

Oracle Fusion Cloud Sales Automation

**How do I implement and configure
Oracle CX Sales Mobile?**



Oracle Fusion Cloud Sales Automation
How do I implement and configure Oracle CX Sales Mobile?

F93967-04

Copyright © 2024, Oracle and/or its affiliates.

Author: Carmen Myrick, Jiri Weiss

Contents

Get Help i

1	How do I implement and configure Oracle CX Sales Mobile?	1
	Introduction	1
	Install	5
	Get Started	9
	Configure Search	19
	Additional Configurations	22
	End-User Tasks	45
	Extend CX Sales Mobile	50

Get Help

There are a number of ways to learn more about your product and interact with Oracle and other users.

Get Help in the Applications

Use help icons  to access help in the application. If you don't see any help icons on your page, click your user image or name in the global header and select Show Help Icons.

Get Support

You can get support at [My Oracle Support](#). For accessible support, visit [Oracle Accessibility Learning and Support](#).

Get Training

Increase your knowledge of Oracle Cloud by taking courses at [Oracle University](#).

Join Our Community

Use [Cloud Customer Connect](#) to get information from industry experts at Oracle and in the partner community. You can join forums to connect with other customers, post questions, suggest [ideas](#) for product enhancements, and watch events.

Learn About Accessibility

For information about Oracle's commitment to accessibility, visit the [Oracle Accessibility Program](#). Videos included in this guide are provided as a media alternative for text-based topics also available in this guide.

Share Your Feedback

We welcome your feedback about Oracle Applications user assistance. If you need clarification, find an error, or just want to tell us what you found helpful, we'd like to hear from you.

You can email your feedback to oracle_fusion_applications_help_ww_grp@oracle.com.

Thanks for helping us improve our user assistance!

1 How do I implement and configure Oracle CX Sales Mobile?

Introduction

Overview of CX Sales Mobile Setup

Here's an overview of the main steps in your CX Sales Mobile setup.

High-Level Steps to Configure the Mobile App

Setup Task	Where to Get More Details
Step 1 Understand the app and review system requirements and other pertinent information.	See: Overview of CX Sales Mobile and What are the system requirements for Oracle CX Sales Mobile?
Step 2 Use the System Status tool to ensure that the app deployment is successful. Optionally, generate a QR code that auto-populates sign-in settings.	Review initial installation tasks. See: Simplified CX Sales Mobile App Deployment
Step 3 The Oracle CX Sales Mobile app is ready to use after you download and install it. So, it's best to try out the app and become familiar with its standard capabilities.	See: Get Started with Your Mobile Implementation
Step 4 Implement more features and functionality for the app, including adding offline mode support, map view, business card scanning, and more.	See: Get Started with Your Mobile Implementation
Step 5 Configure and Extend CX Sales Mobile, including changing layouts and adding or removing UI components.	Start with: Configure and Extend CX Sales Mobile
Step 6 Test your configurations.	See: Test Your Configurations
Step 7	See: Get Started with CX Sales Mobile

Setup Task	Where to Get More Details
Become familiar with the end-user experience and capabilities.	
Step 8 Enable other features, such as embedding links that open custom apps, web apps, or other mobile apps.	See: Embed Links That Open a Web App or Another Mobile App and Embed Links to Custom Apps in the Main Menu

Overview of CX Sales Mobile

The Oracle CX Sales Mobile app helps your users manage their day effectively and develop customer relationships using their phone or tablet.

Read the following sections to learn more.

To begin implementing Mobile, start with [Overview of CX Sales Mobile Setup](#).

Manage Sales Activities

Sales teams can manage Sales activities. Here are the highlights:

- Use the action-driven, configurable home page that contains a sales briefing, letting salespeople track the sales pipeline and stay on top of daily activities.
- Leverage optimized task-based flows for key sales functions: Accounts, contacts, leads, opportunities, activities, assets, service requests, partners, and custom objects.
- View, create, and edit tasks, appointments, and call reports.
- Saved searches that are shared across CX Sales Mobile and Workspace lists, for consistency across applications.
- View your recently accessed records in the app when using the global search, or when sharing notes from an external app.
- Add contacts and leads to CX Sales Mobile from the mobile device's address book.
- Create contacts and leads by using a photo of a business card.
- Scan an asset's bar code to search for an asset and automatically enter the asset serial number.
- Share notes, photos, videos, documents, or files from mobile to Sales accounts, opportunities, or other records.
- Plan your sales activities around your current location by viewing your nearby contacts, accounts, opportunities, and leads plotted on an interactive map. You can also plan future appointments in a city or near an address with an easy location search.
- View your quotes in the app and stay updated on the quote status. View and send quote proposal documents to customers, or share them with your team for feedback.
- View reports and analytics with modern visualizations optimized for mobile displays, such as bar graphs, funnels, bubble charts, and more. Role-based layouts let sales administrators provide unique experiences to salespeople, managers, and others.

Manage Partner Relationship Management Activities

Partners can manage Oracle Partner Relationship Management activities:

- Partner users can effectively manage and improve channel sales on-the-go.
- Channel Managers can track overall channel sales and also easily manage partners, partner contacts, and related information.
- Partner role-based restriction features let you configure the app according to your business requirements.

Oracle Sales Assistant Chatbot

Integrate with the Oracle Sales Assistant chatbot to get these features:

- Oracle Sales Assistant helps salespeople perform their daily sales-related tasks. Using their phones, they can type or use your voice to ask questions, and the sales assistant retrieves the information or acts on the requests.
- Sales Assistant reduces manual data entry and makes it easier to view and update sales records using voice commands or typing.

Note: If you need help enabling the Sales Assistant, send us a message on the [Sales Assistant forum](#) on Oracle Cloud Customer Connect.

New Design for Efficiency on the Go

Take advantage of these new design for efficiencies on-the-go:

- Modern and intuitive user experience, designed to boost productivity with fast and seamless interactions.
- At-a-glance record views for a quick grasp of summary, key, and related information.
- Access contextual actions everywhere for fast and easy record updates.
- Improved picker experience powered by Adaptive Search.

Global Search

These global search features make finding records efficient:

- Fast and simple search across all objects in one place.
- Use the multi-keyword search across multiple fields for faster and easier access to information.
- Search more easily with recent searches and type ahead results that display results as you type.
- Filter search results by object types.

New and Improved Sign-In Experience

This version of the mobile app has a new and improved sign-in experience:

- Scan a QR code to sign in easily using your single sign-on credentials or using basic authentication.
- Use face or fingerprint recognition to securely sign in to the app.
- View new configurations with fast download and instant activation of your configurations.
- Enable OAuth authentication token support to let users stay signed in to the app for longer.

Use a Phone or Tablet

Use either a phone or tablet to access the app. Using a tablet changes the display to 2x portrait mode.

Easy Configuration

Implement Mobile using these easy configuration aspects:

- Configure the app using the new interface designer.
- Create criteria for page layouts for both standard and custom objects.
- Download your configurations to the app quickly and easily. Mobile automatically signs you in to the correct sandbox without restarting the app.
- Create custom scripts that enforce custom validations and business logic for both top-level parent and child objects.

Use Offline

Salespeople can use Mobile in offline mode when they're not able to be online:

- View and edit Sales data in areas with no network connection.
- Sync automatically when a connection is re-established.

What are the system requirements for Oracle CX Sales Mobile?

Here are the system requirements for Oracle CX Sales Mobile:

Apple Devices

- iOS 12.1 or higher

Android Devices

- Android OS 8.0 or higher

Related Topics

- [What are the system requirements for Microsoft integration?](#)

What's the default language for CX Sales Mobile?

CX Sales Mobile uses the language set on your device as the default language.

How do users get CX Sales Mobile upgrades?

CX Sales Mobile is consistently updated every quarter along with the Sales web application. In addition, the mobile app will often, but not always, release monthly updates between the quarterly releases.

If users have automatic updates turned on, then the latest version of the CX Sales Mobile app is automatically downloaded to their devices whenever a new version is made available. If automatic updates aren't turned on, users can always check to see if an update is available in the iOS App Store or the Google Play Store.

Which mobile devices are supported?

To find out which mobile devices and operating systems are supported by CX Sales Mobile, see the System Requirements for Oracle Applications Cloud page at <https://www.oracle.com/system-requirements/>.

Install

Simplified CX Sales Mobile App Deployment

Use the System Status tool to make sure that your CX Sales Mobile app deployment is successful. In the tool, you can check that features have been setup successfully. You can also check whether your deployment is missing recommended features that help salespeople get the best experience of the app. You can also generate a QR code and add it to the Sales home page. The QR code automatically populates the sign-in settings.

Here's an overview of the tasks here:

1. *Navigate to the System Status Tool.*
2. *Understand the System Status Tool Results*
3. *Generate a QR Code*

Navigate to the System Status Tool

Navigate to the System Status tool:

1. Sign in to the Sales application as user with a Sales Administrator role.
2. Create a publishable sandbox that includes Application Composer.
3. Open Application Composer by selecting **Application Composer** in the Configuration category in the Navigator menu.
4. In the Application Composer Common setup menu, or on the Overview page, click **Mobile Application Setup**.
5. Click the **System Status** tab.

The page shows the **Setup Information** pane that gives you information about the host URL for your Sales web application, along with information about when the System Status tool was last run.

Understand the System Status Tool Results

The system status check starts automatically when you click the **System Status** tab. The **System Status Check in Progress** button text changes to **Check System Status** when the check finishes. You can view the results by scrolling down to the Results pane.

The results can either be Success, Error, Hidden, or Recommended. Here a sample image of the icons:



Here are what the results mean:

- **Success**, shown as a green circle with a white check mark. The feature is set up correctly and there aren't any further steps required.
- **Error**, shown as a red circle with a white backslash. This means that the feature is a required feature and that there are problems with the setup.
 - Click the **View Details** button to find out about the issue.
 - In the Status side panel, hover your cursor over the symbols next to the error icon to find out the exact problem.
 - You can also view documentation that will help you to fix the issue by clicking the chain link icon.
- **Recommended**, shown as a circle with an "i" in it. This is a recommended feature that hasn't been set up. Setting up the feature enables your users to get the best experience of the app.
- **Hidden**, shown as an eye with a line through it. This hides the feature in CX Sales Mobile, so that your salespeople can't see it or use it.

Optionally Generate a QR Code

In the QR codes pane, you can generate a QR code that automatically populates the sign-in settings, making signing in to the app easier for your users.

If using the QR code option, create a QR code that uses either single sign-on (SSO) or basic authentication and then click the **Publish** button to publish the QR code to the News and Announcements section in the Sales home page. After your users have downloaded CX Sales Mobile, they just need to scan the QR code to sign in.

Note: If you're using SSO to sign in to the app, then there's no additional SSO setup required for CX Sales Mobile.

See [Get Started with Your Mobile Implementation](#) for more information.

Use Mobile Device Management Software to Install CX Sales Mobile

You can use Mobile Device Management software to install the CX Sales Mobile application, enabling you to manage the CX Sales Mobile app in your organization.

The Mobile Device Management software lets you prepopulate the host URL value and then:

- Require that salespeople sign in with their basic authentication details; or
- Specify that salespeople use their single sign-on details to sign in

Here's how you set it up:

1. Add CX Sales Mobile to your organization's Mobile Device Management application catalog.
2. In the Mobile Device Management app, navigate to the application configuration details so that you can specify configuration values for CX Sales Mobile.

3. Create a new configuration record to set the host URL value. Use these values:
 - o **Configuration Key:** `CXM_HOST`
 - o **Value Type:** `String`
 - o **Configuration Value:** Your organization's host URL value. See the [Get Started with Your Mobile Implementation](#) topic for information about how to find your host URL value.
4. Create another configuration record to specify basic authentication sign in or single sign-on (SSO). Use these values:
 - o **Configuration Key:** `CXM_AUTH_TYPE`
 - o **Value Type:** `String`
 - o **Configuration Value:** Don't enter a value to use SSO. Enter `BASIC` if you want to use basic authentication.

Implementation Considerations

Here are some best practices to follow and things to note when you're implementing Oracle CX Sales Mobile.

The Home Page

The home page is the first page that displays when users sign into the app. For salespeople, the home page can show their pipeline and tasks, and for sales managers it can show navigation shortcuts to sales objects (such as opportunity and account pages).

Here are the aspects to note about the home page:

- The Sales Representative role, Partner Sales roles, and the Sales Administrator role have the sales pipeline and tasks added to their home page by default. The Sales Manager and Partner Sales Manager roles have their team's pipeline added to their home page by default.
- All other job roles don't see sales pipelines and tasks by default. Instead, they see their relevant saved searches and shortcuts to view sales objects (such as viewing opportunities and accounts).
- The pipeline shows the total amount and number of the opportunities owned by the user. The amounts are shown in the corporate currency. However, keep in mind that opportunities might have different currencies and so will use an exchange rate conversion to calculate the corporate currency amount.
- If an accounting calendar is set up with quarters (specified in the `ZCA_COMMON_CALENDAR` profile option), then the pipeline uses the accounting calendar to determine the period for the current and next quarter. If there isn't an accounting calendar defined for Sales use, then the pipeline uses the standard Gregorian calendar quarter definition (January to March, April to June, July to September, October to December). If there's an accounting calendar available for Sales, but it doesn't conform to quarterly periods, or current or next quarters can't be resolved, then the pipeline uses the standard Gregorian calendar quarter definition.
- Ensure that for the `MOO_OPTY_SRCH_CLS_PERIOD` lookup type that the relevant lookup values are enabled, so that the calendar displays the associated close period data:
 - o If you're using the accounting calendar, ensure that the lookup value `CURR_NEXT_QTR_FSCL` is enabled and not end dated.
 - o If you're not using the accounting calendar, ensure the lookup value `CURR_NEXT_QTR` is enabled and not end dated.

For more information about the home page, see the [Configure the Home Page](#) topic.

Search Features

Salespeople can use the global search feature directly from any page to find any type of record quickly and easily. They can also use saved searches for each of the objects that are set up using Workspace and that are shared across the CX Sales Mobile app and the Sales app.

Here are the aspects to note about the global search and Adaptive Search features:

- The global search feature appears at the end of every page and uses Adaptive Search. Search capabilities that are available in other parts of a page also use Adaptive Search, but they only search the object type that the user is viewing. For example, if the user is viewing an opportunity list and uses the search field at the top of the page, then CX Sales Mobile searches for opportunities only.
- CX Sales Mobile uses Adaptive Search and Adaptive Search configuration, so if an object isn't enabled for Adaptive Search, then it won't appear in CX Sales Mobile search results. For example, if you create four custom objects, enable them all for Adaptive Search, but only enable one for mobile, then only the one custom object is included in the CX Sales mobile search results.
- The search is also limited to the standard sales objects that are available in CX Sales Mobile and custom objects. Therefore, if you try to search for a deal for example, it won't appear in the search results.
- You can't configure the information that's shown in the global search results.
- The global search uses a keyword search for all objects enabled for CX Sales Mobile and for all fields that are in those objects. To narrow the search results to a specific object, use the **Show** filter to filter the search results.

Here are the aspects to note about the saved search feature for each of the objects:

- You can enable predefined or custom saved searches for each object in Workspace and make them available for all users generally, or you can restrict them by role. In addition, sales users can also create their own personal saved searches. All the saved searches that are available in Workspace for a user are also available in CX Sales Mobile for that user.
- The last used saved search for an object, and the last used sort option for a saved search, are used by the app until the user closes and restarts the app. For example, if the user had chosen the "All Opportunities" saved search, rather than the default "My Open Opportunities", when the user returns to the opportunity list, the "All Opportunities" saved search is shown, as that was their last used saved search.
- Custom activity saved searches need to be set up with their corresponding activity type criterion, so that they display in the CX Sales Mobile saved search list. For example, custom task saved searches need to be associated with the Task activity type in Workspace, otherwise the saved search won't be shown in the CX Sales Mobile Task saved search list.

FAQs

Here are some FAQs about the features outlined in this topic:

- **Question:** Why can't I see a pipeline or tasks in the home page?
Answer: Only users with the Sales Representative (ORA_ZBS_SALES_REPRESENTATIVE_JOB) and Sales Administrator (ORA_ZBS_SALES_ADMINISTRATOR_JOB) roles see their pipeline and tasks by default.
- **Question:** Why can't I find what I'm searching for?
Answer: If an object isn't enabled for Adaptive Search, then it won't appear on in CX Sales Mobile search results.

Get Started

Get Started with Your Mobile Implementation

The Oracle CX Sales Mobile app is ready to use after you download and install it. It's best to try out the app and become familiar with its standard capabilities.

Prepare

Check these things first:

1. Check that Adaptive Search is deployed in the Sales application, and that search indexing processes are run and regularly scheduled. The CX Sales Mobile app uses the saved searches powered by Adaptive Search.

See the *Records Management* section in this guide for information about how to create useful lists of records for users.

2. Make sure that your device is ready to use with CX Sales Mobile:
 - o Does your mobile device meet the requirements? See the topic *What are the system requirements for Oracle CX Sales Mobile?*.
 - o Is your network connection working?
 - o Does your mobile device include data service?
 - o Have you set up your email account?

Try It Out!

Now you can try out the app and familiarize yourself with the standard capabilities:

1. Sign in to the app and take note of the default objects, fields, and page layouts. While doing that, think about these questions:
 - o What objects are available?
 - o What fields and related objects are displayed with each object?
 - o What do the page layouts look like for each object?
 - o Do you need to configure the home page and reports page more than your users can already do themselves?

Note: See the *Personalize the Home and Reports Pages for End Users* topic for more details about user personalization for the home and reports pages.

2. When you're familiar with the default settings, consider whether you need to change anything to address your organization's needs. For example:
 - o Does CX Sales Mobile display the standard and custom objects that your sales team needs to work with?
 - o Are the fields that you require displayed for each of those objects? Do you want to add or remove other fields, including custom fields?
 - o What about the related objects?
 - o Do you want to configure the page layouts?
 - o Do you need page layouts for users in different geographical regions or with different roles?

To find out how to make these configurations, review the Configure section in this topic. You'll also find out about the other configuration options for the app.

Configure the App

There are two ways that you can configure CX Sales Mobile:

- Using profile options
 - Using the configuration tool in Application Composer
1. Start by enabling and disabling profile options for the app. Profile options lets you, for example, enable the call logging feature. See the [Enable or Disable Features Using Profile Options](#) topic for the full list of profile options.
 2. Use the CX Sales Mobile configuration tool for additional configurations. See the [Configure the App Using the Configuration Tool](#) topic for details.
 3. After making configurations, make sure that you test them and publish the sandbox. See the [Test Your Configurations](#) topic for more details.

On-Board Your Sales Representatives

Now you want to think about how you're going to roll out the application to your salespeople. Here are some tips:

- Consider a phased roll-out, starting with a pilot project that includes a small group of users.
- Collect feedback from the test users and create training materials to address their specific needs.
- Capture screenshots on your mobile device for use in your training materials.
- To create videos of the application, download software that lets you record your screen on your computer.
- To present live training, download software that lets you mirror your mobile device on your computer.
- Consider creating a web page on your intranet with information and links relating to the application.
- Inform your users of the QR code, if you've created one. You could publish the QR code on your intranet web page.
- Make sure that your users have their mobile devices up and running. Remind yourself of the questions that they might have by reviewing the Prepare section of this topic.

Create a QR Code and Install the App

To make signing in to the app easier for your users, create a QR code that automatically populates the sign-in settings. You can create a QR code using any QR code generator. You can find many free ones on the internet.

So that you can associate it with the QR code, you need to create a URL that automatically populates the sign-in settings. Here's how to create the URL:

1. Find out the host URL by signing in to your Sales application, and in your browser's address bar, locate the part of the URL between `https://` and the next slash (`/`). For example, it might be something like: `example-xxxx.exampleleads.com`. Take a note of the host URL, as you'll use it in the next step.
2. Create the URL to use in the QR code. The general format of the URL is `cxm://?host=<host URL>&type=<authentication type>`. Enter the host URL that you made a note of in the first step to the `<host URL>` value. Enter the authentication type of either `basic` or `sso` (single sign-on) to the `<authentication type>` value. For example, if you want to create a URL for basic authentication, use this format: `cxm://?host=<host value>&type=basic`. If you want to create a URL for single sign-on, use this format: `cxm://?host=<host value>&type=sso`.

Note: `<host URL>` and `<authentication type>` represent the host URL, or authentication value, so ensure that you don't include the brackets when entering the actual value.

Install the CX Sales Mobile app and sign in:

1. Open the App Store, or Google Play, and search for Oracle CX Sales Mobile application, and then tap **Install**.
2. Open the app and accept the legal terms.
3. If you've already created your QR code, then you can use that to sign in. Alternatively, tap **I don't have a QR code** on the sign-in page, then tap **Basic Authentication** or **Single Sign On**, and enter the host URL that you made a note of in the Create a QR Code and Install the App section.
4. If you selected the single sign-on option, enter your SSO information. If you selected the Basic Authentication option, enter your CX Sales application user name and password.
5. Tap **Sign In**.

Note: If you're using SSO to sign in to the app there's no additional setup required for CX Sales Mobile. However, ensure that you have SSO set up for the Oracle CX Sales app.

How CX Sales Mobile Continues to Work Offline

When a network connection is lost, a message displays to let your users know that they're offline. They can continue to work as before. After the network connection is restored, their changes are automatically synchronized.

Here's how it works:

Data Downloads When the App is Online

While users are using the application online, up to two different types of data are downloaded for offline use:

- Data that users viewed in the application.
- If you've enabled the Auto Fetch feature, data returned in the default saved search. This includes data for both standard and custom objects. See the [Configure the Offline Settings](#) topic for more information about the Auto Fetch feature.

All data is stored encrypted using the device's encryption. In addition, when a user allows the face or fingerprint recognition sign-in feature, access to data in the offline mode is further protected by the biometric authentication.

Offline Message Displays When the App is Offline

If network connectivity is lost while the app is in use, a message appears to let users know that they're working offline. A banner displays the number of updates that will be synchronized when they're back online. It also shows how long it's been since they last synchronized their data.

Offline Updates are Saved

When users are working offline, they can see all of the data that was saved onto the device while they were online. They can also create new records, or update existing records, which will then be synchronized to the server when they're back online.

When users view records that have been created or updated offline, they'll see a visual indicator on the page indicating that the record will be synchronized to the server when they're back online. If they want to view all of the records that will be updated when the application is back online, then they can navigate to the pending sync queue by tapping the offline banner and then tapping the more icon in the Pending Sync page.

Offline Data is Synchronized

Once the network connection is restored, the records that are pending synchronization are synced to the Sales web app in the order that the transactions were made offline. If a transaction fails to sync, for example due to a validation error, the corresponding error message is displayed against the record in the Pending Sync queue. Once the user has resolved the issue, they can sync the record manually by tapping the more icon in the offline banner and then tapping the **Retry Sync For All** button.

Note: If the application detects a data conflict, then the changes made in CX Sales Mobile will override any changes made in the Sales web application.

Configure the Offline Settings

You can configure Oracle CX Sales Mobile's offline settings, such as whether the offline data synchronization is automatic or manual. You can also select which sales objects' records are available offline.

To configure the general offline settings, navigate to the Mobile Application Setup page in Application Composer, and click **Settings**. Here are the options you can set:

- **Enable Offline Mode:** Enables or disables the offline mode.
- **Enable Offline Create and Edit:** Enables or disables the creating and editing of data offline.
- **Auto Sync:** Lets you specify whether the application automatically syncs offline updates from CX Sales Mobile to the Sales web app when a network connection is detected.
- **Disable Deferred Create and Edit:** Lets you specify whether the application creates and edits transactions asynchronously. If this setting is disabled, then transactions are synchronized asynchronously from the application to Oracle Cloud.
- **Manual Sync:** Lets you specify whether the manual sync makes updates from CX Sales Mobile to the Sales web app and back to CX Sales Mobile (bidirectional sync), or only from CX Sales Mobile to the web app (one-way sync).
- **Maximum Number of Retries:** Lets you specify the maximum number of times CX Sales Mobile tries to apply offline updates to the Sales web app.

- **Online Data Fetch Policy:** Defines which data (local, remote, or both) to display when there's a network connection. Here are the options:
 - **Local:** Only display data stored on the phone.
 - **Remote:** Only display data from the server.
 - **Both:** Display the local data primarily, and then query the server in the background for any remote data that has changed. The changed data is then displayed rather than the local data.
- **Auto Clear Cache:** Defines the maximum duration that data is stored on a user's phone, after which the data is removed from the phone.

You can also configure the data that's available offline. By default, all of the data that users can view in the application when they're online is downloaded for offline use.

In addition to this data, administrators can also enable the Auto Fetch feature for standard and custom sales objects. This feature downloads a subset of records (including the related child records) for each object that has the feature enabled. The application uses the default saved search for the Auto Fetch-enabled object as the criteria for the data subset.

Note: Auto Fetch only downloads child records that are standard objects. Child records that are custom objects won't be downloaded.

Here's how you enable the Auto Fetch feature for an object:

1. Sign in to the Sales application as user with a Sales Administrator role.
2. Create a publishable sandbox that includes Application Composer.
3. Open Application Composer by selecting **Application Composer** in the Configuration category in the Navigator menu.
4. In the Application Composer Common setup menu, or on the Overview page, click **Mobile Application Setup**.
5. Click **Home** within the Application Features menu.
6. In the mobile interface designer, click the object that you want to enable Auto Fetch for.
7. In the Feature Details section, enable the **Enable Auto Fetch** option.
8. Repeat for additional objects.

Define the Default Email Client to Use with CX Sales Mobile

Configure which default email client the CX Sales Mobile app uses when a user taps an email icon or email address.

When a user taps an email icon or address in the app, you can configure your corporate Microsoft Outlook email client to open, rather than the device's default email client.

Note: At present, only Microsoft Outlook is supported as an alternative to the device's default email client.

Here's how to set it up:

1. In a publishable sandbox, navigate to **Application Composer > Mobile Application Setup > CX Sales Mobile Composer > Settings**.
2. Expand the **Security Settings** header.
3. In the **Select Default Email Client** drop-down list, select either Microsoft Outlook or the device's default email client.
4. In the header, click **Save**.

5. Test your update by following the steps in the *Test Your Configurations* topic.
6. Publish the Sandbox.

Note: If the configured email client isn't installed on the user's device, then no email client will open when the user taps an email. If you don't specify an email client, then the device's default email client is used.

How do I enable sales representatives to stay signed in to the CX Sales Mobile app?

You can enable the use of OAuth authentication tokens so that salespeople can stay signed in to the app, and they won't need to sign in again when they launch the application. Using OAuth authentication tokens makes the app simpler to use and is also secure.

To set up this feature, you first need to register the Oracle CX Sales Mobile application with Oracle Identity Cloud Service, by creating a Confidential Application in the Identity Cloud Service console. Then you need to associate the Confidential Application details with Oracle CX Sales Mobile, using Setup and Maintenance in the Oracle CX Sales web app.

Here are the tasks:

1. *Create a Confidential Application for Oracle CX Sales Mobile in Oracle Identity Cloud Service*
2. *Associate the Confidential Application With Oracle CX Sales Mobile*
3. *Use Your Organization's Mobile Device Management Software to Distribute OAuth Authentication Tokens*
4. *Test Your Updates*

Create a Confidential Application for Oracle CX Sales Mobile in Oracle Identity Cloud Service

Here's how you create CX Sales Mobile as a Confidential Application in Identity Cloud Service:

1. Sign in to My Console in Identity Cloud Service, by navigating to **www.oracle.com > View Accounts** and enter the name of the relevant pod in the Cloud Account Name field. You have at least two pods, a test and a production pod, so ensure that you enter the correct pod.
2. Click **Next** and enter your Identity Cloud Service password. If you can't remember the password, contact Oracle Support for help.

Note: For more information about Identity Cloud Service, navigate to <https://docs.oracle.com>, search for Identity Cloud Service, and click Oracle Identity Cloud Service - Get Started.

3. In the Identity Cloud Service console, expand the **Navigation Drawer**, and then click **Applications**.
4. Click **Add**. Then on the **Add Application** page, click **Confidential Application**.
5. In the **Details** tab of the **Add Confidential Application** wizard, enter information in the **Name** and **Description** fields. These fields are used to identify the Confidential Application in Identity Cloud Service only, they don't appear outside of Identity Cloud Service. For example, they don't appear in the Oracle CX Sales app.
6. Click **Next**.
7. In the **Client** tab, click the **Configure this application as a client now** option.
8. In the **Allowed Grant Types** options list, select these options:
 - o **Refresh Token**
 - o **Authorization Code**

9. Select the **Allow non-HTTPS URLs** option.
 - Note:** This option doesn't open a URL that isn't secure, it just sends a request to the CX Sales Mobile app.
10. In the **Redirect URL** field, enter `cxm://?oAuth=true`
11. In Token Issuance Policy section, click **Add Scope** under **Resources**
12. In the Select Resources window, click **Oracle Application Cloud (Fusion)** from the list of resources.
13. Record the **Scope** value of the resource you added.
14. Click **Add**
15. Click **Next**.
16. In the **Resources** tab, click **Skip for later**. No resources are required.
17. Click **Next** to navigate to the **Web Tier Policy** tab. Nothing needs to be entered here
18. Click **Next** to navigate to the **Authorization** tab. Nothing needs to be entered here.
19. Click **Finish**. The CX Sales Mobile application is added in a deactivated state.
20. Record the **Client ID** and **Client Secret** that appear in the **Application Added** dialog box. The **Client ID** and **Client Secret** are equivalent to a credential (for example, an ID and password) that your application uses to communicate with Identity Cloud Service.
21. Click **Close**.
22. At the top of the page, to the right of the application name, click **Activate**.
23. In the **Activate Application?** dialog box, click **Activate Application**.

For more information about creating a Confidential Application, navigate to <https://docs.oracle.com>, search for **Identity Cloud Service**, click **Oracle Identity Cloud Service - Get Started**, and click **Manage confidential Applications** under the Manage Applications heading.

Associate the Confidential Application With Oracle CX Sales Mobile

Here's how you use Setup and Maintenance to associate the Identity Cloud Service Confidential Application with CX Sales Mobile:

1. In the CX Sales application, sign in as an administrator. Open the navigator menu, and click **Setup and Maintenance**.
2. In the Setup and Maintenance work area, click the **Tasks** side panel icon and then click the **Search** link.
3. Search for and select the **Manage Profile Options** task.
4. Create a profile option to enable the OAuth feature. Create the new profile option with these details:
 - o **Profile Option Code:** `CXM_ENABLE_OAUTH`
 - o **Profile Display Name:** OAuth Usage
 - o **Application:** CX Sales Mobile
 - o **Module:** Mobile Sales
 - o **Start Date:** Enter the date you require the OAuth feature to be active
 - o Click **Save and Close**.
 - o In the Profile Values section, for Site level, check the **Enabled** and **Updatable** check boxes.
 - o Click **Save and Close**.
5. Next, you set the values for the profile option:
 - a. Search for and select the **Manage Administrator Profile Values** task.
 - b. Search for the `CXM_ENABLE_OAUTH` profile option.
 - c. In the Profile Values section, add new values with the following details:
 - **Profile Level:** Site
 - **Profile Value:** Yes

6. Click **Save and Close**.
7. Enter a publishable sandbox with Application Composer as one of the tools.
8. Navigate to **Application Composer > Mobile Application Setup > CX Sales Mobile Composer > Settings**.
9. Expand the OAuth Settings header.
10. Provide the Client ID, Client Secret and Scope for the CX Sales Mobile application, as recorded while setting up the Confidential Application in Identity Cloud Service.
11. Provide your Identity Cloud Service host as the Topology URL.
12. In the header, click Save.
13. Test your update by following the steps in the Test Your Configurations topic.
14. Publish the Sandbox.

When OAuth authentication tokens are enabled, the CX Sales Mobile app reads the OAuth parameters when the sales representative signs in. The next time the sales representative closes and restarts the app, or the current session expires, the app prompts them to sign in again. After this second sign-in, the OAuth authentication starts and from then the sales representative stays signed in to the app, as long as they use the app at least once during the time validity of the refresh token configured in Oracle Identity Cloud Service.

Use Your Organization's Mobile Device Management Software to Distribute OAuth Authentication Tokens

You can use your organization's Mobile Device Management (MDM) software to distribute OAuth authentication tokens. Using MDM to distribute the tokens means that your sales representatives won't need to sign in to the app for a second time to trigger the OAuth authentication.

Here's how you enable OAuth authentication tokens for MDM software:

1. Follow the steps in the **Create a Confidential Application for Oracle CX Sales Mobile in Oracle Identity Cloud Service** section of this topic.

Creating a Confidential Application allows you to specify when access tokens expire, and creates a Client ID and Client Secret.

2. Open your MDM software and enter these configuration settings:

- a. **Configuration Key:** `CXM_AUTH_TYPE`

Value Type: String

Configuration Value: `oauth\ssso` or `basic` (enter "SSO" if you want to use Single Sign On, or enter "basic" if you want to use basic authentication)

- b. **Configuration Key:** `CXM_TOPOLOGY_URL`

Value Type: String

Configuration Value: Enter the Oracle Identity Cloud Service instance URL. For more information about Identity Cloud Service, navigate to <https://docs.oracle.com>, search for **Identity Cloud Service**, and click **Oracle Identity Cloud Service - Get Started**.

- c. **Configuration Key:** `CXM_CLIENT_ID`

Value Type: String

Configuration Value: Enter the Client ID that was generated when you created the Confidential Application

- d. **Configuration Key:** `CXM_CLIENT_SECRET`

Value Type: String

Configuration Value: Enter the Client Secret that was generated when you created the Confidential Application

- e. **Configuration Key:** `CXM_HOST`

Value Type: String

Configuration Value: Enter the CX Sales application host URL

Test Your Updates

Test your updates and publish the sandbox when you're happy with the configuration. See the [Test Your Configurations](#) topic for details about how to check your mobile configurations.

Enable or Disable Features Using Profile Options

Here are some of the profile options that you can use to configure Oracle CX Sales Mobile.

Use the **Manage Administrator Profile Values** setup task to set the profile option values.

The CX Sales profile options listed in the table are used in the CX Sales Mobile app to determine the default settings for features that aren't specific to CX Sales Mobile. The Application LOV Value column lists the values that you can select in the Application list on the Manage Administrator Profile Values page.

Profile Option Code	Created by the Administrator?	Purpose	Value and Behavior	Application LOV Value
ZMS_LOG_CALLS	No	Enable the call logging feature.	Set to 'CONFIRM' to ask user permission before initiating log a call flow. Set to 'AUTOMATIC' to automatically start log a call flow. Set to 'DISABLE' to disable call logging.	Mobile Sales
ZEM_CXMOBILE_FORCE_LANGUAGE	Yes	Sets the language in the mobile app, irrespective of device language.	Use the respective language code, for example, "en_US" for US English.	Mobile Sales
ZCA_COMMON_CORPORATE_CURRENCY	No	Set the corporate currency for opportunity records. When an opportunity is created, it uses the CX Sales user preference currency from the user's regional settings. If there's no user preference, then corporate currency is used.	Derived from the CX Sales application.	Common CRM
MOO_REVN_ENABLE_MULTICURRENCY	No	Enable multiple currencies for opportunity revenue lines independently of	If set to Yes, mobile users can create and manage leads and opportunities with currencies that deviate	Opportunity Management

Profile Option Code	Created by the Administrator?	Purpose	Value and Behavior	Application LOV Value
		the opportunity header currency.	from User's Preferred Currency or Corporate Currency (specified in the profile option ZCA_COMMON_CORPORATE_CURRENCY).	
ZEM_DISABLE_FREEFORM	Yes	Disables the free-form picker (a combination of a free entry text box and a picker) in the Leads page. The free form picker is enabled by default.	Set to 'lead' to disable the free-form picker in the Leads page or set to 'none' if you want to keep the picker as enabled.	Mobile Sales

Implement Sales Assistant for CX Sales Mobile

Sales assistant is an individual chatbot that you can use to help you perform your daily sales-related tasks. Using your phone, you can ask questions or commands to the sales assistant and it retrieves the information, or acts on your requests. It reduces manual data entry, and makes it easier to view and update sales records through text or voice commands.

To enable sales assistant for CX Sales Mobile, follow the steps in these two topics: [Set Up Oracle Sales Assistant for CX Sales Mobile](#) and [Enable Sales Assistant for CX Sales Mobile](#).

You can create a new or modify an existing welcome message with some suggestions of what sales representatives can do and say to the sales assistant which are relevant to your business. Modify the message in the Oracle Sales Assistant Common Configuration Tool within Application Composer by following the steps in this topic: [Edit Strings in Oracle Sales Assistant Common Configuration Tool](#)

Enable Notifications for Sales Objects in CX Sales Mobile

You can enable notifications to be triggered automatically when certain business events occur in the CX Sales Mobile app, or in the CX Sales web app. For example, you can set up a notification that notifies a salesperson that they have been assigned a lead or task. You can choose to have the notification pushed to the mobile device, or have it viewed inside the app as in-app notification. When you enable notifications, salespeople receive timely updates to key sales data, so that they can take action on the updates.

There are four parts to enabling notifications:

1. Enable the notifications feature in Setup and Maintenance.
2. Enable notifications for the CX Sales Mobile app.
3. Enable notifications for business events.
4. Configure delivery channels and notifications recipients.

Enable the Notifications Feature in Setup and Maintenance

Follow the instructions in the topic [Enable Notifications](#) in the Notifications chapter of this guide.

Enable Notifications for the CX Sales Mobile App

Next, you need to check that notifications are enabled in the CX Sales Mobile settings. You can also specify whether notifications are available for certain regions and roles only.

1. Navigate to Application Composer and click **Mobile Application Setup**.
2. In the Application Features pane, click **Settings** and expand the **Notifications Settings** menu.
3. Ensure that the **Enabled** setting is on. This option only enables and disables the in-app notification, and is enabled by default.
4. Use the **Assigned Regions** and **Assigned Roles** fields if you want to enable notifications for specific regions and roles only.

Enable Notifications for Business Events

Refer to the topics [Define Notification Scripts](#) and [Sample Notification Scripts](#) in the Notifications chapter of this guide.

Configure Delivery Channels and Notifications Recipients

Follow the instructions in the topic [Configure Delivery Preferences and Text for a Notification](#) in the Notifications chapter of this guide.

Configure Search

Use Saved Searches as Custom Filters and Sort Options in Pickers

Salespeople can use the saved searches created in either classic Sales or Digital Sales in many ways. The saved searches must be those powered by Adaptive Search.

Here's an overview of the search capabilities:

- Use the saved searches as filters and sort options for pickers and dynamic choice lists, both for standard and for custom objects. This enables sales representatives to have the same personalized experience when searching, sorting, and viewing list results for the pickers as they have with the corresponding object list pages.

Note: See the [Use Saved Searches as Filters for Dynamic Choice Lists](#) topic for information about setting up Workspace saved searches as custom filters and sort options for dynamic choice lists.

- Use a default saved search that's specific to their role. As an administrator, you can set the default saved search per object for CX Sales Mobile app users for specific roles (or for all users if preferred). For more details, see the [Set Default Saved Searches for Mobile App Users](#) topic.
- View records related to the parent record in pickers. For example, the opportunity picker shows related opportunities for the account selected for a new task.

FAQs

Here are some FAQs about this feature:

Question: How do I create a custom filter for use in a picker?

Answer: Create a custom Workspace saved search for an object using the CX Sales web app Workspace feature. For example, you could create a saved search called "My Primary Accounts" for the account object.

Question: Why can't I use a saved search as a filter for a picker?

Answer: If an object isn't enabled for adaptive search in workspace, then the picker won't show workspace saved searches as filters.

Question: Why do the resource, competitor, product, industry, and classification pickers only display predefined filters?

Answer: These objects aren't enabled in Workspace by default, so until they're enabled the pickers for these objects will display predefined filters.

Use Saved Searches as Filters for Dynamic Choice Lists

Enable salespeople to use saved searches to filter dynamic choice lists. The application supports the new generation of saved searches powered by Adaptive Search. You can create the saved searches in any work area that uses Adaptive Search, including Workspace and Digital Sales work areas.

Before you start

1. Ensure that the filter criteria field that you'll use for the dynamic choice lists already exists, or has been created.
2. Create the dynamic choice list and enable filter criteria.

For more information about dynamic choice lists, see the [Overview of Dynamic Choice Lists](#) topic.

Enable the Filter Criteria Field

After checking the prerequisites exist, or have been created, you need to enable the field that you want to use as filter criteria in Adaptive Search. Here's how:

1. In the Setup and Maintenance work area, go to the **Configure Adaptive Search** task:
 - o Offering: Sales
 - o Functional Area: Sales Foundation
2. Click the **Setup** tab and in the main page, click the **Advanced** tab.
3. Find the object that contains your filter criteria field. Check that the Enable checkbox is checked for that field. This allows the field to be searched and filtered within Workspace. If it is enabled, proceed to the next section of this document.
4. Find the field that you want to use as a filter criteria and check whether the **Enable** checkbox is enabled. If it's enabled, continue to the "Create Filtered Workspace Search" section. If it isn't enabled, click the **Enable** checkbox, click the **Actions** button, and select **Partial Publish**.
5. Wait for the publish action to complete and then continue to the **Create Filtered Workspace Search** section

Create Filtered Workspace Search

Next you need to create a search that is used to limit the values in your dynamic choice list pick list to a specific list of values.

1. Open the Navigator and click **Workspace** to open the Workspace page.
2. Select the object and saved search that you want to use as a filter in the dynamic choice list.
3. Click **Show Filters** to display the predefined filters for the saved search you selected and click **Edit Filters** (the pencil icon).
4. In the **Search** field, search for the field that you enabled in Adaptive Search and limit values as needed. Click **OK**.
5. Click **Save As** (the floppy disk icon) and name the search how you want it to appear to users in the CX Sales Mobile picker filter.

- In the **Shared** field, choose to make this search available to **Everyone** or **Specific Roles**. Click **Create**.

Add the Filter and the Dynamic Choice List fields to the Mobile Pages

Now you need to add the filter criteria field to its CX Sales Mobile picker page.

- Create a publishable sandbox that includes Application Composer.
- Open Application Composer by selecting **Application Composer** under the Configuration category in the Navigator menu.
- Under the Common Setup Menu, or on the Overview page, click **Mobile Application Setup**.
- Click the **CX Sales Mobile Composer** tab.
- In the Application Features pane, click the object that contains the filter criteria field. For example, if you're filtering the dynamic choice list by Account Type, select the Account object.
- In the object Views pane, click the **Picker** layout. Select an active Picker layout, or create one.
- Click a field that isn't being used and select the filter criteria field. Click **Save**.
- Navigate to the object where the dynamic choice list field exists. In the object Views pane, click the **Summary** page layout. Select an active layout, or create one.
- Click a field where you would like the filter criteria field to appear and select the field. Click **Save**.
- In the object Views pane, click the **Edit** page layout. Select an active layout, or create one.
- Click a field where you would like the filter criteria field to appear and select the field. Click **Save**.

Create Mobile Scripts to Validate Selected Values

Finally, to ensure that users are only selecting values from the filtered record set, two mobile scripts need to be created in Application Composer.

One script runs when the user changes the dynamic choice list field value, ensuring it meets the filter criteria of the saved search. This is important in case the user hasn't taken the one-time action to change the filter on the dynamic choice list to the Workspace Search mentioned above. The second runs when the user attempts to save the record, and also ensures that the dynamic choice list meets the required filter criteria of the saved search.

- Create a publishable sandbox that includes Application Composer.
- Open Application Composer by selecting **Application Composer** under the Configuration category in the Navigator menu.
- Under the Common Setup Menu, or on the Overview page, click **Mobile Application Setup**.
- Click the **CX Sales Mobile Scripts** tab.
- In the CX Object pane, navigate to the object under which you created your filtered dynamic choice list field.
- In the object's Script pane, click the + icon to create a new script, and enter a script name and description. Click **Create**.
- Create the first script with these details:
 - In the object's Events pane, select **On Field Value Change**.
 - In the Select Dependent Field pane, choose your filtered dynamic choice list field.
- For the second script, select **Before Save** in the object's Events pane.
- Here is an example script where the dynamic choice list filter exists in the Account object. The same syntax is used for both the **On Field Value Change** and the **Before Save** scripts::

```
var oracleCxmOutcome = new Result('');
const optyRow = getCurrentRow();
const accountsQuery = query('accounts'); //query accounts or another object, depending on the use case
accountsQuery.setParameters('PartyId', optyRow.getColumn('account_dcl_Id_c')); //be sure to use the ID
field for your DCL here, not the name field
try {
const accountsResponse = accountsQuery.execute();
if (accountsResponse && accountsResponse.length > 0) {
if(accountsResponse[0].getColumn('OrganizationDEO_LSAccountType_c') !== 'SERVICE_CENTER'){ //This checks
to see if the filter criteria applied to the DCL are met
```

```
oracleCxmOutcome.setMessage('MESSAGE_TYPE_ERROR', '', 'You must choose an organization that is a
Service Center. Choose the Service Centers list in the Show filter.');
```

```
oracleCxmOutcome.setOutcomeQualifier("OUTCOME_TYPE_FAILURE"); //The error message here indicates that
Service Centers is the name of the saved Workspace Search, and is the filter the user should select
}
}
} catch (e) {
//failed to get accounts response
}
```

10. Validate the script and click **Save**.

Additional Configurations

Configure Business Card Scan

After you set it up, salespeople can create a new contact or lead in the app by scanning a business card.

Enable Business Card Scanning

Business card scanning is enabled for all geographic regions by default, but you can enable the feature for only specific regions. Here's how:

1. Sign in to the Oracle CX Sales application as user with a Sales Administrator role.
2. Create a publishable sandbox that includes Application Composer.
3. Navigate to **Configuration > Application Composer**.
4. In the Common Setup Menu, or on the Overview page, navigate to **Mobile Application Setup > CX Sales Mobile Composer**.
5. In the Application Features pane, click **Settings**.
6. Expand the **Card Scan Settings** section.
7. To select specific regions, select them in **Regions to Enable Card Scan**.

Note: If you don't select any regions, then the feature is enabled for all regions.

8. Click **Save**.
9. Test and publish your the settings. See *Test Your Configurations*.

Disable Business Card Scanning

You can disable the business card scanning feature. Here's how:

1. Follow steps 1 to 6 above.
2. Disable the **Enable Card Scan** option.
3. Click **Save**.
4. Test and publish the setting. See *Test Your Configurations*.

How do I set up the "What's Around Me?" maps feature?

Salespeople can access the "What's Around Me?" map feature directly from the CX Sales Mobile home page. The application maps the salesperson's location using phone location, displaying nearby leads, accounts, contacts, and opportunities.

How It Works

To display the map, the phone uses the phone's default map. Records display as pins on the map, with icons that distinguish the record type. Tapping a pin displays a card showing record details and the actions that salespeople can do next, such as opening a contact record.

The card's **Get Directions** button uses the phone's native map application to guide salespeople to their destinations. Actions such as **Contact**, **Edit**, and **Convert Lead** are available directly from the cards, meaning that salespeople remain in context when acting on the information that's presented on the map.

Note: Records with addresses in China, Japan, and South Korea aren't geocoded or shown on maps.

Run Scheduled Processes to Enable

The map feature is enabled in the app by default. But, you must run or schedule processes. Here's how you run the scheduled processes:

Prerequisite: Enable geocoding for a country using the steps in [Set Up Geocoding](#).

1. Sign into Sales as a user with a Sales Administrator role.
2. Navigate to **Tools > Scheduled Processes**.
3. In the Overview page, click **Actions > Schedule New Process**.
4. Click the **Name** list button and search for the `Populate Location Latitude and Longitude Information` PROCESS, and then click **OK**. This process populates the latitude and longitude information for every sales object except Leads, which is handled by a different process, as described later in this procedure.
5. Enter the parameters, such as the start date and end date, and schedule the job to run automatically at specified intervals.
6. Click **Submit**.
7. Carry out the relevant steps again. This time search for `Populate Lead Latitude and Longitude Information`. This process populates the latitude and longitude information for leads.

Optionally Enable Saved Search Filters

You can use Workspace saved searches to filter the sales objects that are near a salesperson's location. If you want to enable this filter, follow the steps in the [Enable Salespeople to View Saved Search Results on a Map](#) topic.

Related Topics

- [Set Up Geocoding](#)

Enable Salespeople to View Saved Search Results on a Map

You can make it possible for salespeople to display the records returned by saved searches on a map while using the What's Around Me? map feature. Salespeople can view accounts, contacts, opportunities, and leads nearest to their location.

This feature supports saved searches powered by Adaptive Search, including those saved searches created in Workspace and Digital Sales. For this feature to work, you need to enable the account, contact, and lead objects' GeoLocation field for Adaptive Search. Here's how you do that:

1. Navigate to the **Setup and Maintenance work** area, and use the following:
 - o Offering: Sales
 - o Functional Area: Sales Foundation
 - o Show: All Tasks
 - o Task: Configure Adaptive Search
2. Ensure the account, contact, opportunity, and lead objects are enabled for Adaptive Search. If required, see the [Enable Business Objects for Adaptive Search](#) topic for more information.
3. In the **Configure Adaptive Search** page, click the **Setup** tab and then click the **Advanced** subtab.
4. In the **Advanced** subtab, click **Account > Primary Address > Address** to view the Address fields.
5. In the **Configure Fields** section, find the **GeoLocation** field.
 - a. Click the **Enable** option to index the field for Workspace and Global Search.
 - b. Select the **Include in Object Reference** option to enable the field for use in searches of related objects.
6. In the **Advanced** subtab, click **Contact > Primary Address > Address**.
7. In the **Configure Fields** section, find the **GeoLocation** field. Click the **Enable** and **Include in Object Reference** options.
8. In the **Advanced** subtab, click **Lead**.
9. In the **Configure Fields** section, find the **GeoLocation** field. Click the **Enable** and **Include in Object Reference** options.
10. After making your selections, click **Publish**. Publishing triggers an indexing process that you can monitor by clicking the **Monitor** tab and then the **Publish** subtab. Your new configuration is available when the indexing process completes.

Opportunities use the GeoLocation field of the opportunity's associated account, so no additional configuration is required for the opportunity object.

Enable Sales Recommendations for CX Sales Mobile

Here's how to make the sales recommendations you set up earlier in CX Sales Mobile. Sales recommendations use natural language processing to analyze meeting minutes and recommend salespeople to create contacts and to set up follow-up appointments and tasks.

Note: See the [Tips for Using Meeting Recommendations](#) topic for more information about how the recommendations are generated.

Two steps are required to enable the sales recommendations that you set up as described in the Sales Recommendations Chapter ([Overview of Sales Recommendations Setup](#)):

1. Create a profile option that enables the recommendation feature for the CX Sales Mobile app. (CXM_ENABLE_RECOMMENDATIONS).
2. Enable the Recommendation Feature in the CX Sales Mobile Configuration Tool.

Create the CXM_ENABLE_RECOMMENDATIONS Profile Option

1. Navigate to the Setup and Maintenance work area. Click the **Tasks** side panel icon and then click the **Search** link.
2. Search for and select the **Manage Profile Options** task.
3. Create a new profile option, using these details:
 - o Profile Option Code: CXM_ENABLE_RECOMMENDATIONS
 - o Profile Display Name: CXM_ENABLE_RECOMMENDATIONS
 - o Application: Mobile Sales
 - o Module: Mobile Sales Server
 - o Start Date: Enter today's date
4. Click **Save and Close**.
5. In the **Profile Values** section, for the **Site** level, check the **Enabled** and **Updatable** check boxes.
6. Click **Save**.
7. Next, you need to set the values for the profile option. Search for and select the **Manage Administrator Profile Values** task.
8. Search for the CXM_ENABLE_RECOMMENDATIONS profile option you have just created.
9. In the **Profile Values** section, navigate to the **Site** profile level, and enter the **Profile Value** as **Yes**.

Enable the Recommendation Feature in the CX Sales Mobile Configuration Tool

Enable the recommendation feature in the CX Sales mobile configuration tool, as the recommendation feature isn't enabled by default.

1. In a publishable sandbox, navigate to **Application Composer > Mobile Application Setup > CX Sales Mobile Composer > Settings**.
2. In **Recommended Settings**, activate the **enabled** option. After the feature is enabled, it's available to all CX Sales Mobile users.
3. If required, you can make this feature active for only certain regions or job roles.

Note: If you restricted sales recommendations to specific job roles and regions while setting up the Sales Recommendations feature, those settings take precedence. See the topic: [Associate the Webhook Channel to Sales and Specify Recommendation Options](#).

Tips for Using Meeting Recommendations

Here are some tips for getting the best quality recommendations for CX Sales Mobile and some more information about how it works.

How the Feature Works

- After you have created a call report or an appointment, the sales recommendations feature analyzes the call report meeting notes and the appointment description field and recommends future actions, such as creating follow-up meetings or future tasks. For example, entering “create a follow-up meeting” in the meeting notes triggers a recommendation to create an appointment.

Note: Call report meeting notes and appointment descriptions are only analyzed if the call report is created in the CX Sales Mobile app.

- As salespeople usually enter meeting notes after a meeting has taken place, the appointment description is analyzed after the appointment time has passed. For example, if a salesperson creates an appointment for the next day at 10 AM, the appointment description won't be analyzed until the salesperson enters text in the description after 10:00 AM the next day.
- Because the information in the meeting minutes is interpreted using the app's natural language capabilities, phrases such as "create a meeting for next Friday" are calculated to the correct date.

Note: If the call report meeting notes or appointment description doesn't mention a time zone, the recommendation is created based on the sales representative's time zone selected in the Oracle CX Sales web app. If the mobile device is in a different time zone, CX Sales Mobile converts the time in the recommendation to match the mobile device's time zone.

- When you refer to a person's name in the minutes, such as "give Lisa the project plan", the app searches for the person in the existing sales data as follows:
 - a. The contacts in the call report are searched and if there's a match the app adds the contact to the recommendation.
 - b. If the person hasn't been found, the app searches the account contact list.
 - c. If the app matches the person, then the person is added as a contact to the recommended appointment or task.

Note: In these searches, the app uses the account associated with the call report or appointment.

If more than one contact is found, then a contact won't be added to the recommended appointment or task. Salespeople can create a contact if there isn't one, or pick from a list if there's more than one.

- Create contact recommendations are provided if the name referred to in the note isn't found in the:
 - Call report
 - Appointment's contact list
 - Account's contact list

The app then looks at the smart list for a matching name, and if no match is found, then it recommends creating a contact for the account.

Note: If the call report isn't associated with an account and the contact is found in the smart list, then the contact is added as a contact to the recommended appointment or task.

- If a sales representative accepts the suggested recommendation, they are prompted to review the information before saving. After it's saved, the task or appointment is created with pre-populated information that has been gathered from the appointment description, or the call report meeting minutes.

- Recommendations expire automatically 7 days from when the recommendation was created.

If it is an appointment or task and the recommended appointment/task has a start date/due date, the recommendation for it will expire the next day. It will fall back to 7 days if the appointment/task doesn't have a start date or due date.

Tips

- To receive the highest quality recommendations, meeting minutes need as much detail as possible. For example, entering "create a meeting at 4" could mean creating the meeting at 4 AM or 4 PM, so it's best to enter "create a meeting at 4 PM".
- Because the feature uses natural language processing, there could be times when a recommendation isn't generated, or some of the information isn't interpreted correctly. Therefore, sales representatives should review the recommendation for accuracy.
- If a call report is edited, the recommendations from the original call report are cleared and new recommendations are generated.
- The recommendations feature only supports English at the moment.

Enable or Disable Confirmation Messages Globally, by Region, and by Role

Configure which roles and regions receive confirmation messages when a record is created or a lead is converted. Or, you can disable these messages for all users.

You can choose which roles and locales receive confirmation messages when records are created, or leads are converted. You can also disable confirmation messages for all of your users.

Note: Confirmation messages are enabled by default, so your organization will continue to receive confirmation messages as long as you don't disable them for your organization, or for particular roles or regions.

Here's how you configure who receives confirmation messages:

1. In a publishable sandbox, navigate to **Application Composer** > **Mobile Application Setup** > **CX Sales Mobile Composer**.
2. In the **Application Features** pane, click **Settings**.
3. Expand the **Confirmation Message Settings** header.
4. Ensure that the **Enabled** toggle is active.
5. In the **Assigned Regions** and **Assigned Roles** fields, select the regions and roles that will receive confirmation messages.
6. Alternatively, you can stop confirmation messages for all of your users by making the **Enabled** toggle inactive.
7. In the header, click **Save**.
8. Test your update by following the steps in the [Test Your Configurations](#) topic.
9. Publish the Sandbox.

Embed Links to Custom Apps in the Main Menu

You can add a web link to the CX Sales Mobile application's main menu that opens a custom web or mobile app when you tap the link. Once the web link is set up, it appears just like the other Sales objects in the main menu, so that it integrates seamlessly into the CX Sales Mobile app.

Here's how you add the web link to the main menu:

1. Sign in to the Oracle CX Sales application as user with a Sales Administrator role.
2. Select the sandbox you want use for your configurations.
3. Open Application Composer by selecting **Application Composer** under the Configuration category in the Navigator menu.
4. Under the Common Setup Menu, or on the Overview page, click **Mobile Application Setup**.
5. Click the **CX Sales Mobile Composer** tab.
6. In the Application Features pane, click **Home**.
7. In the mobile interface designer located in the middle of the page, click **+ Add Feature**.
8. Click the **Add Links** button at the bottom of the window.
9. In the Display Label field, enter a name for the link. This name will appear in the main menu along with the existing main menu entries.
10. In the Link field, enter the web link, or the custom app link, that will open when the menu item is tapped.
11. Select an icon for the link and click **Finish**.
12. If you want to re-order the main menu items, drag the entry in the interface designer to the desired position.
13. Click **Save**.
14. Test that the link appears in the main menu and publish the sandbox when you're happy with your updates. See the [Test Your Configurations](#) topic for details about how to check your mobile configurations.

Embed Links That Open a Web App or Another Mobile App

You can create embedded links in CX Sales Mobile that automatically open web apps, or another mobile app, when you tap the link. Here's how:

1. Sign in to the Oracle CX Sales application as user with a Sales Administrator role.
2. Select the sandbox you want use for your configurations.
3. Open Application Composer by selecting **Application Composer** under the Configuration category in the Navigator menu.
4. Expand the menu for a standard or custom object and then select **Fields**.
5. Create a new custom field and select the **Formula** option.
6. Enter the formula field details. Make sure that you select a **Formula Type** of **Text** and that **Include in Service Payload** is selected (this will make the field appear in the mobile interface designer, so that you can add it to a mobile page). Click **Next**.
7. Select the relevant fields in the **Depends On** field.
8. In the **Edit Script** area, create a Groovy script that will generate a dynamic web URL with a prefix of `http://` or `https://`. If you're creating a Groovy script that will open a custom mobile application, you need to use a prefix of the app reference and then `://`, for example `oscm://`. Click **Submit**.
9. Navigate to the main Application Composer page and click **Mobile Application Setup**.
10. Select the object that you created the Groovy script for, and click the **Detail** page layout.
11. Add the formula field you have just created to the detail page and click **Save**.

Note: The field is rendered in a different format depending on where it's placed on the page. For example, if it's placed in a page header or information section, the field label displays as a link. If you add the field to the two column or single column sections, it appears with a label and a URL value.

12. Test your updates and publish the sandbox when you're happy with the configuration. See the [Test Your Configurations](#) topic for details about how to check your mobile configurations.

Track Leads and Opportunities Using the Business Unit Field and Define Page Layouts by Business Unit

You can add the business unit field to an opportunity or lead page layout, enabling you to track opportunities and leads by business unit.

Also, you can define opportunity and lead page layouts by business unit, allowing you to create business unit specific layouts that display relevant fields in order of preference, making it quicker for your sales representatives to enter and update their opportunities or leads. You can also create CX Sales scripts that use the business unit field, meaning that you can, for example, enforce validations or business logic specific to a business unit, or mark fields as mandatory based on selected business unit.

Add the Business Unit Field to an Opportunity or Lead Page Layout

The business unit field is hidden by default, so here's how you add the field to the opportunity or lead List, Summary, or Edit page layout:

1. Sign in to the Oracle CX Sales application as user with a Sales Administrator role.
2. Select the sandbox you want use for your configurations.
3. Open Application Composer by selecting **Application Composer** under the Configuration category in the Navigator menu.
4. Under the Common Setup Menu, or on the Overview page, click **Mobile Application Setup**.
5. In the Application Features pane, click **Leads** or **Opportunities**.
6. In the Layouts pane, select the relevant page layout, or click the **Click to Clone** icon for the standard layout if you want to create a new page layout.
7. In an opportunity layout, add the **Business Unit ID** field to the page layout, in a lead layout add the **Business Unit Identifier** field.
8. Click **Save**.
9. Test and publish your new page layouts. See the [Test Your Configurations](#) topic for details about how to check your mobile configurations.

Once you have added the field to opportunity and lead pages, the business unit value displays automatically for sales representatives who work in one business unit. Sales representatives who work across multiple business units can select the relevant business unit when they create an opportunity or lead.

Create Lead and Opportunity Page Layouts for Business Units

You can create a new page layout that's displayed for specific business units, or assign a business unit to an existing page layout. Here's how:

1. Create a lead or opportunity page layout, or select an existing page layout.
2. Add the Business Unit field to a lead and opportunity page layout (see the steps in the **Enable the Business Unit Field** section above).
3. In **Advance Criteria**, select **Add**.

4. Select **Business Unit ID** for opportunity layouts or **Business Unit Identifier** for leads layouts, the operator, and the relevant business unit field value.
5. Click **Save**.
6. Test and publish your new page layouts. See the [Test Your Configurations](#) topic for details about how to check your mobile configurations.

Create Custom Scripts for Business Unit Page Layouts

Using the CX Sales scripting feature, you can help guide sales representatives when they're creating or updating sales records by enforcing validations or business logic specific to a business unit, or specify field behavior based on selected business units. For example, you can make fields mandatory for a specific business unit page layout, or you can hide the business unit field if you don't want a sales representative to see it. For more details about creating scripts, see the [Configure the App with Custom Scripts](#) topic.

Add Dynamic Choice List Fields for Territories and Languages

Create dynamic choice list fields for territories and languages, and then add them to list, summary, or edit layouts for any standard or custom objects. Here's how:

1. In Application Composer, navigate to **Objects > Standard Objects > Contacts > Fields**.
2. Click **Create Custom Field**. In the dialog window that pops up, click **Choice List (Dynamic)**. Then click **OK**.
3. Enter the required fields, including Display Name and click Next. In the List Data Source section, select Related Object as Sales Territory. The newly created Territory Dynamic Choice List is now available as a custom field.
4. In Application Composer, navigate to **Common Setup > Mobile Application Setup**.
5. Click the **CX Sales Mobile Composer** tab.
6. In the mobile interface designer, scroll to the bottom and click **Add Feature**.
7. In the dialog box, select **Sales Territories**. Then click **Next**.
8. Pick a suitable icon from the list of icons. Then click **Next**.
9. Click **OK**.
10. Navigate to the Sales Territories Layout from the Application Features pane.
11. Select the layout where you want to add the newly created Dynamic Choice List as a field. In this case for example, clone the list layout and the edit or create layout.
12. Add the field that you created in steps 1 through 3. Sales Territory as a Dynamic Choice List is now enabled.

To enable Language Dynamic Choice Lists, follow these steps:

1. In Application Composer, navigate to **Objects > Standard Objects > Contacts > Fields**.
2. Click **Create Custom Field**. In the dialog window that pops up, click **Choice List (Dynamic)**. Then click **OK**.
3. Enter the required fields, including Display Name, and then click **Next**.
4. In the List Data Source section, select the Related Object as **LanguagePVO**. The newly created Dynamic Choice List is now be available as a custom field. You can now add the Dynamic Choice List as a field in any page layout.

Use Filters on Dynamic Choice List Field Pickers

Use simple or advanced filters on dynamic choice list fields, so that the corresponding object picker displays a filtered set of records. When users use the object picker, they see the records that meet the filter criteria, so that they choose from the correct set of records.

Here's how to add a filter to dynamic choice list fields that appear in your Edit page layouts:

1. In a publishable sandbox, navigate to **Application Composer** > **Mobile Application Setup** > **CX Sales Mobile Composer**
2. In the **Application Features** pane, click the relevant object or child object.
3. In the object's **Views** pane, click the **Edit** page layout.
4. In the **Layouts** pane, select an existing page layout, or create a new one by clicking the **Click to Clone** icon.
5. In your page layout, add your dynamic choice list field. If the dynamic choice list field is already in the layout, remove it and add it to the page layout again.
6. Hover over the dynamic choice list field and click the pencil icon to view the options.
7. Configure the following dynamic choice list settings:
 - o **Use Filter Criteria:** Enable this option to use the dynamic choice list's filter criteria. If it's not enabled, CX Sales Mobile uses the Workspace saved search associated with the dynamic choice list target object as the filter.
Note: The app doesn't support advanced filters that have Groovy expressions where the bind variable value is conditionally derived from the Groovy expression. In this instance the filter that's used is the Workspace saved search associated with the dynamic choice list target object.
 - o **Show List by Default:** Enable this option to display the list of records by default when the user taps the picker.
 - o **Choose Search Field:** Select which field is used when the user searches for a record.
 - o **Choose Search Operator:** Choose the appropriate search operator.
8. Click **Ok**.
9. In the header, click **Save**.
10. Test your updates by following the steps in the [Test Your Configurations](#) topic. Make sure that you force close the CX Sales Mobile app, before opening the app again, and verifying that the dynamic choice list filters are being used correctly.
11. Publish the sandbox to make the updates available to all users.

Enable the App to Retain a User's Last Used Saved Search for Dynamic Choice List Pickers

Dynamic choice list pickers now retain the user's last used saved search on each picker separately from the last used saved search for the object's list view.

When you use a saved search on a dynamic choice list picker to filter the records displayed on a picker, each field retains the last used saved search by the user.

Here's how you enable the feature:

1. In a publishable sandbox, navigate to **Application Composer** > **Mobile Application Setup** > **CX Sales Mobile Composer**.
2. In the **Application Features** pane, click **Settings**.
3. Expand the **Dynamic Choice List Settings** heading.
4. Enable the **Use last used saved search**.
5. In the header, click **Save**.
6. Test your updates by following the steps in the [Test Your Configurations](#) topic.
7. Publish the sandbox to make the updates available to all users.

Configure the App for Partner Relationship Management

Channel managers and Partner users can leverage the CX Sales Mobile app to manage and grow channel sales efficiently on the go. Channel managers can manage partners, partner contacts, and other partner related information. Partners can access all of the standard sales features, such as leads and opportunities, so that they can track and manage their partner sales activities.

All of the Partner Relationship Management features are available as standard for the following roles:

- Partner Sales Representative
- Partner Sales Manager
- Partner Administrator
- Channel Account Manager
- Channel Sales Manager
- Channel Operations Manager

The Partner and Partner Contact Management features are available as standard for these roles:

- Channel Account Manager
- Channel Sales Manager
- Channel Operations Manager

You can also configure CX Sales Mobile to suit your particular channel and partner management business requirements, and allow other roles access to PRM features. Here's what you can configure:

Configure the Home Page for Channel Manager Roles

All PRM roles have access to their pipelines, tasks, and appointments on the home page by default. In addition, you can add a partner card containing high-level partner information to the home page, which enables channel managers to get a summary of their partner sales information. Here's how:

1. Sign in to the Oracle CX Sales application as user with a Sales Administrator role.
2. Select the sandbox you want use for your configurations.
3. Open Application Composer by selecting **Application Composer** under the Configuration category in the Navigator menu.
4. Under the Common Setup Menu, or on the Overview page, click **Mobile Application Setup**.
5. Click the **CX Sales Mobile Composer** tab to open the CX Sales Mobile configuration tool.
6. In the Application Features pane, click **Cards**.
7. In the Layouts pane, create a new home page layout by clicking the **Click to Clone** icon.
8. Enter a layout name and click **Ok**.
9. In the Assigned Roles pane, select the channel managers roles that will view the home page.
10. Scroll to the bottom of the interface designer and click **+ Add Custom Card** at the bottom of the home page.
11. Select the Partner object card from the list and then on the next page select the Workspace saved search for the card.
12. Edit the header and title, and add an aggregation field and footer. You can hide the total number of relevant partner objects by clicking the eye icon, ensuring it has a cross through it. You can also change the saved search by clicking the more button (three horizontal dots) on the card.

13. If you want to hide the pipeline, tasks, and appointments cards, click the relevant card in the interface designer, and disable the **Show Card** option in the Card Settings pane.
14. If you want to rename the strings on the cards, you need to use the I18N feature. See the [Rename Child Objects and Strings in the App](#) topic for more details.
15. Click **Save**.
16. Test and publish your new page layouts. See the [Test Your Configurations](#) topic for details about how to check your mobile configurations.

Allow or Restrict Access to the Partner Object in the Main Menu

The Partner object appears in the main menu out of the box for these channel manager roles:

- Channel Account Manager
- Channel Operations Manager
- Channel Sales Manager

If you would like to allow another role to have access to the Partner object in the main menu, or you want to hide the Partner object for a particular channel manager role, follow these steps:

1. Follow the first five steps in the **Configure the Home Page for Channel Manager Roles** section above.
2. In the interface designer, click the main menu and then click the Partners object.
3. In the Feature Details pane, click **Assigned Roles** and add or remove roles by either searching for a role to add a role, or click the cross icon to remove a role.
4. Click **Save**.
5. Test and publish your new page layouts. See the [Test Your Configurations](#) topic for details about how to check your mobile configurations.

Set up Page Layouts for the Partner Object

You can create your own page layouts for the Partner object, so that the pages reflect your business requirements. You can create List and Summary page layouts, and you can also configure the Partner picker and global search. For information about creating these page layouts, see the following topics in the Implementing Sales guide:

- [Configure the App Using the Configuration Tool](#)
- [Configure the List Page Layout](#)
- [Configure the Summary Page Layout](#)
- [Configure the Picker Layout](#)
- [Configure the Global Search Layout](#)

Note: There isn't an Edit page layout available for the Partner object.

Add the Partner Field to Sales Object Page Layouts

You can add partner fields to other sales object page layouts, such as the Primary Partner field in opportunity page layouts and the Partner Party Name field in task page layouts. See the [Configure the App Using the Configuration Tool](#) topic for information about how to add fields to page layouts.

Add the Partner Object to Summary Page Layouts

You can also add the Partner object as a related object in other sales object summary page layouts, such as the opportunity summary page layout, so that channel managers can see the related partner information for the respective

object. See the **Child and Related Object** sections of the [Configure the Summary Page Layout](#) topic for information about how to add a related object to a summary page layout.

Enable Quotes for CX Sales Mobile

Sales representatives can quickly access quotes related to their accounts and opportunities from CX Sales Mobile. Additionally, they can share the proposal document with the team for feedback, or email proposals to customers from the mobile app to speed up the negotiation cycle.

Here are the two steps you must complete to provide access to quotes in CX Sales Mobile:

1. Create a confidential application for CX Sales Mobile in Oracle Identity Cloud Service.

Note: If you already completed this step as described in the topic [How do I enable sales representatives to stay signed in to the CX Sales Mobile app?](#), you can just edit the application you created previously and add the new information. You don't have to create a new application.

2. Display the quotes and lists of quotes in the CX Sales Mobile UI.

Prerequisites: Before you start, you must complete all the steps to integrate Oracle Sales and CPQ Cloud. For details, see the [Integrating Oracle CX Sales with Oracle CPQ](#) guide.

Create a Confidential Application for Oracle CX Sales Mobile in Oracle Identity Cloud Service

Here's how you create CX Sales Mobile as a Confidential Application in Identity Cloud Service:

1. Sign in to My Console in Identity Cloud Service, by navigating to **www.oracle.com > View Accounts** and enter the name of the relevant pod in the Cloud Account Name field. You have at least two pods, a test and a production pod, so ensure that you enter the correct pod.
2. Click **Next** and enter your Identity Cloud Service password. If you can't remember the password, contact Oracle Support for help.

Note: For more information about Identity Cloud Service, navigate to <https://docs.oracle.com>, search for Identity Cloud Service, and click Oracle Identity Cloud Service - Get Started.

3. In the Identity Cloud Service console, expand the **Navigation Drawer**, and then click **Applications**.
4. Click **Add**. Then on the **Add Application** page, click **Confidential Application**.
5. In the **Details** tab of the **Add Confidential Application** wizard, enter information in the **Name** and **Description** fields. These fields are used to identify the Confidential Application in Identity Cloud Service only, they don't appear outside of Identity Cloud Service. For example, they don't appear in the Oracle CX Sales app.
6. Click **Next**.
7. In the **Client** tab, click the **Configure this application as a client now** option.
8. In the **Allowed Grant Types** options list, select these options:
 - o **Refresh Token**
 - o **Authorization Code**
9. Select the **Allow non-HTTPS URLs** option.

Note: This option doesn't open a URL that isn't secure, it just sends a request to the CX Sales Mobile app.

10. In the **Redirect URL** field, enter `cxm://?oAuth=true`
11. Click **Next**.

12. In the **Resources** tab, enter a value in the **Access Token Expiration** field. This defines how long the access token associated with CX Sales Mobile remains valid.
13. Select the **Is Refresh Token Allowed** option and enter a value in the **Refresh Token Expiration** field. This defines how long the access token can be refreshed until the sales representative needs to sign into the CX Sales Mobile app again.

When an access token expires, the CX Sales Mobile app uses a valid refresh token to get a new pair of access and refresh tokens. If the refresh token isn't valid, or has expired, the sales representative has to sign in again. If the refresh token is valid the CX Sales Mobile app uses it to get a new access token, which is used to validate subsequent access to the app.

Note: It's recommended to keep the access token validity to a short duration and the refresh token to a long duration. For example, 30 minutes for the access token and 7 days or more for the refresh token.

14. In the **Primary Audience** field, add your host URL (see the [Get Started with Your Mobile Implementation](#) topic for more information about finding out your host URL).
15. In the **Secondary Audience** field, fill in your CPQ domain (for example, `https://xxxx-xxx.bigmachines.com`).
16. In the **Scopes** section, click **Add**, and enter the value `/cxmobile`.
17. In the **Scopes** section, click **Add**, and your CPQ scope. The CPQ scope will be in the form `<CPQ_domain>/api` (for example, `https://xxxx-xxx.bigmachines.com/api`).
18. Under Resources, select **Add Scope**.
19. Select the **Resource Client Oracle Applications (Fusion)**, which is the available resource provided by Oracle.
20. Add all required scopes from the resource client, including the CX Sales Mobile and CPQ scopes you just created.
21. Click **Next** to navigate to the **Web Tier Policy** tab. Nothing needs to be entered here.
22. Click **Next** to navigate to the **Authorization** tab. Nothing needs to be entered here.
23. Click **Finish**. The CX Sales Mobile application has been added in a deactivated state.
24. Record the **Client ID** and **Client Secret** that appear in the **Application Added** dialog box. The **Client ID** and **Client Secret** are equivalent to a credential (for example, an ID and password) that your application uses to communicate with Identity Cloud Service.
25. Click **Close**.
26. At the top of the page, to the right of the application name, click **Activate**.
27. In the **Activate Application?** dialog box, click **Activate Application**.

For more information about creating a Confidential Application, navigate to <https://docs.oracle.com>, search for **Identity Cloud Service**, click **Oracle Identity Cloud Service - Get Started**, and click **Manage confidential Applications** under the Manage Applications heading.

Display Quotes and Quote Lists in the Mobile App

1. Enable Quote lists in the Navigator to display the Quote and Orders saved searches setup in Workspace:

Note: These steps are optional. You only need to enable Quote lists in the Navigator if you want your users to have access to all of their quotes in aggregate.

- a. Enter a sandbox with Application Composer and click to **Application Composer > Mobile Application Setup**.
- b. Click the **CX Sales Mobile Composer** tab.
- c. In the Application Features pane, click **Home**.
- d. Scroll down in the mobile interface designer pane and select the **Quotes and Orders** feature.
- e. In the Feature Details, toggle **Show On Navigator** to On.

2. Quotes and Orders are enabled by default in the summary views of Accounts and Opportunities. If associated quote records exist for the parent records, then quote pickers appear in the Account and Opportunity Summary pages automatically. However, if you created page layouts before release 21C and you want quotes to appear in these pages, then you need to enable quotes by following these steps:
 - a. In Summary view of the Accounts feature, select your custom page layout.
 - b. In the mobile interface designer pane, select **Add Child or Related**.
 - c. From the list of options, select `SalesOrders`.
 - d. Repeat these steps for your Opportunity Summary page layouts.
3. Click **Save**.
4. Test and publish your configurations. See the [Test Your Configurations](#) topic for details.

Enable Price Books for Opportunities

Improve the speed and accuracy of sales representatives' pricing discussions with their customers by bringing Account and Opportunity based pricing into CX Sales Mobile.

To get started with enabling this feature, first follow all of the steps listed in the [Price Books in Opportunities and Accounts](#) section of the Price Books chapter, available in the Implementation Reference guide.

After completing this, continue with these steps to enable for CX Sales Mobile.

Add the Price List Fields You Created to Mobile Account and Opportunity Pages

Add the Price List fields you created in the “Create Price Book Field for Accounts and Opportunities” topic to their respective CX Sales Mobile Account and Opportunity Summary and Edit pages. Here’s how to add the fields:

1. Sign in to the Oracle CX Sales application as user with a Sales Administrator role.
2. Create a publishable sandbox that includes Application Composer.
3. Open Application Composer by selecting **Application Composer** under the Configuration category in the Navigator menu.
4. Under the Common Setup Menu, or on the Overview page, click **Mobile Application Setup**.
5. Click the **CX Sales Mobile Composer** tab.
6. In the Application Features pane, click **Accounts**.
7. In the Accounts Views pane, click **Summary**.
8. Click the **Click to Clone** icon next to the default Account Detail layout and name the new layout. Click **Ok**.
9. Open the Assigned Roles pane and search for the custom Sales Representative role with the Manage Price Books role. Add the role by clicking the role name.
10. Add the Price List field to this layout by selecting **+ Add Field** in the location you would like the field to appear. Search for the Price List field and click the field name to add it to the layout.
11. In the header, click **Save**.
12. In the Accounts Views pane, click **Edit**
13. Click the **Click to Clone** icon next to the default Account Detail layout and name the new layout. Click **Ok**.
14. Open the Assigned Roles pane and search for the custom Sales Representative role with the Manage Price Books role. Add the role by clicking the role name.
15. Add the Price List field to this layout by selecting **+ Add Field**. Search for the Price List field and click the field name to add it to the layout.
16. In the header, click **Save**.
17. Repeat steps 6 to 16 for Opportunities.

Create a Script to Apply Updated Pricing When the Price List Field Value Is Changed

Next, create a mobile script that updates the unit pricing for Revenue Line Items, multiplies them by Quantity to get the Line Item Totals, and updates the Opportunity Revenue when the Price List field value is changed. Here's how to create the script:

1. Click the **CX Sales Mobile** Scripts tab.
2. In the CX Objects pane, click **Opportunities**
3. In the Opportunities Event pane, click **On Field Value Change**.
4. In the Select Dependent Field pane, choose **Price List**.
5. The Price List Scripts pane appears above the Select Dependent Field pane. Click the **+ icon** in the Price List Scripts pane to add a script.
6. In the Script Name field, enter **OnFieldValueChange_PriceList** and add a suitable description in the Script Description field.
7. Click **Create**.
8. Enter the following script:


```
const row = getCurrentRow();
let priceListName = row.getColumn('PriceList_c');
if(!(priceListName == null)){
let q = queryWithParameters("priceBookHeaders", [{"field": 'Name', 'operator': 'contains', 'value': priceListName}]);
let rows = q.execute();
let priceBookItems = rows[0].getColumn('PriceBookItem').items;
let prodGroupItems = row.getColumn('ChildRevenue').items;
for(var i =0; i< prodGroupItems.length; i++){
for(var j =0; j< priceBookItems.length; j++){
if(prodGroupItems[i].Description === priceBookItems[j].ItemDescription){
prodGroupItems[i].UnitPrice = priceBookItems[j].ListPrice;
prodGroupItems[i].RevAmount = priceBookItems[j].ListPrice * prodGroupItems[i].Quantity;
break;
}
}
}
}
```
9. Click **Validate**.
10. In the header, click **Save**.

Create a Script to Apply Updated Pricing When the Revenue Line Item Is Changed or Added

Lastly, create a mobile script that updates or fills in the unit price, multiplies them by Quantity to get the Line Item Totals, and updates the Opportunity Revenue when a Revenue Line Item is changed or an additional item is added. Here's how to create the script:

1. Select the **CX Sales Mobile Scripts** tab.
2. In the CX Objects pane, click **Opportunities**.
3. In the Opportunities Child Objects pane, click **Opportunity Revenue**.
4. In the Opportunity Revenue Events pane, select **On Field Value Change**.
5. In the Select Dependent Field pane, search for and select **Product Group**.
6. The On Field Value Change Scripts pane appears above the Select Dependent Field pane. Click the **+ icon** in the On Field Value Change Scripts pane to create a new script.
7. In the Script Name field, enter **onfieldvaluechange_productgroup** and add a suitable description in the Script Description field.
8. Click **Create**.
9. Enter the following script:


```
const row = getCurrentRow();
```

```
let priceListName = row.getColumn('PriceList_c');
let q = queryWithParameters("priceBookHeaders", [{'field': 'Name', 'operator': 'contains', 'value':
priceListName}]);
let rows = q.execute();
let priceBookItems = rows[0].getColumn('PriceBookItem').items;
const childRow = getCurrentChildRow();
let productName = childRow.getColumn('Description');
for(var i =0; i< priceBookItems.length; i++){
if(priceBookItems[i].ItemDescription === productName){
childRow.setColumn('UnitPrice',priceBookItems[i].ListPrice);
break;
}
else {
childRow.setColumn('UnitPrice',null);
}
}
```

10. Click **Validate**.
11. In the header, click **Save**.

Test Your Configurations

Test your updates and publish the sandbox when you're happy with the configuration. See the [Test Your Configurations](#) topic for details about how to check your mobile configurations.

Prevent Users Copying Data from CX Sales Mobile

You can disable the copy, cut, and paste actions in the CX Sales Mobile app, so that you can prevent user from copying data from CX Sales Mobile to other applications on their mobile devices.

Disabling the copy, cut, and paste actions ensures data security by helping prevent data being copied and viewed outside of the app. Here's how you set it up:

1. In a publishable sandbox, navigate to **Application Composer > Mobile Application Setup > CX Sales Mobile Composer > Settings**.
2. Expand the **Security Settings** header.
3. Click the **Disable Cut, Copy, Paste** toggle.
4. In the header, click **Save**.
5. Test your update by following the steps in the [Test Your Configurations](#) topic.
6. Publish the Sandbox.

Disable App Screen Recording or Taking Screenshots of the App

Configure CX Sales Mobile so that users can't record the CX Sales Mobile app pages, or take screenshots of the pages in the app. This helps prevent data from being copied and viewed outside of the app, making the data secure.

Here's how you can disable screen recording and screenshot capture using Application Composer:

Note: This feature is available for Android devices only at present.

1. In a publishable sandbox, navigate to **Application Composer > Mobile Application Setup > CX Sales Mobile Composer > Settings**.
2. Expand the **Security Settings** header.

3. Click the **Disable Screen Capture** toggle to prevent your users recording the CX Sales Mobile pages and prevent them from taking screenshots of the pages.
4. In the header, click **Save**.
5. Test your update by following the steps in the [Test Your Configurations](#) topic.
6. Publish the Sandbox.

Use Oracle BI to Track CX Sales Mobile Usage Across Your Organization

You can learn who in your organization uses CX Sales Mobile, how frequently and in which channels by using the analytics built into the Transactional Business Intelligence tool.

To enable usage tracking, complete these 4 steps:

1. Enable the profile option that tracks usage.
2. Enable usage tracking in CX Sales Mobile Composer.
3. Track mobile usage by active/inactive days, users, and sign-ins.
4. Track mobile record create, read, and update by object.

Enable Profile Option That Tracks Usage

To monitor usage in Transactional Business Intelligence, you must verify that the FND_TRACK_USER_ACTIVITY profile option is enabled:

1. Navigate to Setup and Maintenance.
2. Click Tasks (the drawer on the right side of the page).
3. Click **Search**.
4. Search for and select the **Manage Administrator Profile Values** task.
5. Search for the Profile Option Code **FND_TRACK_USER_ACTIVITY**
6. Under Profile Option: Profile Values, ensure that Profile Level is **Site** and Profile Value is **ENABLED** (all caps).
7. Click **Save and Close**.
8. To apply these changes, run the process Refresh BI Reports Audit Data for User Adoption Reporting. This scheduled process summarizes and aggregates data used for user adoption reporting.

Enable Usage Tracking in CX Sales Mobile Composer

Verify that CX Sales Mobile usage tracking hasn't been disabled:

1. Enter a publishable sandbox.
2. Navigate to Application Composer.
3. Click **Mobile Application Setup**.
4. Under CX Sales Mobile Composer, Application Features, select **Settings**.
5. Expand **Usage Settings**
6. Ensure that the **Track Usage** switch is enabled.
7. Optionally, select the geographical regions where you want to track usage. Otherwise, leave this set to All.
8. Click **Save**.
9. Publish the sandbox if you made changes.

Track Mobile Usage by Active/Inactive Days, Users, and Sign-ins

To gain insights into Oracle CX Sales Mobile usage frequency, navigate to Oracle Transactional Business Intelligence, create a new Analysis, and select the **Resource System Usage** subject area.

This subject area allows you to track the following:

- **Facts:** Measures usage including Counts of Days, Counts of Users, and measures of Session Duration.
- **Channel:** Breaks out usage by channel, including Web, Mobile, and Email.

You can combine these measures with other subject areas to extend the reports and expand your insights. You can include items such as user region and division, for example.

Note that Mobile usage frequency uses Date, and not Datetime.

Track Create, Read, and Update by Object

To gain insights into Oracle CX Sales Mobile object interactions, navigate to Oracle Transactional Business Intelligence, create a new Analysis, and select the Sales – CRM Object Activity subject area.

Notes about folders in this subject area:

- The Facts table contains Insert, Read, and Update columns. You can optionally break these down by channel, such as Mobile.
- The Object Type table contains an Object Name column that delivers insights into what kinds of records are being most frequently viewed and updated.

For more information on Oracle Sales usage analytics, see [Subject Areas for Adoption and Usage Reporting](#)

For more information on Multiple Subject Area reporting, see [Overview of Cross-Subject Area Joins](#)

Configure Separate Create and Edit Pages for CX Sales Mobile

Oracle CX Sales Mobile allows Sales Administrators to configure List, Summary, Edit, Picker, and Global Search Layouts in CX Sales Mobile Composer. Using CX Sales Mobile Scripts, Sales Administrators can also choose which fields on the Edit Layout to show to or hide from a user when they create a record.

Use CX Sales Mobile Scripts to Configure a Separate Create Layout

Follow these steps to apply additional conditions to the Edit Layout a user sees when they create a new record:

1. Enter a publishable sandbox with Application Composer as one of the tools.
2. Navigate to **Application Composer > Mobile Application Setup > CX Sales Mobile Scripts**
3. Under CX Objects, select the object for the Create layout. For example, select **Opportunities**.
4. Under [Object Name] Events, select **On Create**.
5. To the right of On Create Scripts, click **Create** (the plus icon).
6. Enter a unique script name, then move the cursor out of the field to validate.
7. Enter a description for the script, then move the cursor out of the field to validate.
8. Ensure **Is Active?** is enabled.
9. Optionally, select geographic regions where script applies.
10. Enter a script to specify which fields are hidden in the edit page. Here's a script example to hide the Account field from the opportunity edit page on create:


```
const row = getCurrentRow();
row.setColumnVisible("TargetPartyName", false);
```
11. Click **Validate**.
12. Click **Save**.

13. After verifying the script behaves as expected in CX Sales Mobile, publish the sandbox.

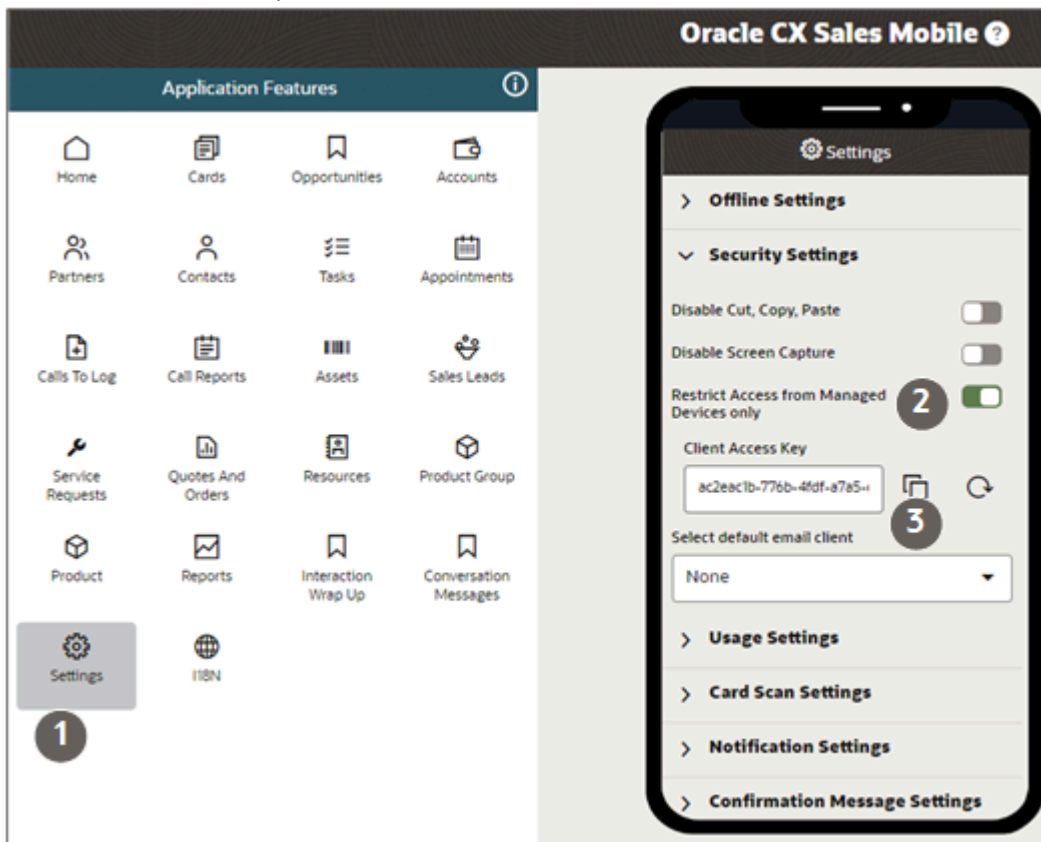
Related Topics

- [Library of System Functions for Custom Scripts](#)

Restrict Access to Managed Mobile Devices Only

You can restrict access to CX Sales Mobile to managed mobile devices. Here's how to obtain the key that you enter in your Mobile Device Management (MDM) application.

1. Create and enter a sandbox with Application Composer.
2. Click **Mobile Application Setup**.
3. Click **Settings** (callout 1 in the following screenshot).
4. Click **Security Settings**.
5. Turn on the setting **Restrict Access from Managed Devices Only** (callout 2).
6. Copy the **Client Access Key** by clicking on the **Copy** icon (callout 3).
7. In your MDM software, set the CXM_ACCESS_KEY value to the client access key you copied.
8. Deploy the App to managed devices using the MDM software.
9. Test and then publish sandbox.



Use URL Patterns That Open Leads and Opportunities Within CX Sales Mobile

Oracle Sales supports URL formats that open leads and opportunities within the CX Sales Mobile app. These deep links are embedded in emails when a CX Sales Mobile user selects a record, clicks the Share action, and then selects Email. Review sample patterns for these deep links below.

Here's a sample pattern for leads:

```
https://xx.xx..oraclecloud.com/crmUI/oraclecxm/redirect.html?redirectUrl=entity=leads&key=<LeadId>
```

Here's a sample pattern for opportunities:

```
https://xx.xx..oraclecloud.com/crmUI/oraclecxm/redirect.html?redirectUrl=entity=opportunities&key=<OptyId>
```

Override the Device Language for the CX Sales Mobile App

You can override the language set in mobile devices exclusively for the CX Sales Mobile app. For example, if the device language is set to Spanish, you may want to use CX Sales Mobile in English. Information pulled directly out of the phone operating system, such as camera and microphone permissions, is not affected. You can override the language for everyone in the organization or for select users.

To override the mobile language for CX Sales Mobile, create and set a profile option. Before you start, make sure you have the appropriate Oracle language packs implemented in your Sales application. See the topic: [Language Packs on Your Application](#).

Create the Profile Option

1. In the Setup and Maintenance work area, click on the right-side panel icon to open it.
2. Click **Search**.
3. Search for the task **Manage Profile Options** and click the name link.
4. On the page, click **Actions > New**.
5. On the Create Profile Option page, fill in the following:

Field	What to Enter
Profile Option Code	Enter ZEM_CXMOBILE_FORCE_LANGUAGE
Profile Display Name	Enter ZEM_CXMOBILE_FORCE_LANGUAGE
Application	Select Mobile Sales .
Module	<ol style="list-style-type: none"> a. Click the Search Mobile drop-down icon and click the Search link at the bottom. b. Search for Mobile Sales c. Select the Mobile Sales module.
Description	Entry isn't required.

Field	What to Enter
SQL Validation	Leave blank.
Start Date	Select a date.
End Date	Entry isn't required.

6. Click **Save and Close**.
7. On the Manage Profile Options page, search for the newly created profile option and from the results, select it.
8. In the Profile Option Levels section, you can enable this profile option for everyone (**Site**) and for specific users (**User**). If you enable both levels, then the user level overrides the site level.
 - o To enable the profile for everyone, select **Enabled** and **Updatable** for the **Site** level.
 - o If you want to specify different languages for different users, select **Enabled** and **Updatable** for the **User** level.
9. Click **Save and Close**.

Set the Profile Option

Now set the profile option you just created.

1. In the Setup and Maintenance work area, click on the right-side panel icon to open it.
2. Click **Search**.
3. Search for the task **Manage Administrator Profile Values** and click the task name link.
4. On the Manage Administrator Profile Values page Search section: enter the following
 - o In the **Application** field, select **Mobile Sales**
 - o
 - i. Click the **Module** drop-down icon and click the **Search** link at the bottom.
 - ii. Search for **Mobile Sales**
 - iii. Select the **Mobile Sales** module.
5. Click **Search**.
6. Select the profile ZEM_CXMOBILE_FORCE_LANGUAGE.
7. In the Profile Values section, click **New** (the plus sign icon).
8. Select the **Profile Level**, either Site or User.
9. If you selected **User**, then search for and enter the user name in the **User Name** field.
10. Enter the code for the language in the **Profile Value** field.
The language codes are listed in the section that follows.
11. For the user level, you can add as many users as needed by clicking **New** (the plus sign icon).
12. When you're done, click **Save and Close**.

Languages and Their Codes

Enter the language code in the profile corresponding to one of the languages listed here.

Language	Code
Arabic -	ar

Language	Code
Chinese (Simplified)	zh-CN
Chinese (Traditional)	zh-TW
Croatian	hr
Czech	cs
Danish	da
Dutch	nl
Finnish	fi
French (Canadian)	fr-CA
French (European)	fr
German	de
Hebrew	iw
Hungarian	hu
Italian	it
Japanese	ja
Korean	ko
Norwegian	no
Portuguese (Brazilian)	pt-BR
Polish	pl
Romanian	ro
Russian -	ru
Slovak -	sk

Language	Code
Spanish -	es
Swedish -	sv
Thai -	th
Turkish -	tr

End-User Tasks

Get Started with CX Sales Mobile

The Oracle CX Sales Mobile app helps you manage your day effectively and develop customer relationships using your phone or tablet. An action-driven home page helps you prepare for meetings with the clients, stay on top of key activities for your accounts, and track your pipeline anytime or anywhere.

Install and Sign In to the App

First of all, you need to install the app and sign in. Here's how:

1. Open the App Store, or Google Play, and search for Oracle CX Sales Mobile application, and then tap **Install**.
2. Open the app and accept the legal terms.
3. If you have a QR code, tap the **QR Code** button and scan the QR code using your mobile device's camera. Then go to step 5.
4. If you don't have a QR code, tap either **Single Sign On** (if you have SSO access), or tap **Basic Authentication**. Enter the host name your administrator has given you and then tap **Next**.

Note: If you have problems entering the host name, make sure that the auto-correct isn't changing the values you're entering.

5. Enter either your SSO details, or your CX Sales user name and password. Tap **Next**.

Once you have signed in, you will be given the option of setting up fingerprint recognition, or face recognition, to sign in to the app in the future. If you would like to enable this at another time, tap the menu icon on the bottom left-hand side of the app and then tap **Settings**.

Navigate Around the App

Use the home page to manage your pipeline and tasks. To show the list of sales objects, such as Opportunities, Accounts, and Contacts, tap the menu icon on the bottom left-hand side of the app. Then tap the feature to retrieve a list of items - for example, tap Opportunities to get a list of your open opportunities. To view the item details, tap any item in the list view. In the list view, tap the more icon on an item to access frequent actions, such as adding a note.

When you're viewing a sales object's details, not only can you view and edit the details - such as updating the account address, or opportunity win probability - you can also quickly access other actions for the sales object, such as logging a call or creating a task, by just tapping the more icon in the details page.

You can also use Oracle Sales Assistant to help you perform your daily sales-related tasks. Sales Assistant is a chatbot that you can send questions or commands to and it retrieves the information, or acts on your requests. Tap the chat icon on the bottom right hand side of the app and in the chat screen just type your request, or use the microphone to say your request.

Add Contacts

You can sync individual contacts from your mobile device's address book to the app by tapping the plus icon in the **Contacts** list page and then tapping **From Address Book**.

You can also add contacts by using a photo of a business card. Just take a photo of a business card using your mobile device's camera, or use an existing photo, and a new contact is created in the app using the information on the business card. Here are the exact steps::

1. Tap the plus icon in the **Contacts** list page.
2. Tap **From Business Card**.
3. Tap either **From Photo Gallery** to use an existing photo, or tap **Take Photo** to take a photo of the card using your mobile device.
4. Check and, if required, edit the contact details, and then tap **Save**.

Note: Here are some points to note about the business card feature:

- Only business cards in English are supported at this time.
- Only US address formats are supported at this time.
- The business card needs to contain an email address. The domain name (@example.com) is used to search for an existing account. If a single account is found, then the account details are populated automatically in the new record. If more than one is account found, then the account field is left empty, and you will need to select an account.
- The app has a list of job titles that are used to match the job title on the business card. If a match can't be made, then you can add it to the record, and the job title is added to the job title list so that it can be used in the future. Also, if you edit a job title then this is added to the job title list as a new job title, so that you can use it again.
- For best results, you should use a photo of the actual business card. Don't use a photo which is a picture of a business card on a monitor, for example, as the image quality won't be good enough.
- Ensure that there is a good color contrast between the business card background and the text on the card, as this enables a better scan.

Offline Usage

Can I use CX Sales Mobile when I am not connected to Wi-Fi and I haven't got a cell phone signal?

Yes, you can view, create, and edit records in the app even when you're offline. When you lose your network connection, a message is displayed to indicate that you're working offline.

At the top on the page, a banner displays how many updates will be synchronized when you're back online and also how long it's been since you last synchronized your data.

When the network connection is restored, the application automatically synchronizes the updates you have made on your mobile device to Oracle CX Sales. Alternatively, you can manually sync the records by tapping the offline banner at the top of the page and then tapping the more icon in the Pending Sync page.

What data can I see in the offline mode?

All of the data that you view in the application when you're online will be available when you're offline. Also, all of the sales objects that your administrator has specified for offline use are available in the offline mode.

The data you created, edited, or deleted when you were offline is synchronized with the CX Sales web app when CX Sales Mobile detects a network connection. Alternatively, you can sync the records manually by tapping the offline banner at the top of the page and then tapping the more icon in the Pending Sync page.

How can I synchronize my offline data?

Your data synchronizes automatically with the CX Sales web app when your mobile device detects a network connection and goes online.

Alternatively, you can manually sync the records by tapping the offline banner at the top of the page and then tapping the more icon in the **Pending Sync** page.

What happens if there's an issue when your offline updates synchronize with the CX Sales web app?

You can find a list of records that have synchronization issues by tapping the synchronization banner at the top of the app.

Once the issues have been resolved, you can sync the records manually by tapping the more icon and then the **Retry Sync For All** button.

Note: If Oracle CX Sales Mobile detects a data conflict between CX Sales Mobile and the CX Sales web application, then the updates you have made in CX Sales Mobile override the updates made in the Oracle CX Sales web app.

End-User FAQs

What note apps can I use with CX Sales Mobile?

As well as the note app that's provided with your mobile device, you can use other notes apps - such as Google Keep or OneNote - to create and share notes.

Can I scan a business card to create a contact or lead?

Yes, you can add contacts and leads by scanning a business card using your mobile device's camera, or by using an existing photo of a business card. The app creates a new contact or lead using the information on the business card. Here are the steps:

1. Tap the plus icon in the **Contacts** or **Leads** list page.
2. Tap **From Business Card**.
3. Tap either **From Photo Gallery** to use an existing photo, or tap **Take Photo** to take a photo of the card using your mobile device.
4. Check the newly created contact or lead details and, if required, edit the details. Then tap **Save**.

Note: Here are some points to note about the business card feature:

- Only business cards in English are supported at this time.
- Only US address formats are supported at this time.
- The business card needs to contain an email address. The domain name (@example.com) is used to search for an existing account. If a single account is found, then the account details are populated automatically in the new record. If more than one is account found, then the account field is left empty, and you will need to select an account.
- The app has a list of job titles that are used to match the job title on the business card. If a match can't be made, then you can add it to the record, and the job title is added to the job title list so that it can be used in the future. Also, if you edit a job title then this is added to the job title list as a new job title, so that you can use it again.
- For best results, you should use a photo of the actual business card. Don't use a photo which is a picture of a business card on a monitor, for example, as the image quality won't be good enough.
- Ensure that there is a good color contrast between the business card background and the text on the card, as this enables a better scan.

Can I scan a barcode to view an asset or create an asset record?

Yes, click **Assets** in the navigator, click the barcode icon, and take a photo of the barcode. The asset record is displayed and you can create an asset record containing the asset details.

Why can't the bar code scanner find an asset?

Ensure that the Equipment ID field is configured for Keyword Search in your organization's Adaptive Search configuration.

How can I log a call report?

Open the main menu and tap Calls to Log. Tap the activity you want to log, enter the information, and tap Save.

Alternatively, when you view your list of opportunities, accounts, tasks, and so on, just tap the more icon (three horizontal dots) on a record, and then tap **Create Call Report**.

Lastly, you can also create a call report from a sales object's detail page (such as an opportunity, lead, or task detail page). Open the detail page, scroll down to the bottom of the page, and in the **Add** section, tap **Call Reports**.

When you create or edit a call report, you can record many aspects of the activity, including logging the objectives that you have achieved, and adding resources and attachments.

How can I view my call reports?

Open the main menu and tap Call Reports.

You can also see a call report that's related to an opportunity, lead, task, and so on, by opening the details page and scrolling to the **Call Report** section.

Why has the note I created become a PDF attachment?

Some note apps, for example the OneNote app, create a note as a PDF file. CX Sales Mobile recognizes the PDF as an attachment, rather than a note, and adds the file to the record as an attachment.

What type of files can I attach to a record?

You can attach photos, videos, audio recordings, Word documents, PowerPoint presentations, Excel files, plain text files, and PDFs.

You can attach one video to a record, and up to three other file types.

How can I use Sales Assistant to help me with my sales-related activities?

You can use Sales Assistant to help you perform your daily sales-related tasks via your phone.

Type or use your voice to ask questions and sales assistant retrieves the information, or acts on your requests. Sales Assistant reduces manual data entry, and makes it easier to view and update sales records using voice commands or typing.

See the Sales Assistant chapter in the Using Sales guide for more information about how you can use Sales Assistant.

Related Topics

- [Overview of Sales Assistant](#)
- [Features of Sales Assistant](#)
- [CX Sales Mobile Specific Features](#)

What records do you see when you use the org filter?

The Org filter uses the `ORA_MYMANAGERANDPEERS` RecordSet, which returns the user's manager or peer records.

Use the **All** filter if you want to see all of the records in the organization.

When does a message appear to confirm that you've created a record?

A confirmation message appears when you've successfully created a new record, your lead conversion is in-progress, and your lead conversion is complete. These messages remain briefly before closing automatically. You also have the option to close the message.

The confirmation message appears for the following sales objects:

- Leads
- Opportunities
- Accounts
- Contacts
- Tasks
- Appointments
- Call reports
- Assets
- Service requests
- Custom objects

When a new record is created, the confirmation message contains a **Review** button which opens the new record if you tap it.

Note: This doesn't apply to lead conversion **in-progress** or **complete** messages.

If you're offline when you create a new record, or convert a lead to an opportunity, the confirmation message appears when you're back online.

Extend CX Sales Mobile

Configure and Extend CX Sales Mobile

In this section, you learn how to configure and extend Oracle CX Sales Mobile.

How can I create a custom field and add it to CX Sales Mobile?

You can create a custom field and add it to the CX Sales Mobile application.

The configuration involves three major steps:

1. Create the custom field for the Sales application.
2. Make the field available in Adaptive Search.
3. Add the field to CX Sales Mobile UI using the CX Sales Mobile Composer tool in Application Composer.

Create the Custom Field

1. Enter a sandbox where Application Composer is an active tool
2. Open Application Composer.
For example, you can open Application Composer by selecting it from the sandbox **Tools** menu.
3. In the left pane, search for the object for which you want to create the field.
4. Expand the object node.
5. Click **Fields**.
6. In the Fields page, click **Create**.
7. On the Select Field Type window, select the type of field you are creating.
8. Enter the required information for the field. What you enter is different for each field type. For details, see the topic [Field Types and Field Properties](#).
9. Make sure the **Include in Service Payload** check box is selected.
10. Click **Save and Close**.
11. Click on the sandbox name and select **Publish**.
12. On the Sandbox Details page, click **Publish** again.

Make the Field Available in Adaptive Search

Follow the instructions in the topic [Make Additional Fields Searchable](#).

Add the Field to the CX Sales Mobile UI

Follow the instructions in the topic [Configure the App Using the Configuration Tool](#) and related topics in the Implementation Reference guide.

Configure the Page Layouts

Configure the App Using the Configuration Tool

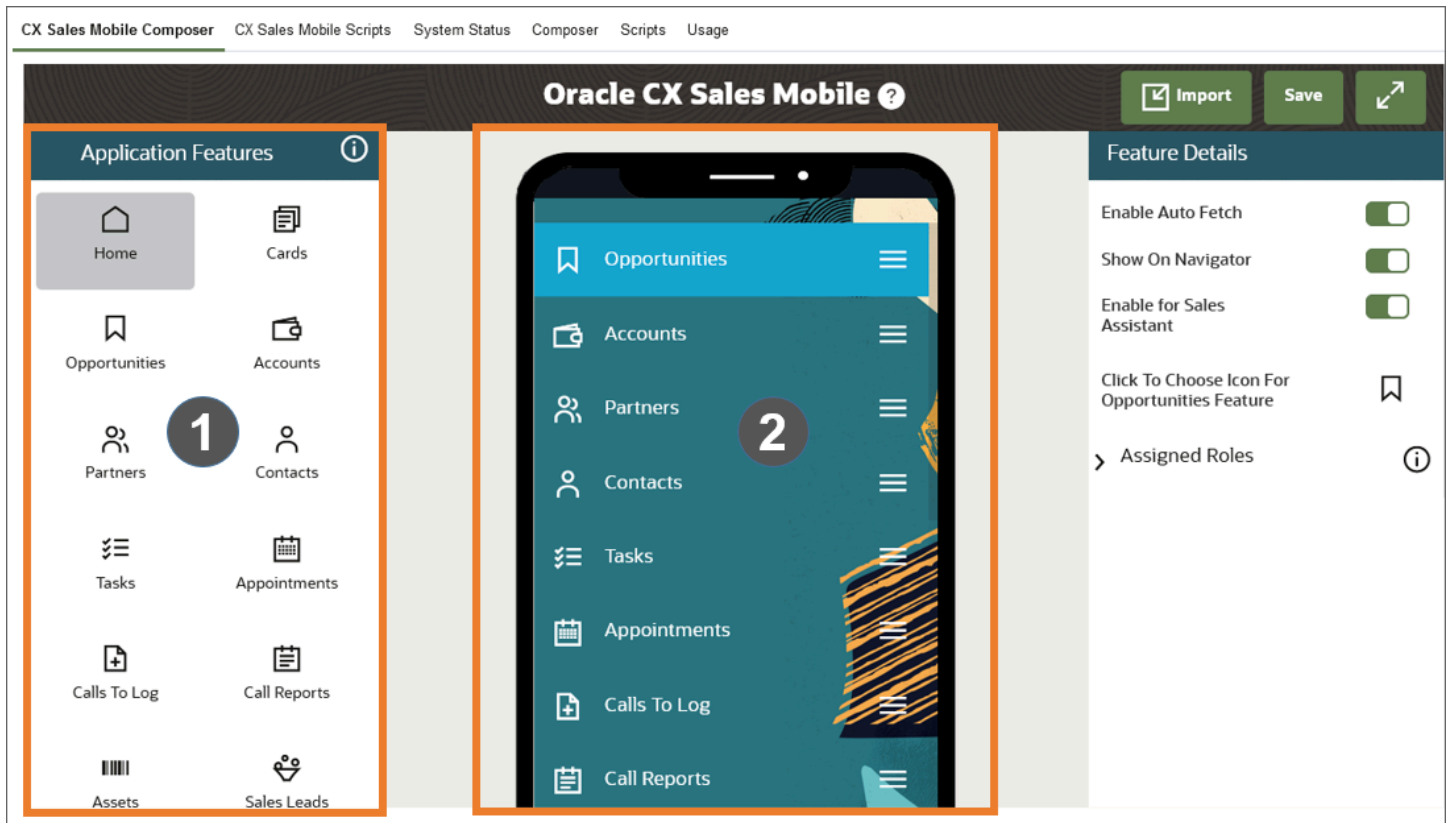
You can configure the Oracle CX Sales Mobile Apple and Android applications using an easy-to-use interface designer in Application Composer. Using the interface designer, you can configure the app to match your business requirements, such as selecting which objects and fields are visible in the application, and defining when a page layout displays.

Navigate to the Configuration Tool

First, navigate to the configuration tool:

1. Select or create a sandbox that you want to use for your configurations. Make sure that your sandbox includes Application Composer as an active tool.
2. Open Application Composer by selecting **Application Composer** in the Configuration category in the Navigator menu.
3. Under the Common Setup Menu, or on the Overview page, click **Mobile Application Setup**.
4. Click the **CX Sales Mobile Composer** tab to open the CX Sales Mobile configuration tool.

Here's what the configuration tool looks like:



On the left-hand side of the configuration tool, you can see the Application Features pane which lists the standard and custom objects that are enabled for CX Sales Mobile in the current sandbox (callout 1 in the screenshot). This pane also includes the settings that you can set up for CX Sales Mobile, which include the offline settings and user settings.

In the center of the configuration tool, you can see a visual representation of your configured version of the CX Sales Mobile app (callout 2 in the screenshot). Scroll to the bottom of the page and you can see the **+ Add Feature** button, which will add another standard or custom object to the app.

Note: If you're adding a custom object, ensure that the **Include in Service Payload** option is enabled for an object in Application Composer. For details about how to do this, see the "Add Objects and Fields in Application Composer" chapter in the Extending CX Sales and Fusion Service guide.

Create Page Layouts for Standard and Custom Objects

If you decide that you want to change the standard layout for an object, you can create a new page layout and configure it to suit your company's requirements. There are four different types of page layouts that you can create: A List page, a Summary page, an Edit page, and a Picker layout. Here are the basic configuration steps that will enable you to make simple changes to a layout:

1. In the Application Features pane, click the standard or custom object that you want to create a layout for.
2. Select the relevant type of page layout, such as the List or Summary.
3. In the Layouts pane, click the **Click to Clone** icon for the standard layout and enter a layout name.
4. Change the fields that appear by clicking the relevant field and selecting a new field from the **Select Field** pop-up page.
5. You can remove fields by clicking the field and then clicking the cross icon.
6. To move a field, click on the field that's positioned where you want your field to appear, and select or search for the relevant field in the **Select Field** pop-up page. If you want to re-position the field that you have just removed, click on the field that's positioned where you want your field to appear, and select the relevant field.
7. Click **Save** in the top right-hand corner of the configuration tool page.
8. Test and publish your new page layouts. See the [Test Your Configurations](#) topic for details about how to check your mobile configurations.

If you want to make more extensive changes to a page layout, you will need to familiarize yourself with the additional capabilities of the different types of page layouts. See these topics in the **Configure the Page Layouts** section in the *Implementing Sales* guide for more details:

- [Configure the List Page Layout](#)
- [Configure the Summary Page Layout](#)
- [Configure the Edit Page Layout](#)
- [Configure the Picker Layout](#)
- [Configure Child Object Page Layouts](#)
- [Configure the Global Search Layout](#)

Configure the List Page Layout

The List page layout displays a list of an object's records, with each record showing a set of fields that you can configure. You can make simple changes to the List page layout, as described in the [Configure the App Using the Configuration Tool](#) topic.

If you want to make more extensive changes to the List page layout, make note of these points:

- When you configure the top card in the List page layout, your changes will be reflected in every card in the list when you view them in the app.
- You can add up to two new fields to the card, as well as changing the existing fields. Just click the field and select a field from the list.
- You can add or hide actions for the List page, such as whether you can create a task, add a note, or edit the object. Click the more icon (three horizontal dots) on the page for the list of available actions, then click **+** to add an action and click an action's **X** icon to hide the action. You can also hide or add actions for child objects by clicking the child object and following the same steps.
- You can also add or hide actions for the card, such as **Edit**, **Add Note**, and **Share**. Click the more icon on the card for the list of actions. Click **+** for additional actions and click an action's **X** icon to hide the action.

- Roles and geographical regions can be assigned to List page layouts. These enable you to restrict who can view the page layout by their role and their geographical region. You can find more details about how to do this in the [Create Criteria for Page Layouts](#) topic.
- List page layouts are used by Oracle Sales Assistant. For more information about using and setting up Sales Assistant, see the [Implement Sales Assistant for CX Sales Mobile](#) topic in the Implementing Sales guide.

After making updates to your page layout, make sure that you click **Save** in the configuration tool page, and test and publish your new page layout. See the [Test Your Configurations](#) topic for details about how to check your mobile configurations.

Configure the Summary Page Layout

The Summary page layout displays an object's details.

You can make simple changes to the Summary page layout, as described in the [Configure the App Using the Configuration Tool](#) topic. To make more extensive changes to the Summary page layout, read the following sections:

The Top Section

Here are the points to note about configuring the top section:

- You can change the fields, remove fields, and move fields using the steps in the **Create Page Layouts for Standard and Custom Objects** section of this topic.
- Choose a relevant field type of either **Phone**, **Mobile**, or **Email** to match the icon. Note that you can't change the icon for these types of fields.
- You can't move the top section, because this section is designed to contain the top-level information for the object.

The Aggregates Section

The aggregates section displays aggregated information for a sales record, such as at the number of open opportunities with the predicted total value, and the number of critical open service requests for an account. Here are some points to note about this section:

- The aggregates section is available for all child and related standard objects, and custom objects associated to the parent standard object.
- You can't move this section, however you can add up to six aggregates by clicking **Add Aggregate**.
- You can move aggregates around by placing your cursor over the aggregate, clicking and holding the three horizontal lines icon in the aggregate, and dragging it to your preferred position.
- You can delete aggregates by clicking the red cross in the top right hand corner of the aggregate.
- The format of an aggregated field is **xx.xx** and the format can't be changed.

Appointment Section (Appointments Summary Page Only)

In the Appointment Summary page, there is a section below the Aggregates section that displays the appointment recurrence details, the accept or reject option, and the option to mark the appointment time as busy in your calendar. You can hide the accept or reject option, and the **Show Time As Busy** option, so that they don't appear in the Appointment Summary page. Here's how:

1. Edit an existing Appointment Summary page layout, or create a new page layout by clicking the **Click to Clone** icon.
2. In the interface designer, scroll to the appointment section and hover over the accept or reject, and **Show Time As Busy** options. Note that an eye icon appears.

3. Click the eye icon to hide the section and click **Save** in the top right-hand corner of the configuration tool page.
4. Test and publish your new page layouts. See the [Test Your Configurations](#) topic for details about how to check your mobile configurations.

The Information Sections

Here some points to note about the information sections:

- The information sections (there are up to four) are called the following: **Information**, **Address**, **Info 1 Column**, and **Info 2 Column**.
- You can reorder any of the sections by using the blue up and down arrows.
- You can add fields by clicking the **+ Add Field** button at the bottom of each section. There's no limit as to the number of fields you can add.
- The **Information** section will only display email, phone, and address icons and not field labels, so you should only select email, phone, and address fields that can have an icon, and don't require a field label.
- In the Address section of Account and Contact Summary page layouts, you can remove the actions that are available for addresses, such as editing or deleting an address. See the [Remove Address Actions on the Account and Contact Page Layouts](#) topic for more details.
- The **Info 1 Column** has a one-column layout and displays only one field across the page.
- The **Info 2 Column** has a two-column layout, so two columns will be displayed side-by-side across the page.

The Child and Related Object Sections

The section for adding child and related objects is located at the bottom of the Summary page layout.

- This section can't be moved, however you can add as many child and related objects as you require by clicking **+ Add Child or Related**.
- When you add a child or related object, you can choose from five different templates, which you can then configure in the child objects' List page layout. See the **Child Object Page Layouts** section of this topic for more information.

Other Configuration Options

Other configuration options for the Summary page include:

- Add or hide actions for the Summary page, such as whether you can create a task, add a note, or edit the object. Click the more icon (three horizontal dots) on the page for the list of available actions, then click **+** to add an action and click an action's **X** icon to hide the action. You can also hide or add actions for child objects by clicking the child object and following the same steps.
- Assign roles and geographical regions to Summary page layouts. These enable you to restrict who can view the page layout by their role and their geographical region. You can also define a set of conditions that have to be met before the page layout is displayed. For example, if you create criteria as follows: **Win probability is greater than 50%**, then any opportunities with a win probability greater than 50% will use your layout. You can find more details about how to set these up in the [Create Criteria for Page Layouts](#) topic.

After making updates to your page layout, make sure that you click **Save** in the configuration tool page, and test and publish your new page layout. See the [Test Your Configurations](#) topic for details about how to check your mobile configurations.

Configure the Edit Page Layout

The Edit page layout displays when you're editing an object's details in the app. You can make simple changes to the Edit page layout, as described in the [Configure the App Using the Configuration Tool](#) topic.

If you would like to make more extensive changes to the Edit page layout, you will need to note these additional points:

- To add a field, scroll down and click **+ Add Field**.
- To move a field, click on the field that's positioned where you want your field to appear, and select or search for the relevant field in the **Select Field** pop-up page. If you want to re-position the field that you have just removed, click on the field that's positioned where you want your field to appear, and select the relevant field.
- In the Account and Contact Edit page layouts, you can remove the actions that are available for addresses, such as editing or deleting an address. See the [Remove Address Actions on the Account and Contact Page Layouts](#) topic for more details.
- Add child objects to the bottom of the page layout by clicking **+ Add Child**.
- You can also assign roles and geographical regions. These enable you to restrict who can view the page layout by their role and their geographical region. You can also define a set of conditions that have to be met before the page layout is displayed. For example, if you create criteria as follows: **Win probability is Greater than 50%**, then any opportunities with a win probability greater than 50% will use your layout. You can find more details about how to set these up in the [Create Criteria for Page Layouts](#) topic.
- Edit page layouts are used by Oracle Sales Assistant (OSA) to determine which fields to ask questions about during task, call report, and appointment record creation. OSA asks for values for mandatory fields, or fields without defaults, in the order that they appear in the edit layout page.

If a field in the edit page layout isn't mandatory, or doesn't have a default value, but you still want OSA to ask for a value, you can specify that the field is prompted by adding the field in the **Prompted Fields on Assistant Create** pane.

After making updates to your page layout, make sure that you click **Save** in the configuration tool page, and test and publish your new page layout. See the [Test Your Configurations](#) topic for details about how to check your mobile configurations.

Remove Address Actions on the Account and Contact Page Layouts

You can remove the actions that are available for addresses on the Account and Contact Summary and Edit pages, such as deleting or editing an address. Here's how:

1. Sign in to the Oracle CX Sales application as user with a Sales Administrator role.
2. Select or create a sandbox for your configurations.
3. Open Application Composer by selecting **Application Composer** under the **Configuration** category in the **Navigator** menu.
4. Under the **Common Setup Menu**, or on the **Overview** page, click **Mobile Application Setup**.
5. Click the **CX Sales Mobile Composer** tab.
6. In the Application Features pane, click either **Account** or **Contact**.
7. In the **Children** pane, click the **Address** child object.
8. Click the **List** page layout.
9. In the Layouts pane, click the **Click to Clone** icon to create a new page layout.
10. Enter a layout name and click **Ok**.
11. In the interface designer, click the more icon (three horizontal dots) in the **Address List** card and remove the options that you don't want to appear by clicking the cross icon. This removes the options from the **Address** section in the main account or contact **Summary** or **Edit** page.

12. If you want to remove the **Create** option, click the more icon (three horizontal dots) in the header and remove the **Create** option by clicking the cross icon. This removes the **Create** option in the main account or contact **Summary** or **Edit** page.
13. Click **Save**.
14. Test and publish your new page layouts. See the [Test Your Configurations](#) topic for details about how to check your mobile configurations.

Configure the Picker Layout

The Picker page layout displays whenever you're picking an object from a selection list, such as choosing an opportunity from an opportunity search. You can make simple changes to the Picker page layout, as described in the [Configure the App Using the Configuration Tool](#) topic.

If you would like to make more extensive changes to the Picker page layout, you will need to note these additional points:

- You can change the fields, add up to six fields, remove fields, and move fields using the steps in the [Configure the App Using the Configuration Tool](#) topic.
- You can also enable or disable the ability to create an object from the picker by adding or hiding the page actions. Click the more icon on the page and either add the Create action if it isn't there, or remove the action using the cross icon that appears when you hover over the action.

After making updates to your page layout, make sure that you click **Save** in the configuration tool page, and test and publish your new page layout. See the [Test Your Configurations](#) topic for details about how to check your mobile configurations.

Configure the Global Search Layout

Sales representatives can use the global search feature directly from any page in the app to locate any type of record quickly and easily. You can configure which fields appear in the global search results page, so that your sales representatives view the most important information for their specific business needs.

See the [Configure the App Using the Configuration Tool](#) topic for instructions about how to create a new page layout. Note these additional points about the Global Search layout:

- You can change the fields, add up to six fields, and remove fields using the steps in the [Configure the App Using the Configuration Tool](#) topic.
- You can't assign criteria, such as user roles or geographical regions, for Global Search page layouts.

After making updates to your page layout, make sure that you click **Save** in the configuration tool page, and test and publish your new page layout. See the [Test Your Configurations](#) topic for details about how to check your mobile configurations.

Configure Child Object Page Layouts

Not only can you add child objects to the Summary and Edit page layouts (see the [Configure the Summary Page Layout](#) and [Configure the Edit Page Layout](#) topics for more details), but you can also create page layouts for the child objects themselves.

Here are points to note when configuring page layouts for child objects:

- When you add a child object to a parent object's page layout, the child object is added to the **Children** pane. To create a page layout for the child object, click the child object and clone the standard page layout, as you would for the parent object.
- Configuring a child object page layout is the same as configuring a parent object's page layout, so you can follow the same steps you're already familiar with in this topic.

- The template that's used for a child object's List page layout is selected when you add the child object to a parent page. You can then further configure the List page when you select the child object and clone the standard page layout.
- You can assign roles and geographical regions to child object page layouts. You can also assign criteria for displaying the page layout, known as the Advanced Criteria feature. When you using Advanced Criteria, you can select fields from the parent object, as well as the child object. You can find more details about this in the [Create Criteria for Page Layouts](#) topic.

After making updates to your page layout, make sure that you click **Save** in the configuration tool page, and test and publish your new page layout. See the [Test Your Configurations](#) topic for details about how to check your mobile configurations.

Configure the Home Page

CX Sales Mobile contains preconfigured Sales Representative and Sales Manager home page layouts by default. You can edit the preconfigured layouts, such as adding roles, re-ordering the cards, and adding Workspace saved searches to the pipeline and custom cards. You can also create entirely new home page layouts and configure them in the same way as the preconfigured home pages.

Here's how you configure the home pages:

1. Sign in to the Oracle CX Sales application as user with a Sales Administrator role.
 2. Select the sandbox you want use for your configurations.
 3. Open Application Composer by selecting **Application Composer** under the Configuration category in the Navigator menu.
 4. Under the Common Setup Menu, or on the Overview page, click **Mobile Application Setup**.
 5. In the Application Features pane, click **Cards**.
 6. To edit a preconfigured home page, navigate to the Layouts pane, click either the **Sales Rep Card Layout** or the **Sales Manager Card Layout**, and click the **Click to Clone** icon.
 7. Enter a layout name and click **Ok**.
 8. All the cards are displayed by default, but if you want to hide a card, click the card in the interface designer, and disable the **Show Card** option in the Card Settings pane.
 9. You can add additional cards, as follows:
 - a. Click **+ Add Custom Card** at the bottom of the home page.
 - b. Select the object card from the list and then on the next page select the Workspace saved search for the card.
 - c. Edit the header and title, and add an aggregation field and footer. You can hide the total number of relevant sales objects by clicking the eye icon, ensuring it has a cross through it. You can also change the saved search by clicking the more button (three horizontal dots) on the card.
- Note:** You can only have one card per object on a page layout.
10. As well as adding a Workspace saved search to a custom card, you can also add a saved search to the pipeline card.
 11. If you want to assign a role to your home page layout, add the role in the Assigned Roles pane.
 12. If you want to re-order the cards, drag the card in the interface designer to the position you would like.
 13. If you want to rename the strings on the cards, you need to use the I18N feature. See the [Rename Child Objects and Strings in the App](#) topic for more details.
 14. If you would like to create an entirely new home page layout, follow the previous steps, but select the **Default Navigator Layout** in step 6.
 15. Click **Save**.
 16. Test and publish your new page layouts. See the [Test Your Configurations](#) topic for details about how to check your mobile configurations.

Personalize the Home and Reports Pages for End Users

Your users can show, hide, and reorder the cards that appear on home and reports pages.

Here's how to personalize the home and reports pages:

Personalize the Home Page

1. On the home page, tap **Personalize**.
2. Tap **Hide this card** to hide a card.
3. Reorder the cards by using the arrow buttons on the card.
4. Tap **Done** to save the changes.

Personalize Reports

1. On the Reports page, tap the more button (three horizontal dots), and then tap **Personalize**.
2. Hide and reorder the cards by following the same steps for personalizing the home page.
3. Tap **Done** to save the changes.

Reset the Home and Reports Pages

To return to how the home page and reports pages were originally configured, follow these steps:

1. On the home page, tap **Personalize**.
On the Reports page, tap the more button (three horizontal dots), and then tap **Personalize**.
2. Tap **Reset to Default**.
3. Tap **Reset** to confirm.

Migrate Home Page Cards from Test to Production

When you move your configurations from a test environment to a production environment using the Configuration Set Migration (CSM) tool, home page card layouts in CX Sales Mobile are moved to the Production environment. However, any custom Workspace searches referenced by the home page card layouts won't be migrated.

To ensure that home page cards display correctly for your users, follow these steps after migrating from a test to production environment:

1. Re-create your test environment's custom Workspace searches in your production environment.
2. In a publishable sandbox, navigate to **Application Composer > Mobile Application Setup > CX Sales Mobile Composer > Cards**.
3. Within each custom layout, ensure your home page cards reference the appropriate saved searches as configured in your test environment.
4. In the header, click **Save**.
5. Publish the Sandbox.

Configure the Reports Page Layouts

The reporting feature provides tailored insights into the state of your business, with compelling visualizations optimized for mobile displays, including bar graphs, funnels, and area, doughnut and pie charts. As well as selecting which reports your users can view and interact with, you can also create layouts based on role and geographical region, so that users can view the reports that are most relevant to them.

Users can view these reports easily by clicking the **View my reports** button on their home page, or they can view them using the main menu.

There are two page layouts for reports provided by default: One for Sales Representatives and one for Sales Managers. The Sales Representative layout has a **My Pipeline** funnel chart. The Sales Manager layout has a **My Team's Pipeline for Sales Manager** funnel chart, a **My Direct Team's Performance for Sales Manager** bar chart, and a **My Team's Deal Size by Resource for Sales Manager** bar chart. If you want to add reports to the report pages, or configure report pages for roles or geographical regions, just follow the relevant steps in this topic.

Set Up Your Reports in Visualization Configuration

To deliver custom reports to your mobile users, use Visualization Configuration to create responsive, mobile-optimized visualizations that reference Oracle Transactional Business Intelligence reports in any shared folders.

Add Reports to Page Layouts

Once you've created the right visualizations, you can add them to report pages so that your users can view the reports that are most relevant to them. Here's how:

1. Sign in to the Oracle CX Sales application as user with a Sales Administrator role.
2. Select the sandbox you want use for your configurations.
3. Open Application Composer by selecting **Application Composer** under the Configuration category in the Navigator menu.
4. Under the Common Setup Menu, or on the Overview page, click **Mobile Application Setup**.
5. Click the **CX Sales Mobile Composer** tab.
6. In the Application Features pane, click **Reports**.
7. In the Layouts pane, select either the Sales Representative or Sales Manager layout, and click the **Click to Clone** icon.
8. Enter a name and click **Ok**.
9. In the interface designer, click **+ Add Report**.
10. Search for the report that you want to add and click on it. The list displays reports that are active and have been set up in Visualization Configuration.
11. Add as many reports as you require to the page layout. You can re-order the reports by dragging them to your preferred position in the interface designer. To delete reports, move your cursor to the report in the interface designer, and then click the delete icon.
12. Click **Save** in the top right-hand corner of the configuration tool page.
13. Make sure that you prioritize the order of your page layouts, so that the page layout at the top is the first layout that's displayed to the user, if they fit the criteria that you have set up. The second page layout is displayed if the user fits the second criteria, and so on. See the **Prioritize Your Page Layouts** section of the [Create Criteria for Page Layouts](#) topic for more details.

Add Roles to Page Layouts

If you want to create report pages for different roles, or you want to add other roles to the Sales Representative or Sales Manager page layouts, you will need to create a copy of the default layouts. Here's how:

1. Follow the first six steps in the **Add Reports to Page Layouts** section of this topic.
2. In the Layouts pane, select the page layout that you want to add roles to and that you have created, or click the **Click to Clone** icon for either the Sales Representative or Sales Manager layout.
3. In the Assigned Roles pane, search for the role that you want to add, and click the role. The role is added to the page layout.
4. Add as many roles as you require and click **Save**.
5. Ensure that you have prioritized the order of your page layouts. See the **Prioritize Your Page Layouts** section of the [Create Criteria for Page Layouts](#) topic for more details.

Add the Report Privilege to Custom Roles

If you have assigned custom roles to your reports pages, you need to add the visualizations privilege ZCA_VIEW_DATA_VISUALIZATION_CONFIGURATION_PRIV to the custom roles.

1. Navigate to the Security Console by clicking **Navigator** > **Tools** > **Security Console**.
2. Click the Roles tab and find your custom role.
3. Edit the role and click **Next** to get to the Function Security Policies page
4. Click **Add Function Security Policy** and search for the ZCA_VIEW_DATA_VISUALIZATION_CONFIGURATION_PRIV privilege.
5. Select the privilege and click **Add Privilege to Role**.
6. Click **Next** through the wizard and finally click **Save and Close**.
7. Now you need to run three processes to assign the privilege to the custom role. Navigate to Scheduled Processes by clicking **Navigator** > **Tools** > **Scheduled Processes**.
8. Click **Schedule New Process**.
9. Set the Type as **Job** and in the **Name** field, search for **Retrieve Latest LDAP Changes**.
10. Click **OK** and in the Process Details page, click **Submit**.
11. Repeat for the following processes:
 - o **Send Pending LDAP Requests**
 - o **Import User and Role Application Security Data**
12. Open the CX Sales Mobile app and check that your reports pages are visible in the navigation menu for your custom roles.

Add Geographical Regions to Page Layouts

You can add geographical regions to reports page layouts, so that you can select a set of reports for roles within particular regions. You can add regions to page layouts as well as roles.

1. Follow the first six steps in the **Add Reports to Page Layouts** section of this topic.
2. In the Layouts pane, select the page layout that you want to add geographical regions to and that you have created, or click the **Click to Clone** icon for either the Sales Representative or Sales Manager layout.
3. In the Assigned Regions pane, search for the geographical region that you want to add, and click the region. The region is added to the page layout.
4. Add as many regions as you require and click **Save**.
5. Ensure that you have prioritized the order of your page layouts. See the **Prioritize Your Page Layouts** section of the [Create Criteria for Page Layouts](#) topic for more details.

Migrate Report Lists from Your Test to Production Environments

The Visualization Configuration tool generates Reference IDs automatically for visualizations created in the tool. Also, from the 22A release, visualizations are migrated from Test environments to Production environments. As a result, you don't need to rebuild Report Lists if you perform a Configuration Set Migration (CSM). Here are the steps to perform the mobile report migration:

1. Complete the Oracle Transactional Business Intelligence migration steps in the [Archive and Move Analytics](#) topic in the *Creating and Administering Analytics and Reports* guide.
2. Migrate your Test Environment Visualizations to your Production Environment. In the [Configure Visualizations for CX Sales Mobile](#) topic, follow the steps in the **Import and Export Visualization Configurations** section.
3. Run the Configuration Set Migration (CSM). This allows configurations made in Application Composer - including configured mobile report lists, privileges, and references to Visualization Configuration - to be migrated from one environment to another. Additional resources on CSM can be found in the [Migrate Your Configurations](#) topic in the *Configuring and Extending Applications* guide.

Note: If you rebuilt your visualizations instead of migrating them, or did not successfully run CSM, follow these steps to finish the migration:

1. Create a publishable sandbox in your production environment.
2. Navigate to **Navigator > Application Composer > Mobile Application Setup**.
3. Click the **CX Sales Mobile Composer** tab and select **Reports** in the Application Features pane.
4. If you didn't run CSM, re-create the configured report lists for roles and regions as necessary.
5. Select **Add Report** to select reports that reference items that are in the production environment's Visualization Configuration tool.
6. Click **Save**.
7. Publish the sandbox.

Display Visualizations from Saved Searches in Home Page Cards and the Reports List

Enhance the level of insight available to salespeople and managers by displaying saved searches as visualizations on home page cards and in the reports list. Tapping a segment of a parent report reveals a list of records associated with the respective segment. Selecting a list item reveals its associated record.

1. **Note:** If you've already created a visualization from a saved search and want to display it on a home page card, you can skip to step 12.

In any of the work areas that use saved searches powered by Adaptive Search, create or find the saved search you want to use for the visualization. The saved search must be made visible to the roles of the user performing configurations and to the users who will consume the visualizations.

Note that attribute dimensions are determined by what is made available for **Group By under Manage Adaptive Search** in Setup and Maintenance. Also, search-based visualizations use **Count of Results** as their measure dimensions.

2. Outside of a sandbox, navigate to **Application Composer > Visualization Configuration**.
3. In the Visualization Configurations page, click **Add**.
4. In the **Source Type** field in the **Create Configuration** page, select **Adaptive Search**.
5. In the **Source** field, select the saved search that you created or found in the first step. The saved search is used as the data source for the visualization.
6. In the **Visualization Type** field, choose your preferred visualization, such as a bar chart, doughnut chart, and so on.
7. In the **Details** section, enter the dimensions and other preferences for the visualization.
8. In the **Drill Down** section, change the **Action** option from **None** to **Link**. This enables drill down to the search list.
9. Set the **Status** field to **Active** and click **Create**.
10. Return to the main Oracle CX Sales home page and enter a sandbox.
11. Add visualizations to the Reports page as specified in the [Configure the Reports Page Layouts](#) topic. Note that the Reports page supports both Oracle Transactional Business Intelligence (OTBI) reports and visualizations from saved searches.

12. Add visualizations to the mobile home page. Note that the home page supports only search-based visualizations. Here's how you add the visualizations:
 - a. Navigate to **Application Composer > Mobile Application Setup > CX Sales Mobile Composer > Cards**. Select a custom layout, or clone a standard layout.
 - b. Scroll to the bottom of the layout and select **+ Add Custom Card**.
 - c. In the **Add Card** screen, select the **Visualizations** tab.
 - d. Select your visualization to add it to the home page. Click **Save**.
13. Test your update by following the steps in the [Test Your Configurations](#) topic.

Ensure Users with Custom Roles See Visualizations

To see visualizations, salespeople with custom job roles must have the `ZCA_VIEW_DATA_VISUALIZATION_CONFIGURATION_PRIV`. Here's how you can grant that privilege.

1. Navigate to **Security Console > Roles** and search for **ZCA_VIEW_DATA_VISUALIZATION_CONFIGURATION_PRIV** (make sure that enabled all role types and privileges in the search filters).
2. From the **Expand Toward** list, select **Users**.
3. From the **Show** list, select **Users** to see which users have the privilege and from what role.
4. To add the privilege, edit the custom role:
 - a. Go to **Function Security Policies > Privileges > Add Function Security Policy** and search for privilege **ZCA_VIEW_DATA_VISUALIZATION_CONFIGURATION_PRIV**.
 - b. Add the privilege to the role.
 - c. Save.

Enable Visualizations That Drill Down to OTBI Reports

You can embed visualizations related to a record in its detail page. In addition, you can enable drill down to Oracle Transactional Business Intelligence (OTBI) reports from visualizations that appear in both record detail pages and in the reports list.

This feature enables sales representatives to improve outcomes of customer interactions by having customer and deal 360 views available in any setting.

Here's how to enable the feature:

Build a Report in OTBI

1. Navigate to Oracle Transactional Business Intelligence (OTBI) and click **Add** and create a top-level report that the record-based visualization will reference.
2. (Optional) Add a prompted filter. You might use this if, for example, your organization wants to view reports on the Account Detail page that only show information related to the displayed account. Here's how:
 - a. Add the Customer Row ID column to the report.
 - b. Enable the ID as a filter where the **Operator** is set to **is prompted**. This allows the Account Party ID to be passed to the report as a filter value.
3. Save the top-level report.
4. Without navigating away from the top-level report, click **Save As**, and change the name to represent the drill down report.
5. Add additional columns to the report table as required, to provide sales users with more granular insights.
6. Add additional prompts that represent the fields that delineate the elements of the top-level report. For example, if the top-level report will be visualized as a bar graph where the X-axis is **Enterprise Quarter** and the series is **Product Group**, add both of these as Filters where the **Operator** equals **is prompted**.

7. Save the drill-down report once all your changes are made.

Create a Data Visualization and Reference the Report

1. Outside of a sandbox, navigate to **Application Composer > Visualization Configuration**.
2. Click **Add** and in the **Source Type** field, select **OTBI Analysis**.
3. In the **Source** field, find and select the top-level report that you have just created.
4. In the **Visualization Type** field, choose your preferred visualization, such as a bar chart.
5. In the **Details** section, select the **X Axis**, **Value**, and **Categorized By** fields.
6. In the **Drill Down** section, change the Action from **None** to **Link**. If the **Search for target analysis** field appears, search for and select the drill down report that you just created.
7. Set the **Status** to **Active** and then click **Create**.

Display the Visualization in CX Sales Mobile

1. Return to the main Oracle CX Sales home page and enter a sandbox.
2. Navigate to **Application Composer > Mobile Application Setup > CX Sales Mobile Composer**.
3. Navigate to the object where you want to add the visualization, and click the **Summary** page.
4. Select a custom layout, or clone a standard layout.
5. Scroll to the **Reports** section of the layout preview and click **+ Select and Re-Order Reports**.
6. Click **+ Add Report** and choose the top-level visualization created in the previous section.
7. In the **Report Filters** pane in the right-hand side of the page, select the prompted field from the report that will receive a filter value from the page. For the example in step 3 of the **Build a Report in OTBI** section in this topic, this would be the **Customer Row ID** field. Also select the object field whose value will be passed to the report. For the example above, this would be the **PartyId** field.
8. Click **Save**.
9. Test your update by following the steps in the [Test Your Configurations](#) topic.

Rename Child Objects and Strings in the App

You can change the names of child objects and strings in the app -- such as page headers, section headers, actions, and titles -- to suit your business requirements.

Here's how you do that:

1. Sign in to the Oracle CX Sales application as user with a Sales Administrator role.
2. Select the sandbox you want use for your configurations.
3. Open Application Composer by selecting **Application Composer** under the Configuration category in the Navigator menu.
4. Under the Common Setup Menu, or on the Overview page, click **Mobile Application Setup**.
5. In the Application Features pane, click **I18N**.
6. Select the relevant language and then select the string you want to change.
7. Enter the new name in the **Edit String** box.
8. Click **Save**.
9. Test and publish your new page layouts. See the [Test Your Configurations](#) topic for details about how to check your mobile configurations.

Note: If the string you want to change isn't available in the choice list, you can submit a request to add the string using the **Ideas** tab in Oracle Cloud Customer Connect.

Create Criteria for Page Layouts

You can create criteria for a standard and custom object's page layout, so that a condition, or a set of conditions, must be met before the page layout is displayed. You can restrict the page layout by role, geographical region, and by field value.

Assign User Roles to a Page Layout

You can create a page layout for a standard or custom object that will only be displayed to one, or multiple user roles. This can be very useful if, for example, you want to create a page layout for a Sales Manager role, which will require certain fields on an opportunity detail record that other sales team members won't need. Here's how you set it up:

1. Sign in to the Oracle CX Sales application as user with a Sales Administrator role.
2. Select the sandbox you want use for your configurations.
3. Open Application Composer by selecting **Application Composer** under the Configuration category in the Navigator menu.
4. Under the Common Setup Menu, or on the Overview page, click **Mobile Application Setup**.
5. In the Application Features pane, click the feature that contains the page layout that you want to add roles to, for example, Opportunities.
6. Select the relevant type of page view, such as the List or Summary view.
7. In the Layouts pane, select the relevant page layout, or create a new page layout (see the [Configure the App Using the Configuration Tool](#) topic for details about how to create a page layout).
8. In the Assigned Roles pane, select the role, or roles, that you want to add.
9. Click **Save**.

Assign Geographical Regions to a Page Layout

Adding geographical regions to a standard or custom object's page layout, enables you to restrict its availability to users from a selected set of geographical regions. For example, if you add the United States and United Kingdom regions to a page layout, then only users from these countries can view the layout. Here's how you set it up:

1. Repeat steps 1 to 7 in the Assign User Roles to a Page Layout section.
2. In the Assigned Regions pane, select the geographical regions by clicking on the region. Select as many regions as you require.
3. Click **Save**.

Assign Advanced Criteria to a Page Layout

Creating advanced criteria enables you to define a set of conditions that have to be met before the page layout is displayed for a standard or custom object's Summary or Edit views. For example, if you create criteria for an Opportunity Summary layout as follows: `Opportunity Type is Equal to Finance`, then any opportunities with a Finance opportunity type will use your layout in the Summary view.

1. Repeat steps 1 to 7 in the Assign User Roles to a Page Layout section.
2. In the Advanced Criteria pane, create your criterion by selecting a field from the list, an operator, and then entering the relevant field value.
3. To add a conditional statement to your criterion, click **Add Criteria**, and select AND or OR. Enter the field, operator, and relevant field value.

Note: If you want to have multiple AND/OR conditions, it's best to use custom scripting to specify your criteria. See the "Configure the App with Custom Scripts" for information about creating a custom script.

4. Click **Save**.

If you're creating criteria for a child object's page layout, then you can choose fields from the child and parent object. For example, you can create an advanced criteria where an opportunity's Opportunity Type value determines the page layout for opportunity revenue page.

Prioritize Your Page Layouts

Page layouts display in priority order in the Layouts pane. The page layout at the top is the first layout that's displayed to the user, if they fit the criteria that you have set up. The second page layout is the second layout that displays if the user doesn't fit the first criteria, but does fit the second page layout's criteria, and so on. You can move the layouts around by dragging them to your preferred priority position.

It's recommended that you create a default layout that's applicable to everyone and place the default layout at the bottom of the Layouts pane. Setting a default layout means that users will see at least one of the page layouts if they don't fit the criteria you have set up.

Test Your Page Layouts

It's important to test your configurations before publishing the sandbox. See the [Test Your Configurations](#) topic for details about how to check your mobile configurations.

Add Criteria to Actions in List and Summary Page Layouts

You add criteria to actions, such as create and edit actions, to List and Summary pages, so that the actions will only appear when the conditions have been met. For example, you can add a criteria to the Create Appointment action, so that the action will appear only when the Account Score field is greater than a set value.

Here's how you set up the criteria:

1. Sign in to the Oracle CX Sales application as user with a Sales Administrator role.
2. Select or create the sandbox you want use for your configurations.
3. Open Application Composer by selecting **Application Composer** under the Configuration category in the Navigator menu.
4. Under the Common Setup Menu, or on the Overview page, click **Mobile Application Setup**.
5. Click the **CX Sales Mobile Composer** tab.
6. In the Application Features pane, click the relevant feature.
7. Click either the **Summary** or **List** page layout.
8. In the Layouts pane, select an existing page layout, or create a new page layout by clicking the **Click to Clone** icon.
9. In the interface designer, click the more icon (three horizontal dots) in the header and select the relevant action, such as the Create Task action.
10. Click **Add Criteria**.
11. Enter the Field, Operator, and Value.
12. If you want to add more criteria click **Add Criteria** and enter the criteria values.
13. Add as many criteria as you require.
14. Click **OK**.
15. Click **Save**.
16. Repeat the steps above for each action that you want to add criteria to.
17. Test and publish your new page layouts. See the [Test Your Configurations](#) topic for details about how to check your mobile configurations.

Use Your Many-to-Many Relationship Setup When You Create Page Layouts

Add your many-to-many relationship intersection objects to your page layouts, so that you can display the intersection object fields with the respective target object fields in the page layout.

Here's how you set this up:

1. Sign in to the Oracle CX Sales application as user with a Sales Administrator role.
2. Select the sandbox you want use for your configurations.
3. Open Application Composer by selecting **Application Composer** under the Configuration category in the Navigator menu.
4. Under the **Common Setup Menu**, or on the **Overview** page, click **Mobile Application Setup**.
5. Click the **Source** object of the Many-to-Many relationship, such as the Opportunity object.
6. Edit or create a new page layout, and click **+ Add Child or Related** at the bottom of the page layout.
7. Search for your intersection object and select it.
8. Add fields from the intersection object, or the Source object, in the page layout.
9. Click **Save**.
10. Test and publish your new page layouts. See the *Test Your Configurations* topic for details about how to check your mobile configurations.

For more information about many-to-many relationships, see the "Many-to-Many Relationships" topic in the "Configuring Applications Using Application Composer" guide.

Note:

- You can add the source object fields only in the page layout view. Editing the layout of the intersection object only allows adding fields from the intersection object.
- Use the edit layout of the source object to edit fields on the source object.
- Tapping the summary page layout on the mobile app navigates users to the source object summary page layout. The edit page layout only displays the details of the intersection object.

Related Topics

- [Many-to-Many Relationships](#)

Configure the App with Custom Scripts

You can further configure the app to match your business requirements by writing custom scripts using JavaScript that enforce validations and rules, whether you're online or offline. This powerful feature lets you write scripts that can trigger on an event of your choosing, for any top-level parent object and child object. For example, you can create a script that will make the product field mandatory when you create an opportunity.

How to Create a Script

You can create custom scripts for all objects, or for a specific object. Here's how you create a custom scripts for a specific object:

1. Select the sandbox you want use for your configuration.
2. Select **Navigator > Application Composer > Mobile Application Setup**.
3. Click the **CX Sales Mobile Scripts** tab at the top of the **Mobile Application Setup** page.
4. To create a script for a top-level parent object, select the object you want to write a script for, and then select the event that will trigger your script. You can choose one of these events:

- **On Create** (triggered when a new record is created in the mobile app).
 - **Before Save** (triggered before a record is saved).
 - **After Save** (triggered after the app has saved the record).
 - **On Field Value Change** (triggered when a user changes the value of a field and then taps out of the field).
 - **On Edit** (triggered when a user edits a record).
5. To create a script for a child object, select the top-level parent object, then select the relevant child object in the **Opportunities Children** section. Finally, select the event that will trigger your script.

Note: If you want to create validation for a child object's **Before Save** or **After Save** event, you have to use the parent object's **Before Save** or **After Save** event.

6. Click the plus icon in the **<Event Name> Scripts** section.
7. Enter the script name and description.
8. (Optional) If you want the script to work for particular regions, select geographical regions for the script. The script will work for users who have your selected regions as their country preference setting in the CX Sales app.

Note: A script can have multiple regions associated to it, but the same region can't be assigned to more than one active script for the same object and event.

If you want to restrict your script to trigger for certain user roles, you will need to specify these by writing a JavaScript script

9. Click **Create** and create your script.
10. Click **Validate** to check whether your script has any errors. The validation checks include checks for nested functional calls and anything else in the script that will cause it to run forever in an infinite loop.
11. When you're finished, click **Save**.

How to Create a Custom Function

You can also create a custom function (known as a utility function) that will be available for all objects in all of your scripts. Here's how:

1. Follow steps 1 to 6 in the **How to Create a Script** section, and then click the plus icon in the **Utility Functions** section.
2. Create your function. Make sure that you include "_c" in the function, so that the function is designated as a utility function, and is included in the System Functions list (see the **Feature That Helps You to Create Your Scripts** section below for details about System Functions). Once it's in the System Functions list, you can access your utility function from any custom script. Here's an example of the format:

```
function example_c() {
  var a = 100;
  a = 100+a
}
```

3. Click **Validate** and then **Save**.

Feature That Helps You to Create Your Scripts

When you're creating your scripts it's important that you enter the correct object and field values, so to help you with this you can type **Control + Space** in the script box and a list of objects and fields are displayed. Pick the object or field that you want to include and the correct value is added to your script.

This feature also enables you to use the System Functions that auto-populate APIs into your script. Type **Control + Space**, scroll down to the **System Functions** section, and you will see a list of APIs that you can use. You can use these APIs, for example, to find out the device's operating system and the current position of the user. In the **System Functions** section, you can also see the Utility Functions that you have created. See the [Library of System Functions for Custom Scripts](#) topic for a list of the System Functions that are available and how you can use them.

Test Your Script

To help you test your script, you can use the `CXCoreLogger` class methods in your script to write warning, error, or info messages. You can then review the messages to debug the script.

After debugging the script, test the script in CX Sales Mobile, and then publish the sandbox when you're done. See the [Test Your Configurations](#) topic for details about how to check your configurations in CX Sales Mobile.

Script Examples

Here are some examples of how you can create custom validation and rules:

Object, Event, and Requirements	Example Script
Object: Opportunity Event: OnCreate Requirement: Auto-populate the name and owner field for the Opportunity.	<pre>var oracleCxmOutcome = new Result(""); var currentRow = await getCurrentRow(); var userPref = await getUserPreferences(); var partyName = userPref.getPartyName(); currentRow.setColumn('PartyName1', partyName); currentRow.setColumn('Name', 'StandardScript'+new Date().toISOString().substring(0, 10)); oracleCxmOutcome.setMessage('MESSAGE_TYPE_SUCCESS', '', 'OOB Script for oppty-OnCreate executed');</pre>
Object: Opportunity Event: BeforeSave Requirement: Update the Opportunity Revenue amount based on child revenue amounts.	<pre>var oracleCxmOutcome = new Result(''); var currentRow = await getCurrentRow(); var childRev = currentRow.getColumn('ChildRevenue'); let totalAmount = 0; if(childRev) { let items = currentRow.getColumn('ChildRevenue').items; if(items) { items.forEach((item) => { totalAmount = totalAmount + item.RevnAmount; }); } } currentRow.setColumn('Revenue', totalAmount);</pre>
Object: Opportunity Event: AfterSave Requirement: Add a note to the opportunity, if the opportunity isn't newly created.	<pre>var oracleCxmOutcome = new Result(''); var optiRow = await getCurrentRow(); if(!optiRow.isNew()) { var opptyNote = await createNewRow(true, 'opportunities', 'Note'); var noteTxt = 'VGhpcyB0b3RlIHdhcyBjcmVhdGVkIHRocm91Z2ggc2NyaxB0IGZvcjBvcHB0eSAtIA=='; opptyNote.setColumn('NoteTxt', noteTxt); optiRow.setColumn('Note', opptyNote); oracleCxmOutcome.setModifiedObject(optiRow); }</pre>
Object: Opportunity	<pre>var oracleCxmOutcome = new Result(''); var opptyRow = await getCurrentRow(); var statusCode = opptyRow.getColumn('StatusCode');</pre>

Object, Event, and Requirements	Example Script
<p>Event: OnFieldValueChange - Status Code</p> <p>Requirement: If the status code value is changed to WON/LOST, and the win/loss reason isn't specified, then the ReasonWonLostCode field is made mandatory, and a message is displayed 'Please specify win/loss reason.'</p>	<pre>var reasonWonLostCode = optyRow.getColumn('ReasonWonLostCode'); if((statusCode === 'WON' statusCode === 'LOST') && (!reasonWonLostCode)) { oracleCxmOutcome.setMessage('MESSAGE_TYPE_SUCCESS', '', 'You have selected' + statusCode); optyRow.setColumnMandatory('ReasonWonLostCode', true); oracleCxmOutcome.setMessage('MESSAGE_TYPE_ERROR', '', 'Please specify win/ loss reason'); } if(statusCode !== 'WON' && statusCode !== 'LOST') { optyRow.setColumn('ReasonWonLostCode', null); optyRow.setColumnMandatory('ReasonWonLostCode', false); }</pre>
<p>Object: Opportunity Revenue</p> <p>Event: OnCreate</p> <p>Requirement: The OnCreate event for the child object called Opportunity Revenue auto-populates its custom field values.</p>	<pre>var optiRow = getCurrentRow(); const optyRow = getCurrentRow(); const childRev = getCurrentChildRow(); var oracleCxmOutcome = new Result(""); childRev.setColumn('CXM_Text_c', 'Single Text'); childRev.setColumn('CXM_Date_c', "2020-03-20"); childRev.setColumn('CXM_CC_c', true); childRev.setColumn('CXM_Perc_c', .32); childRev.setColumn('CXM_Num_c', 5000); childRev.setColumn('CXM_LText_c', 'RGTvbyB0ZXh0IHRvIGJlIGVudGVyZWQ='); oracleCxmOutcome.setMessage("MESSAGE_TYPE_SUCCESS", "", "OnCreate on Child:");</pre>
<p>Object: Opportunity</p> <p>Event: OnFieldValueChange</p> <p>Field: Dynamic choice list whose filter value is being checked.</p> <p>Requirement: Trigger error if user attempts to select a dynamic choice list value that doesn't meet filter criteria.</p> <p>Prerequisite: Include filter criteria field (in this example, the Account Type) in filter object Picker page on CX Sales Mobile</p>	<pre>var oracleCxmOutcome = new Result(''); const optyRow = getCurrentRow(); const accountsQuery = query('accounts'); //query accounts or another object, depending on the use case accountsQuery.setParameters('PartyId', optyRow.getColumn('account_dcl_Id_ c')); //be sure to use the ID field for your DCI here, not the name field try { const accountsResponse = accountsQuery.execute(); if (accountsResponse && accountsResponse.length > 0) { if(accountsResponse[0].getColumn('OrganizationDEO_LSAccountType_c') !== 'SERVICE_CENTER'){ //This checks to see if the filter criteria applied to the DCI are met oracleCxmOutcome.setMessage('MESSAGE_TYPE_ERROR', '', 'You must choose an organization that is a Service Center. Choose the Service Centers list in the Show filter.');</pre>
<p>Object: Opportunity</p> <p>Event: BeforeSave</p> <p>Requirement: Trigger error if user attempts to save a record with an invalid dynamic choice list value.</p>	

Object, Event, and Requirements	Example Script
Prerequisite: Include filter criteria field (in this example, the Account Type) in filter object Picker page on CX Sales Mobile	

```

var oracleCxmOutcome = new Result('');
const opptyRow = getCurrentRow();
const accountsQuery = query('accounts'); //query accounts or another object, depending on the use case
accountsQuery.setParameters('PartyId', opptyRow.getColumn('account_dcl_Id_c')); //be sure to use the ID field for
your DCL here, not the name field
try {
const accountsResponse = accountsQuery.execute();
if (accountsResponse && accountsResponse.length > 0) {
if(accountsResponse[0].getColumn('OrganizationDEO_LSAccountType_c') !== 'SERVICE_CENTER'){ //This checks to see
if the filter criteria applied to the DCL are met
oracleCxmOutcome.setMessage('MESSAGE_TYPE_ERROR', ' ', 'You must choose an organization that is a Service
Center. Choose the Service Centers list in the Show filter. ');
oracleCxmOutcome.setOutcomeQualifier("OUTCOME_TYPE_FAILURE"); //The error message here indicates that Service
Centers is the name of the saved Workspace Search, and is the filter the user should select
}
}
} catch (e) {
//failed to get accounts response
}

```

Library of System Functions for Custom Scripts

When you're creating your custom scripts, you can use a feature that enables you to use System Functions that auto-populate APIs into your script. Type **Control + Space**, scroll down to the System Functions section, and you'll see a list of APIs that you can use. Here are more details about the System Functions and their related classes.

System Functions

These are the System Functions that are available:

Method Signature	Definition	Usage
<code>getCurrentRow()</code>	Returns the current row. For information about the Row class methods, refer to the table below.	<pre>const taskRow = getCurrentRow(); const statusCode = taskRow.getColumn('StatusCode');</pre>
<code>getCurrentChildRow()</code>	Returns the current child row if we are in context of a child row, else it returns null. Refer to the Row class methods in the next table.	<pre>const opptyRow = getCurrentRow(); const childRev = getCurrentChildRow(); childRev.setColumn('RevAmountCurcyCode', opptyRow.getColumn('CurrencyCode'));</pre>
<code>createNewRow(isChild: boolean, featureName: string, childType: string)</code>	Creates new row or child row for the given resource and child type. Only supported in AfterSave event. Refer to the Row class methods in the next table.	<pre>const opptyRow = createNewRow(false, "Opportunity", null); opptyRow.setColumn("Name", "New oppty");</pre>

Method Signature	Definition	Usage
		<p>Note: The use of this API is recommended for creating a row for the top level resource only. For creating a child of a top level resource, please refer to <code>addNewChildRow</code> under Row class.</p>
<code>query(resource: string)</code>	<p>Returns Query object for the given resource. This can be used to query rows of the given resource by setting query parameters. This function only queries for data stored locally on the device.</p> <p>Refer to the Query class methods in the next table.</p>	<pre>const currentRow = getCurrentRow(); const userPrefProvider = getUserPreferences(); const resourceQuery = query('resources'); resourceQuery.setParameters('PartyId', userPrefProvider.getPartyId()); let cxCoreLogger = getCXCoreLogger().getLogger(); try { const resourceResponse = resourceQuery.execute(); if (resourceResponse.length > 0) { currentRow.setColumn('JobName', resourceResponse[0].getColumn('JobMeaning')); } } catch (e) { //failed to get resource response cxCoreLogger.error("Query on resources failed"); }</pre> <p>Note: This API is for querying on the basis of primary key only. To query using any other field please refer to the <code>queryWithParameters</code> API below.</p>
<code>getUserPreferences()</code>	<p>Returns <code>UserPreferences</code> object. This can be used to retrieve different user preferences.</p> <p>For information about the <code>UserPreferences</code> class methods, refer to the table below.</p>	<pre>const row = getCurrentRow(); const userPrefProvider = getUserPreferences(); const closeDate = userPrefProvider.getProfileOptionValue('MOO_DEFAULT_CLOSE_WINDOW');</pre>
<code>getDeviceInformation()</code>	<p>Returns <code>DeviceInformation</code> object. This can be used to retrieve different device related information.</p> <p>Refer to the <code>DeviceInformation</code> class methods in the next table.</p>	<pre>const deviceInfo = getDeviceInformation(); const os = deviceInfo.getOs();</pre>
<code>getCXCoreLogger().getLogger()</code>	<p>Returns an instance of <code>CXCoreLogger</code> which can be used for logging info, error, warnings, and so on.</p> <p>Refer to the <code>CXCoreLogger</code> class methods in the next table.</p>	<pre>const currentRow = getCurrentRow(); const userPrefProvider = getUserPreferences(); const resourceQuery = query('resources'); resourceQuery.setParameters('PartyId', userPrefProvider.getPartyId());</pre>

Method Signature	Definition	Usage
		<pre>let cxCoreLogger = getCXCoreLogger().getLogger(); try { const resourceResponse = resourceQuery.execute(); if (resourceResponse && resourceResponse.length > 0) { currentRow.setColumn('JobName', resourceResponse[0].getColumn('JobMeaning')); } } catch (e) { //failed to get resource response cxCoreLogger.error("Query on resources failed"); }</pre>
<p><code>getParentRow()</code></p>	<p>Returns the parent object from which the current object was accessed. Only to be used in context of top level objects being visited via another top level object.</p> <p>Note: This function doesn't return the parent for a child row.</p> <p>Refer to the Row class methods in the next table.</p>	<pre>/*Appointments - OnCreate*/ const row = getCurrentRow(); const parentRow = getParentRow(); if (parentRow && parentRow.getResourceName() === 'leads') { row.setColumn('LeadId', parentRow.getColumn('LeadId')); row.setColumn('LeadName', parentRow.getColumn('Name')); if (parentRow.getColumn('PrimaryContactId')) { row.setColumn('PrimaryContactId', parentRow.getColumn('PrimaryContactId')); row.setColumn('PrimaryContactName', parentRow.getColumn('PrimaryContactPartyName')); } if (parentRow.getColumn('CustomerId')) { row.setColumn('AccountId', parentRow.getColumn('CustomerId')); row.setColumn('AccountName', parentRow.getColumn('CustomerPartyName')); } }</pre>
<p><code>getLovDataProvider()</code></p>	<p>Returns local list of values (LOV) data provider. The LOV data can be fetched using this provider.</p>	<pre>const row = getCurrentRow(); const userPrefProvider = getUserPreferences();</pre>

Method Signature	Definition	Usage
		<pre>const territory = userPrefProvider.getTerritory(); const lovProvider = getLovDataProvider(); const countryCodeLov = lovProvider.getLovData('WorkPhoneCountryCode'); if (countryCodeLov) { let locale; for(let i =0 ; i < countryCodeLov.length ; i++) { if(countryCodeLov[i].TerritoryCode === territory) { locale = countryCodeLov[i]; } } if(locale) { if (row.getColumn('MobileCountryCode') == null row.isNew()) { row.setColumn('MobileCountryCode', locale.PhoneCountryCode); } if (row.getColumn('WorkPhoneCountryCode') == null row.isNew()) { row.setColumn('WorkPhoneCountryCode', locale.PhoneCountryCode); } } }</pre>
<pre>queryWithParameters (resource: string, dynamicParams: array)</pre>	Used to query for records by passing dynamic fields, operator, and values.	<pre>let currentRow = getCurrentRow(); let q = queryWithParameters('opportunities', [{field: 'Name', 'operator': 'contains', 'value': 'testQ'}]); let rows = q.execute(); if(rows && rows.length > 0) { let account = rows[0].getColumn('TargetPartyId'); currentRow.setColumn('TargetPartyId', account); }</pre>

Class Methods

Here are the class methods for the System Functions outlined in the previous table and examples of how you can use them.

- Row Class Methods:

Method Signature	Definition	Usage
<pre>setColumn(name: string, value: string)</pre>	Used to set value of a particular column of the row.	<pre>const optyRow = getCurrentRow(); const childRev = getCurrentChildRow();</pre>

Method Signature	Definition	Usage
		<pre>childRev.setColumn('RevAmountCurcyCode', optyRow.getColumn('CurrencyCode'));</pre>
<code>getColumn(name: string)</code>	Used to get value of a particular column of the row.	<pre>const taskRow = getCurrentRow(); const statusCode = taskRow.getColumn('StatusCode');</pre>
<code>isNew(): boolean</code>	Returns true if the row is new else it returns false.	<pre>const taskRow = getCurrentRow(); const isNew = taskRow.isNew();</pre>
<code>getResourceName(): string</code>	Returns the resource to which the row belongs.	<pre>const parentRow = getParentRow(); const res = parentRow.getResourceName();</pre>
<code>setColumnMandatory(name: string, value: boolean)</code>	Used to set value of mandatory property for a particular column. Note: Use this function instead of the CX Cloud Mobile function if you're copying your CX Cloud Mobile script.	<pre>const optyRow = getCurrentRow(); const childRev = getCurrentChildRow(); childRev.setColumnMandatory('UnitPrice', true);</pre>
<code>setColumnUpdatable(name: string, value: boolean)</code>	Used to set value of updatable property for a particular column.	<pre>const optyRow = getCurrentRow(); const childRev = getCurrentChildRow(); childRev.setColumnUpdatable('RevAmountCurcyCode', canEditCurrency === 'Y');</pre>
<code>setColumnVisible(name: string, value: boolean)</code>	Used to display/hide a particular column.	<pre>const row = getCurrentRow(); row.setColumnVisible('ReasonWonLostCode', false);</pre>
<code>addNewChildRow(childName: string)</code>	Adds new child to the row.	<pre>const row = getCurrentRow(); const resources = row.addNewChildRow('ActivityAssignee'); resources.setColumn('AssigneeName', userPrefProvider.getPartyName()); resources.setColumn('AssigneeId', userPrefProvider.getPartyId());</pre>
<code>setColumnDisplayType(fieldName: string, DisplayType.TEXT DisplayType.TEXTSCAN DisplayType.NUMBER)</code>	Used to change the display type of a field to text, scan or number.	<pre>let row = getCurrentRow(); row.setColumnDisplayType("MyCustomSerialNumber", DisplayType.TEXTSCAN);</pre>
<code>disableAction(action: string, disable: boolean)</code>	Used to disable or enable the save action in the edit view.	<pre>let optyRow = getCurrentRow(); optyRow.disableAction('Save', true);</pre>
<code>getOriginalValues()</code>	Used to get the original values of a field before it was modified.	<pre>const row = getCurrentRow(); let oriVal = row.getOriginalValues(); let oriName = oriVal.Name;</pre>

Method Signature	Definition	Usage
		<code>let currName = row.getColumn('Name');</code>

- Query Class Methods:

Method Signature	Definition	Usage
<code>getParameters()</code>	Used to get all available parameters for this query.	<pre>let oracleCxmOutcome = new Result(''); let currentRow = getCurrentRow(); let id = currentRow.getColumn('OptyId'); let qT = query('opportunities'); let param = qT.getParameters()[0]; qT.setParameters(param, id); let rows = qT.execute(); if(!rows rows[0].getColumn('OptyId') !== id) { oracleCxmOutcome.setErrorMessage("Query on current row failed"); } else { oracleCxmOutcome.setMessage("MESSAGE_ TYPE_SUCCESS", "", "Query successful"); }</pre>
<code>setParameters(name: string, value: string)</code>	Used to set value of a particular query parameter.	<pre>const resourceQuery = query('resources'); resourceQuery.setParameters('PartyId', userPrefProvider.getPartyId());</pre>
<code>execute()</code>	Used to perform the query. It returns the set of rows that satisfy the query criteria.	<pre>const resourceQuery = query('resources'); resourceQuery.setParameters('PartyId', userPrefProvider.getPartyId()); try { const resourceResponse = resourceQuery.execute(); if (resourceResponse && resourceResponse.length > 0) { resources.setColumn('JobName', resourceResponse[0].getColumn('JobMeaning')); } } catch (e) { //failed to get resource response }</pre>

- UserPreferences Class Methods:

Method Signature	Definition	Usage
<code>getUserSettings ()</code>	Returns user settings.	<pre>const userPref = getUserPreferences (); const userSettings = userPref.getUserSettings ();</pre>
<code>getUserName ()</code>	Returns the user name for the current user.	<pre>const userPref = getUserPreferences (); const userName = userPref.getUserName ();</pre>
<code>getPartyName ()</code>	Returns name of the current user.	<pre>const userPref = getUserPreferences (); const partyName = userPref.getPartyName ();</pre>
<code>getDateFormat ()</code>	Returns date format set by the current user.	<pre>const userPref = getUserPreferences (); const dateFormat = userPref.getDateFormat ();</pre>
<code>getPartyId ()</code>	Returns ID of the current user.	<pre>const userPref = getUserPreferences (); const partyId = userPref.getPartyId ();</pre>
<code>getCurrency ()</code>	Returns currency set by the current user.	<pre>const userPref = getUserPreferences (); const currency = userPref.getCurrency ();</pre>
<code>getTimezone ()</code>	Returns time zone set by the current user.	<pre>const userPref = getUserPreferences (); const timeZone = userPref.getTimezone ();</pre>
<code>getNumberFormat ()</code>	Returns number format set by the current user.	<pre>const userPref = getUserPreferences (); const numberFormat = userPref.getNumberFormat ();</pre>
<code>getLanguage ()</code>	Returns language set by the current user.	<pre>const userPref = getUserPreferences (); const language = userPref.getLanguage ();</pre>
<code>getRoles ()</code>	Returns roles assigned to the current user.	<pre>const userPref = getUserPreferences (); const roles = userPref.getRoles ();</pre>
<code>getProfileOptions ()</code>	Returns all the profile options for the current user.	<pre>const userPref = getUserPreferences (); const profileOptions = userPref.getProfileOptions ();</pre>

Method Signature	Definition	Usage
<code>getProfileOptionValue (profileOption)</code>	Returns profile option value of the specified profile option for the current user if the profile option exists, else it returns null.	<pre>const userPref = getUserPreferences(); const profileOptionValue = userPref.getProfileOptionValue (\$PROFILE_ OPTION_STRING)</pre>
<code>isRoleAssigned (role)</code>	Returns true if the specified role is assigned to the current user else it returns false.	<pre>const userPref = getUserPreferences(); const isRoleAssigned = userPref.isRoleAssigned (\$USER_ROLE_ STRING);</pre>
<code>getDefaultBU ()</code>	Used to fetch the default BU of the current user. Refer to the Business Unit Class Method below.	<pre>let userPrefProvider = getUserPreferences(); let defaultBU = userPrefProvider.getDefaultBU();</pre>

- **DeviceInfo Class Methods:**

Method Signature	Definition	Usage
<code>getCurrentPosition (maximumAge : Number , timeout : Number , enableHighAccuracy : Boolean)</code>	<p>Returns the location of the device.</p> <p>Parameters:</p> <ul style="list-style-type: none"> • maximumAge: The time in milliseconds that's acceptable for cached position. Enter 0 if a cached position isn't acceptable. • timeout: The maximum length of time (milliseconds) that's allowed to pass from the call to <code>navigator.geolocation.getCurrentPosition</code> until the corresponding <code>geolocationSuccess</code> callback executes. If the <code>geolocationSuccess</code> callback isn't invoked within this time, the <code>geolocationError</code> callback is passed a <code>PositionError.TIMEOUT</code> error code. • enableHighAccuracy: Provides a hint that the application needs the best possible results. By default, the device attempts to retrieve a 'Position' using network-based methods. Setting this property to 'true' tells the framework to use more accurate methods, such as satellite positioning. 	<pre>const deviceInfo = getDeviceInformation(); const position = deviceInfo.getCurrentPosition (3000, 5000, true); const latitude = position.coords.latitude; const longitude = position.coords.longitude;</pre>
<code>getOs ()</code>	Returns the operating system name for the device.	<pre>const deviceInfo = getDeviceInformation(); const os = deviceInfo.getOs();</pre>
<code>getPlatform ()</code>	Returns platform name of the device.	<pre>const deviceInfo = getDeviceInformation();</pre>

Method Signature	Definition	Usage
		<pre>const platform = deviceInfo.getPlatform();</pre>
<code>getVersion()</code>	Returns device version.	<pre>const deviceInfo = getDeviceInformation(); const version = deviceInfo.getVersion();</pre>
<code>getModel()</code>	Returns device model name.	<pre>const deviceInfo = getDeviceInformation(); const model = deviceInfo.getModel();</pre>
<code>isDeviceOnline()</code>	Returns true if the device is online, else false.	<pre>const deviceInfo = getDeviceInformation(); const online = deviceInfo.isDeviceOnline();</pre>

- **CXCoreLogger** Class Methods:

Method Signature	Definition	Usage
<code>setLogLevel(level: string)</code>	Set logging level. Level values can be : info , error , warn , log , and none .	<pre>const row = getCurrentRow(); row.setColumn("Name", "New oppty"); let cxCoreLogger = getCXCoreLogger().getLogger(); cxCoreLogger.setLogLevel("info"); cxCoreLogger.info("On create event executed")</pre>
<code>log(...args: any[])</code>	Logs the message at log level.	<pre>const row = getCurrentRow(); row.setColumn("Name", "New oppty"); let cxCoreLogger = getCXCoreLogger().getLogger(); cxCoreLogger.setLogLevel("log"); cxCoreLogger.log("On create event executed");</pre>
<code>info(...args: any[])</code>	Logs the message at info level.	<pre>const row = getCurrentRow(); row.setColumn("Name", "New oppty"); let cxCoreLogger = getCXCoreLogger().getLogger(); cxCoreLogger.setLogLevel("info"); cxCoreLogger.info("On create event executed");</pre>
<code>warn(...args: any[])</code>	Logs the message at warn level.	<pre>const row = getCurrentRow(); row.setColumn("Name", "New oppty"); let cxCoreLogger = getCXCoreLogger().getLogger(); cxCoreLogger.setLogLevel("warn"); cxCoreLogger.warn("On create event executed");</pre>

Method Signature	Definition	Usage
<code>error(...args: any[])</code>	Logs the message at error level.	<pre>const row = getCurrentRow(); row.setColumn("Name", "New oppty"); let cxCoreLogger = getCXCoreLogger().getLogger(); cxCoreLogger.setLogLevel("error"); cxCoreLogger.error("On create event executed");</pre>

- Result Class Methods:

Method Signature	Definition	Usage
<code>setModifiedObject()</code>	Used to include a newly created row, or updated row, in the set of rows that the script has modified. This action will ensure that the included row is committed by event-handler implementation.	<pre>var oracleCxmOutcome = new Result(''); var row = getCurrentRow(); row.setColumn("Name", "Sample Obj"); oracleCxmOutcome.setModifiedObject(row);</pre> <p>Note: Any row created or modified in an After Save event needs to be added using the Result object's <code>setModifiedObject</code> API, so that the changes are saved.</p>
<code>setMessage(key, stringBundle, message)</code>	Used to add a message that can be returned to the event handler implementation. The key can be: "MESSAGE_TYPE_SUCCESS" or "MESSAGE_TYPE_ERROR".	<pre>var oracleCxmOutcome = new Result(''); var row = getCurrentRow(); row.setColumn("Name", "Sample Obj"); oracleCxmOutcome.setModifiedObject(row); oracleCxmOutcome.setMessage("MESSAGE_ TYPE_SUCCESS", "", "Row modified");</pre>
<code>setErrorMessage(message)</code>	Used to set the error message and to mark the outcome qualifier as "OUTCOME_TYPE_FAILURE".	<pre>var oracleCxmOutcome = new Result(''); var row = getCurrentRow(); var a = row.getColumn("A"); var b = row.getColumn("B"); if(a>b){ oracleCxmOutcome.setErrorMessage("Row validation failed"); } else { oracleCxmOutcome.setMessage("MESSAGE_ TYPE_SUCCESS", "", "Row validated successfully"); }</pre>
<code>setOutcomeQualifier()</code>	Used to set qualifier to be handled post script execution. Returned string could be either "OUTCOME_TYPE_SUCCESS" or "OUTCOME_TYPE_FAILURE". If the qualifier "OUTCOME_TYPE_FAILURE" is used in the <code>beforeSave</code> event, then the page will show the error message set using <code>setMessage/setErrorMessage</code> .	<pre>var oracleCxmOutcome = new Result(''); var row = getCurrentRow(); var a = row.getColumn("A"); var b = row.getColumn("B"); if(a>b){ oracleCxmOutcome.setMessage("MESSAGE_ TYPE_ERROR", "", "Row validation failed");</pre>

Method Signature	Definition	Usage
		<pre>oracleCxmOutcome.setOutcomeQualifier("OUTCOME TYPE_FAILURE"); }</pre>

- Business Unit Class Methods:

Method Signature	Definition	Usage
<code>getName ()</code>	Used to get name of the Business Unit.	<pre>let userPrefProvider = getUserPreferences(); let defaultBU = userPrefProvider.getDefaultBU(); let name = defaultBU.getName();</pre>
<code>getBusinessUnitId ()</code>	Used to get BusinessUnitID.	<pre>let userPrefProvider = getUserPreferences(); let defaultBU = userPrefProvider.getDefaultBU(); let businessUnitId = defaultBU.getBusinessUnitId();</pre>
<code>getBUName ()</code>	Used to get the Business Unit name.	<pre>let userPrefProvider = getUserPreferences(); let defaultBU = userPrefProvider.getDefaultBU(); let buName = defaultBU.getBUName();</pre>
<code>getBUId ()</code>	Used to get BU ID.	<pre>let userPrefProvider = getUserPreferences(); let defaultBU = userPrefProvider.getDefaultBU(); let buID = defaultBU.getBUId();</pre>
<code>getDefaultBUFlag ()</code>	Used to get default BUFlag.	<pre>let userPrefProvider = getUserPreferences(); let defaultBU = userPrefProvider.getDefaultBU(); let defaultBUFlag = defaultBU.getDefaultBUFlag();</pre>

Test Your Configurations

After you configure the Oracle CX Sales Mobile using Application Composer, you should test your configurations before distributing them to your user's mobile devices.

Here's how:

1. After making your configurations, keep the Oracle CX Sales application open and make sure that the sandbox where you made the changes is active in the application.

2. Open CX Sales Mobile on your mobile device and sign in as the same user that you used to make your configurations. If you have already signed in to CX Sales Mobile, you need to force close the app and open it again. This is because the app picks up the latest updates in the sandbox when you restart the app, or when you sign in.
3. Once you have restarted the app and signed in, your configurations are ready to view. The sandbox where you made your configurations is linked to your user details, and the configurations are automatically downloaded when you restart the app. However, you can double-check that you're in the correct sandbox by tapping **Settings > System** in the CX Sales Mobile menu and checking the **Sandbox Name** and the **Last Download Time**.
4. Check your configurations. If you have more than one pod and you want to check your configurations in another pod, then you need to connect to the other pod by changing the host URL. Navigate to **Settings** in the CX Sales Mobile app, tap **Reset Application**, and enter the host URL of the relevant pod. For details about finding your host URL, see the Prepare section of the [Get Started with Your Mobile Implementation](#) topic.
5. If you make further updates in the sandbox, you need to force close the app, and sign in again to see the updates.
6. When you're happy with your configurations, publish the sandbox to distribute the configurations to your users.