

Developer Course

Customization

T400 Basic Customization of the Mobile App 2025 R1

Revision: 4/9/2025

Contents

Copyright.....	3
How to Use This Course.....	4
Preparing the Environment.....	6
Part 1: Introduction to the Acumatica Mobile App.....	8
Lesson 1.1: Install the Acumatica Mobile App.....	8
Lesson 1.2: Sign In to the Acumatica Mobile App.....	9
Part 2: Tools and Languages.....	11
Lesson 2.1: Explore a WSDL Schema.....	11
Lesson 2.2: Add a Screen to the Mobile App.....	13
Step 2.2.1: Open the Mobile Application Page.....	13
Step 2.2.2: Add a Screen.....	14
Step 2.2.3: Update the Mobile Site Map.....	15
Step 2.2.4: Add a Screen To a Workspace.....	16
Step 2.2.5: Test the Added Functionality in the Mobile App.....	18
Lesson Summary.....	20
Part 3: Configuration of Dashboards and Generic Inquiries.....	22
Lesson 3.1: Configure a Dashboard Screen.....	23
Lesson 3.2: Add a Generic Inquiry with Parameters.....	26
Lesson 3.3: Map the Inquiry With Tabs.....	29
Lesson 3.4: Add a Generic Inquiry Form by Using the Browser Version of Acumatica ERP.....	32
Lesson 3.5: Add a KPI Widget.....	35
Part 4: Configuration of the Extended Functionality.....	37
Lesson 4.1: Update a Screen.....	37
Lesson 4.2: Configure the Attachment Capabilities of a Screen.....	40
Lesson 4.3: Configure the Ability to Enhance and Attach Receipts.....	41
Lesson 4.4: Configure a Report.....	43
Lesson 4.5: Configure the Signature Capabilities.....	46
Lesson 4.6: Map an Action.....	48

Copyright

© 2025 Acumatica, Inc.

ALL RIGHTS RESERVED.

No part of this document may be reproduced, copied, or transmitted without the express prior consent of Acumatica, Inc.

3075 112th Avenue NE, Suite 200, Bellevue, WA 98004, USA

Restricted Rights

The product is provided with restricted rights. Use, duplication, or disclosure by the United States Government is subject to restrictions as set forth in the applicable License and Services Agreement and in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c)(1) and (c)(2) of the Commercial Computer Software-Restricted Rights at 48 CFR 52.227-19, as applicable.

Disclaimer

Acumatica, Inc. makes no representations or warranties with respect to the contents or use of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Acumatica, Inc. reserves the right to revise this document and make changes in its content at any time, without obligation to notify any person or entity of such revisions or changes.

Trademarks

Acumatica is a registered trademark of Acumatica, Inc. HubSpot is a registered trademark of HubSpot, Inc. Microsoft Exchange and Microsoft Exchange Server are registered trademarks of Microsoft Corporation. All other product names and services herein are trademarks or service marks of their respective companies.

Software Version: 2025 R1

Last Updated: 04/09/2025

How to Use This Course

The *T400 Basic Customization of the Mobile App* course introduces the basic functionality for customizing the Acumatica mobile app by using the Acumatica Customization Platform.

This course is intended for business and implementation consultants who are learning how to customize the Acumatica mobile app or other Acumatica Framework-based mobile apps.

After you complete the course, you will have an understanding of how to perform simple customization tasks of the Acumatica mobile app or other apps developed with Acumatica Framework. Upon completion of the course, you will have learned how to perform the following types of customization activity:

- Using the Customization Project Editor to customize the mobile app
- Adding general inquiries and dashboards to the mobile app
- Configuring additional functionality of the mobile app screens to give users the ability to do the following:
 - Adding signatures to a record
 - Scanning receipts
 - Generating reports
 - Attach files to a record
- Testing the customized mobile app



We recommend that you complete the examples in the order in which they are provided in the course, because some examples use the results of previous ones.

What the Course Prerequisites Are

Before you begin this course, we recommend that you work through the lessons of the *S130 Reporting: Data Retrieval and Analysis* course. To learn more about the functionality of the Acumatica Customization Platform, we recommend that you complete the *T190 Quick Start in Customization* or *W140 Customization Projects* courses before completing the current course.

What Is in a Part

Each of the course parts introduces you to various ways to customize the Acumatica mobile app and consists of lessons you are to complete. Each part begins with an explanation of the subject area that you are going to use in the lessons.

What Is in a Lesson

Each lesson consists of steps that outline the procedures you are completing and describe the related concepts you are learning.

Each lesson ends with the *Lesson Summary* topic, which summarizes the concepts you have learned during the lesson.

Where the Source Code Is

You can find the source code of the customization described in this course and code snippets for the course in the MobileDevelopment\T400 folder of the [Help-and-Training-Examples](#) repository in Acumatica GitHub.



Code snippets with code on MSDL have the JSON extension only so that the syntax can be highlighted.

What the Documentation Resources Are

The complete Acumatica ERP and Acumatica Framework documentation is available at <https://help.acumatica.com/> and is included in the Acumatica ERP instance. While viewing any form used in the course, you can click the **Open Help** button in the top pane of the Acumatica ERP screen to bring up a form-specific Help menu; you can use the links on this menu to quickly access form-related information and activities and to open a reference topic with detailed descriptions of the form elements.

Which License You Should Use

For the educational purposes of this course, you use Acumatica ERP under the trial license, which does not require activation and provides all available features. For the production use of the Acumatica ERP functionality, an administrator has to activate the license the organization has purchased. Each particular feature may be subject to additional licensing; please consult the Acumatica ERP licensing policy for details.

Preparing the Environment

In this training, you will learn how to customize the Acumatica mobile app. Before you start the practical steps of the course, you need to prepare the development environment, as described in *Preparation of the Development Environment* below, and consider the recommendations described in *Required Role for Users*.

Preparation of the Development Environment

Before you install Acumatica ERP, review [System Requirements for the Acumatica ERP Installation](#), and prepare the environment, as described in [Preparing for Installing Acumatica ERP](#) in the same guide.

You need to deploy an instance of Acumatica ERP with a company that contains specific data that you will use for the training.

To prepare the development environment, perform the following actions:



You can use the Acumatica ERP instance you used to pass the S130 training course.

1. Install Acumatica ERP, as described in [Acumatica ERP Installation On-Premises: To Install the Acumatica ERP Configuration Wizard](#).
2. In the Acumatica ERP Configuration wizard that appears after the installation is complete, select **Deploy a New Acumatica ERP Instance** to create a local instance of Acumatica ERP for the development environment.
3. To fill in the database with the data needed for the training course, on the Tenant Setup page, specify the following settings for the tenant:
 - **Tenant Name:** MyCompany
 - **New:** Selected
 - **Insert Data:** SalesDemo
 - **Parent Tenant ID:** 1
 - **Visible:** Selected
4. On the Instance Configuration page, for the **Local Path to the Instance**, specify a path that is not in the C :\Program Files (x86) and C :\Program Files folders to avoid an issue with permission to work in these folders. For example, you can enter C :\AcumaticaSites\T400.

The system creates a new Acumatica ERP instance, adds a new company, and loads the selected data.
5. Use the following initial credentials to sign in to the new company:
 - **User Name:** admin
 - **Password:** setup

Change the password when the system prompts you to do so.
6. Make sure your instance of Acumatica ERP can be accessed from other devices in your local network. To do this, on a computer in your local network, try to open the following webpage: https://<COMPUTER_IP_ADDRESS>/<INSTANCE_NAME> or http://<COMPUTER_IP_ADDRESS>/<INSTANCE_NAME>, where <COMPUTER_IP_ADDRESS> is the IP address of the computer in your local network that is running the Acumatica ERP instance and <INSTANCE_NAME> is the name of the Acumatica ERP instance you have installed. For more information, see [To Access an Acumatica ERP Instance Running Locally from the Acumatica Mobile App](#)
7. If the Sign-In page of Acumatica ERP opens, the instance is accessible in your local network.



You can run the `ipconfig` command in the Command Prompt program of the computer that is running the Acumatica ERP instance to find its IP address.

Course Materials on GitHub

You can clone or download the course materials from the [Help-and-Training-Examples](#) repository in Acumatica GitHub to a folder on your computer. The course materials include the following items:

- Files that you will need to complete the course activities
- Code snippets
- The customization project resulting from the course activities

The course materials are located in the `MobileDevelopment\T400` folder of the repository. By using these materials, you can copy code instead of entering it manually and compare your resulting project with the one located on GitHub.



Code snippets with code on MSDL have the JSON extension only so that the syntax can be highlighted.

Required Role for Users

Specialists who will work on customization projects should be assigned the *Customizer* role in the application instance that is to be customized and tested, as well as in the production application that should be updated with the customization project. With this role assigned, the specialists can use the customization tools and facilities of the Acumatica Customization Platform, and upload and publish customization projects.

A user with the *Administrator* role can assign the *Customizer* role to the needed users by using the [Users](#) (SM201010) form.

The *admin* account has already been assigned the *Administrator* role, so you do not need to assign the *Customizer* role to this account. In a production environment, however, you need to assign the *Customizer* role to all developers who will work on customization projects.

Related Links

- [Users](#)
- [To Assign the Customizer Role to a User Account](#)
- [User Access Rights for Customization](#)

Part 1: Introduction to the Acumatica Mobile App

In this part, you will become acquainted with the Acumatica mobile app. You will install it on your mobile device, sign in to a sample account, and explore the base functionality.

Lesson 1.1: Install the Acumatica Mobile App

In this lesson, you will install the Acumatica mobile app from an app store. The Acumatica mobile app is available for the following operating systems:

- Android
- iOS

You perform the step below that corresponds to your operating system.

Step: Install Acumatica Mobile App for Android

Before installing the Acumatica mobile app on your mobile device running Android, make sure the device satisfies the following requirements:

- Android 5.1 or later
- Sufficient free disk space (the amount depends on the particular device)

Also, you need to make sure that your mobile device is connected to the same local network as the computer hosting the development environment of your Acumatica ERP instance.

To install the Acumatica mobile app on an Android device, do the following:

1. On your mobile device, open the Play Store app.
2. Find the Acumatica mobile app.
3. Tap **Install**.

After the app installation is finished, you can open it.

Step: Install Acumatica Mobile App for iOS

Before installing the Acumatica mobile app on your mobile device running iOS, make sure the device satisfies the following requirements:

- iOS 15 or later
- Sufficient free disk space (the amount depends on the particular device)

To install the Acumatica mobile app on an iOS device, do the following:

1. On your mobile device, open the App Store app.
2. Find the Acumatica mobile app.
3. Tap **Get**.
4. Confirm the installation.

After the app installation is finished, you can open the app.

Lesson Summary

In this lesson you learned what is required to installed the Acumatica mobile app, and you have downloaded and installed it.

Lesson 1.2: Sign In to the Acumatica Mobile App

In this lesson, you will sign in to the Acumatica mobile app.



The instructions in this guide are designed for the Acumatica mobile app for Android. Instructions for the Acumatica mobile app for iOS may differ.

Step 1: Sign In to the Acumatica Mobile App

To use the Acumatica mobile app, you need to sign in to it by using the Acumatica instance URL and your credentials. In this training course, you can use the URL and credentials for the instance you prepared as described in the [Preparing the Environment](#) topic.

Do the following to sign in to the Acumatica mobile app:

1. Open the Acumatica mobile app.
2. In the **Server URL** section, select the version of the protocol and type address to the Acumatica ERP instance deployed in your network.
The name of the account is displayed after the server URL box.
3. Click **Next**.
4. In the **Username** box, enter `admin`.
5. In the **Password** box, enter your password.
6. Tap **Sign In**.

The app connects to your Acumatica instance and opens the Home screen of the app.

Before the mobile app is customized, it contains basic functionality and includes several screens of Acumatica ERP.



Connection to the mobile app may be blocked if the computer with your Acumatica instance has a firewall enabled.

Step 2: Explore the Acumatica Mobile App

When you sign in to the app, you see the Home screen of the app, which is one of the major parts of the app. The Home screen of the Acumatica mobile app includes the following (see the screenshot below):

- The screen toolbar, which includes the name of the tenant (Item 1 in the following screenshot)
To select a company or a branch, tap the name of the tenant.
- The KPI area where dashboards from Acumatica ERP can be added (Item 2)
- The workspace menu (Item 3)
If you tap a workspace name in the menu, you can navigate to a workspace (which corresponds to an Acumatica ERP workspace).
- The navigation bar (Item 4)

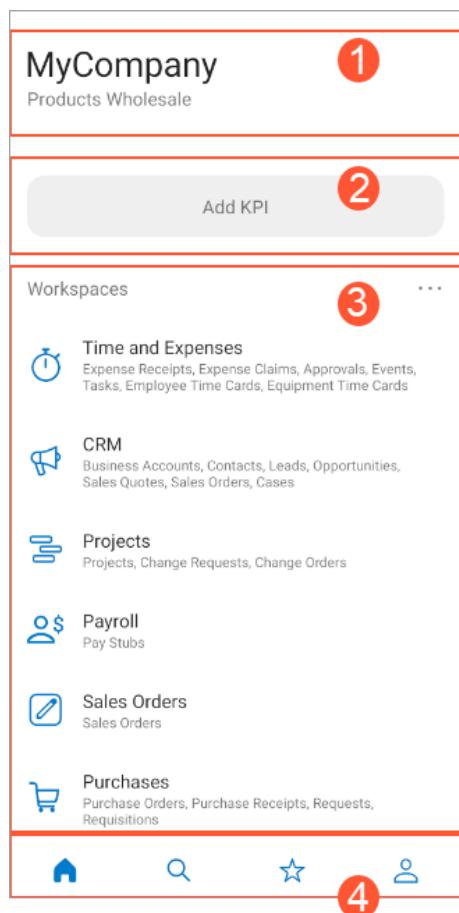


Figure: The Acumatica mobile app Home screen



The contents of the Acumatica mobile app include only those screens that are available to your user account, based on its assigned user roles.

The navigation bar (Item 4) contains access to the following:

- The Home screen of the app
- Search in the documents and mapped screens
- The list of favorite workspaces and screens
- Settings of the app where you can sign out of the instance

Lesson Summary

In this lesson, you have learned how to sign in to the mobile app and explored the main parts of the app.

Part 2: Tools and Languages

In this part, you will explore the tools and languages that are used to customize the Acumatica mobile app by using the example of the DB Sales Activities (GI000023) generic inquiry form. You will use the Mobile Application page of the Customization Project Editor and MSDL, a special language used for customizing the Acumatica mobile app.

Mobile Application Page of the Customization Project Editor

You use the Mobile Application page of the Customization Project Editor to manage the customization of screens in the Acumatica mobile app. On the page, you can add, update, or remove screens; you can also remove the customization for any tenants without removing the customization code.

MSDL

An Acumatica mobile client application uses the Mobile API to access the data of the forms that are mapped to mobile apps in the Acumatica ERP instance. The metadata of the mobile site map is used to configure the user interface of the mobile client app. You can expose any form of Acumatica ERP on your mobile device if the mobile site map includes the metadata for the form.

The metadata of the form is described by using Mobile Site Map Definition Language (MSDL). Configuring the user interface of the mobile app by using MSDL is called *mapping*.

MSDL code is a set of predefined instructions. An instruction can have attributes and contain multiple nested instructions within braces. An instruction can contain commands that assign values to the various attributes of the instruction. For more information about the MSDL syntax, see [MSDL](#).

Lesson 2.1: Explore a WSDL Schema

In this lesson, you will learn what the WSDL schema is and explore the WSDL schema of the DB Sales Activities (GI000023) generic inquiry form in Acumatica ERP.

About the WSDL Schema

The WSDL schema of an Acumatica ERP webpage is an XML document that describes all possible functionality that this webpage provides, including form and table toolbar buttons, record fields, and table columns. By reviewing the WSDL schema, you can learn what to map on the mobile screen that corresponds to an Acumatica ERP form.

Step: Explore the WSDL Schema

To explore the WSDL schema, do the following:

1. Open the DB Sales Activities (GI000023) generic inquiry:
 - a. In Acumatica ERP, open the [Generic Inquiry](#) (SM208000) form.
 - b. In the **Inquiry Title** box, select **DB-SalesActivities**.
 - c. On the form toolbar, click **View Inquiry**.

The DB Sales Activities (GI000023) generic inquiry form opens.

2. On the title bar of the inquiry form, click **Tools > Web Service**.
3. On the screen with the web service links, click *Service Description*.

The WSDL schema of the DB Sales Activities generic inquiry form opens (see the following screenshot).

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

<wsdl:definitions xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:tns="http://www.acumatica.com/typed" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://www.acumatica.com/typed/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://www.acumatica.com/typed/">
      <s:element name="Clear">
        <s:complexType/>
      </s:element>
      <s:element name="ClearResponse">
        <s:complexType/>
      </s:element>
      <s:element name="GetProcessStatus">
        <s:complexType/>
      </s:element>
      <s:element name="GetProcessStatusResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="GetProcessStatusResult" type="tns:ProcessResult"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="ProcessResult">
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="Status" type="tns:ProcessStatus"/>
          <s:element minOccurs="1" maxOccurs="1" name="Seconds" type="s:int"/>
          <s:element minOccurs="0" maxOccurs="1" name="Message" type="s:string"/>
        </s:sequence>
      </s:complexType>
      <s:simpleType name="ProcessStatus">
        <s:restriction base="s:string">
          <s:enumeration value="NotExists"/>
          <s:enumeration value="InProcess"/>
          <s:enumeration value="Completed"/>
          <s:enumeration value="Aborted"/>
        </s:restriction>
      </s:simpleType>
      <s:element name="GetScenario">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="scenario" type="s:string"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetScenarioResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="GetScenarioResult" type="tns:ArrayOfCommand"/>
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:complexType name="ArrayOfCommand">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="unbounded" name="Command" nillable="true" type="tns:Command"/>
        </s:sequence>
      </s:complexType>
      <s:complexType name="Command">
        <s:sequence>
          <s:element minOccurs="0" maxOccurs="1" name="FieldName" type="s:string"/>
          <s:element minOccurs="0" maxOccurs="1" name="ObjectName" type="s:string"/>
          <s:element minOccurs="0" maxOccurs="1" name="Value" type="s:string"/>
          <s:element minOccurs="0" maxOccurs="1" default="false" name="Commit" type="s:boolean"/>
        </s:sequence>
      </s:complexType>
    </s:schema>
  </wsdl:types>

```

Figure: The WSDL Schema of the Sales Activities generic inquiry form

You map a generic inquiry form to a mobile app screen by adding the elements described in the complex types of the WSDL schema. This practice is also applied for other kinds of forms. To map a screen, you get the WSDL schema for it, find elements you want to map, and map them by using MSDL in the Customization Project Editor.

```

▼<s:complexType name="Result">
  ▼<s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="DisplayName" type="s:string"/>
    <s:element minOccurs="0" maxOccurs="1" name="Selected" type="tns:Field"/>
    <s:element minOccurs="0" maxOccurs="1" name="RowNumber" type="tns:Field"/>
    <s:element minOccurs="0" maxOccurs="1" name="SalespersonID" type="tns:Field"/>
    <s:element minOccurs="0" maxOccurs="1" name="Name" type="tns:Field"/>
    <s:element minOccurs="0" maxOccurs="1" name="StartDate" type="tns:Field"/>
    <s:element minOccurs="0" maxOccurs="1" name="EndTime" type="tns:Field"/>
    <s:element minOccurs="0" maxOccurs="1" name="ActivityID" type="tns:Field"/>
    <s:element minOccurs="0" maxOccurs="1" name="Subject" type="tns:Field"/>
    <s:element minOccurs="0" maxOccurs="1" name="Body" type="tns:Field"/>
    <s:element minOccurs="0" maxOccurs="1" name="Status" type="tns:Field"/>
    <s:element minOccurs="0" maxOccurs="1" name="Class" type="tns:Field"/>
    <s:element minOccurs="0" maxOccurs="1" name="Type" type="tns:Field"/>
    <s:element minOccurs="0" maxOccurs="1" name="NoteText" type="tns:Field"/>
    <s:element minOccurs="0" maxOccurs="1" name="ServiceCommands" type="tns:ResultServiceCommands"/>
  </s:sequence>
</s:complexType>

```

Figure: The Result complex type of the WSDL schema

- Find the Result complex type, and explore its contents.

The Result complex type of the generic inquiry's WSDL schema always corresponds to the grid with the results of the generic inquiry. We will use this part of the schema later in [Lesson 2.2: Add a Screen to the Mobile App](#) to add fields to the mobile app screen.

Lesson Summary

In this lesson, you have learned what a WSDL schema is, and how to view and explore it for an Acumatica ERP form.

Lesson 2.2: Add a Screen to the Mobile App

In this lesson, you will add to the mobile site map a new screen that corresponds to the DB Sales Activities (GI000023) generic inquiry form. Adding a new screen includes the following actions:

- Opening the Mobile Application Page of the Customization Project Editor. This page lists the added and customized screens.
Because the changes made on this page are saved as a part of a specific customization project, you need to open the customization project to be used and then open this page. If needed, you create a new customization project and then open the page.
- Adding a new screen that corresponds to this generic inquiry to the mobile site map.
- Declaring this screen in the site map of the mobile app.
- Testing your changes in the mobile app.

In this lesson, you will add the customization project to be used and then perform these actions.

Step 2.2.1: Open the Mobile Application Page

You perform the customization of the Acumatica mobile app on the Mobile Application page of the Customization Project Editor, so you should start the process of adding a screen by opening this page.

All changes made on this page are saved as a part of the customization project. Thus, to begin customizing a mobile app, you should first create a customization project or open an existing project.

In this lesson, you will create a new customization project and open the Mobile Application page to become familiar with it.

Step: Open the Mobile Application Page of the Customization Project Editor

To open the Mobile Application Page of the Customization Project Editor, do the following:

1. Open the [Customization Projects](#) (SM204505) form, and on the form toolbar, click **Add Row**.
 2. In the **Project Name** column for the added row, enter **SalesActivities**.
 3. Save your changes.
 4. In the table, click the name of the project you created.
- The system opens the project in the Customization Project Editor.
5. In the navigation pane of the Customization Project Editor, click **Mobile Application**.
- The Mobile Application page opens (see the following screenshot).

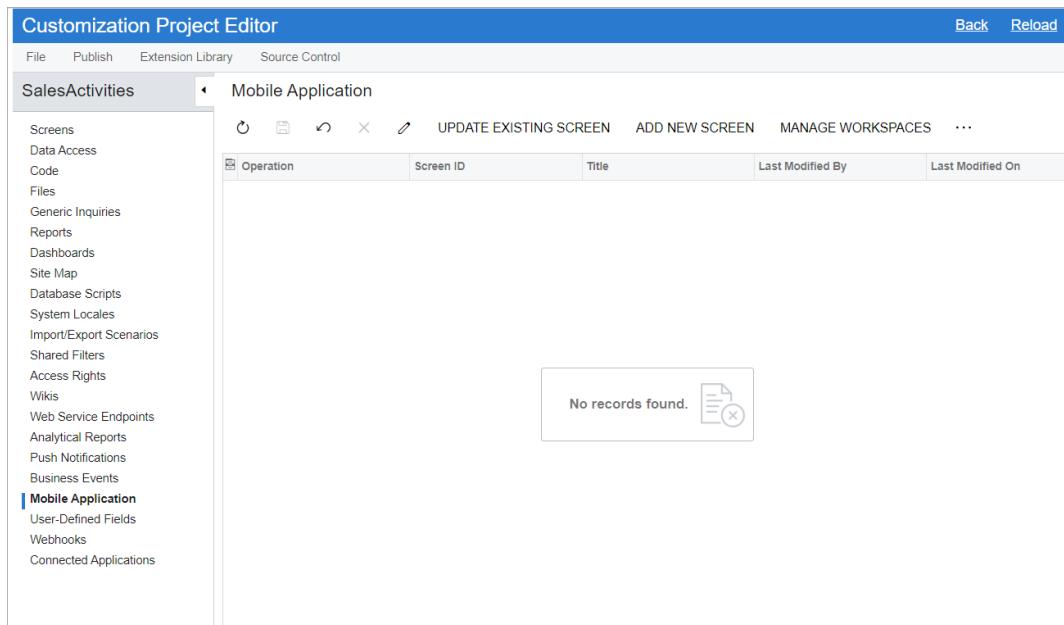


Figure: The Mobile Application page

Now you can explore the Mobile Application page, which includes the following parts:

- The page toolbar
- The table that will contain the list of added and customized screens

When you add a new customization of a screen, the name of the screen will also appear in the navigation pane of the Customization Project Editor, under the **Mobile Application** node.

Step 2.2.2: Add a Screen

One of the ways to customize a mobile app is to add a new screen to it. In this step, you will add a screen corresponding to the DB Sales Activities (GI000023) generic inquiry form.

Step: Add a Screen to the Mobile App

1. In the Customization Project Editor, open the Mobile Application page.

2. On the page toolbar, click **Add New Screen**.
3. In the **Add New Screen** dialog box, which opens, select *GI000023*, and click **OK**.

The Add: *GI000023* DB Sales Activities page opens. It consists of the following areas:

- The **Commands** area.
You add the new MSDL code in this area.
- The **Errors** area.
The area shows any syntax errors after you save your changes in the **Commands** area.
- The **Result Preview** area.
The area shows the resulting MSDL code of the screen after your changes have been applied successfully.

4. In the **Commands** area, notice that the initial code of the **Commands** area includes the `add` instruction (see the following code), which indicates that the *GI000023* screen should be added to the mobile site map.

```
add screen GI000023 {  
}
```

5. Specify the columns of the *GI000023* screen that you want to see in the mobile app screen, as shown in the following code.

```
add screen GI000023 {  
    add container "Result" {  
        add field "Name"  
        add field "Subject"  
        add field "Status"  
        add field "Type"  
    }  
}
```

You get the names of containers and fields from the WSDL schema of the DB Sales Activities (*GI000023*) generic inquiry form (see [Lesson 2.1: Explore a WSDL Schema](#)). The **Result** container always corresponds to the table with the results of the inquiry.

6. On the page toolbar, click **Save**.

If your changes were saved without errors, you see the code you added in the **Result Preview** area of the page. The changes are not yet visible in the mobile app because the customization project has not been published.

Related Links

- [To Add a Screen to the Mobile Site Map \(Example\)](#)
- [add](#)
- [screen](#)
- [container](#)
- [field](#)

Step 2.2.3: Update the Mobile Site Map

For a new screen to be displayed in the Acumatica mobile app, you need to add it to the site map of the app.

In this step, you will update the site map of the mobile app by adding information about the new screen from [Step 2.2.2: Add a Screen](#).

Step: Update the Mobile Site Map

To update the mobile site map, do the following:

1. In the Customization Project Editor, open the Mobile Application page.
2. On the More menu (under **Actions**), click **Update Main Menu**.
The Update: MENU page opens.
3. In the **Result Preview** area of this page, explore the original code of the site map.
4. In the **Commands** area, insert the following code, which defines the new screen.

```
update sitemap {
    add item "GI000023" {
        displayName = "Sales Activities"
    }
}
```

In the code above, you have updated the mobile site map by adding an item that corresponds to the screen you have added in the previous step. You have specified the name on the Home screen tile (the `displayName` attribute) for this item.

5. Save your changes.
6. Publish your customization project.

To publish a customization project, on the Customization Project Editor menu, click **Publish > Publish Current Project**.



After the customization project is published, you can close the **Compilation** pane which appears during the publication process.

Related Links

- [update](#)
- [item](#)

Step 2.2.4: Add a Screen To a Workspace

You can add a new screen to a workspace in the Acumatica mobile app. By default, a new screen is displayed in the **Other** workspace of the mobile app.



You can add a new screen to the mobile workspace only after you have published the changes that include adding the new screen to the mobile site map.

In this step, you will add the *GI000023* screen to the CRM workspace, which is predefined in the Acumatica mobile app.

Adding a Screen to the Workspace

To add the *GI000023* screen to the workspace, do the following:

1. In the Customization Project Editor, open the Mobile Application page.
2. On the page toolbar, click **Manage Workspaces**.

The Mobile Workspaces page of the Customization Project Editor opens.

3. On the page toolbar, click **Manage Workspaces**.

The [Mobile Workspaces](#) (AU220012) form opens in a new window.



You can skip Instruction 2 and open the Mobile Workspaces (AU220012) form directly in Acumatica ERP.

4. In the table, click the *CRM* workspace.

The [Mobile Workspace](#) (AU220013) form opens.

5. On the **Screens** tab, click **Add Row** on the table toolbar, and specify the following settings in the added row:

- **Visible:** Selected
- **Item Name:** G1000023

The **Display Name** and **Item Type** columns are populated automatically.

6. Save your changes.

7. Save your changes in the customization project as follows:

- a. Return to the Customization Project Editor.

The Mobile Workspaces page is opened, as shown in the following screenshot.

Object Name	Description	Last Modified By	Last Modified On
No records found. Try to change filter to see records here.			

Figure: The Mobile Workspaces page

- b. On the page toolbar, click **Add New Record**.

- c. In the **Add Workspace** dialog box, which opens, select the *CRM* workspace as shown in the following screenshot, and click **Save**.

The workspace is saved to the customization project.

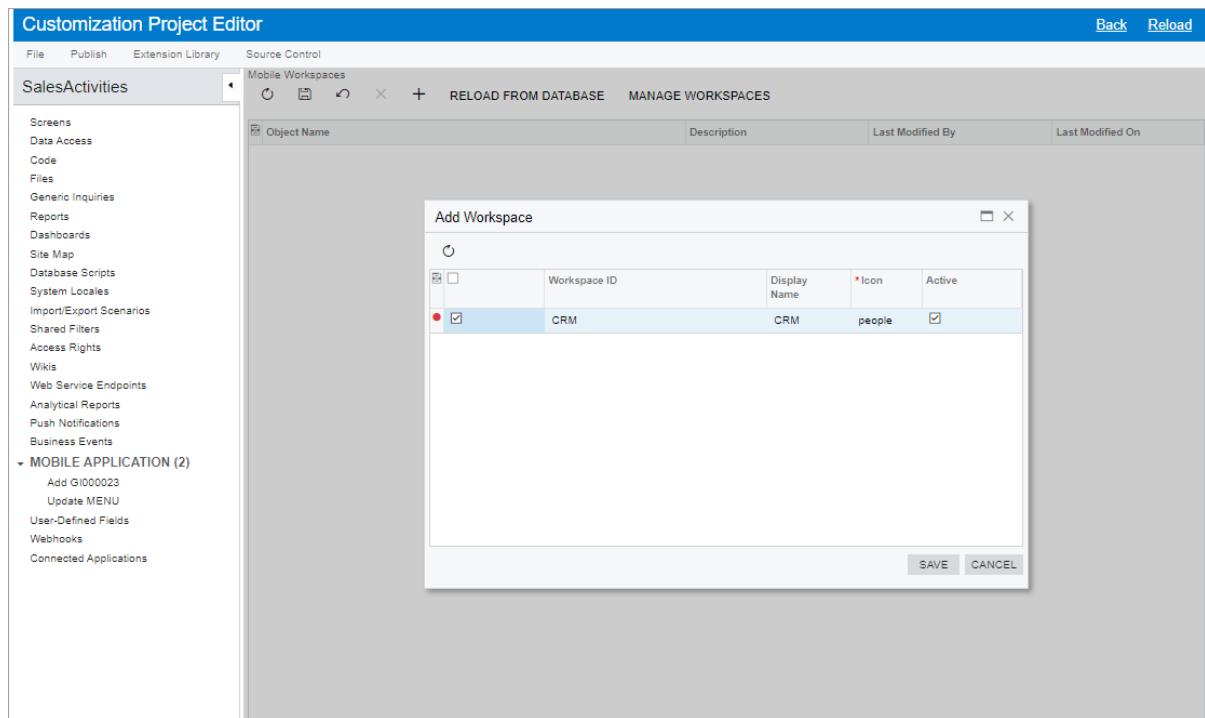


Figure: The Add Workspace dialog box



A screen will appear in the mobile app even if you do not perform Instruction 7. The actions in this instruction are necessary if you need to apply these changes to another instance.

Related Links

- [To Manage the Workspaces of the Mobile App](#)

Step 2.2.5: Test the Added Functionality in the Mobile App

After you have updated the mobile site map and published your customization project, you can check your changes in the mobile app.

Step: Check the Changes in the Mobile App

To check the changes in the mobile app, do the following:

1. On your mobile device, open Acumatica mobile app and sign in.

If you were already signed in, you need to sign out and sign in again to view your changes to the mobile app.

2. On the Home screen, tap **CRM**.

The CRM workspace is displayed as shown in the following screenshot. At the end of the list of screens of the workspace, the link to the Sales Activities generic inquiry is displayed.

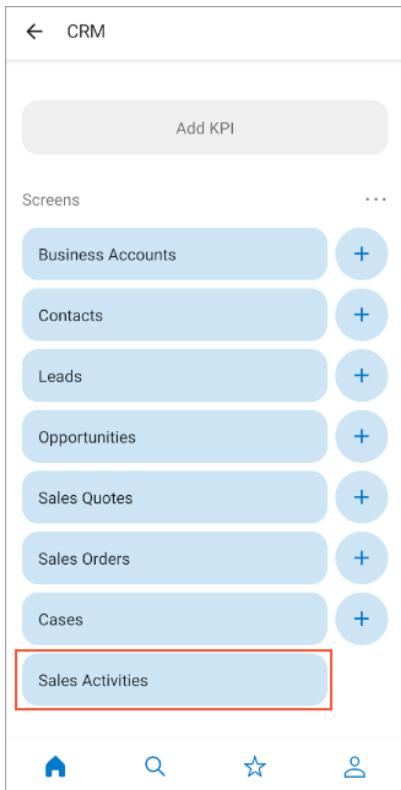


Figure: The CRM workspace

3. Tap **Sales Activities**.

The Sales Activities screen is opened as shown in the following screenshot.

← Sales Activities		Q	:
Sort By			
Jason Mendenhall	Contact Deyo about laptop pricing		
Open		Task	
Michal Bujacek	Prepare new proposal		
Processing		Task	
Michal Bujacek	received inbound sales call		
Open		Phone Call	
Michal Bujacek	Setup project contract with client		
Completed		Phone Call	
Steve Church	Talk to prospect		
Open		Phone Call	
Steve Church	Held organizational meeting		
Open		Appointment	
Steve Church	Follow up on pricing		
Open		Chat	
Steve Church	Create PPT for decision makers		
Open		Work Item	

Figure: The Sales Activities screen

4. Tap any record in the list.

A screen appears with the same fields (along with the title of each field) as were shown for the selected record on the Sales Activities screen (see the following screenshot.)

X Sales Activities	
Name	Jason Mendenhall
Subject	Contact Deyo about laptop pricing
Status	Open
Type	Task

Figure: The screen with the details of a record

Lesson Summary

In this lesson, you have learned how to use the Customization Project Editor to customize the mobile app and added a new screen to the mobile app. In particular, you have done the following:

- Installed the Acumatica mobile app on your mobile device
- Signed in to the Acumatica mobile app by using the credentials of your Acumatica ERP instance

- Explored the WSDL schema of the DB Sales Activities (GI000023) generic inquiry form
- Created a new customization project in your instance of Acumatica ERP
- Added a new screen corresponding to the DB Sales Activities (GI000023) generic inquiry form to the mobile site map
- Added the new screen to a mobile workspace
- Tested the added screen in the mobile app

Part 3: Configuration of Dashboards and Generic Inquiries

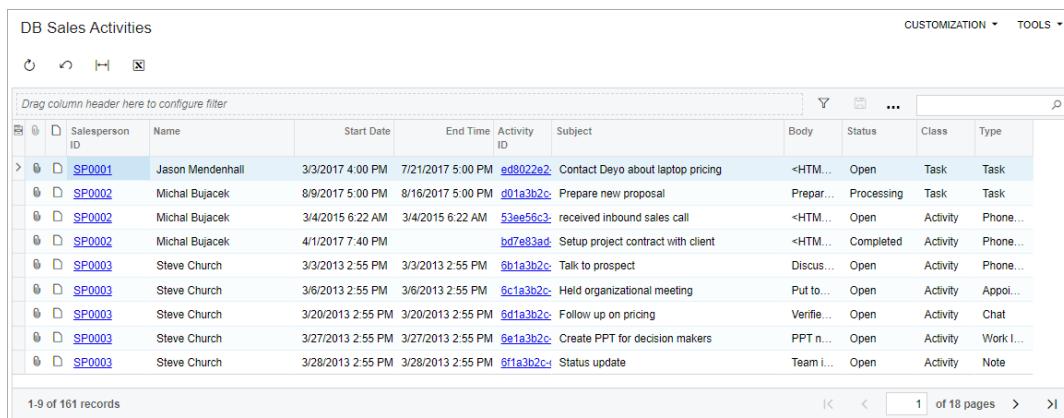
Acumatica Mobile Framework provides the ability to add generic inquiries and dashboards to the Acumatica mobile app. In this part, you will learn about different ways to map generic inquiries to the mobile app and the way to map dashboards to the mobile app.

Generic Inquiries With and Without Parameters

In Acumatica ERP, a generic inquiry can be configured to have parameters, which correspond to elements in the Selection area of the resulting generic inquiry form (above the results grid). By using these elements, users can narrow the results shown on the form. Whether or not the generic inquiry has parameters determines how it is mapped for the mobile app as follows:

- Without parameters: This generic inquiry contains only the results grid, as shown in the following screenshot.

To map such an inquiry for the mobile app, you declare only one container: **Result**. You have mapped this type of generic inquiry in [Lesson 2.2: Add a Screen to the Mobile App](#).



The screenshot shows a generic inquiry titled "DB Sales Activities". The interface includes a toolbar with icons for back, forward, search, and refresh, along with "CUSTOMIZATION" and "TOOLS" dropdown menus. Below the toolbar is a filtering bar with the placeholder "Drag column header here to configure filter". The main area is a results grid with the following columns: Salesperson ID, Name, Start Date, End Time, Activity ID, Subject, Body, Status, Class, and Type. The grid displays 1-9 of 161 records. The first few rows show entries for Jason Mendenhall, Michal Bujacek, and Steve Church, each with a unique ID and corresponding activity details.

Salesperson ID	Name	Start Date	End Time	Activity ID	Subject	Body	Status	Class	Type
SP0001	Jason Mendenhall	3/3/2017 4:00 PM	7/21/2017 5:00 PM	g08022e...	Contact Deyo about laptop pricing	<HTM...	Open	Task	Task
SP0002	Michal Bujacek	8/9/2017 5:00 PM	8/16/2017 5:00 PM	d01a3b2c...	Prepare new proposal	Prepar...	Processing	Task	Task
SP0002	Michal Bujacek	3/4/2015 6:22 AM	3/4/2015 6:22 AM	53ae56c3...	received inbound sales call	<HTM...	Open	Activity	Phone...
SP0002	Michal Bujacek	4/1/2017 7:40 PM		b07e83ad...	Setup project contract with client	<HTM...	Completed	Activity	Phone...
SP0003	Steve Church	3/3/2013 2:55 PM	3/3/2013 2:55 PM	6f1a3b2c...	Talk to prospect	Discus...	Open	Activity	Phone...
SP0003	Steve Church	3/6/2013 2:55 PM	3/6/2013 2:55 PM	6f1a3b2c...	Held organizational meeting	Put to...	Open	Activity	Appoi...
SP0003	Steve Church	3/20/2013 2:55 PM	3/20/2013 2:55 PM	6f1a3b2c...	Follow up on pricing	Verifie...	Open	Activity	Chat
SP0003	Steve Church	3/27/2013 2:55 PM	3/27/2013 2:55 PM	6f1a3b2c...	Create PPT for decision makers	PPT n...	Open	Activity	Work I...
SP0003	Steve Church	3/28/2013 2:55 PM	3/28/2013 2:55 PM	6f1a3b2c...	Status update	Team ...	Open	Activity	Note

Figure: A generic inquiry without parameters

- With parameters: This generic inquiry contains the filtering area and the results grid, as shown in the following screenshot.

To map such an inquiry, you declare two containers: **Filter_** and **Result**. You will map this kind of generic inquiry in [Lesson 3.2: Add a Generic Inquiry with Parameters](#).

Date	Customer ID	Customer Name	Order Nbr.	Quantity	Sales Order Total	Status
> 0 1/1/2024	FDIAGRI	Agrilink Food	SO008035	11,668.000000	239,570.70	Completed
0 1/1/2024	FDIMINN	Minneapolis Food Distribution	SO008078	1,529.000000	38,540.00	Completed
0 1/1/2024	ABARTENDE	USA Bartending School	SO008055	0.000000	160,000.00	Completed
0 1/2/2024	CASHCONNEC	Cash Connection	SO008087	420.000000	4,550.00	Completed
0 1/2/2024	ELEVATION	Elevation Computers	SO008041	2,423.800000	509,230.00	Completed
0 1/2/2024	TOYONEILL	O'Neill's Trading	SO008046	18,342.800000	309,882.70	Completed
0 1/2/2024	WESTERNTRC	Western Star Trucks	SO008091	0.000000	17,244.00	Completed
0 1/3/2024	STREAMRAY	Streamray Inc	SO008072	13,010.720000	116,760.16	Completed
0 1/4/2024	INDSACRAME	Sacramento Industrial Supply	SO008043	3,676.800000	62,999.97	Completed
0 1/5/2024	ELITEANSW	Elite Answering	SO008042	190.400000	35,000.00	Completed

Figure: A generic inquiry with parameters

Preliminary Steps

In this part, you will use the Open Sales Orders by Customer (GI400001) generic inquiry form, which was created in the *S130 Reporting: Data Retrieval and Analysis* training course. If you do not have the inquiry in your Acumatica ERP instance, you need to import the SO-OpenByCustomer.xml file, which is included in the MobileDevelopment\T400\FilesForTraining folder of the GitHub repository that you downloaded in [Preparing the Environment](#). To import this generic inquiry, do the following:

1. Open the [Generic Inquiry](#) (SM208000) form.
2. On the form toolbar, click **Clipboard > Import from XML**.
3. In the **Upload XML file** dialog box, which opens, select the SO-OpenByCustomer.xml file provided with the training course.
4. Click **Upload**.

The SO-OpenByCustomer inquiry opens.

Lesson 3.1: Configure a Dashboard Screen

In this lesson, to configure a dashboard screen in the mobile app, you will add the Sales Operations dashboard screen with its widgets to the mobile app. Each widget that can be used in a dashboard screen is based on a generic inquiry. For details on how to manage dashboards in a customization project, see [Dashboards](#).

A mobile screen of the *Dashboard* type can display the following types of dashboard widgets:

- Chart
- Data table
- Scorecard KPI
- Trend card KPI

Widgets of other types are not available in the mobile app.

Step: Configure a Dashboard Screen

1. In the browser version of Acumatica ERP, open the *Sales Operations* dashboard.
Notice that the form already contains several widgets.
2. As a self-guided exercise, add a widget (of one of the supported types listed above) to the dashboard for each of the following generic inquiry forms:
 - DB Sales Activities (GI000023)
 - Open Sales Orders by Customer (GI400001)



To learn how to add widgets, see [Configuring Widgets](#).

When you add a widget for the Open Sales Orders by Customer generic inquiry form, you may see a placeholder with a grey lock icon instead of the widget on the dashboard. This means that you do not have the appropriate access rights configured for this generic inquiry form; hence, you are not able to view its data in the widget. To configure the necessary access rights for the Open Sales Orders by Customer generic inquiry form, do the following:

- a. Open the [Access Rights by Screen](#) (SM201020) form. In the left pane, expand the **Data Views** node; find the **Open Sales Orders by Customer** node in the expanded list of items, and click it.
- b. With the **Open Sales Orders by Customer** node selected in the left pane, in the table on the right pane, click the row with *Administrator* in the **Role** column, and change the value in the **Access Rights** column from *Revoked* to *Delete*.
- c. Click the row with *Customizer* in the **Role** column, and change the value in the **Access Rights** column from *Revoked* to *Delete*.
- d. On the form toolbar, click **Save**



Refresh your browser tab to ensure that the modified access rights have taken effect.

The following screenshot shows an example of the dashboard with the widgets added on the bottom of the dashboard.

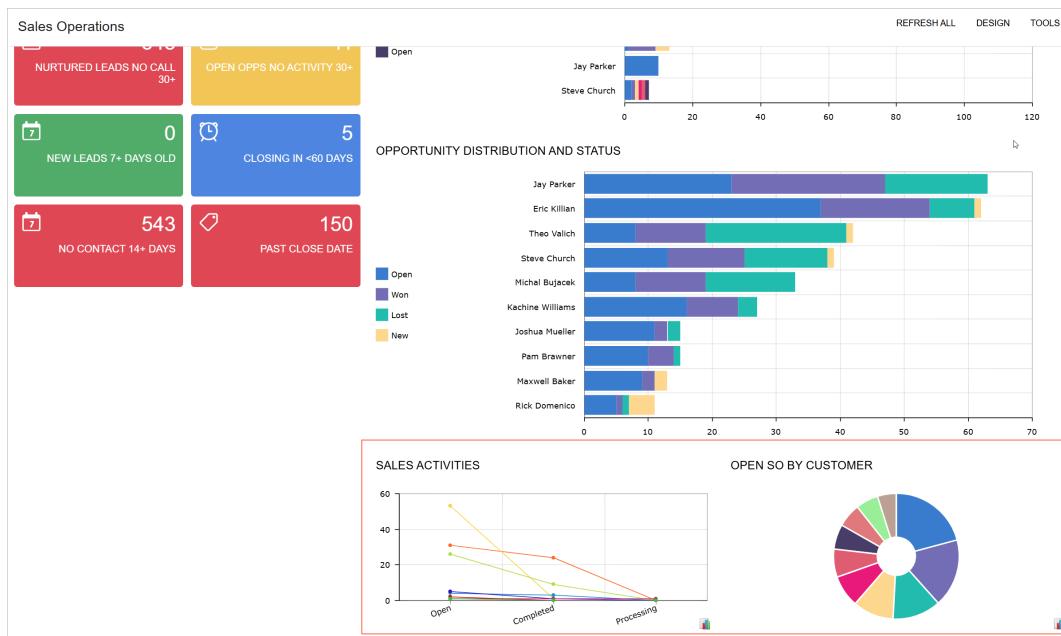


Figure: Added widgets on the Sales Operations dashboard

- Add the Sales Operations screen, which has the DB000012 screen ID, to the mobile site map, as described in [Step 2.2.2: Add a Screen](#) and [Step 2.2.3: Update the Mobile Site Map](#). The code of the screen should look as follows.

```
add screen DB000012 {
    type = Dashboard
}
```

The code for the mobile site map should look as follows.

```
add item "DB000012" {
    displayName = "Sales Operations"
}
```

- Save your changes on the Add: DB000012 Sales Operations page and on the Update: MENU page.
- Publish your customization project.
- Add the Sales Operations (DB000012) dashboard to the CRM workspace, as described in [Step 2.2.4: Add a Screen To a Workspace](#).
- Save changes made in the CRM workspace to the customization project by doing the following:
 - Open the Mobile Application page of the Customization Project Editor.
 - On the page toolbar, click **Manage Workspaces**.

The Mobile Workspaces page opens.

 - On the page toolbar, click **Reload from Database**.



You need to use the **Reload from Database** button because the CRM workspace is already added to the customization project.

- Test the mapped dashboard by doing the following:
 - In the Acumatica mobile app, to open the instance with the published customization project, sign out and sign in again.

- b. Navigate to the **CRM** workspace and open the Sales Operations dashboard screen. Scroll until you see the widgets you have added. They should look as shown in the following screenshot.

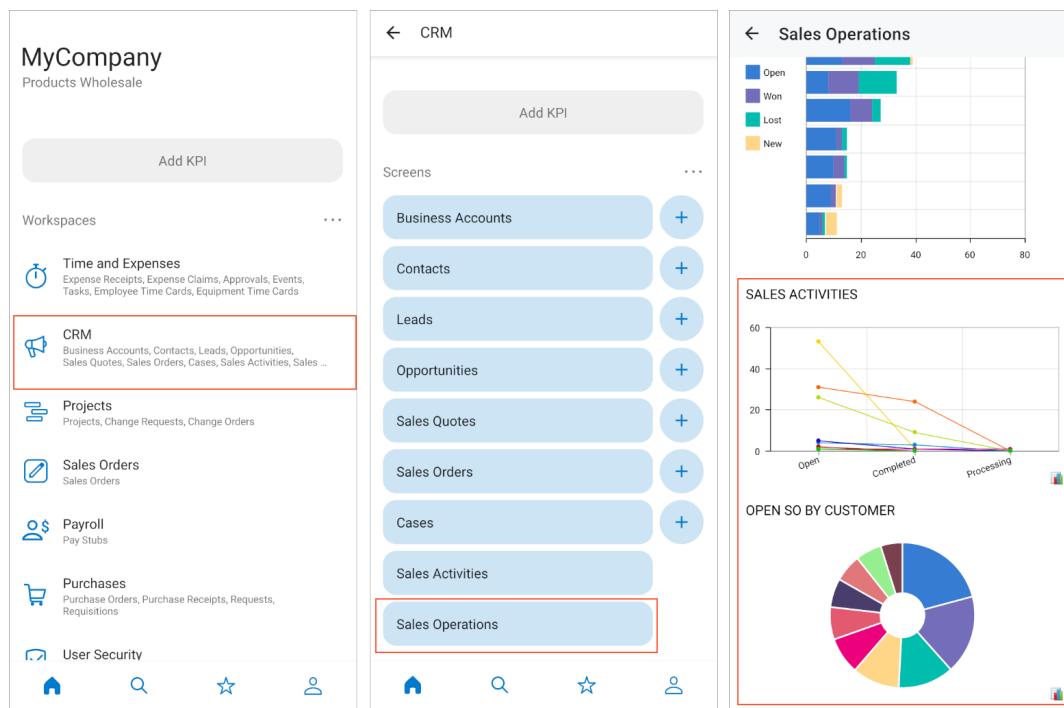


Figure: Opening the Sales Operations dashboard screen

- On the Sales Operations dashboard screen, tap the name of the *Sales Activities* widget, which shows data from the Sales Activities generic inquiry. The Sales Activities screen, which you added it in the previous lesson, opens.
- Return to the Sales Operations dashboard screen by tapping the back arrow.
- On the Sales Operations dashboard screen, tap the name of the *Open SO by Customer* widget, which shows data from the Open Sales Orders by Customer generic inquiry.

The app does not open a screen when you tap the name of the *Open SO by Customer* widget because you did not add the screen corresponding to the Open Sales Orders by Customer generic inquiry to the mobile site map. You will do it in the next lesson.

Lesson Summary

In this lesson, you have learned how to configure a dashboard screen in the mobile app by using the `type` property of the `screen` object. You have added and configured the Sales Operations dashboard screen and learned how to open generic inquiries from the dashboard in the mobile app.

Related Links

- [Designing Dashboard Contents](#)

Lesson 3.2: Add a Generic Inquiry with Parameters

In this lesson, you will explore how to add to the mobile app a generic inquiry that has parameters, which give users the ability to make selections and view the results that match the selection criteria. You will use a generic inquiry that is not included in the default data set, so make sure you uploaded it (see the *Preliminary Steps* section of [Part 3: Configuration of Dashboards and Generic Inquiries](#)).

Step: Add a Generic Inquiry with Parameters

To add a generic inquiry with parameters, do the following:

1. On the [Generic Inquiry](#) (SM208000) form, open the *SO-OpenByCustomer* generic inquiry.
2. On the **Parameters** tab, review the parameters of the inquiry, which serve as selection criteria to filter the results.
3. Explore the WSDL schema of the inquiry, as described in [Lesson 2.1: Explore a WSDL Schema](#).
4. In the Customization Project Editor, open the Mobile Application page.
5. Add the *GI400001* screen by using the process described in [Step 2.2.2: Add a Screen](#).

Because the inquiry has parameters, you have to define not only the **Result** container but also the **Filter_** container, as shown in the code below. The **Filter_** container has to include all required parameters of the inquiry.

```
add screen GI400001 {
    type = FilterListScreen
    add container "Filter_" {
        add field "DateFrom"
        add field "DateTo"
        add field "Customer"
        add field "OpenOnly"
    }
    add container "Result" {
        add field "CustomerID"
        add field "OrderNbr"
        add field "Quantity"
        add field "SalesOrderTotal"
        add field "Status"
    }
}
```



You need to set the `type` attribute of the screen to *FilterListScreen* to indicate that the screen should include the filtering functionality.

6. Add the generic inquiry to the main menu of the mobile app, as described in [Step 2.2.3: Update the Mobile Site Map](#).
Specify the `displayName` attribute value that is short enough to fit the mobile app tile and toolbar, such as *Open SO by Customer*.
7. Publish your customization project.
8. Add the Open Sales Orders by Customer (*GI400001*) inquiry to the **Sales Orders** workspace as described in [Step 2.2.4: Add a Screen To a Workspace](#).
9. Save changes made in the Sales Orders workspace to the customization project as described in instruction 7 of [Lesson 3.1: Configure a Dashboard Screen](#).
10. Check the generic inquiry in the mobile app as follows:
 - a. On the main menu, open the **Sales Orders** workspace.
 - b. In the Sales Order workspace, tap **Open SO by Customer**.
The Open Sales Orders by Customer generic inquiry opens.
 - c. On the screen toolbar, tap the filter button (see the following screenshot).

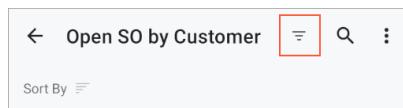


Figure: The screen toolbar with the filtering button

The screen with fields corresponding to the inquiry parameters opens (see the following screenshot).

A screenshot of a "Filter" screen. It has a header with an "X" and the word "Filter". Below are four input fields: "Date From *" (set to "Jan 1, 2024"), "Date To *" (set to "Aug 31, 2024"), "Customer" (dropdown menu), and "Open Only" (checkbox). The "Customer" field is currently empty.

Figure: The screen where users can filter the results

- d. In the **Customer** field, select the AACustomer value.
- e. On the Filter screen, click **Update**, as shown in the following screenshot.

A screenshot of the same "Filter" screen as before, but with the "Customer" field now containing "Alta Ace". The "Update" button at the top right is highlighted with a red box.

Figure: The changed Filter screen

- f. Check the inquiry results (see the following screenshot).

Open SO by Customer		Sort By	Search	More
<hr/>				
Alta Ace	S0008088			
266.520000	6,670.52			
Completed				
Alta Ace	S0008146			
235.960000	5,899.73			
Completed				
Alta Ace	S0008191			
27,644.900000	394,926.40			
Completed				
Alta Ace	S0008203			
235.960000	5,899.73			
Completed				
Alta Ace	S0008260			
235.960000	5,899.73			
Completed				
Alta Ace	S0008262			
0.000000	5,600.00			
Completed				
Alta Ace	S0008317			
239.150000	5,984.29			

Figure: The inquiry results

11. In the **CRM** workspace, tap **Sales Operations** to open the view with the dashboard screen you added in the previous lesson, and tap the name of the *Open SO by Customer* widget.
- Make sure that you are now redirected to the Open SO by Customer screen.



When you open the Open SO by Customer generic inquiry from the widget, the app opens it with the filter data saved in Instruction 10e.

Lesson Summary

In this lesson, you have learned how to map a generic inquiry with parameters and tested it in the mobile app.

Lesson 3.3: Map the Inquiry With Tabs

Sometimes, a single generic inquiry can contain multiple tabs that show the same table but with built-in filtering. In this lesson, you will learn how to map such generic inquiry.

Now that you have completed Lessons 2.1 through 3.2, you know how to map a generic inquiry to the mobile site map. As a practice exercise, you will now map the Invoiced Items (GI000008) generic inquiry form to the mobile site map.

The Invoiced Items generic inquiry should be displayed in the **Inventory** workspace which does not exist in the mobile app. So, you will add this workspace first.

Step 1: Add a New Workspace

To add the **Inventory** workspace to the mobile app, do the following:

1. In Acumatica ERP, open the [Mobile Workspaces](#) (AU220012) form.
2. On the form toolbar, click **Add Row** and enter the following values:
 - **Workspace ID:** INVENTORY
 - **Display Name:** Inventory
 - **Visible:** Selected
3. Save your changes.
4. Configure the Inventory workspace:
 - a. Click the *Inventory* workspace ID to open settings for the workspace.
The [Mobile Workspace](#) (AU220013) form opens.
 - b. In the **Icon** box, select *local shipping*.
 - c. On the **Screens** tab, notice that several screens are already added. These screens have been copied automatically from the **Inventory** workspace of the browser version of Acumatica ERP. Delete all these screens because you do not need them.
 - d. Save your changes.
5. Add the new workspace to the customization project by doing the following:
 - a. In the Customization Project Editor, open the Mobile Application page.
 - b. On the form toolbar, click **Manage Workspaces**.
The Mobile Workspace page opens.
 - c. On the page toolbar, click **Add New Record**.
 - d. In the **Add Workspace** dialog box which opens, select the *Inventory* workspace and click **Save**.
The Inventory workspace is saved to the customization project.

Step 2: Map the Inquiry

To map the Invoiced Items (GI000008) generic inquiry form, do the following:

1. In your Acumatica ERP instance, open the Invoiced Items (GI000008) generic inquiry form.
Explore the WSDL schema of the form. Decide which fields you want to map.
2. On the Mobile Application page of the Customization Project Editor, perform the following operations:
 - a. Add the *GI000008* screen to the customization project, and map the fields you have chosen.
 - b. Include the *GI000008* screen on the site map of the mobile app.
 - c. Publish your customization project.
3. In the Acumatica mobile app, open the **Other** workspace.

Notice that instead of one new line with the title of the generic inquiry, three new lines have been added. These lines correspond to the tabs of the Invoiced Items generic inquiry as shown in the following screenshot.

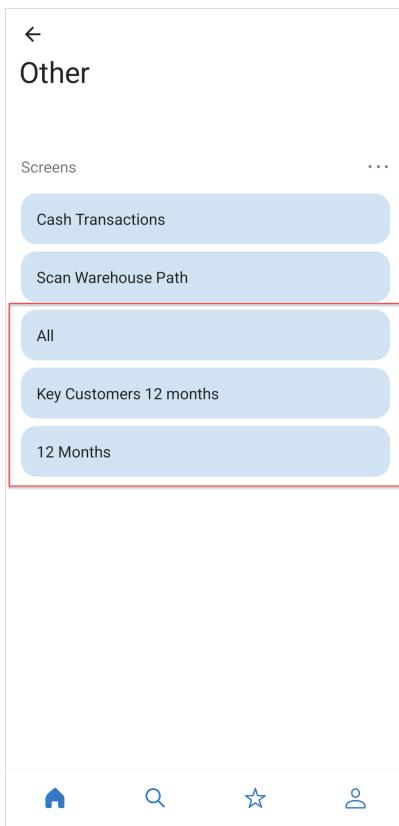


Figure: Tiles of the added inquiry

4. To avoid these lines being used to represent the generic inquiry, group the tabs as follows:
 - a. Return to the Customization Project Editor, and open the Update: MENU page.
 - b. Replace the code that includes the inquiry in the mobile site map with the following code.

```
add folder "InvoicedItems" {
    type = HubFolder
    displayName = "Invoiced Items"

    add item "GI000008" {
        displayName = "Invoiced Items"
    }
}
```

This code unites the tabs into one screen. Notice that the `type` of the folder should be set to `HubFolder`, which causes the tabs to be united into one screen.

5. Publish your customization project again.
6. Include the `GI000008` screen to the **Inventory** workspace.



You should type `InvoicedItems` in the **Item Name** column of the Mobile Workspace form to add the `GI000008` screen to the **Inventory** workspace. This is because we added this screen inside the `InvoicedItems` folder in the previous instruction.

7. Save changes made to the **Inventory** workspace to the customization project.
8. In the Acumatica mobile app, open the **Inventory** workspace.



You can also notice that the three generic inquiries you have seen in instruction 3 are no longer displayed in the Other workspace.

- Verify that the **Invoiced Items** line appears, and tap it to view the Invoiced Items screen.

The Inventory workspace and the screen should look as shown in the following screenshot.

The screenshot displays two views of the Acumatica mobile application. On the left is the 'Inventory' workspace, which includes a 'Screens' section where the 'Invoiced Items' tile is highlighted with a red box. On the right is the 'Invoiced Items' screen, which shows a list of invoices with columns for Customer ID, Date, and Description. The first few rows of the list are:

Customer ID	Date	Description
USA Bartending School AR013938	Aug 31, 2024	Special or custom order
ABC Capital Ventures AR013953	Aug 31, 2024	Labor - Junior Consultant
Active Staffing Service AR013862	Aug 31, 2024	Software SaaS, base product
Active Staffing Service AR013862	Aug 31, 2024	Software SaaS, add-on product
Artcages AR013945	Aug 31, 2024	Project Consulting
New York International Beauty School Ltd AR013949	Aug 31, 2024	Project Consulting
New York International Beauty School Ltd AR013863	Aug 31, 2024	Software maintenance

Figure: The Invoiced Items tile and screen

Related Links

- [folder](#)

Lesson 3.4: Add a Generic Inquiry Form by Using the Browser Version of Acumatica ERP

You can map a generic inquiry form to the mobile app by using the browser version of Acumatica ERP without using the Customization Project Editor. In this case, your changes will not be saved in the customization project and will stay in only your instance of Acumatica ERP. You won't be able to configure the resulting appearance of the generic inquiry screen, such as the tabs, filters, title, fields (if applicable) to be used for filtering, and the mobile workspace where the generic inquiry should be located.

In this lesson, you will map an existing generic inquiry form in the instance to the Acumatica mobile app.

Step: Map a Generic Inquiry Form by Using the Browser Version of Acumatica ERP

To map a generic inquiry form to the mobile app by using the browser version of Acumatica ERP, do the following:

1. In Acumatica ERP, open the [Generic Inquiry](#) (SM208000) form.
2. In the **Inquiry Title** box, select *SalespersonSales*.
3. Select the **Expose to the Mobile Application** check box.
4. Save your changes.



There are other ways to set the **Expose to the Mobile Application** state of a generic inquiry on the [Generic Inquiry](#) form. First, the **Expose to the Mobile Application** state of a generic inquiry is included in a customization project if you export the generic inquiry into the project. Also, the **Expose to the Mobile Application** state is imported when you import a generic inquiry in XML format.

5. Save your changes to the generic inquiry in the customization project so that the changes are available on other instances where you publish the project:
 - a. In the Customization Project Editor, open the Generic Inquiries page.
 - b. On the page toolbar, click **Add New Record**.
 - c. In the **Add Generic Inquiries** dialog box, which opens, select the *SalespersonSales* generic inquiry and click **Save**.

The line for the *SalespersonSales* generic inquiry appears on the Generic Inquiries page.
6. In the Acumatica mobile app, notice the **Data Views** workspace that is shown in the following screenshot. The **Data Views** workspace includes all generic inquiries mapped using the **Expose to Mobile Application** check box.

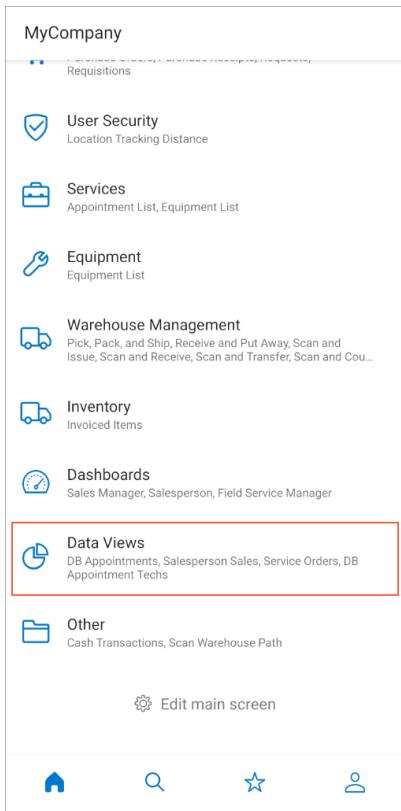


Figure: The Data Views workspace on the Home screen



If you scroll down to the end of the main screen, you can see the Edit Main Screen button which allows you to organize order and visibility of sections on the main screen.

7. Tap **Data Views > Salesperson Sales**.

The screen with the generic inquiry opens (see the following screenshot).

← Salesperson Sales		
≡ Salesperson Sales		
Salesperson ID		
Jason Mendenhall	Jason Mendenhall	10,000.00
Dec 1, 2023		
Jason Mendenhall	Jason Mendenhall	50,000.00
Jan 1, 2024		
Jason Mendenhall	Jason Mendenhall	20,000.00
Jan 1, 2024		
Jason Mendenhall	Jason Mendenhall	37,500.00
Jan 1, 2024		
Jason Mendenhall	Jason Mendenhall	10,000.00
May 1, 2024		
Jason Mendenhall	Jason Mendenhall	35,000.00
Jun 1, 2024		
Jason Mendenhall	Jason Mendenhall	25,000.00
Jul 1, 2024		
Jason Mendenhall	Jason Mendenhall	12,933.00
Dec 12, 2023		

Figure: The Salesperson Sales screen



You do not need to publish a customization project that contains this generic inquiry because the **Expose to Mobile Application** functionality is not related to a customization project.

Lesson Summary

In this lesson, you have learned how to add a generic inquiry form to the Acumatica mobile app without using a customization project.

Lesson 3.5: Add a KPI Widget

On the Home screen of the mobile app and in each of the workspaces, you can add KPI widgets of the following types:

- Scorecard
- Meter
- Trend card

By default, no KPI widgets are displayed.

In this lesson, you will add the *Orders to Ship* KPI widget from the Warehouse Manager (IN3015DB) dashboard to the Home screen of the mobile app.

Step: Add a KPI Widget to the Home Screen

To add the *Orders to Ship* KPI widget to the mobile app, do the following:

1. In the Acumatica mobile app, open the Home screen.
2. On the Home screen, tap **Add KPI**.
If the tenant already has added KPIs, in the menu of the KPIs section, tap **Add KPI**.
3. In the Search box, enter **My Opportunities**.
4. Tap the **My Opportunities** KPI widget (Item 1 in the first screenshot below) on the Salesperson dashboard.
Notice that the mobile app marks it as a favorite.
5. Click the back arrow icon (Item 2) to save your changes and return to the workspace.

The KPI widget is displayed on the Home screen, as you can see in the second screenshot.

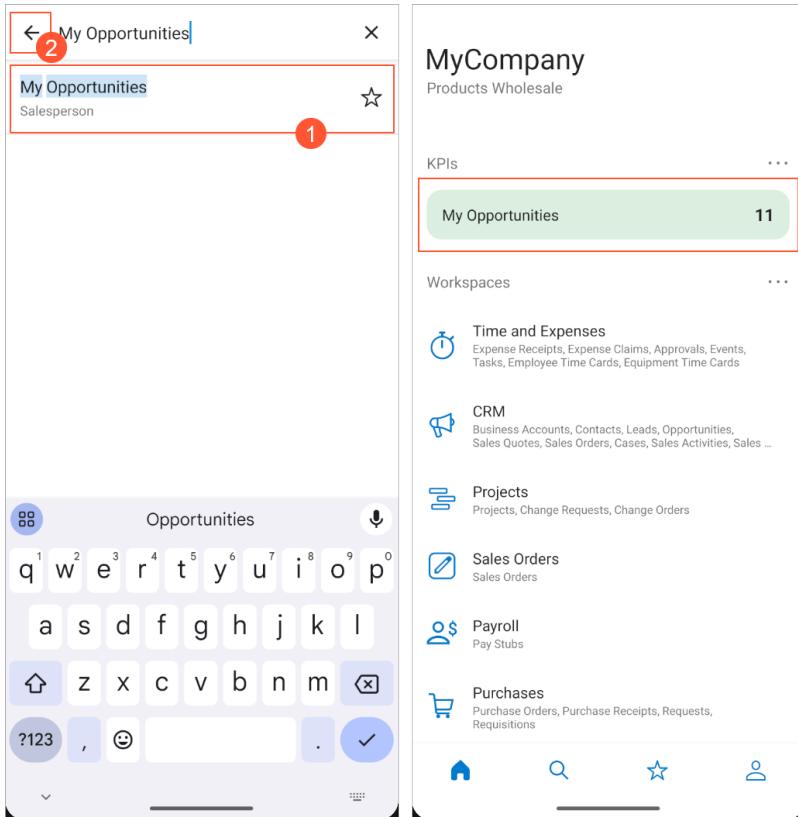


Figure: Adding a KPI widget

6. Tap **My Opportunities** to view the list of records.

The My Opportunities generic inquiry is displayed.

Part 4: Configuration of the Extended Functionality

In this part, you will learn how to configure extended functionality on the screens that have already been defined for the mobile app (that is, those that are included in the base functionality). Depending on the Acumatica ERP form that a screen is based on, you can add functionality to the screen so that users can do the following:

- Add an attachment to a screen
- Add a signature to a record
- Scan and format paper receipts
- Generate and download reports

Also, you will learn how to organize the layout of fields and map actions to an existing screen of the mobile app.

In the lessons of this part, you will customize the following screens:

- The Sales Order screen, which corresponds to the [Sales Orders](#) (SO301000) form
- The Projects screen, which corresponds to the [Projects](#) (PM301000) form

Lesson 4.1: Update a Screen

In this lesson, you will learn how to update the Sales Order screen, which already exists in the mobile site map. You will add and remove some fields in different parts of the screen.

Step 1: Prepare to Update the Screen

To prepare for the screen update, do the following:

1. Review the WSDL schema of the [Sales Orders](#) (SO301000) form, as described in [Lesson 2.1: Explore a WSDL Schema](#).
2. In the Customization Project Editor, open the Mobile Application page.
3. On the page toolbar, click **Update Existing Screen**.
4. In the **Update Existing Screen** dialog box, which opens, select SO301000.
5. Click **OK**.

The Update: SO301000 Sales Orders page opens.

6. In the **Result Preview** area of the page, explore the original code of the screen.

Notice that the original `add_screen` instruction includes containers that correspond to the complex types of the screen's WSDL schema and to parts of the mobile screen.

Step 2: Update the Header Container

The *header* is a part of the screen that should contain the fields of the Summary area of the form that cannot be edited manually. The header of the Sales Order screen is shown in the following screenshot.

The screenshot shows the 'Sales Order' screen. At the top, there's a summary box containing financial details:

Order Nbr.: SO008320	Order Total: 17,244.00
Status: Completed	Line Discounts: 1,916.00
Ordered Qty.: 40.00	Document Discounts: 0.00
Est. Margin (%): 0.00	Freight Total: 0.00
Est. Margin Amount: 0.00	Tax Total: 0.00

Below the summary is a blue button labeled 'Create Payment'. The main content area has tabs: 'Summary' (selected), 'Details', and 'Settings'. The 'Summary' tab contains the following fields:

- Order Type *: SO
- Currency: USD
- Date *: May 9, 2024
- Requested On *: May 9, 2024
- Customer *: KRKCONSULT - KRK Consulting Service
- Location *: MAIN - Primary Location
- Contact: Alexey Novikov
- Description: Software
- Notes

A dropdown menu at the bottom shows 'SHIPPING'.

Figure: The Sales Order screen header

You can organize the content of the header by using layout objects. In this step, you will add two new fields of the [Sales Orders](#) (SO301000) form to the Sales Order screen header. Do the following:

1. In the WSDL schema of the [Sales Orders](#) form, which you reviewed in the previous step, find the `OrderSummary` complex type, and the `Date` and `RequestedOn` fields.
2. Open the Update: SO301000 Sales Orders page of the Customization Project Editor. In the **Result Preview** area, find the `add layout "OrderHeader"` instruction.
Notice that each new row of the header corresponds to a layout object.
3. In the **Commands** area of the page, insert the following code, which adds a row with two new fields.

```
update screen SO301000 {
    update container "OrderSummary" {
        update layout "OrderHeader" {
            add layout "OrderHeaderDates" {
                displayName = "OrderHeaderDateRow"
                layout = "Inline"
                add field "Date"
                add field "RequestedOn"
            }
        }
    }
}
```

To place two new fields in one row, you set the `layout` property of the `layout` object to `Inline`.



If you need to add fields from a complex type that is different from the one specified in the `update container` instruction, then in the `add field` instruction specify the name of the complex type and the field name separated by the # sign, for example, `TotalsVATTotals#VATExempt`.

4. Save your changes.

Your commands are applied to the mobile site map. If any errors have occurred, you can see them in the **Errors** area of the page. If your changes have been applied successfully, you can see the updated site map of the mobile app in the **Result Preview** area of the form.

5. Publish the customization project.

6. Check your changes in the mobile app.

The screen header should look as shown in the following screenshot.

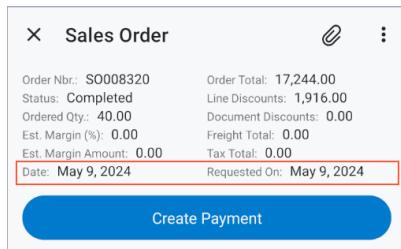


Figure: Fields added to the Sales Order screen header

Step 3: Remove Fields from the Screen

In this step, you will remove some of the fields that are displayed in the **Bill-To Info** section of the **Settings** tab.



Be cautious when removing fields from the original mobile site map. Do not remove fields that are marked with an asterisk and thus are required for a record.

Do the following:

1. In the navigation pane of the Customization Project Editor, click **Mobile Application > Update SO301000**.

The Update: SO301000 Sales Orders page opens.

2. In the **Result Preview** area of the page, find the `add group "BillToInfoGroup"` instruction.

Learn in which objects the instruction is nested and what fields it contains. You will remove the `AddressesBillToAddress#AddressLine2` field. Notice that the name of the field is composed of two parts separated with #: the name of the `AddressesBillToAddress` complex type, and the name of the field itself. You can find the name of the complex type and the name of the field in the WSDL schema.

```
<s:complexType name="AddressesBillToAddress">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="DisplayName" type="s:string"/>
    <s:element minOccurs="0" maxOccurs="1" name="OverrideAddress" type="tns:Field"/>
    <s:element minOccurs="0" maxOccurs="1" name="AddressLine1" type="tns:Field"/>
    <s:element minOccurs="0" maxOccurs="1" name="AddressLine2" type="tns:Field"/>
    <s:element minOccurs="0" maxOccurs="1" name="City" type="tns:Field"/>
    <s:element minOccurs="0" maxOccurs="1" name="Country" type="tns:Field"/>
    <s:element minOccurs="0" maxOccurs="1" name="State" type="tns:Field"/>
    <s:element minOccurs="0" maxOccurs="1" name="PostalCode" type="tns:Field"/>
    <s:element minOccurs="0" maxOccurs="1" name="Validated" type="tns:Field"/>
    <s:element minOccurs="0" maxOccurs="1" name="ServiceCommands" type="tns:AddressesBillToAddressServiceCommands"/>
  </s:sequence>
</s:complexType>
```

Figure: Part of the WSDL schema of the Sales Orders form

3. In the **Commands** area of the page, add the following code, which removes the `AddressLine2` field.

```

update screen SO301000 {
    update container "OrderSummary" {
        ...
        update layout "OrderSettingsTab" {
            update group "BillToInfoGroup" {
                remove field "AddressesBillToAddress#AddressLine2"
            }
        }
    }
}

```

4. Save your changes.

Your commands are applied to the mobile site map. If any errors have occurred, you can see them in the **Errors** area of the page. If your changes have been applied successfully, you can see the updated site map of the mobile app in the **Result Preview** area of the form.

5. Publish your customization project.

6. To check your changes in the mobile app, do the following:

- a. In the **Sales Orders** workspace of the mobile app, open the Sales Orders screen.
- b. In the Sales Orders screen, open any sales order.
The Sales Order screen opens.
- c. On the **Settings** tab, find the **Bill-To Info** section.
- d. In the **Bill-To Info** section, make sure that the **Address Line 2** field is not displayed.

Lesson Summary

In this lesson, you have learned how to add, update, and remove different elements of a mobile app screen.

Related Links

- [layout](#)
- [group](#)
- [remove](#)

Lesson 4.2: Configure the Attachment Capabilities of a Screen

You can configure a screen so that a user can attach a file to a record displayed on the screen. All screens that are included in the basic functionality of the mobile app have the instructions to allow the attachment capabilities added to their screen code. This means that a user can attach any type of files specified on the [File Upload Preferences](#) (SM202550) form.

If you want to narrow the list of types that may be attached to the record, you need to configure the attachment types in the customization of the mobile app. In this lesson, you will modify the Sales Order screen to accept only files with the jpg and png extensions.

Step: Configure the Attachment Capabilities of a Screen

To configure the attachment capabilities of a screen, do the following:

1. In the navigation pane of the Customization Project Editor, click **Mobile Application > Update SO301000**.
2. Explore the **Result Preview** area of the page, and find the attachments instruction.

The screen code includes several attachments instructions, each corresponding to its container.

You need to modify the `attachments` instruction in the `OrderSummary` container, which corresponds to the Summary view of the [Sales Orders](#) (SO301000) form. This container's Attachment button is displayed on the screen toolbar, as shown in the following screenshot.



Figure: The Attachment button on the Sales Order screen toolbar

3. In the **Commands** area of the page, update the `attachments` instruction as shown in the following code.

```
update screen SO301000 {
    update container "OrderSummary" {
        ...
        attachments {
            add type "jpg" {
                extension = "jpg"
            }
            add type "png" {
                extension = "png"
            }
        }
    }
}
```

4. Save your changes.

Notice that in the **Result Preview** area, the `attachments` instruction has changed.

5. Publish your customization project.
6. Open the mobile app, and navigate to the Sales Order screen.
7. Try to attach files of different types by tapping the Attachment button.

If you attempt to attach files of types other than `jpg` or `png`, you will get an error message.

Lesson Summary

In this lesson, you have learned how to configure the types of files that can be attached in the mobile app.

Related Links

- [Configuring Attachments](#)
- [attachments](#)

Lesson 4.3: Configure the Ability to Enhance and Attach Receipts

Users often need to attach to entities not only images or photos but also receipts. Photos of receipts taken from a mobile app camera, however, are usually hard to read, and not all details can be viewed. To address this problem, the Acumatica mobile app provides image enhancement capabilities that are designed for documents with small print, such as receipts. With these capabilities in use, the mobile app enhances images taken from the camera of a mobile device, making the images look clearer and more readable. You can configure particular screens to offer this functionality.

In this lesson, you will configure the Sales Order screen to have the functionality that allows a user to enhance and attach receipts.

Step: Configure the Enhancing and Attaching of Receipts

To configure the enhancing and attaching of receipts, do the following:

1. In the navigation pane of the Customization Project Editor, click **Mobile Application > Update SO301000**. You will modify the same attachments instruction of the OrderSummary container as was modified in the previous lesson.
2. In the **Commands** area of the page, update the attachments instructions as shown in the following code block.

```
update screen SO301000 {
    update container "OrderSummary" {
        ...
        attachments {
            imageAdjustmentPreset = Receipt
        }
    }
}
```

When the `imageAdjustmentPreset` attribute is set to `Receipt` for a screen, a special camera mode that enhances captured receipts is switched on in the Acumatica mobile app for the specified screen. In this mode, when the user captures a receipt, the mobile app automatically enhances the image as follows:

- The image is cropped by the bounding box of the detected edges.
- Any image distortion is removed.
- The image is converted into black and white.
- The contrast of the image is maximized.

The user can make manual adjustments to the image, such as rotating and resizing. The automatic changes cannot be undone.

3. Save your changes.

Notice that in the **Result Preview** area, the `attachments` instruction has changed.

4. Publish your customization project.
5. Open the mobile app, and navigate to the Sales Order screen.
6. Tap the Attachment button on the screen toolbar and then tap the Camera icon. Explore the enhancement functionality (see the screenshots below).

The first screenshot shows the preview of a receipt. If you click  (the image capture button), the mobile app captures the image and improves it (as you can see in the second screenshot) by using the image enhancement functionality. You can manually adjust the size of the captured area by dragging the blue dots. By clicking the palette icon (Item 1 in the second screenshot), you can open enhancement mode, which is shown in the third screenshot. By clicking the check mark icon (Item 2 in the second screenshot), you save your changes.

In the third screenshot, you can see the final (black and white) version of the receipt.



Figure: The enhancement mode functionality

After you perform all the needed modifications to the image and click the Save button, the enhanced image is attached to the sales order.

Lesson Summary

In this lesson, you have learned what the enhanced mode of the Acumatica mobile app is and how to configure it in the mobile app.

Lesson 4.4: Configure a Report

In this lesson, you will configure the viewing and downloading of the Sales Order report from the Sales Order screen. Before you start customizing the mobile app, make sure that this report form already exists in Acumatica ERP.



Users cannot use this functionality if they do not have appropriate access rights to the report.

Step: Configure the Report in the Mobile App

To configure the report, do the following:

1. On the **Sales Orders** (SO301000) form, select a sales order, and on the More menu (under **Printing and Emailing**), click **Print Sales Order**.

The report that corresponds to the form is shown below, along with the menu command that causes the system to open the report. Make a note of the ID of the report: SO641010.

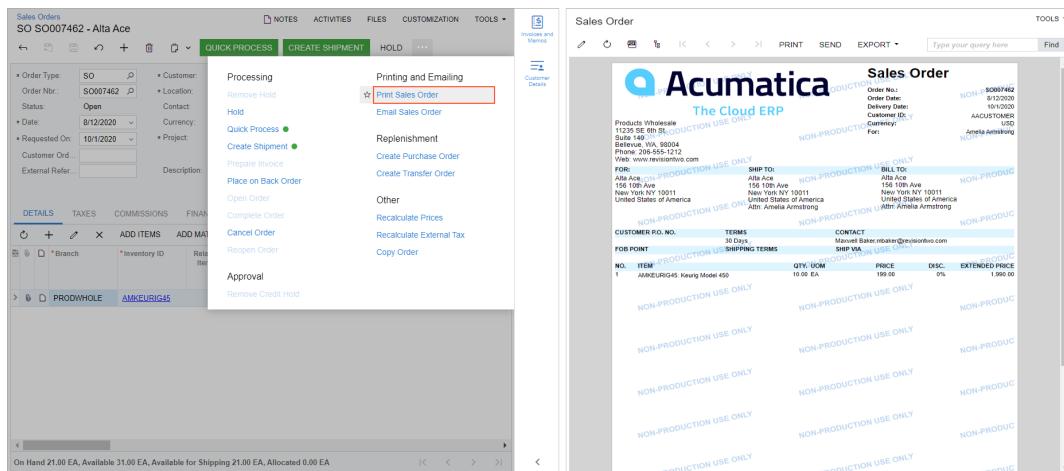


Figure: The Sales Order report

2. Go back to the **Sales Orders** form and open its WSDL schema.
3. In the WSDL schema, find the action that generates the report, which is shown below.

This action is located in the **Actions** complex type of the WSDL schema.

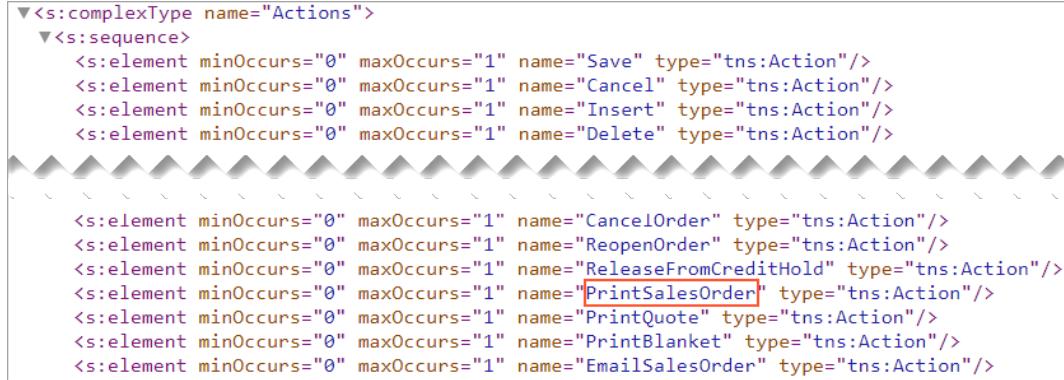


Figure: The Print Sales Order action in the WSDL schema

4. In the navigation pane of the Customization Project Editor, click **Mobile Application > Update SO301000**. The Update: SO301000 Sales Orders page opens.
5. In the **Commands** area of the page, add the following code, which maps the action to generate the report.

```
update screen SO301000 {
    update container "OrderSummary" {
        ...
        add recordAction "PrintSalesOrder" {
            redirect = True
        }
    }
}
```



Because the action is performed on a record, the `recordAction` object is used.

In the mobile app, this action will redirect the user to the page with the report itself, as it does in the browser version of Acumatica ERP. If you published your customization project now, you would see that a context menu appears on the Sales Order screen with the **Print Sales Order** item. However, you will not publish the project yet, because the report screen has not been added to the mobile site map.

- Add the SO641010 report screen to the mobile site map, as described in [Step 2.2.2: Add a Screen](#) and [Step 2.2.3: Update the Mobile Site Map](#), taking the following recommendations into consideration:

- When adding the screen, specify the `type` attribute of the screen as *Report*, as shown below.

```
add screen SO641010 {
    type = Report
}
```

- When modifying the mobile site map, make the screen with the report invisible from the Home screen as shown below.

```
add item "SO641010" {
    visible = False
}
```

- Publish your customization project.

- Open the mobile app, and navigate to the Sales Order screen.

On the screen menu, make sure that you can see the new menu command (**Print Sales Order**) as shown in the screenshot below.

- Tap **Print Sales Order**.

The generated printed document opens as shown in the following screenshot.

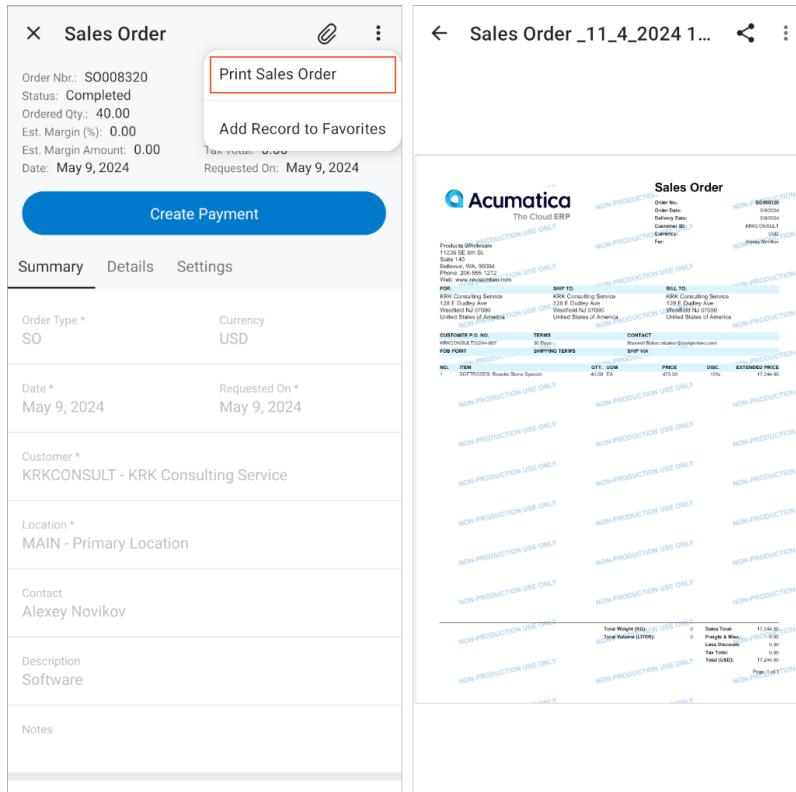


Figure: Opening the Sales Order report

- Download the report: Open the page menu and tap **Open With**.

The PDF version of the report is downloaded to your mobile device and the PDF version is displayed on the screen as shown in the following screenshot.



Figure: The PDF version of the report

Lesson Summary

In this lesson, you have learned how to set up a report corresponding to a screen in the mobile app by mapping an action that generates a report, and mapping the page with the report.

Related Links

- [recordAction](#)

Lesson 4.5: Configure the Signature Capabilities

In the Acumatica mobile app, you can add the signature functionality to any screen that supports attachments. In this lesson, you will configure the Sales Order screen so that a user will be able to sign a sales order by using the touch screen of the device.

Adding a signature is an action performed on a record, so to add this functionality to the screen, you will add the `recordAction` object to the `OrderSummary` container.

Step: Configure the Signature Capabilities

To configure the signature capabilities, do the following:

1. In the navigation pane of the Customization Project Editor, click **Mobile Application > Update SO301000**.
The Update: SO301000 Sales Orders page opens.

2. Update the OrderSummary container to the code shown below.

```
update screen SO301000 {
    update container "OrderSummary" {
        ...
        add recordAction "SignReport" {
            behavior = SignReport
            displayName = "Sign"
        }
    }
}
```

3. Save your changes.
4. Publish your customization project.
5. Open the mobile app, and navigate to the Sales Order screen.

On the screen menu, you can see the new menu command (**Sign**), as shown in the following screenshot.

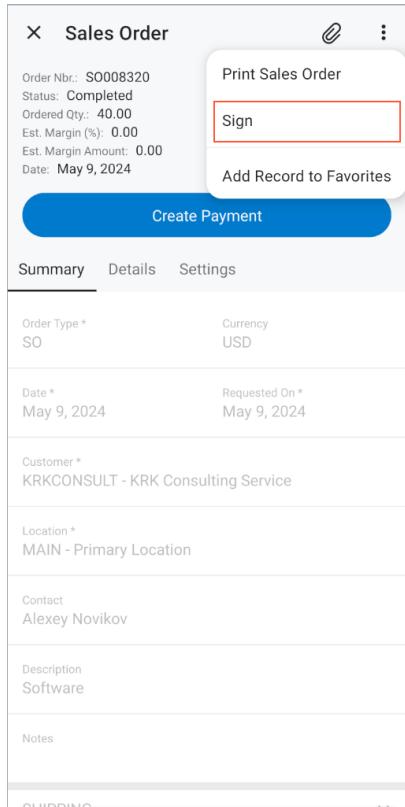


Figure: The Sign menu command

6. Tap **Sign**.

The app displays a blank pop-up with the **Cancel** and **OK** buttons and the words *Sign here*. The user then adds the signature, as shown in the following screenshot.



Figure: Creation of a signature

7. Sign on the screen and tap **OK**.

The signature is added to the list of attached items and saved automatically. A user can add multiple signatures to one record by attaching multiple signatures.

Lesson Summary

In this lesson, you have learned how to configure the signature functionality on a mobile screen and use this functionality.

Lesson 4.6: Map an Action

You can use the [Actions](#) page of the Customization Project Editor to map an action on a form to the mobile app screen.

In this lesson, you will map the **Lock Budget** and **Unlock Budget** actions of the [Projects](#) (PM301000) form to the Projects screen of the mobile app.



The screen where you are mapping the action should also be mapped to the Acumatica mobile app. If the screen has not already been mapped, you should map it, as described in [Lesson 2.2: Add a Screen to the Mobile App](#). In this lesson, it is not necessary to map the Projects screen because it has already been mapped to the Acumatica mobile app by Acumatica developers.

Process of Mapping an Action to a Mobile Screen

The entire process of mapping an action to the mobile screen consists of the following actions:

1. You make sure the screen is mapped to the mobile app. If not, you map it by using MSDL, as described in [Lesson 2.2: Add a Screen to the Mobile App](#).



You can find out whether the screen is already mapped by clicking **Update Existing Screen** on the [Mobile Application](#) page. If a screen is present in the lookup table of the dialog box, then the mapping for the screen has been defined by Acumatica developers.

2. If you need to see the results of the action in the mobile app, you make sure that all elements of the mobile screen that may be affected by the action are also mapped to the mobile app screen. In the Customization Project Editor, you can do this by analyzing the mapping of the screen in the **Result Preview** area on the Update page, which is accessed from the [Mobile Application](#) page for the form.
3. You learn the internal name of the action by using the Element Inspector on the Acumatica ERP form.
4. In the Customization Project Editor, you select the **Expose to Mobile** check box for the action in the **Actions Properties** dialog box, which you open on the [Actions](#) page.

If the action is not displayed on the [Actions](#) page for the form, you can add it manually by clicking the **Add Existing Action** command on the More menu.



You can also create a new action on the [Actions](#) page and map it by using the **Expose to Mobile** check box. For details, see [Configuring Actions](#).

5. You publish the customization project.



You can also map an action using MSDL. For details, see Lesson 1.1 of the *T410 Advanced Customization of the Mobile App* course.

Before You Proceed

Locking and unlocking of a project budget affects ability to edit the **Original Budgeted Quantity**, **Unit Rate**, and **Original Budgeted Amount** columns on the **Revenue Budget** and **Cost Budget** tabs of the [Projects](#) (PM301000) form. So to be able to test the **Lock Budget** and **Unlock Budget** commands in the mobile app, you need at least one of these tabs of the [Projects](#) form to be mapped in the mobile app. You can map the **Revenue Budget** tab by updating the Projects screen, as described in [Lesson 4.1: Update a Screen](#), and using the following code.

```
update screen PM301000 {
    add container "RevenueBudget" {
        add field "ProjectTask"
        add field "AccountGroup"
        add field "Description"
        add field "OriginalBudgetedQuantity"
        add field "UnitRate"
        add field "OriginalBudgetedAmount"
    }
}
```



To map a tab, you use the same container object as you use to map the Summary area of the form. The object should be located at the same level as the object for the Summary area of the form.

Step 1: Mapping the Lock Budget Action in the Customization Project Editor

To map the **Lock Budget** command of the [Projects](#) (PM301000) form to the Acumatica mobile app, do the following:

1. Find out the internal name of action related to the **Lock Budget** command as follows:
 - a. Open the [Projects](#) form.
 - b. On the More menu, locate the **Lock Budget** command, which is located in the **Budget Operations** category.
 - c. To open the Element Inspector for the command, click Ctrl + Alt and then click the command name.
The **Element Properties** dialog box opens.
 - d. In the dialog box, find the internal name of the **Lock Budget** command in the **Action Name** box:
LockBudget.
2. Add the form to the customization project as follows:
 - a. Open the [SalesActivities](#) customization project in the Customization Project Editor.
 - b. In the navigation pane, click **Screens**.
The [Customized Screens](#) page opens.
 - c. On the page toolbar, click **Customize Existing Screen**.
The **Customizing Existing Screen** dialog box opens.
 - d. In the dialog box, select *Projects*. You can open the lookup table and search for the screen by its name or its ID (PM301000).
 - e. Click **OK**.
The Screen Editor: PM301000 (Projects) page opens.

Tip: In the name that appears on the [Screen Editor](#) page, *Screen Editor:* is followed by the screen ID and then the screen name in parentheses.
3. Open the [Actions](#) page for the [Projects](#) form: In the Navigation pane, click **Screens > PM301000 > Actions**.
The PM301000 (Projects) Actions page opens.



In the name that appears on the page, *Actions* is preceded by the screen ID and then the screen name in parentheses.

4. In the table, click the *lockBudget* link in the **Action Name** column.
The **Action Properties** dialog box opens.
5. In the dialog box, select the **Expose to Mobile** check box, as shown in the following screenshot.

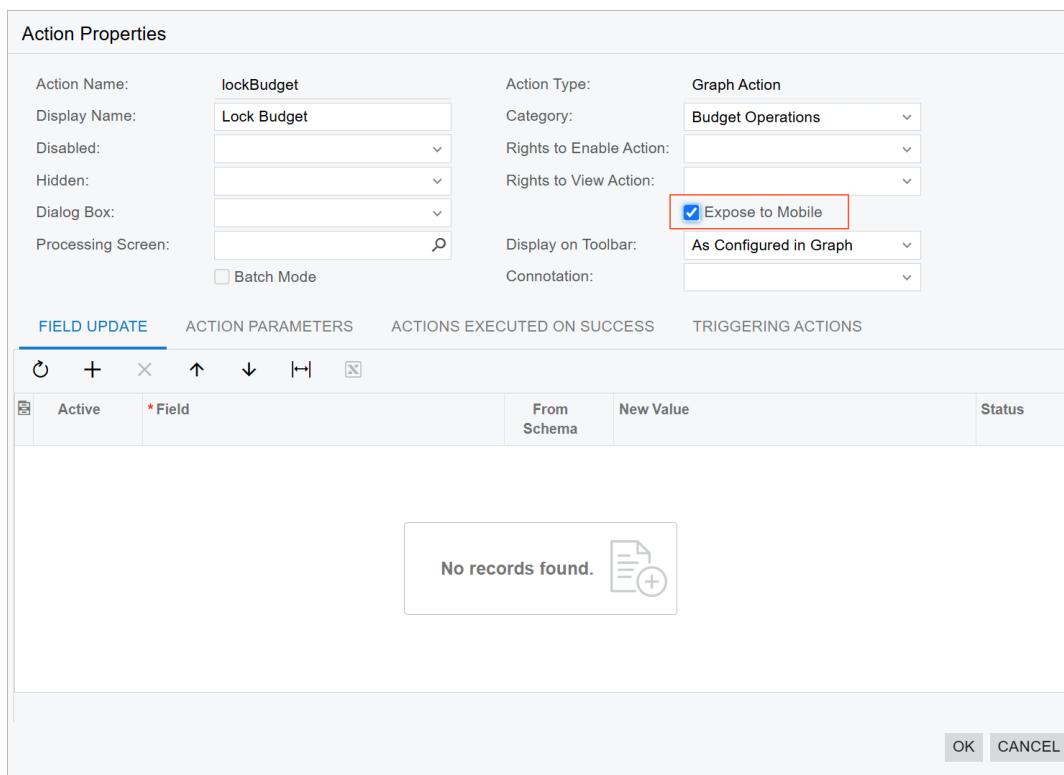


Figure: The Actions Properties dialog box

6. Click **OK** to save your changes and close the dialog box.
The action is mapped to the mobile screen.
7. Publish the customization project.

Step 2: Mapping the Unlock Budget Action in the Customization Project Editor (Self-Guided Exercise)

In this step, you will map the **Unlock Budget** action to the Projects screen of the mobile app as a self-guided exercise by performing the same instructions you did to map the **Lock Budget** action.

On the More menu of the [Projects](#) (PM301000) form, the **Unlock Budget** command is located in the same category as the **Lock Budget** command is.



The **Unlock Budget** command appears in the screen menu only when the project budget is locked.

Step 3: Testing the Mapped Actions

To test the **Lock Budget** and **Unlock Budget** commands in the mobile app, do the following:

1. Open the mobile app, and tap **Projects**.
The Projects workspace opens.
2. In the Projects workspace, tap **Projects**.
The Projects list view opens.
3. Open the **BUDGETBYM** project.

In the form view, notice the **Revenue Budget** tab; also, on the screen menu, notice the new menu command (**Lock Budget**). (See the following screenshot.)



The **Lock Budget** command may not be visible for some projects because locking of the project budget is not allowed, based on the project status. The **Unlock Budget** command is not visible for this record because the project budget is not locked.

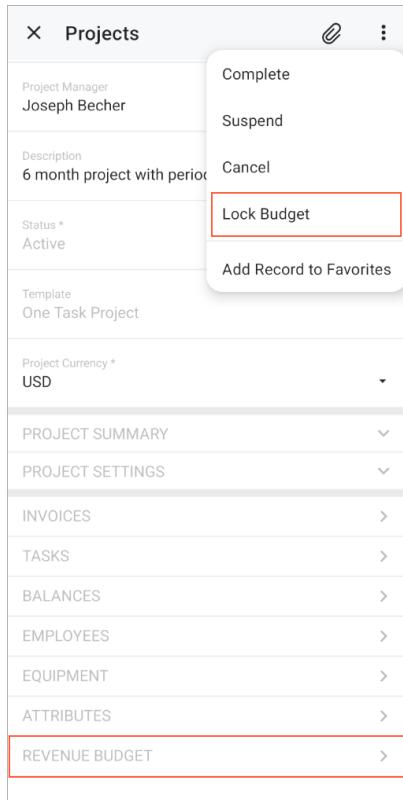


Figure: The Projects screen menu

4. Make sure the editing of the project budget is available as follows:
 - a. On the Projects screen, tap **Revenue Budget**.
 - The Revenue Budget screen opens with the list of balances.
 - b. Tap a line.
 - c. Make sure you can edit the **Original Budgeted Quantity** box, as shown in the following screenshot.

The screenshot shows the 'Revenue Budget' screen with the following fields:

- Project Task ***: 01
- Account Group ***: REVENUE
- Description**: 01 Main Task
- Original Budgeted Quantity**: 6.00 (highlighted with a blue border)
- Unit Rate**: 1,000.00
- Original Budgeted Amount**: 6,000.00

Below the form is a numeric keypad with a pink delete key.

Figure: The Original Budgeted Quantity box, which can be edited

5. In the **Original Budgeted Quantity** box, enter 7, and tap **Update** on the screen toolbar to save your changes.

The Revenue Budget screen opens with all affected values updated, as shown in the following screenshot.

The screenshot shows the 'Revenue Budget' screen with the following data for the first row:

Project Task	Account Group	Original Budgeted Quantity	Original Budgeted Amount
01 01 Main Task 1,000.00	REVENUE	7.00	7,000.00

Figure: The settings of the affected line

6. Return to the Projects screen and tap **Save** on the screen toolbar.
7. On the screen menu, tap **Lock Budget**.
8. On the Revenue Budget screen, open the edited line.
9. Make sure that the **Original Budgeted Quantity**, **Unit Rate**, and **Original Budgeted Amount** boxes are unavailable for editing, as shown in the following screenshot.

The screenshot shows a mobile application interface for a 'Revenue Budget' entry. At the top, there's a header bar with an 'X' icon, the title 'Revenue Budget', and a save/cancel icon. Below the header, there are three fields: 'Project Task *' with value '01', 'Account Group *' with value 'REVENUE', and 'Description' with value '01 Main Task'. A large red rectangular box highlights the following three fields: 'Original Budgeted Quantity' with value '7.00', 'Unit Rate' with value '1,000.00', and 'Original Budgeted Amount' with value '7,000.00'. The rest of the screen is mostly blank.

Figure: The boxes that are unavailable for editing

10. Return to the Projects screen, and, on the screen menu, tap the **Unlock Budget** command, which is now available because the project budget has been locked.
11. On the Revenue Budget screen, open a line.
12. Make sure the editing of line details is available again.

Lesson Summary

In this lesson, you have learned how to use the Customization Project Editor to map an action to a mobile form.