

MYOB Advanced

Customising the Mobile App - Basic

Last Updated: 23 October 2019

myob

Contents

Introduction	1
Document Objectives	1
Preparing the Environment	2
Required Role for Users.....	2
Preparation of a Mobile Device	2
Part 1: Introduction to the MYOB Advanced Mobile App	3
Lesson 1.1: Installing the MYOB Advanced Mobile App.....	3
Installing MYOB Advanced Mobile App for Android.....	3
Installing MYOB Advanced Mobile App for iOS.....	3
Lesson Summary	4
Lesson 1.2: Signing in to the MYOB Advanced Mobile App	4
Signing in to the MYOB Advanced Mobile App.....	4
Exploring the MYOB Advanced Mobile App	4
Lesson Summary	5
Part 2: Tools and Languages	6
Lesson 2.1: Exploring a WSDL Schema	7
About the WSDL Schema.....	7
To Explore the WSDL Schema.....	7
Lesson Summary	9
Lesson 2.2: Adding a Screen to the Mobile App.....	10
Step 2.2.1: Explore the Mobile Application Page.....	10
Step 2.2.2: Add a Screen	11
Step 2.2.3: Update the Main Menu	12
Updating the Main Menu.....	12

Step 2.2.4: Check the Added Functionality in the Mobile App	13
To Check Your Changes in the Mobile App.....	13
Lesson Summary	15
Part 3: Configuring Dashboards and Generic Inquiries	15
Types of Generic Inquiries	15
Preliminary Steps.....	16
Lesson 3.1: Configuring a Dashboard Screen	17
To Configure a Dashboard Screen	17
Lesson Summary	18
Lesson 3.2: Adding a Generic Inquiry with Parameters	19
To Add a Generic Inquiry with Parameters	19
Lesson Summary	21
Practice Exercise	21
Mapping the Inquiry.....	21
Lesson 3.3: Adding a Generic Inquiry from the Instance.....	23
To Map a Generic Inquiry from the Instance.....	23
Lesson Summary	24
Part 4: Configuring Extended Functionality	25
Lesson 4.1: Updating a Screen	26
Step 1: Preparing to Update the Screen.....	26
Step 2: Updating the Header Container	26
Step 3: Removing Fields from the Screen.....	27
Lesson Summary	28
Lesson 4.2: Configuring Attachment Capabilities of a Screen.....	29
To Configure Attachment Capabilities of a Screen.....	29
Lesson Summary	30

Lesson 4.3: Configuring the Ability to Enhance and Attach Receipts.....	30
To Configure the Enhancing and Attaching of Receipts	30
Lesson Summary	31
Lesson 4.4: Configuring Reports.....	32
To Configure Reports in the Mobile App	32
Lesson Summary	33
Test Use Case 1: Downloading a Report in the Mobile App	34
Lesson 4.5: Configuring the Signature Capabilities.....	36
To Configure the Signature Capabilities	36
Lesson Summary	37

Introduction

This document introduces the basic functionality for customising the MYOB Advanced mobile app by using the MYOB Advanced Customisation Platform.

This document is intended for business and implementation consultants who are learning how to customise the MYOB Advanced mobile app. The document is based on an end-to-end customisation task that demonstrates the general approach to customising the MYOB Advanced mobile app. The document consists of four parts, which take a total of about 6 hours to complete.

Before starting to work with the document, you should prepare the environment needed for performing the customisation and final testing, as described in “Preparing the Environment”.

Document Objectives

After you complete the document, you will have an understanding of how to perform simple customisation tasks of the MYOB Advanced mobile app. Upon completion of the document, you will have learned how to perform the following types of customisation activity:

- Using the customisation Customisation Project Editor to customise the mobile app
- Adding general inquiries and dashboards to the mobile app
- Configuring additional functionality of the mobile app screens to allow users to do the following:
 - Adding signatures to a record
 - Scanning receipts
 - Generating reports
 - Attach files to a record
- Testing the customised mobile app

Preparing the Environment

In this training, you will learn how to customise the MYOB Advanced mobile app. Before you start the practical steps of the document, you have to prepare the development environment.

Required Role for Users

Specialists who will work on customisation projects should be assigned the *Customiser* role in the application instance that is to be customised and tested, as well as in the production application that should be updated with the customisation project. With this role assigned, the specialists can use the customisation tools and facilities of the MYOB Advanced Customisation Platform, and upload and publish customisation projects.

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

Preparation of a Mobile Device

To test a customised mobile app, you will have to install an MYOB Advanced mobile app to a mobile device that is connected to the same local network as the computer hosting the development environment of your MYOB Advanced instance.

Part 1: Introduction to the MYOB Advanced Mobile App

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

Lesson 1.1: Installing the MYOB Advanced Mobile App

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

- Android
- iOS

Installing MYOB Advanced Mobile App for Android

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

- Android 4.0 or higher
- Sufficient free disk space (the amount depends on the particular device)

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

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

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

Installing MYOB Advanced Mobile App for iOS

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

- iOS 10.0 and higher
- 15.7 MB of free disk space

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

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

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

Lesson Summary

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

Lesson 1.2: Signing in to the MYOB Advanced Mobile App

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

Note: The instructions in this guide are designed for the MYOB Advanced Mobile App for Android. Instructions of the MYOB Advanced Mobile App for iOS may differ.

Signing in to the MYOB Advanced Mobile App

To use the MYOB Advanced mobile app, you need to sign in to it by using the MYOB Advanced instance URL and your credentials. In this training document, you can use the URL and credentials for the instance you prepared as described in “Preparing the Environment” on page 2.

Do the following to sign in to the MYOB Advanced mobile app:

1. Open the MYOB Advanced mobile app.
2. In the **Server URL** section, select the version of the protocol and type address to the MYOB Advanced instance deployed in your network.
3. Tap **Next**.
4. In the **Username** field, enter your username.
5. In the **Password** field, enter your password.
6. Tap **Sign In**.

The app connects to your MYOB Advanced instance and opens the main screen of the app.

Before the mobile app is customised, it contains basic functionality and includes several screens of MYOB Advanced.

Exploring the MYOB Advanced Mobile App

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

- The screen toolbar, which includes the name of the tenant (Item 1 on the following screenshot) and a button to open the sidebar menu (2). To select a company or a branch, tap the name of the tenant.
- The main menu of the app, which consists of tiles. If you tap a tile, you can navigate to a screen (which corresponds to an MYOB Advanced form) or to a folder that groups screens.

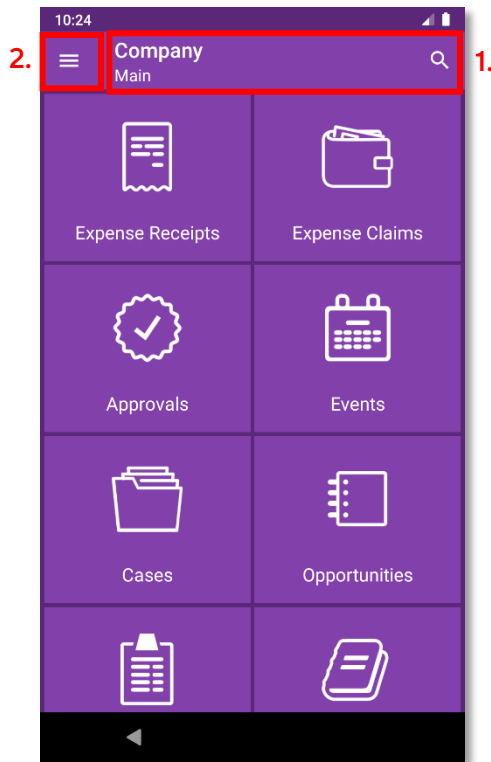


Figure: The MYOB Advanced Mobile App main screen

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

The other major part of the mobile app is the sidebar menu, which contains the following:

- Information about the accounts that you used to sign in the app. You can switch between accounts by selecting any of them in the sidebar menu.
- Links to screens marked as favourites.
- The About screen of the mobile app, where you can learn the version of the app and other service information.

Lesson Summary

In this lesson, you 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 customise the MYOB Advanced mobile app by using the example of the DB Sales Activities (GI000023) generic inquiry. You will use the Mobile Application page of the Customisation Project Editor and a special language, MSDL.

Mobile Application Page of the Customisation Project Editor

The Mobile Application page of the Customisation Project Editor is designed to manage the customisation of screens in the **MYOB Advanced** mobile app. On the page, you can add, update, or remove screens; you can also remove the customisation for a tenant or all tenants without removing the customisation code.

MSDL

An MYOB Advanced mobile client application uses the Mobile API to access the data of the forms that are mapped for mobile apps in the MYOB Advanced 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 MYOB Advanced on your mobile device if the mobile site map includes the metadata for the form.

The metadata of the form is described using Mobile Site Map Definition Language (MSDL). Configuring the user interface of the mobile app 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. Inside an instruction there can be commands that assign values

to the various attributes of the instruction. For more information about the MSDL syntax, see “MSDL reference” in the MYOB Advanced Help.

Lesson 2.1: Exploring 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 MYOB Advanced.

About the WSDL Schema

The WSDL schema of an MYOB Advanced 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 a mobile screen that corresponds to an MYOB Advanced form.

To Explore the WSDL Schema

1. In MYOB Advanced, open the DB Sales Activities (GI000023) generic inquiry form.
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 (GI000023) generic inquiry, shown below, opens.

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

```
<?xml version="1.0" encoding="UTF-8" ?>
<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="InProgress"/>
          <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:element minOccurs="0" maxOccurs="1" default="false" name="IgnoreError" type="s:boolean"/>
          <s:element minOccurs="0" maxOccurs="1" name="LinkedCommand" type="tns:Command"/>
        </s:sequence>
      </s:complexType>
    </s:schema>
  </wsdl:types>

```

Figure: The WSDL Schema of the Sales Activities generic inquiry

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

Find the "Result" complex type and explore its contents. The Result complex type of the generic inquiry's WSDL schema always corresponds the grid with the results of the generic inquiry. We will use this part of the schema later in Lesson 4 to add fields to the mobile app screen.

```

▼<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

Lesson Summary

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

Lesson 2.2: Adding 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:

1. Creating a customisation project. You will do this only once. After that, you will continue customising your mobile app in the same customisation project.
2. Adding a new screen corresponding to this generic inquiry to the mobile site map.
3. Declare this screen in the main menu of the mobile app.

Step 2.2.1: Explore the Mobile Application Page

The customisation of the MYOB Advanced mobile app is performed on the Mobile Application page of the Customisation Project Editor. All changes made on that page are saved as a part of the customisation project. Thus, to begin customising a mobile app, you should first create a customisation project or proceed in the one you already have. In this lesson, you will create a new customisation project and find the page where you will perform all changes to the mobile site map.

Exploring the Mobile Application Page of the Customisation Project Editor

1. Create a new customisation project.

Note: To create a customisation project, on the Publish Customisation (SM204505) form, add a row, enter project properties, and save your changes.

2. Open the created project in the Customisation Project Editor by selecting the created project's name in the table on the Customisation Projects form.
3. In the navigation pane of the Customisation Project Editor, click **Mobile Application**. The Mobile Application page, shown in the following screenshot, opens.

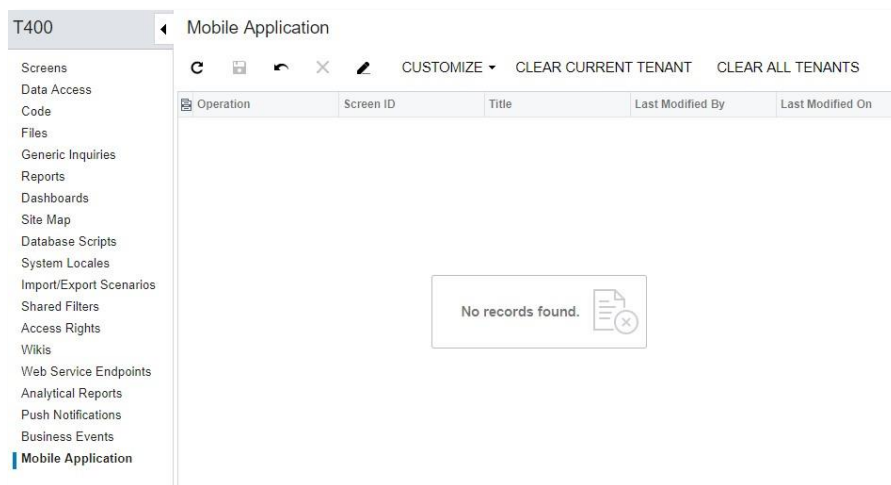


Figure: The Mobile Application page

The Mobile Application page includes the following parts:

- A toolbar
- A table with the list of added and customised screens

When you add a new customisation of a screen, the name of the screen also appears in the navigation pane of the Customisation Project Editor, under the **Mobile Application** node.

Step 2.2.2: Add a Screen

One of the ways to customise 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.

Adding a Screen to the Mobile App

1. In the Customisation Project Editor, open the Mobile Application page.
2. On the toolbar of the Mobile Application page, click **Customise > Add New Screen**.
3. In the **Add New Screen** dialog box, which opens, enter GI000023, and click **OK**.

The Add: GI000023 DB Sales Activities page opens.

The Add: GI000023 DB Sales Activities page 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. Notice that the initial code of the **Commands** area includes the add instruction, 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 3). The "Result" container always corresponds to the table with the results of the inquiry.

6. Save your changes.

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 customisation project has not been published.

Note: The `attachments` instruction is automatically added to the page code. Thus, the attachment functionality can be applied to this type of container although the functionality does not work yet. You will explore the attachment functionality in Lesson 9.

Step 2.2.3: Update the Main Menu

In this lesson you will update the main menu of the mobile app by adding information about the new screen from “Step 2.2.2: Add a Screen”.

Note: Adding a screen to the main menu is a mandatory step when you add a screen to the mobile app.

Updating the Main Menu

1. In the Customisation Project Editor, open the Mobile Application page.
2. On the page toolbar of the Mobile Application page, click **Customise > Update Main Menu**. The Update: MENU page opens.
3. In the **Result Preview** area of the Update: MENU page, explore the original code of the main menu.
4. In the **Commands** area of the Update: MENU page, insert the following code that describes the new screen.

```
update sitemap {
  add item "GI000023" {
    displayName = "Sales Activities"
    icon = "system://Box1"
  }
}
```

In the code above, you update mobile site map by adding an item that corresponds to the screen you added in the previous step. For the item, you specify the name of the main menu tile (the `displayName` attribute) and an icon of the tile (the `icon` attribute). For possible values of the main menu icons, see “Icons” in the MYOB Advanced Help.

1. Save your changes.
2. Publish your customisation project.

Note: To publish a customisation project, in Customisation Project Editor menu, select **Publish > Publish Current Project**.

Step 2.2.4: Check the Added Functionality in the Mobile App

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

To Check Your Changes in the Mobile App

1. On your mobile device, open MYOB Advanced 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 main menu, scroll until you see the **Sales Activities** tile, which is shown in the following screenshot.

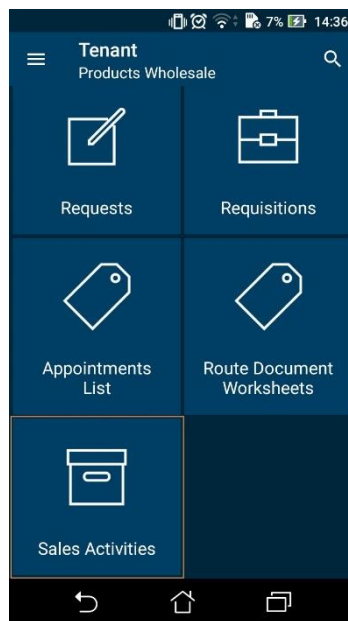


Figure: The Sales Activities tile on the main menu of the mobile app

3. Tap the tile.

The Sales Activities screen, shown on the following screenshot, is opened.

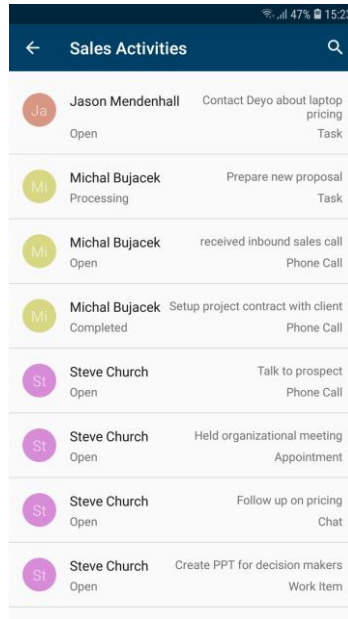


Figure: The Sales Activities screen

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 example.)

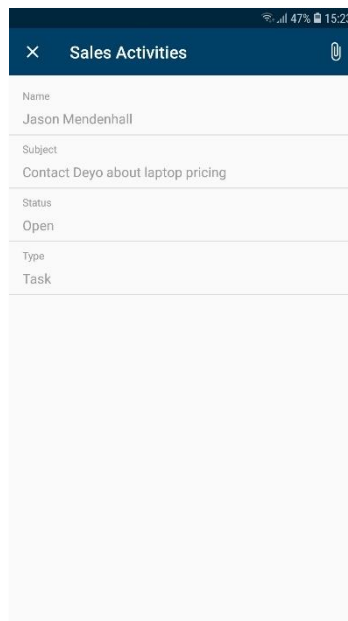


Figure: The screen with the details of a record

Lesson Summary

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

- Installed the MYOB Advanced mobile app on your mobile device
- Signed in to the MYOB Advanced mobile app by using the credentials of your MYOB Advanced instance
- Explored the WSDL schema of the DB Sales Activities (GI000023) generic inquiry form
- Created a new customisation project in your instance of MYOB Advanced
- Added a new screen corresponding to the DB Sales Activities (GI000023) generic inquiry form to the mobile site map and declared it in the app's main menu
- Tested the added screen in the mobile app

Part 3: Configuring Dashboards and Generic Inquiries

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

Types of Generic Inquiries

In MYOB Advanced, the type of the generic inquiry—that is, whether it has parameters that appear in the Selection area above the Results Grid and narrow the results displayed—determines how it is mapped for the mobile app. There are two types of generic inquiries:

- Without parameters: These generic inquiries contain only the Results Grid, as shown below in the following screenshot. To map such an inquiry, you should declare only one container, `Result`. You will map this type of generic inquiry in Lesson 4.

Owner ID	Owner Description	Created Date	Summary	Activity Details	Activity Class	Type Code	Activity Type	Task ID	Status
EP00000002	Baker Maxwell, Mr.	11/26/2018	TEst Test		Task			c9a7bd2d-6c	Open
EP00000002	Baker Maxwell, Mr.	11/22/2018	Revision Two ...	<html><head ><style>*/ RTE style begin */ol, ul { ...	Email			2cb5d749-6c	Open
EP00000002	Baker Maxwell, Mr.	11/22/2018	Revision Two ...	<html><head ><style>*/ RTE style begin */ol, ul { ...	Email			82359ed9-5f	Open
EP00000002	Baker Maxwell, Mr.	11/22/2018	Revision Two ...	<html><head ><style>*/ RTE style begin */ol, ul { ...	Email			feba9655-3d	Open
EP00000002	Baker Maxwell, Mr.	11/15/2018	Revision Two ...	<html><head ><style>*/ RTE style begin */ol, ul { ...	Email			0ac29eab-f6	Open
EP00000008	Bujacek, Michal	11/7/2018	Consultant task		Activity	W	Work Item	d41ba4f5-70	Completed
EP00000002	Baker Maxwell, Mr.	11/6/2018	Draft invoice # ...	<html><head ><style>*/ RTE style begin */.rich ol...	Email			6a4ef74a-e8	Open
EP00000002	Baker Maxwell, Mr.	11/2/2018	(SERVICE)Dat...		Task			fa284907-ad	Open
EP00000002	Baker Maxwell, Mr.	10/31/2018	Company Sale...	<html><head ></head><body ><p class="richp">...	Email			5e7e59c4-27	Open
EP00000002	Baker Maxwell, Mr.	10/24/2018	TEST		Activity	W	Work Item	c7de472a-87	Completed

- With parameters: These generic inquiries contain the filtering area and the Results Grid, as shown below in the following screenshot. To map such an inquiry, two containers are required: `Filter_` and `Result`. You will map this kind of generic inquiry in Lesson 6.

Open Sales Orders by Customer ★

* Date From: Customer:

* Date To: Open Only

Drag column header here to configure filter

		Date	Customer ID	Customer Name	Order Nbr.	Quantity	Sales Order Total	Status
>	<input type="button" value="U"/>	8/9/2015	INDSUPPLY	Industrial Supply Inc	SO003263	100,634.120000	16,766.99	Completed
	<input type="button" value="U"/>	8/9/2015	SNOSHORT	Shortstop Sports	SO003268	6,970.320000	46,116.00	Completed
	<input type="button" value="U"/>	8/12/2015	ABCSTUDIOS	ABC Studios Inc	SO003272	133.280000	46,924.30	Completed
	<input type="button" value="U"/>	8/12/2015	ELITEANSW	Elite Answering	SO003261	43.520000	11,996.48	Completed
	<input type="button" value="U"/>	8/14/2015	FDIAGRI	Agrilink Food	SO003256	1,288.200000	19,135.92	Completed
	<input type="button" value="U"/>	8/14/2015	BIBIMBAB	Bibimbab Korean Restaurant	SO003269	1,025.200000	4,439.20	Completed

Preliminary Steps

In this part, you will use the Open Sales Orders by Customer generic inquiry form, which was created in the S130 training. If you don't have the inquiry in your MYOB Advanced instance, import it to the instance. To do this, import the `SO-OpenByCustomer.xml.xml` file, which is included in the files of this training document. To import a generic inquiry, on the Generic Inquiry (SM208000) form, select **Clipboard > Import from XML**. In the dialog box that opens, specify the XML file with the generic inquiry, and click **Upload**.

Lesson 3.1: Configuring a Dashboard Screen

In this lesson, to configure a dashboard screen in the mobile app, you will add the Sales Operations (DB000012) dashboard screen with its widgets to the mobile app. Each widget corresponds to a generic inquiry. For details, 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.

To Configure a Dashboard Screen

1. In the browser version of MYOB Advanced, open the Sales Operations (DB000012) dashboard. The page already contains several widgets.
2. On the Sales Operations dashboard, add a widget (of one of the supported types listed above) for each of the following generic inquiry forms:
 - o DB Sales Activities (GI000023)
 - o Open Sales Orders by Customer (GI000082)

Note: To learn how to add widgets, see instructions in the “Configuring Dashboards” section of the Getting Started Guide. While adding a widget, you specify an inquiry's path in the site map. The DB Sales Activity inquiry is located in **Company > Hidden > Hidden Generic Inquiries**, the Open Sales Orders by Customers inquiry is located in **Company > Distribution > Sales Orders > Reports > Audit**.

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

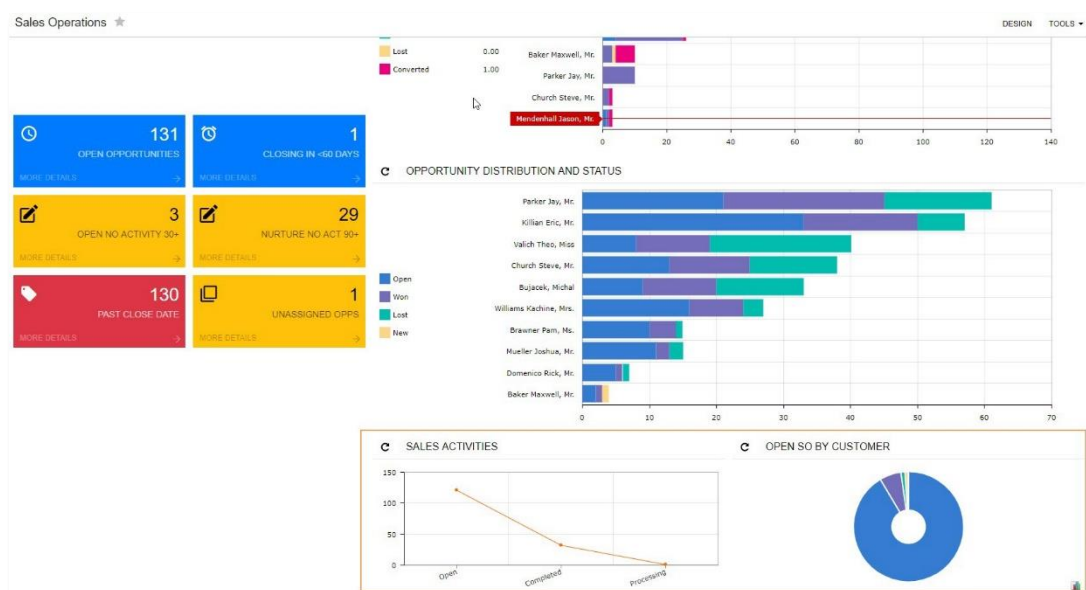


Figure: Added widgets on the Sales Operations dashboard

3. Add the Sales Operations (DB000012) screen to the mobile site map, as described in “Step 2.2.2: Add a Screen” and “Step 2.2.3: Update the Main Menu”. The code of the screen should look the following.

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

The code for the main menu should look as follows.

```
add item "DB000012" {
  displayName = "Sales Operations"
  icon = "system://Graph1"
}
```

4. Save your changes on the Add screen page and on the Update Menu.
5. Publish your customisation project.
6. In the mobile app, open the Sales Operations dashboard screen. Scroll until you see the widgets you added. They should look as follows.

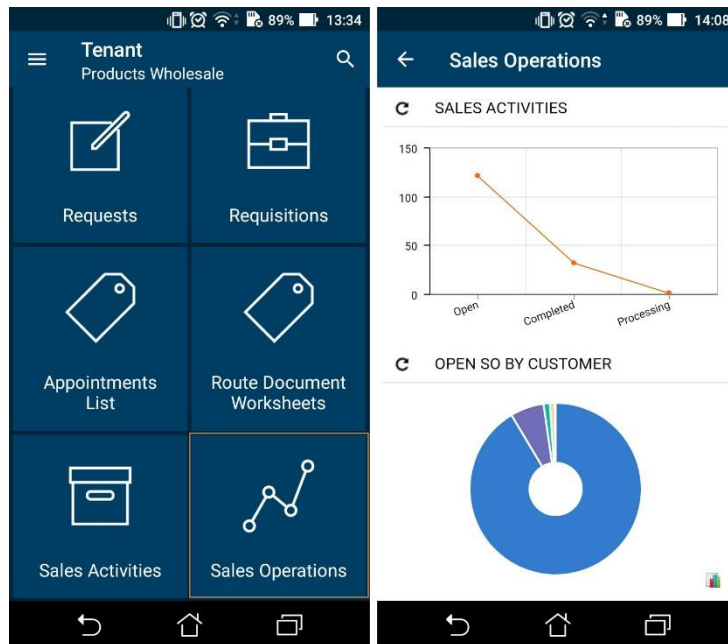


Figure: Opening the Sales Operations dashboard screen

7. On the Sales Operations dashboard screen, tap 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.
8. Tap the *Open Sales Orders by Customer* widget, which shows data from the Open Sales Orders by Customer generic inquiry. On an Android device, the app returns an error because you didn't add the screen corresponding to the Open Sales Orders by Customer generic inquiry to the mobile site map.

Lesson Summary

In this lesson, you learned how to configure a dashboard screen in the mobile app by adding and configuring the Sales Operations dashboard screen.

Lesson 3.2: Adding 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 Preliminary Steps for Part 3).


To Add a Generic Inquiry with Parameters

1. On the Generic Inquiry form (SM208000), open the Open Sales Orders by Customer generic inquiry (GI400001).
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: Exploring a WSDL Schema”.
4. In the Customisation 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 `Results` container, but also the `Filter_` container, as shown below. The `Filter_` container has to include all required parameters of the inquiry.

```
add screen "GI400001" {
  type = FilterListScreen
  add container "Filter_" {
    add field "DateFrom" # required parameter
    add field "DateTo" # required parameter
    add field "Customer" # optional parameter
  }
  add container "Result" {
    add field "CustomerName"
    add field "OrderNbr"
    add field "Quantity"
    add field "SalesOrderTotal"
    add field "Status"
  }
}
```

Note: The `type` attribute of the screen has to be set 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 Main Menu](#). Specify the `displayName` attribute value that fits the mobile app tile and toolbar, such as *Open SO by Customer*.
7. Publish your customisation project.
8. Check the generic inquiry in the mobile app as follows:
 - a. On the main menu, tap the **Open SO by Customer** tile.
 - b. On the screen toolbar, tap the filter button ().

The screen with fields corresponding to the inquiry parameters, shown in the following screenshot, opens.

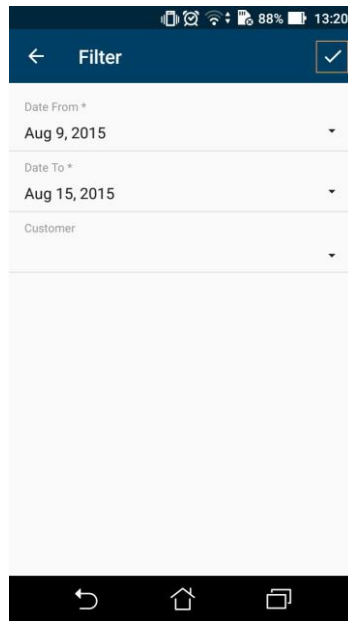


Figure: The screen with the fields that can filter the results

- c. Select different values for the fields or leave the default values. If you have changed anything, tap the Save button on the screen toolbar.
- d. Check the inquiry results.

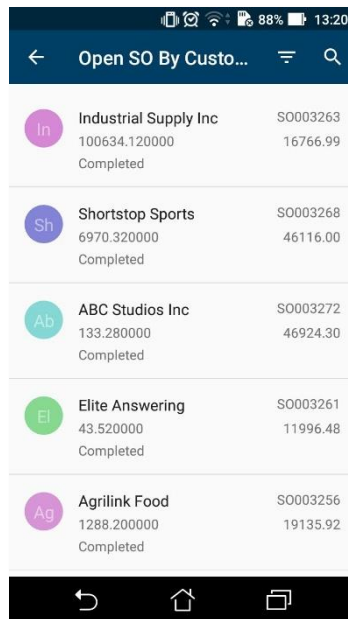


Figure: The inquiry results

9. On the main menu of the mobile app, tap the **Sales Operations** tile to open the view with the dashboard screen you added in the previous lesson, and tap the *Open Sales by Customer* widget. Make sure that you are now redirected to the corresponding screen.

Lesson Summary

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

Practice Exercise

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

Mapping the Inquiry

To map the Invoiced Items generic inquiry, do the following:

1. In MYOB Advanced, open the Invoiced Items (GI000008) generic inquiry form, and explore its WSDL schema. Decide which fields you want to map.
2. In the Customisation Project Editor, open the Mobile Application page, and perform the following actions:
 - a. Add the GI000008 screen to the customisation project, and map the fields you have chosen.
 - b. Include the GI000008 screen in the mobile app's main menu.
 - c. Publish your customisation project.
3. On the main menu of the mobile app, check to see what has changed since you added the generic inquiry. Note that instead of one new tile with the title of the generic inquiry, three new tiles have been added. These tiles correspond to the tabs of the Invoiced Items generic inquiry.

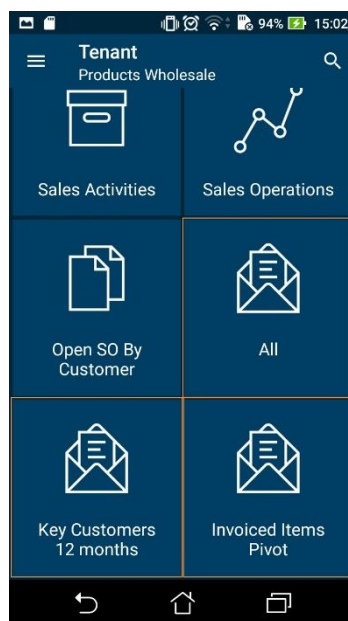


Figure: Tiles of the added inquiry

4. To avoid three tiles being used to represent the generic inquiry, group the tabs: Return to the Customisation Project Editor, and on the Update Main Menu page, replace the code that includes the inquiry in the main menu with the following code.

```

add folder "InvoicedItems" {
  type = HubFolder
  displayName = "Invoiced Items"
  icon = "system://MailOpenFile"

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

```

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

5. After publishing the project again, open the main menu on the mobile app, verify that the **Invoiced Items** tile appears, and tap it to view the Invoiced Items screen. The main menu and the screen should look as follows.

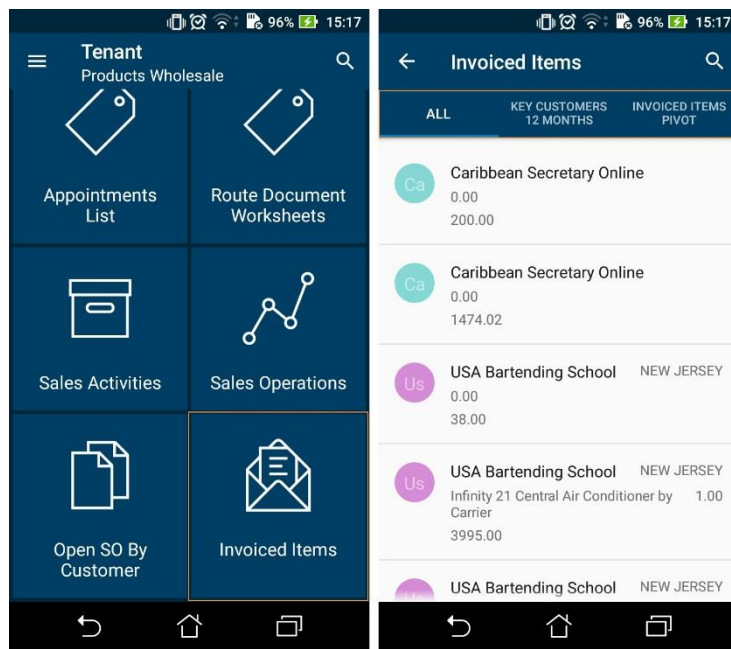


Figure: The Invoiced Items tile and screen

Lesson 3.3: Adding a Generic Inquiry from the Instance

You can map a generic inquiry to the mobile app by using the browser version of MYOB Advanced without using the Customisation Project Editor. In this case, your changes will not be saved in the customisation 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 fields to be displayed, tabs, filters, title, and icon on the main menu.

Note: There are other ways to save the **Expose to Mobile** setting of a generic inquiry on the Generic Inquiry (SM208000) form. First, the **Expose to Mobile** setting of a generic inquiry is included in a customisation project if you export the generic inquiry into the project. Also, the **Expose to Mobile** setting is saved when you import the generic inquiry in XML format.

To Map a Generic Inquiry from the Instance

1. In MYOB Advanced, open the Generic Inquiry (SM208000) form, and select the *Salesperson Sales (G1000003)* generic inquiry.
2. Select the **Expose to Mobile** check box.
3. Save your changes.
4. In the MYOB Advanced mobile app, on the main menu, make sure you see the **Salesperson Sales** tile, shown in the following screenshot.

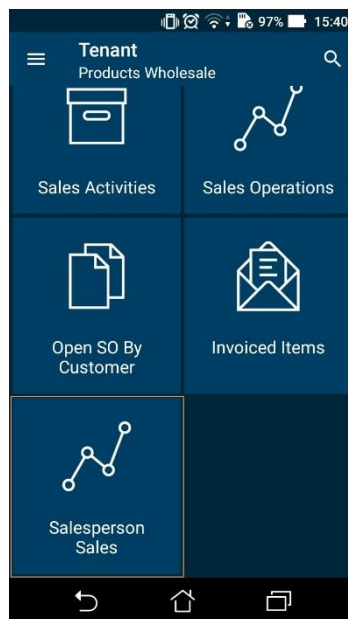


Figure: The Salesperson Sales tile on the main menu

5. Tap the tile.

The screen with the generic inquiry opens:

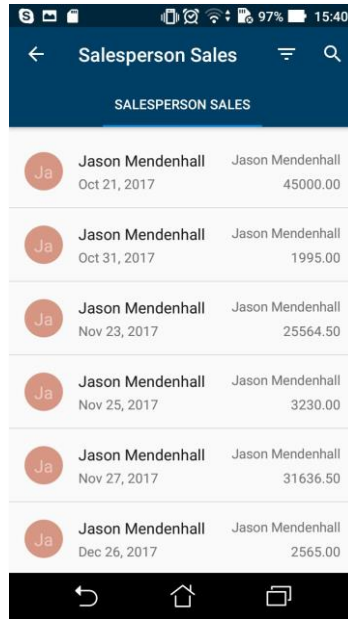


Figure: The Salesperson Sales screen

Note: You don't need to publish a customisation project because the **Expose to Mobile** functionality is not related to a customisation project.

Lesson Summary

In this lesson, you have learned how to add a generic inquiry without using a customisation project.

Part 4: Configuring Extended Functionality

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

- The ability to add an attachment to a screen
- The ability to add a signature to a record
- The ability to scan and format paper receipts
- The ability to generate and download reports

Also, you will learn how to add more UI elements to an existing screen of the mobile app.

In all lessons of this part, you will customise the Sales Orders screen, which corresponds to the Sales Orders form (SO301000).

Lesson 4.1: Updating 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 at different parts of the screen.

Step 1: Preparing to Update the Screen

1. Review the WSDL schema of the Sales Orders screen, as described in “Lesson 2.1: Exploring a WSDL Schema”.
2. In the Customisation Project Editor, open the Mobile Application page.
3. On the page toolbar of the Mobile Application page, select **Customise > Update Existing Screen**.
4. In the Update Existing Screen dialog box, which opens, type SO301000, and click **OK**. The Update: SO301000 Sales Orders page opens.
5. In the **Result Preview** area of the Update page, explore the original code of the screen. Note that the original `add screen` instruction includes containers that correspond to complex types of the screen's WSDL schema and parts of the mobile screen.

Step 2: Updating the Header Container

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

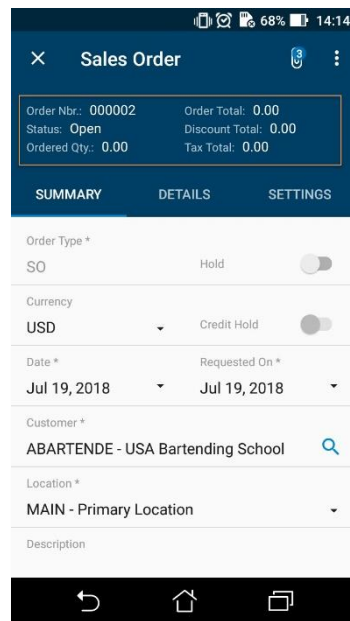


Figure: The Sales Order screen header

You can organise 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.

1. In the WSDL schema of the Sales Orders form, which you reviewed in the previous step, find the `OrderSummary` complex type and the `VATExemptTotal` and `VATTaxableTotal` fields.

- In the **Result Preview** area of the Update: SO301000 page, find the add layout "OrderHeader" instruction. Note that each new row of the header corresponds to a layout object.
- In the **Command** area of the Update: SO301000 page, insert the following code, which adds a row with two new fields.

```
update screen SO301000 {
  update container "OrderSummary" {
    update layout "OrderHeader" {
      add layout "OrderHeaderVATRow" {
        displayName = "OrderHeaderVATRow"
        layout = "Inline"
        add field "VATExemptTotal"
        add field "VATTaxableTotal"
      }
    }
  }
}
```

- Save your changes.

Your commands are applied to the 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 main menu in the **Result Preview** area of the form.

- Publish the customisation project.
- Check your changes in the mobile app. The screen header should look as follows.

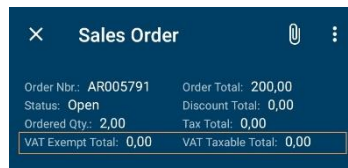


Figure: The modified Sales Order screen header

Step 3: Removing Fields from the Screen

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

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

- In the navigation pane of the Customisation Project Editor, select **Mobile Application > Update SO301000**.
- In the **Result Preview** area of the page, find the add group "BillToInfoGroup" instruction.

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

```

▼<s:complexType name="FinancialSettingsBillToAddress">
  ▼<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="Validated" 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="ServiceCommands" type="tns:FinancialSettings"/>
  </s:sequence>
</s:complexType>

```

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

- 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 "FinancialSettingsBillToAddress#AddressLine2"
      }
    }
  }
}

```

- Save your changes.

Your commands are applied to the 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 main menu in the **Result Preview** area of the form.

- Publish your customisation project.
- To check your changes in the mobile app, do the following:
 - Open the Sales Orders screen.
 - In the **Settings** tab, find the **Bill-To Info** group.
 - In the **Bill-To Info** group, 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.

Lesson 4.2: Configuring 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 form (SM202550).

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

To Configure Attachment Capabilities of a Screen

1. In the navigation pane of the Customisation 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 `attachment` instruction in the `OrderSummary` container, which corresponds to the Summary view of the Sales Orders form. This container's Attachment button is displayed on the screen toolbar, as shown below.



Figure: The Sales Order screen toolbar

3. In the **Commands** area of the page, update the `attachments` instruction to the following.

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

4. Save your changes. Note that in the **Result Preview** area, the `attachments` instruction has changed.
5. Publish your customisation 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 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.

Lesson 4.3: Configuring 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, you can configure the enhancement of images taken from the camera of a mobile device that is implemented in the MYOB Advanced mobile app. This image enhancement makes the image look clearer and more readable.

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

To Configure the Enhancing and Attaching of Receipts

1. In the navigation pane of the Customisation 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 follows.

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

When the `imageAdjustmentPreset` attribute is set to *Receipt* for a screen, a corresponding button appears in the attachment area. When a user taps the Attachment button, a special camera mode is switched on in the MYOB Advanced mobile app. In this mode, the following enhancements of the image captured by the camera are performed automatically:

- 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 maximised.

The user can make manual adjustments to the image. The automatic changes cannot be undone.

3. Save your changes. Note that in the **Result Preview** area, the `attachments` instruction has changed.
4. Publish your customisation project.
5. Open the mobile app, and navigate to the Sales Order screen.

6. Tap the Attachment button on the screen toolbar, and explore the enhancement functionality. An example is presented below.

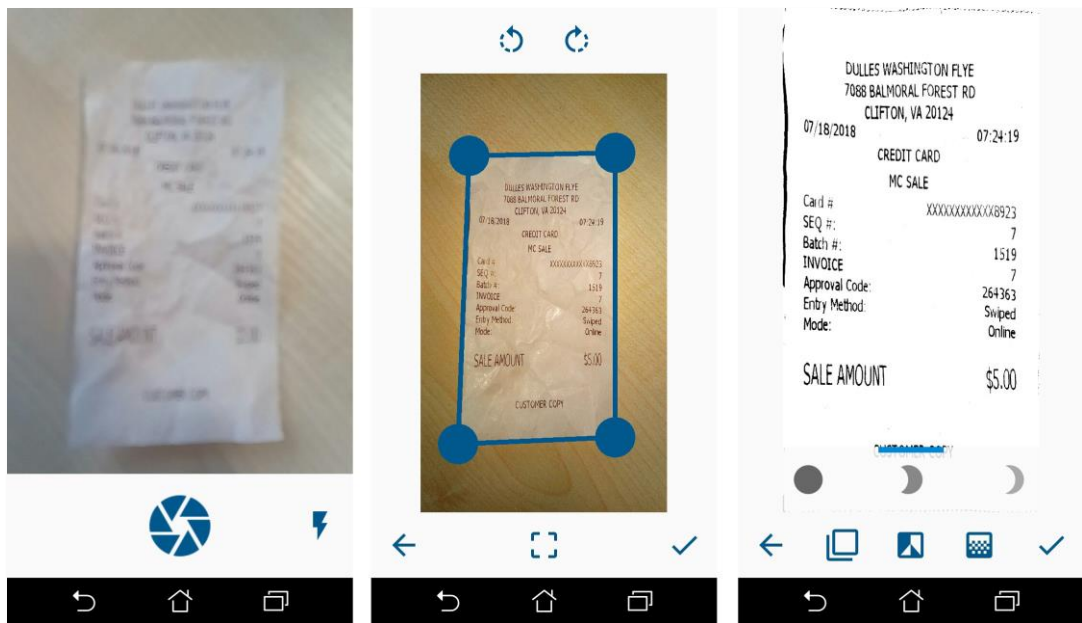


Figure: The enhancement mode functionality

Lesson Summary

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

Lesson 4.4: Configuring Reports

In this lesson, you will configure viewing and downloading the Print Sales Order/Quote report from the Sales Order screen. Before you start customising the mobile app, make sure that this report form already exists in MYOB Advanced.

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

To Configure Reports in the Mobile App

1. On the Sales Orders (SO301000) form, select a sales order and click **Reports > Print Sales Order/Quote**. 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*.

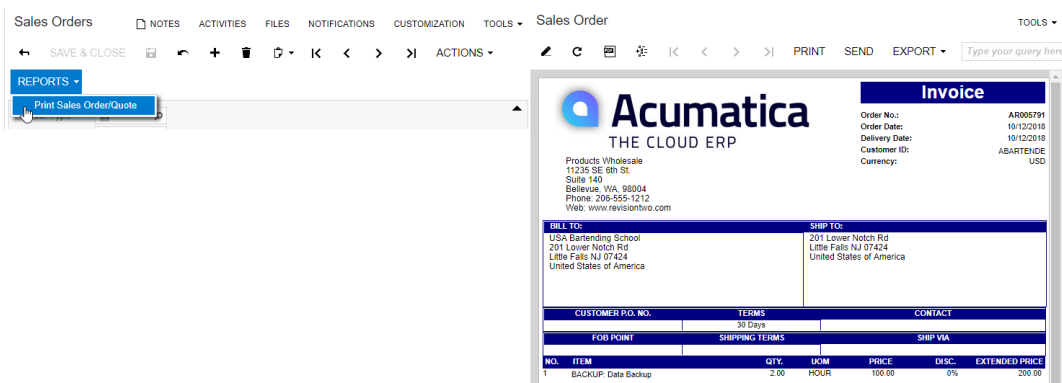


Figure: The Print Sales Order/Quote report

2. In the WSDL schema of the Sales Orders form, find the action that generates the report, which is shown below. Generating a report is an action, so it is located in the `Actions` complex type of the WSDL schema.

```

<s:complexType name="Actions">
  <s:sequence>
    <s:element minOccurs="0" maxOccurs="1" name="CancelCloseToList" type="tns:Action"/>
    <s:element minOccurs="0" maxOccurs="1" name="SaveCloseToList" type="tns:Action"/>
    <s:element minOccurs="0" maxOccurs="1" name="Save" type="tns:Action"/>
    <s:element minOccurs="0" maxOccurs="1" name="Cancel" type="tns:Action"/>
    <s:element minOccurs="0" maxOccurs="1" name="Insert" type="tns:Action"/>
    <s:element minOccurs="0" maxOccurs="1" name="Delete" type="tns:Action"/>
    <s:element minOccurs="0" maxOccurs="1" name="CopyDocumentCopyPaste" type="tns:Action"/>
    <s:element minOccurs="0" maxOccurs="1" name="PrintSalesOrderQuoteReport" type="tns:Action"/>
    <s:element minOccurs="0" maxOccurs="1" name="PrepareInvoice" type="tns:Action"/>
  </s:sequence>
</s:complexType>

```

Figure: The Print Sales Order/Quote action in the WSDL schema

3. In the navigation pane of the Customisation Project Editor, click **Mobile Application > Update SO301000**.
4. In the **Commands** area of the page, add the following code, which maps the action to generate the report.

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

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

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

5. 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 Main Menu”, heeding the following recommendations:

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

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

- o When modifying the main menu, make the screen with the report invisible from the main menu as shown below.

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

6. Publish your customisation project.

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.

Test Use Case 1: Downloading a Report in the Mobile App

In the previous lesson, you have configured a report for the Sales Order screen. Now you can generate and download a report in the mobile app for a particular sales order.

To test the use case, perform the following actions:

1. Open the MYOB Advanced mobile app.
2. Sign in using your credentials.
3. Open the Sales Order screen.
4. On the screen toolbar, open the screen menu and select **Print Sales Order/Quote**.

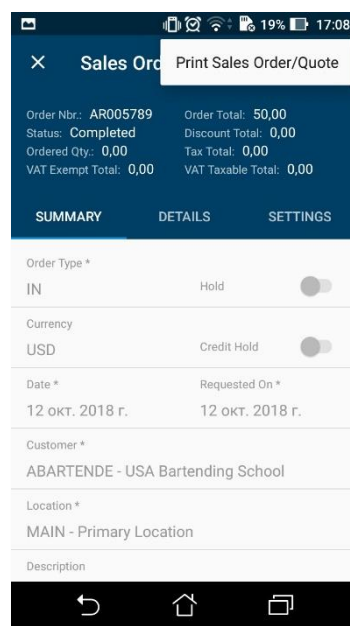


Figure: The screen menu of the Sales Orders screen

The screen with the generated report opens.



Figure: The generated report

5. Open the page menu and select **Download**.

The PDF version of the report is downloaded to your mobile device.

Lesson 4.5: Configuring the Signature Capabilities

In the MYOB Advanced mobile app, you can add the signature functionality to any screen that supports attachments. In this lesson, you will configure the Sales Order (SO301000) 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 such functionality to the screen you will add the `recordAction` object to the `OrderSummary` container.

To Configure the Signature Capabilities

1. In the navigation pane of the Customisation Project Editor, click **Mobile Application > Update SO301000**.
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, and publish your customisation project.
4. Open the mobile app, and navigate to the Sales Order screen. On the screen menu, you see the new menu command, **Sign** as shown in the following screenshot

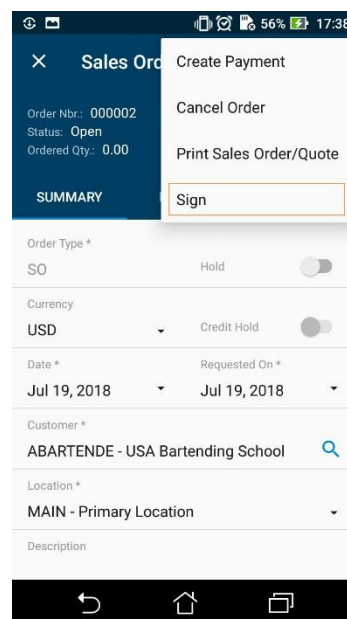


Figure: The Sign menu command

5. Tap **Sign**.

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



Figure: Creation of a signature

6. Sign on the screen and tap **OK**. The signature is added to the list of attached items. A user can add multiple signatures to one record by attaching multiple signatures.
7. Tap the **Save** button to save the signature.

Lesson Summary

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