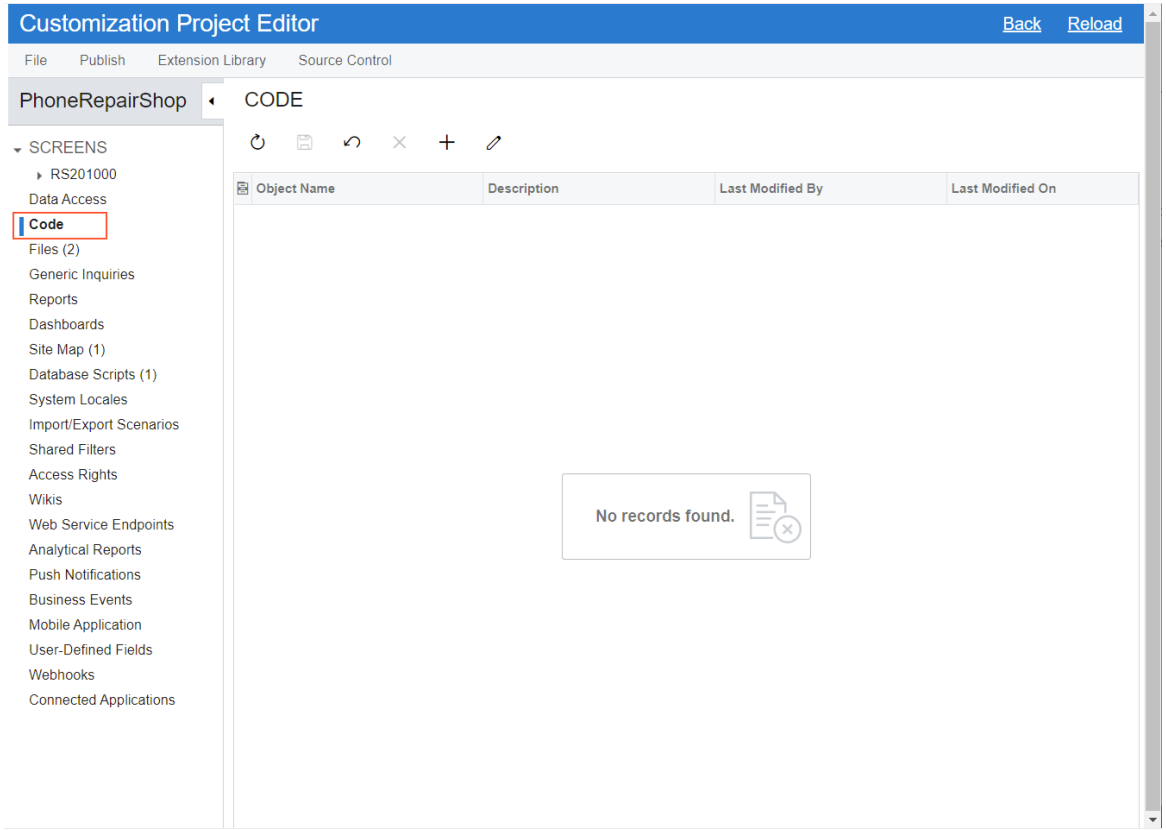
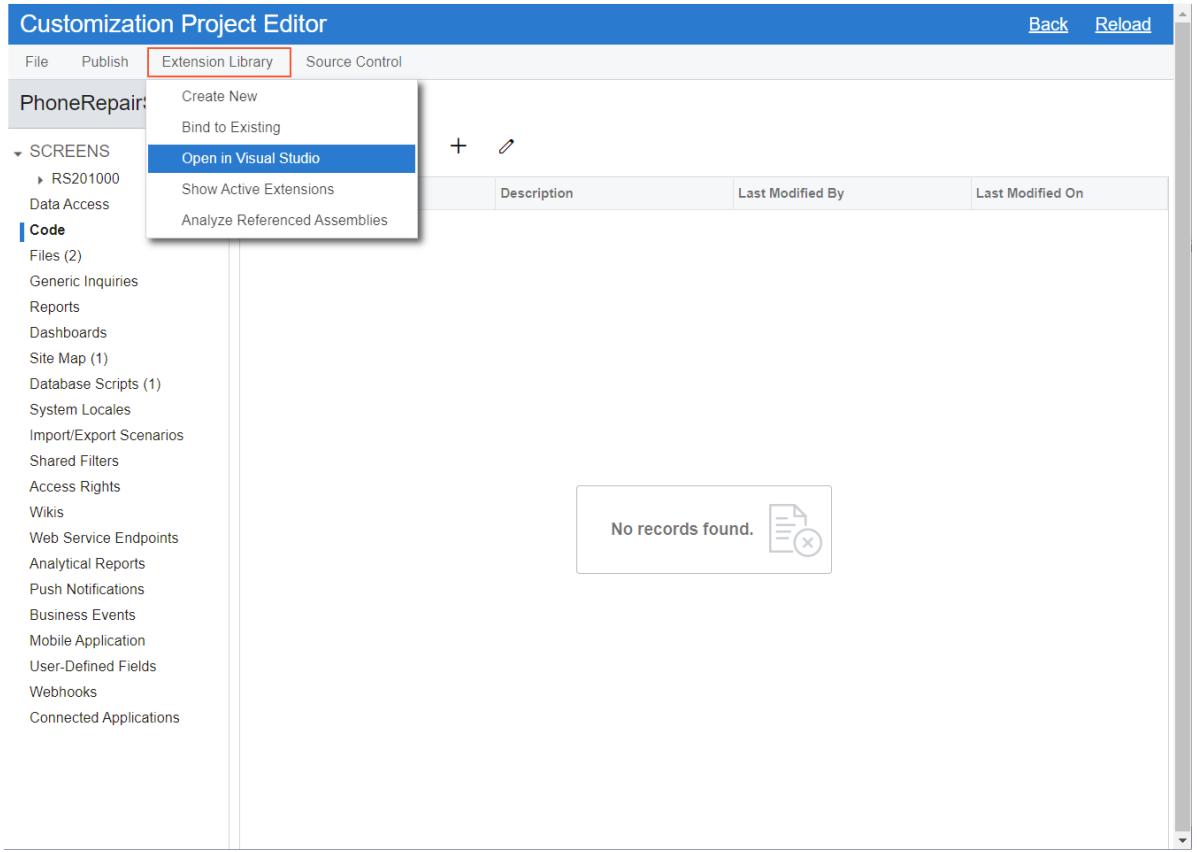


Figure: Opening of the solution from the Customization Project Editor



Lesson Summary

In this lesson, you have learned how to create an extension library, move customization code to the extension library, and open the code items of a customization project in Visual Studio in order to start developing customization code in Visual Studio.

You have completed the following steps:

1. Created an extension library

2. Moved all code items from the customization project to the Visual Studio project

3. Opened the created project in Visual Studio

4. Built the project in Visual Studio

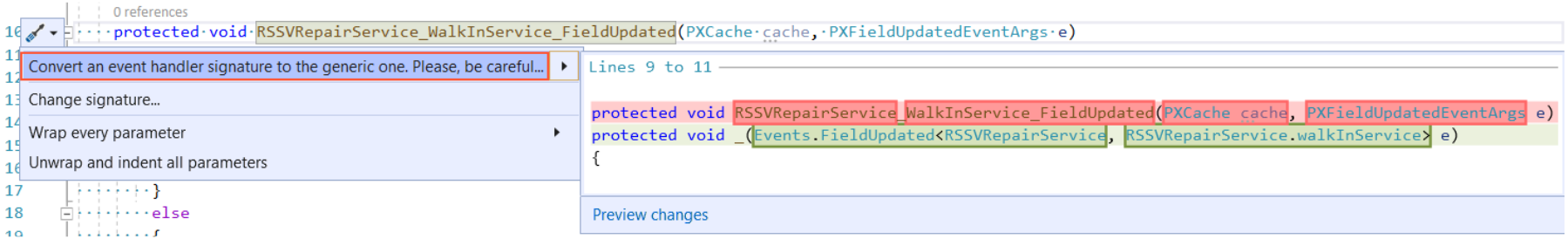
You have also included the extension library in the customization project as a File item.

Lesson 1.9: Add an Event Handler In Visual Studio

Learning Objectives

In this lesson, you will learn how to add an event handler in Visual Studio and use Acuminator to refactor existing code.

Figure: Refactoring of the event handler



Lesson Summary

In this lesson, you have added an event handler in Visual Studio and learned how to use Acuminator to refactor the existing code.

Part 2: Serviced Devices Maintenance Form

Lesson 2.1: Create a Graph and a DAC in Visual Studio

Learning Objectives

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

- Define a graph and DAC in Visual Studio
- Configure a selector for a field and a drop-down menu for a field

Lesson Summary

In this lesson, you have learned how to create a graph and a DAC in Visual Studio. Also, you learned how to configure a lookup control by using the PXSelector attribute and a drop-down list by using the PXStringList attribute.

Day 4

Curran Young Construction Company
Acumatica Customer Since 2018

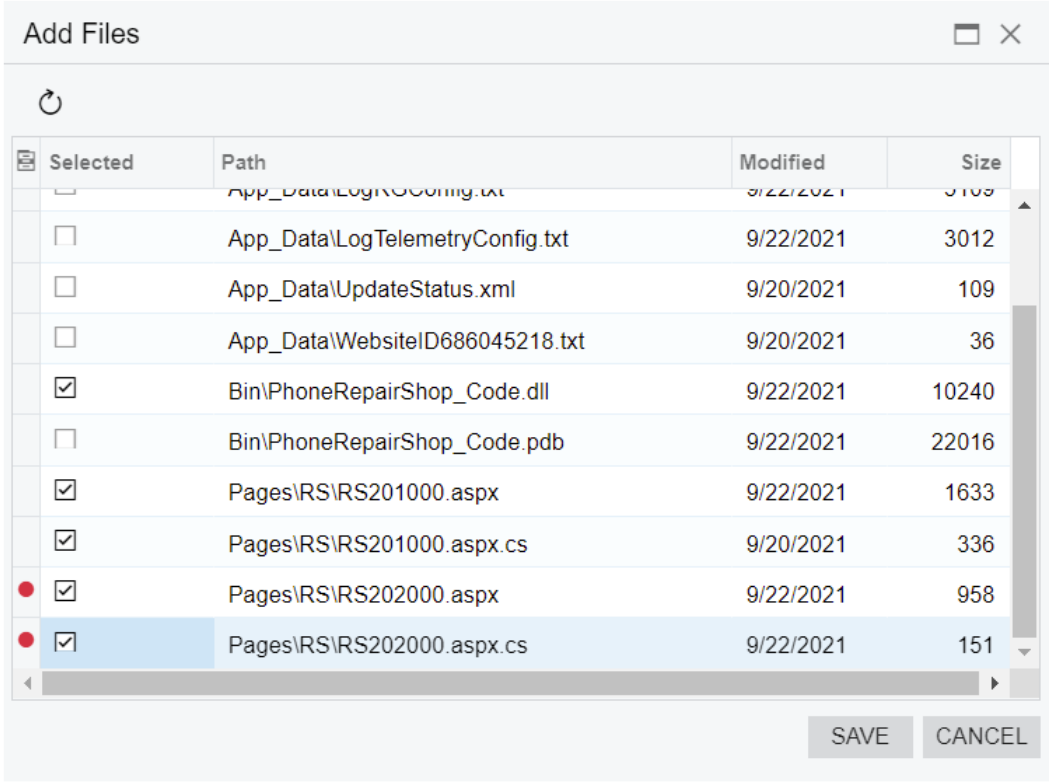
Lesson 2.2: Create an ASPX Page in Visual Studio

Learning Objectives

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

- Create an ASPX file
- Configure a data member for an Acumatica ERP form
- Add the ASPX and ASPX.CS files to the customization project

Figure: The Add Files dialog box



Lesson Summary

In this lesson, you have learned how to create an ASPX page in Visual Studio and configure a data member for it.

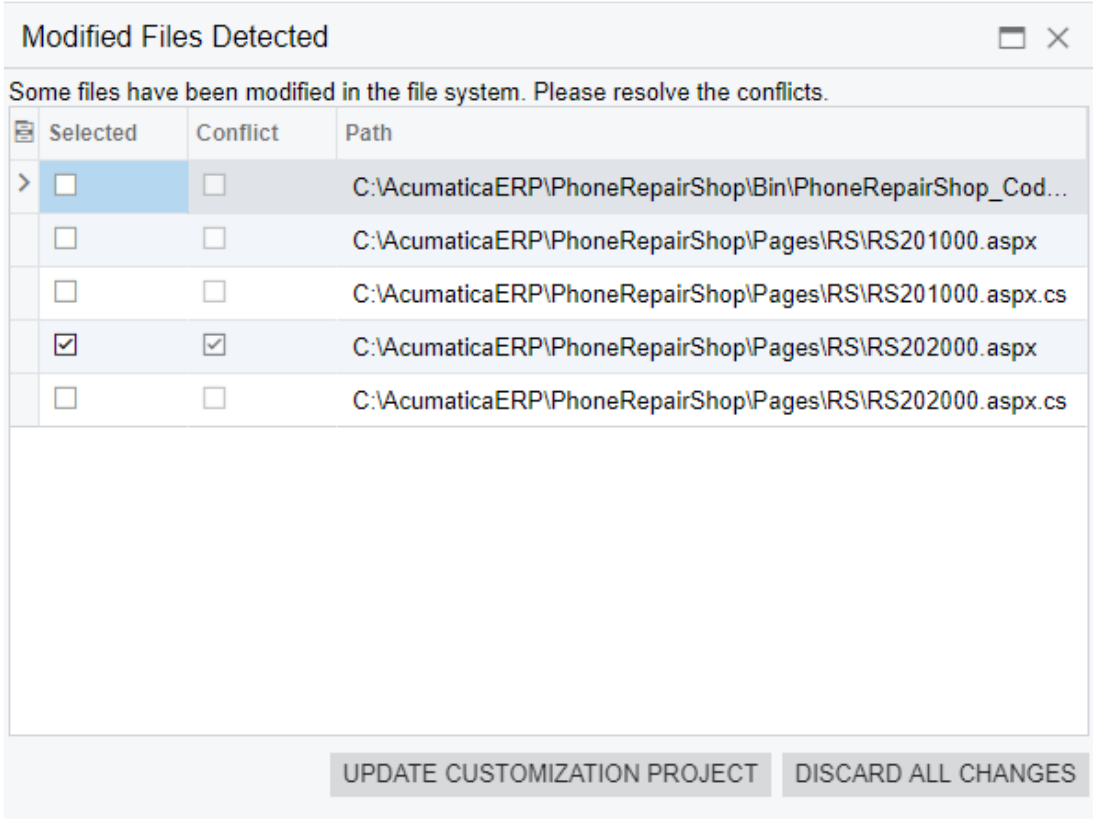
Also, you have added created files to the customization project.

Lesson 2.3: Configure a Form in Visual Studio

Learning Objectives

In this lesson, you will learn how to use Visual Studio to add controls and configure layout of a form.

Figure: The Modified Files Detected dialog box



Lesson Summary

In this lesson, you have learned how to configure input controls on a form in Visual Studio and organize the layout of the form.

To configure controls on the Serviced Devices (RS202000) form, you have used PXSelector, PXTextEdit, PXCheckBox, and PXDropDown tags in ASPX. To organize the layout of the form, you have added the PXLayoutRule tags. You have also updated the modified ASPX file in the customization project.

Lesson 2.4: Add the Form to the Site Map and Workspace

Learning Objectives

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

- Create a site map item for a custom form
- Save the created site map item to the customization project
- Add a form created in Visual Studio to the Screen Editor

Figure: The selected site map item

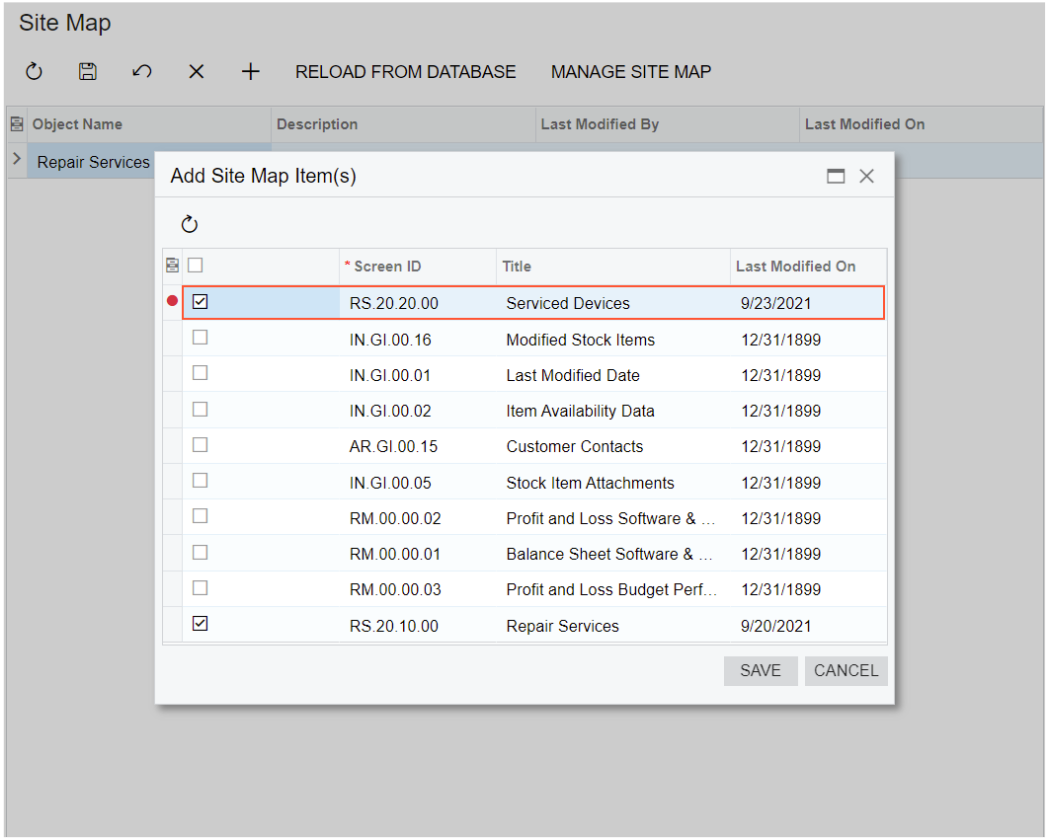


Figure: The Site Map page

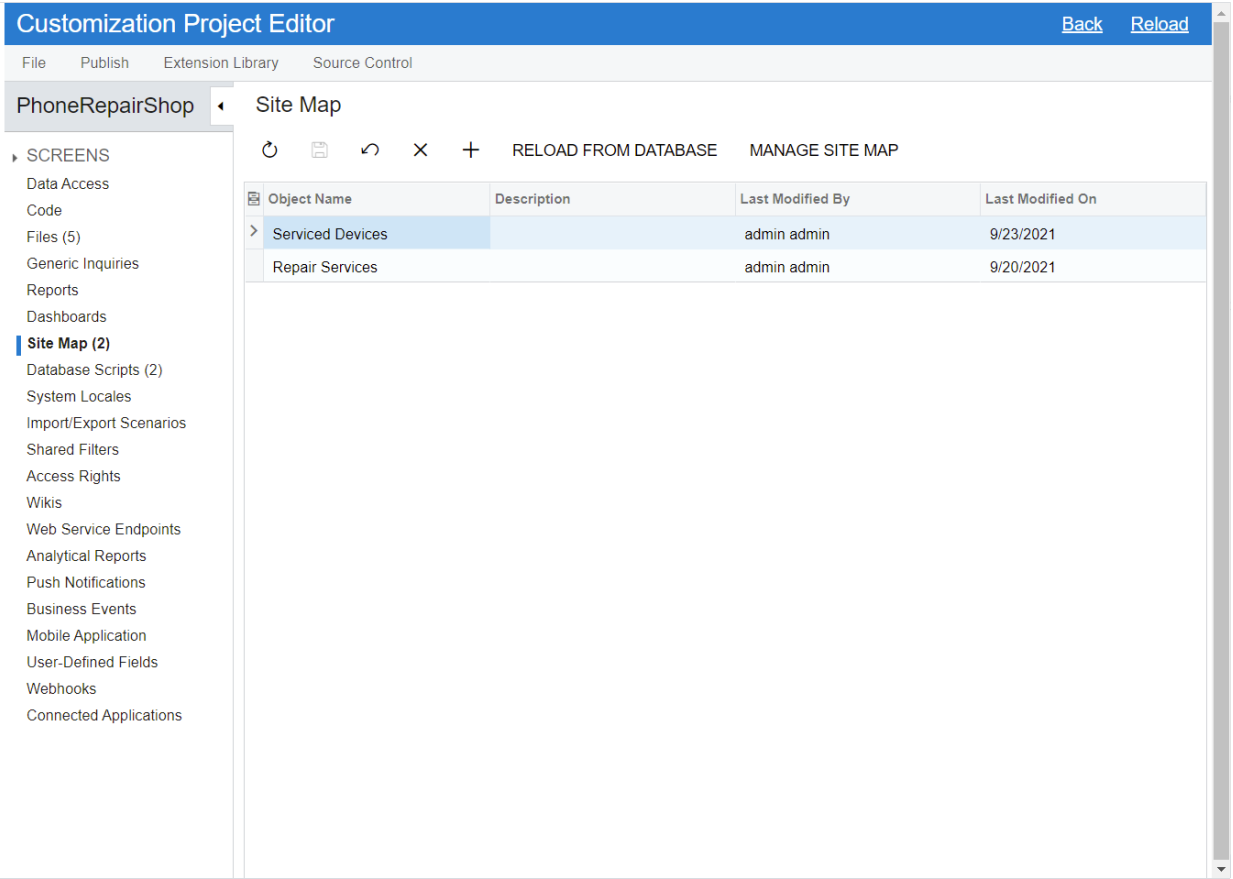


Figure: Selecting the Serviced Devices form

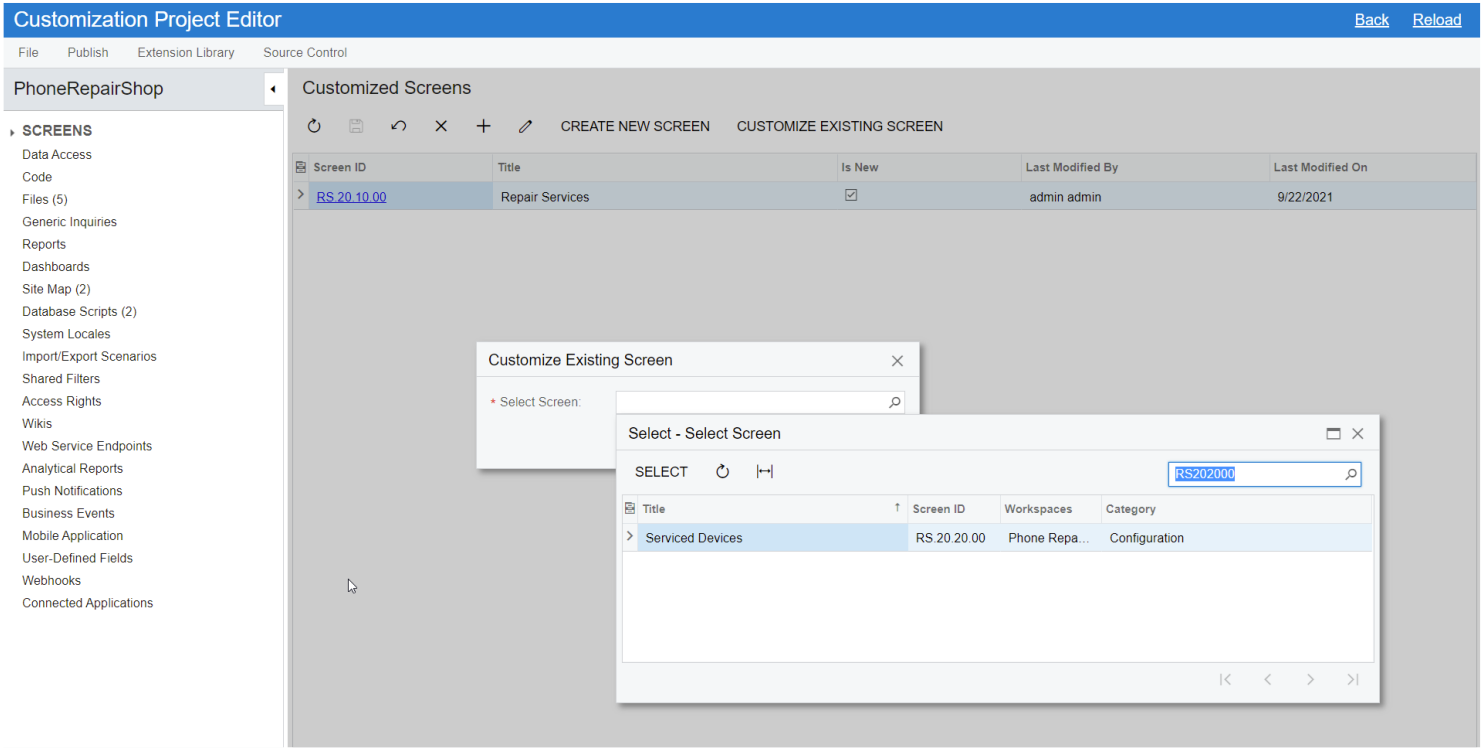


Figure: Viewing the Serviced Devices form in the Screen Editor

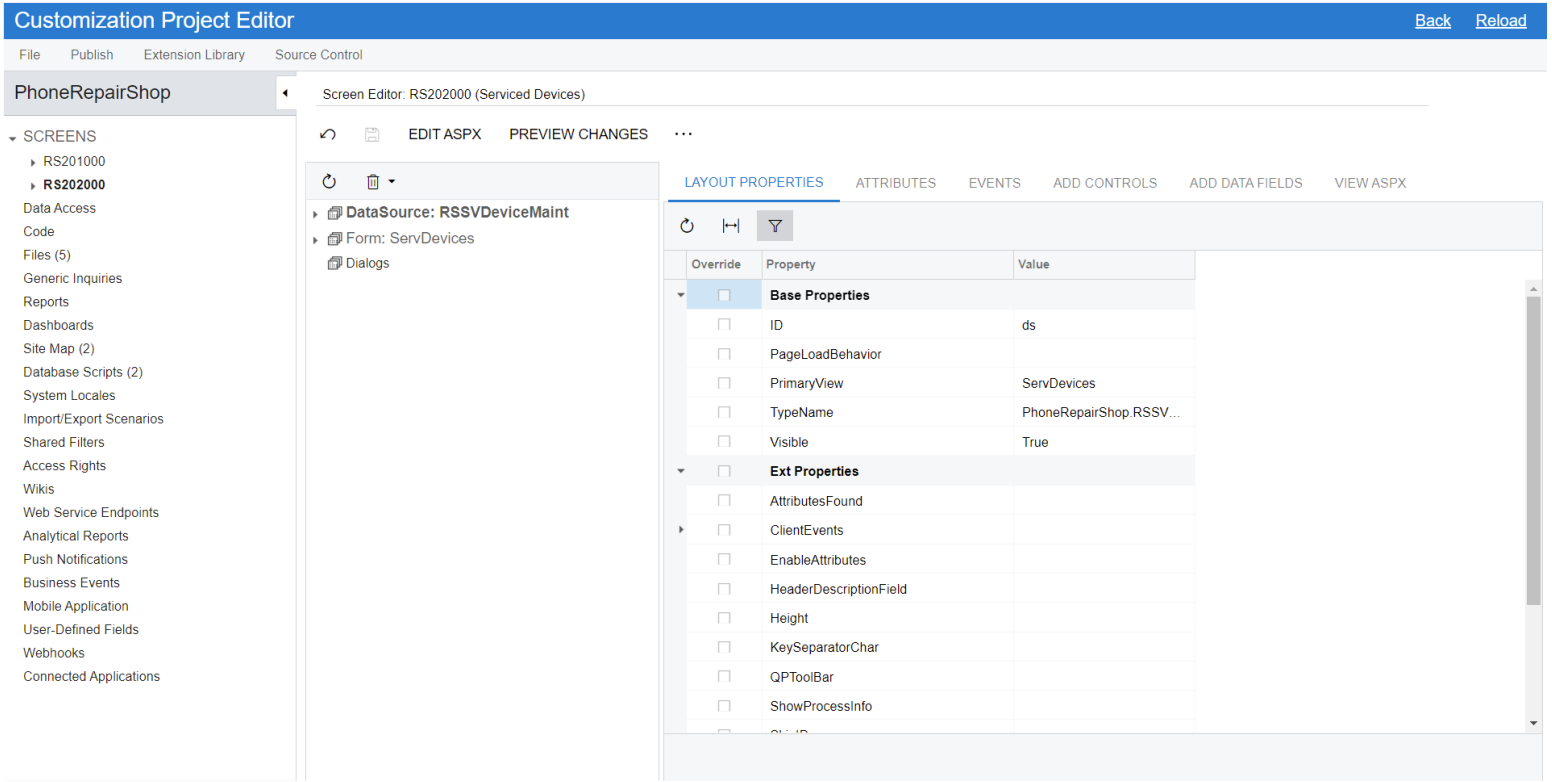


Figure: The Serviced Devices form

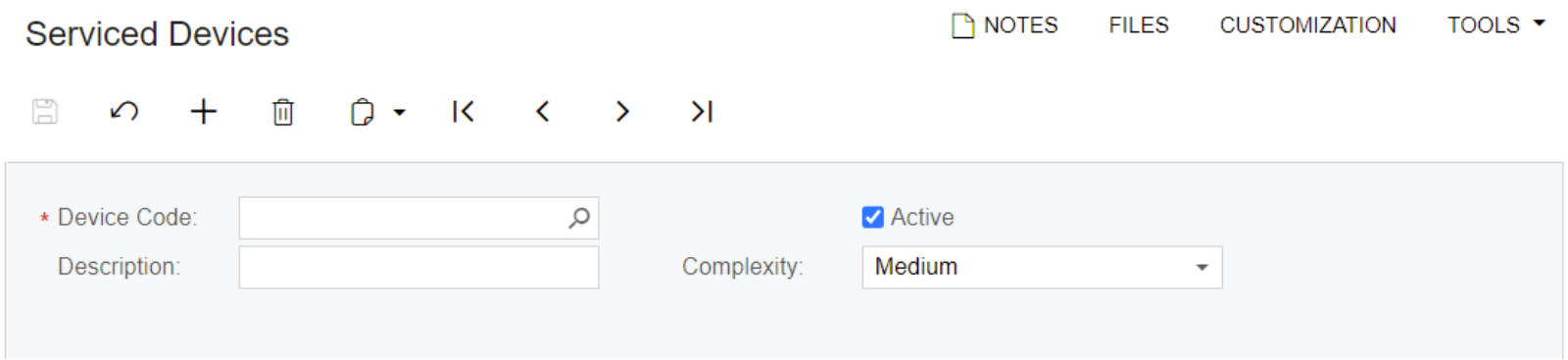


Figure: The lookup table

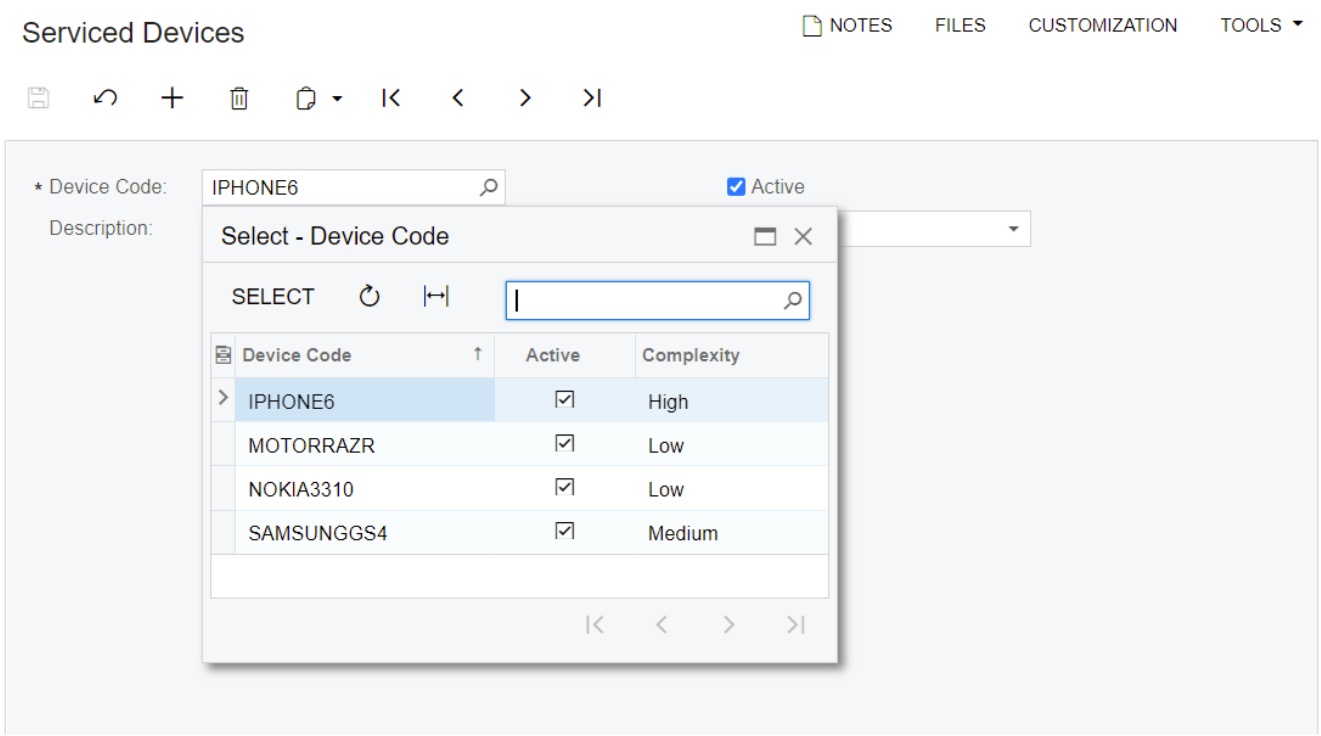
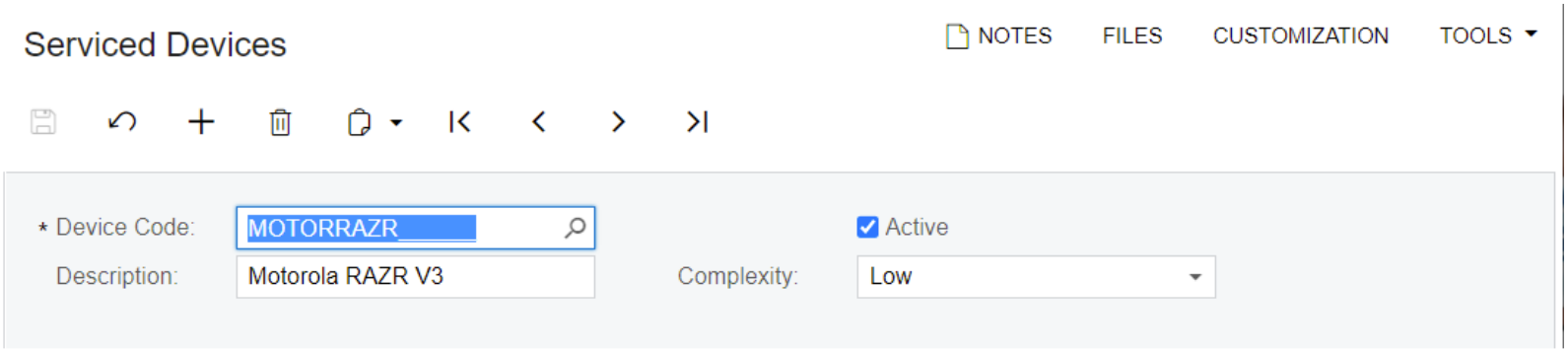


Figure: The device properties



Lesson Summary

In this lesson, you have added the Serviced Devices (RS202000) form to a workspace, saved a site map item to the customization project, added the form to the Screen Editor, and tested the form.

Lesson 2.5: Create a Substitute Form

Learning Objectives

In this lesson, you will learn how to create a substitute form for a custom form and save it to the customization project.

Figure: The Serviced Devices generic inquiry

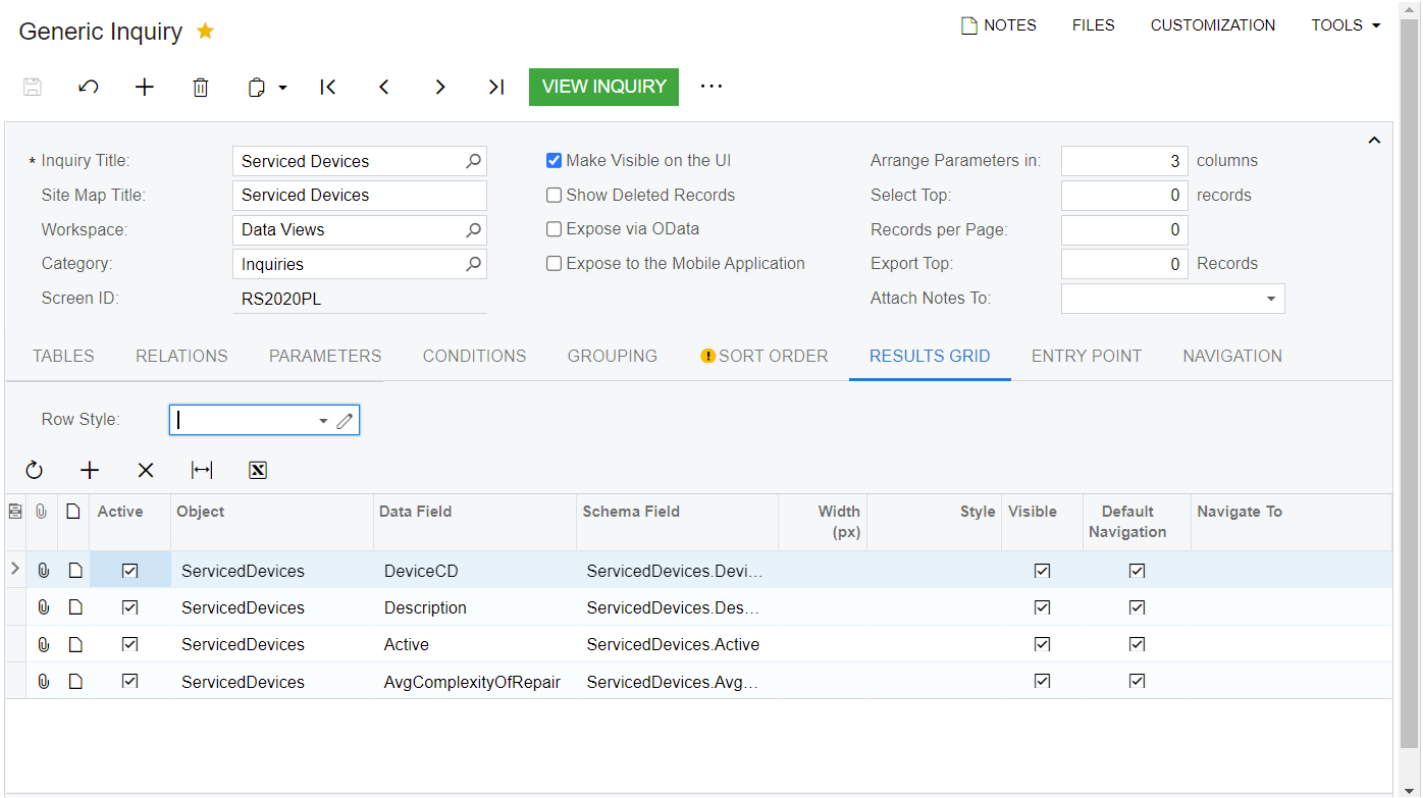


Figure: Selecting the Serviced Deviced form

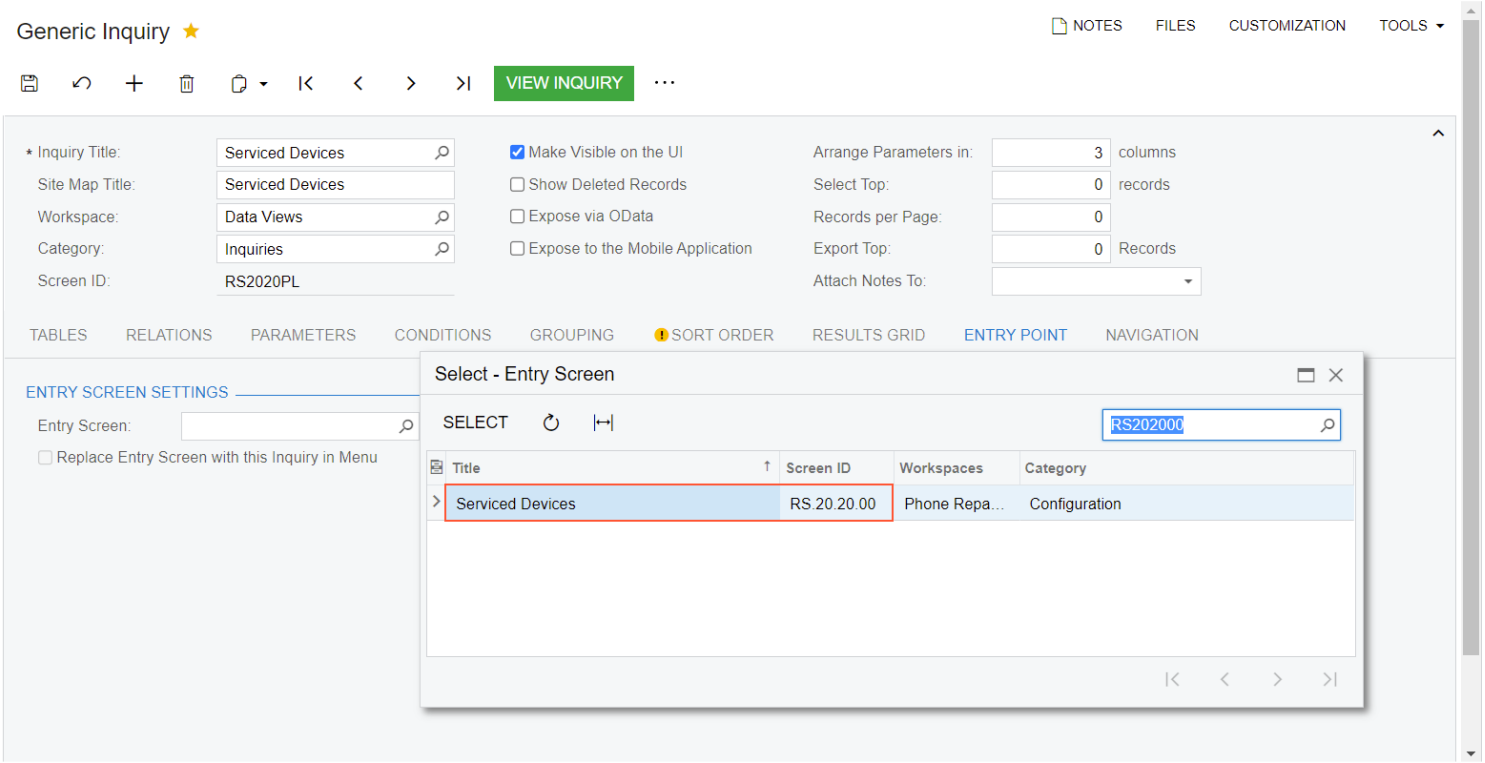


Figure: The Add Generic Inquiries dialog box

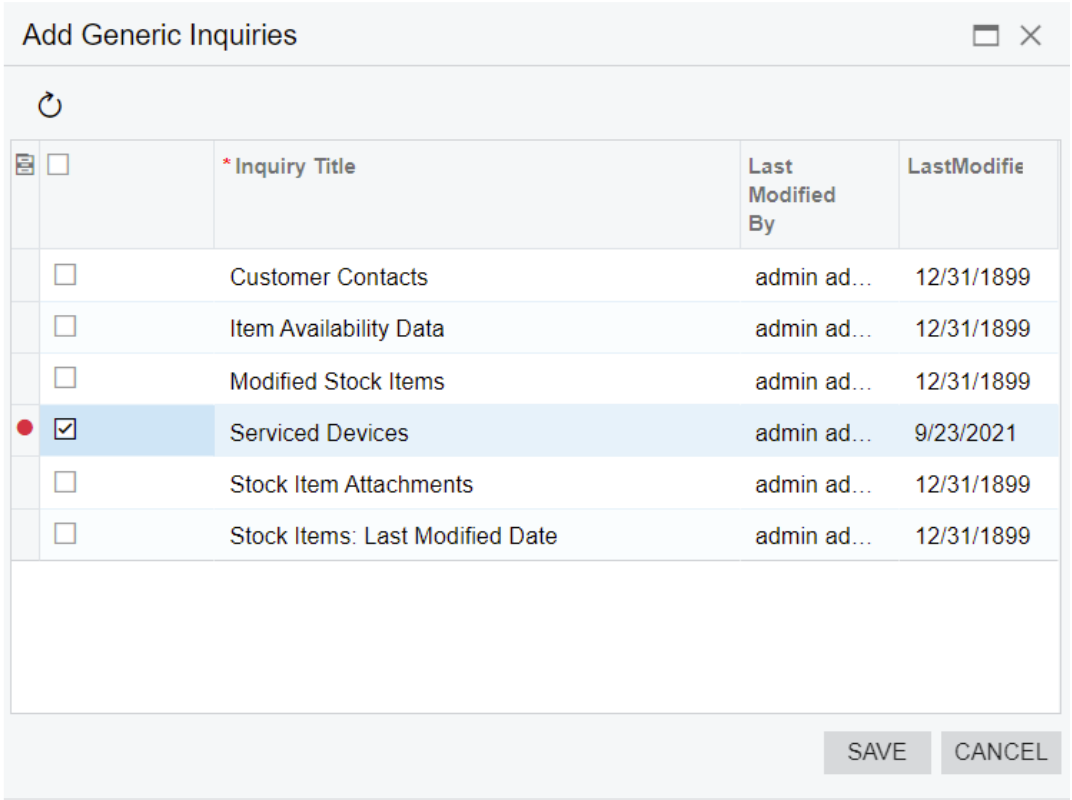
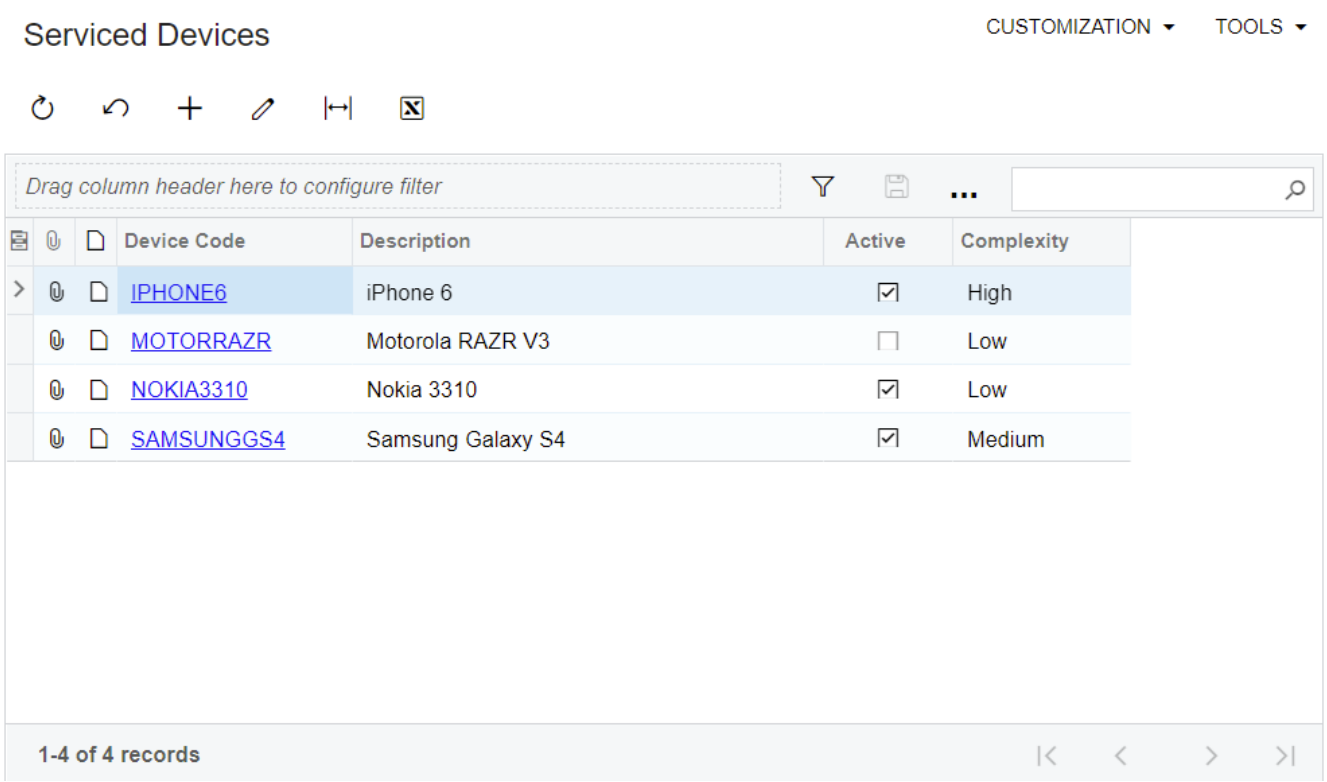


Figure: The Serviced Devices substitute form



Lesson Summary

In this lesson, you have learned about substitute forms and have created one for the Serviced Devices (RS202000)

form.

To configure the substitute form, you have completed the following steps:

1. Loaded a predefined generic inquiry on the Generic Inquiry (SM208000) form.
2. Configured the properties of the generic inquiry on the **Entry Point** tab of this form.
3. Saved the generic inquiry to the customization project.

Course Summary

In this course, you have learned the basic concepts of developing a custom form in Acumatica ERP using the Customization Project Editor and Visual Studio. Now you know the following:

- How to create a maintenance form
- How to access and manipulate data by using DACs and graph members
- How to implement business logic on a form by using event handlers
- How to configure controls that display data on a form and adjust the layout of controls on the form

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

Vidhyalakshmi Hariharasubramanian